

PROCESSUS MÉTIER ET COMPOSANTS LOGICIELS POUR LA GESTION INTÉGRÉE DES EAUX EN MILIEU URBAIN

THÈSE N° 2334 (2001)

PRÉSENTÉE AU DÉPARTEMENT DE GÉNIE RURAL

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES TECHNIQUES

PAR

VINCENT MOTTIER

Ingénieur du génie rural diplômé EPF
originaire de Gy (GE)

acceptée sur proposition du jury:

Prof. A. Musy, directeur de thèse
Prof. J.-C. Deutsch, rapporteur
Prof. F. Golay, rapporteur
Prof. W. Gujer, rapporteur
Prof. S. Pierre, rapporteur

Lausanne, EPFL
2001

Résumé

La gestion des eaux dans les agglomérations est une tâche complexe, en raison des multiples interactions qui existent entre les composantes du système hydrologique urbain - eaux de surface et souterraines, infrastructures d'assainissement et d'approvisionnement en eau potable, ménages, industries, territoire - et des nombreuses catégories d'intervenants impliqués. Cette complexité, ainsi que la prise en compte de la notion de développement durable, font que les différentes activités touchant de près ou de loin à la gestion des eaux en milieu urbain doivent être abordées non plus de manière sectorielle, mais selon une approche globale et intégrée.

Cela implique que soient pris en compte, lors de chaque décision ou de chaque intervention, de nombreux paramètres et que soient manipulées d'énormes quantités d'informations. Les outils informatiques apportent dans un tel contexte un soutien bienvenu. La mise en oeuvre de ces outils n'est cependant pas chose aisée : il apparaît que les échanges d'informations entre intervenants restent difficiles, que les fonctionnalités spécifiques au domaine de la gestion des eaux manquent, et que la complexité même du domaine rend la mise sur pied d'outils adaptés difficile et coûteuse.

Partant de ces constatations, la présente recherche a pour objectif de proposer des concepts et une démarche permettant de mettre en place des outils logiciels à même de soutenir efficacement une gestion intégrée des eaux en milieu urbain. Une première étape a consisté à identifier les critères qui doivent être pris en compte par de tels outils. Il s'est avéré que ces derniers doivent : 1°) être en mesure d'évoluer parallèlement aux méthodes de gestion des eaux; 2°) être adaptables aux besoins spécifiques des différents utilisateurs; 3°) être compatibles avec différentes configurations informatiques; 4°) être ouverts sur les applications logicielles existantes; 5°) pouvoir être construits rapidement et modifiés facilement; 6°) pouvoir être mis en oeuvre simplement; et 7°) ne pas être trop coûteux.

Compte tenu de ces exigences, notre réflexion nous a conduits à proposer que les outils informatisés d'aide à la gestion des eaux soient conçus par assemblage d'entités logicielles indépendantes, les composants métier. Véritables briques logicielles, de tels composants autorisent en effet la construction d'outils adaptés aux besoins des utilisateurs tout en permettant de profiter des avantages liés à une approche par intégration de composants préexistants : construction rapide, coûts réduits, fiabilité élevée. Précisons que les composants métier sont des assemblages autonomes de composants logiciels qui rendent possible la manipulation d'objets informatiques à forte granularité, les objets métier. Ces derniers constituent des abstractions de concepts utilisés par les intervenants dans le contexte de leur activité professionnelle.

D'un point de vue théorique, notre proposition repose sur trois idées principales :

- la complexité inhérente à la problématique considérée - la gestion des eaux en milieu urbain - peut être maîtrisée, au niveau de la conception d'outils logiciels, par segmentation : chaque composant métier permet de manipuler un ensemble cohérent et clairement délimité de concepts. A partir de là, un outil logiciel peut être construit par assemblage des composants métier nécessaires, cette complexité étant alors réintégrée de manière maîtrisée;
- le territoire et ses composantes peuvent jouer un rôle d'articulation entre les métiers des différentes catégories d'intervenants. Dans le domaine de la gestion des eaux, les intervenants manipulent, dans le contexte de leurs processus métier respectifs, des abstractions des mêmes entités "réelles".

Certains composants métier permettent spécifiquement de manipuler ces abstractions, en les présentant sous des formes adaptées aux besoins des utilisateurs. Ainsi, un composant métier se rapportant à un ensemble de concepts particuliers sera mis en oeuvre par l'ensemble des intervenants qui manipulent ces concepts;

- les composants métier peuvent être conçus de manière à ce que leur potentiel de réutilisation dans des applications différentes soit maximal. En effet, plus ce potentiel est élevé, plus il vaut la peine de réaliser le composant métier considéré.

D'un point de vue technique, les notions qui sous-tendent notre recherche sont :

- le paradigme orienté objet, qui autorise l'établissement d'une correspondance entre système réel et système logiciel et facilite la mise en place de systèmes logiciels modulaires, évolutifs et intuitifs.
- les composants logiciels qui, regroupant au sein d'une même unité logicielle un ensemble de fonctionnalités et de concepts logiquement corrélés, offrent un niveau de granularité adéquat en regard de l'objectif de réutilisation qui est le nôtre.
- l'interopérabilité entre composants, qui est rendue possible par la nette séparation entre interfaces (soit les définitions des services offerts) et implémentations (soit les manières d'accomplir ces services) ainsi que par la standardisation de ces interfaces.

Nous proposons de déterminer les composants métier nécessaires à la construction d'outils informatisés d'aide à la gestion des eaux à partir de l'analyse des processus métier menés par les différents intervenants. Deux types principaux de composants métier sont distingués, les seconds dépendant des premiers : les composants métier de type entité, qui permettent la manipulation d'abstractions d'entités du monde réel, et les composants métier de type processus, destinés à soutenir directement les processus métier spécifiques aux différents intervenants, et qui mettent à disposition pour cela les fonctionnalités adéquates. Nous nous sommes également penchés sur la manière de structurer les composants métier de façon à ce qu'ils soient évolutifs et adaptables à de multiples configurations informatiques.

Destinés à être réutilisés, les composants métier doivent être construits très soigneusement, et la mise en oeuvre d'une méthode de développement est pour cela indispensable. Par conséquent, nous en proposons une, dont les caractéristiques sont les suivantes : 1°) rendre possible des cycles de développement rapides, chaque cycle conduisant à la livraison de composants métier opérationnels; 2°) conduire à la mise en place, au fil des itérations, d'un référentiel commun, le modèle du domaine de la gestion des eaux, qui est progressivement affiné et permet que soient construits successivement des composants métier cohérents entre eux; et 3°) fournir une technique qui maximise le potentiel de réutilisation de ces composants.

Tant les concepts que la méthode de développement proposés ont été validés par le biais de prototypes. Il ressort qu'une approche par composants métier des outils logiciels d'aide à la gestion des eaux constitue une réelle opportunité d'améliorer les processus de gestion dans ce contexte, ainsi que de disposer de véritables moyens de suivi et de pilotage du système "Eaux en milieu urbain". Autorisant une valorisation des données disponibles et facilitant les échanges d'informations entre intervenants, la mise en oeuvre de composants métier doit être encouragée. C'est pourquoi ce travail se conclut par la proposition d'un processus organisationnel à même de stimuler et de coordonner la mise sur pied d'un véritable "marché" de composants métier destinés à la gestion des eaux en milieu urbain.

Abstract

Water management in urban areas is a complex task because of the several interactions that exist between the different facets of the urban water system (surface and subsurface water, sewer system, water supply system, water and wastewater treatment plant, etc) and the many stakeholders in this context. It is necessary to reconsider current practices in order to shift from a sectorised approach towards an integrated and global one.

Such an approach implies that for each decision that has to be taken or each operation that has to be planned, numerous parameters must be considered and huge amounts of data have to be handled. Software tools offer in this context very interesting potentialities because of their ability to store, manage and exploit large volumes of spatial, thematic and temporal data. However, it turns out that such tools are not so easy to put into practice : exchange of information remains difficult, a lack of functions specific to water management has been stated, and the complexity of the domain itself makes the construction of dedicated software systems difficult and costly.

The objectives of the present study are thus to develop concepts and a methodology allowing the development of software systems that support the management of urban water in an integrated and sustainable way. First, an identification of the criteria that must be fulfilled by such systems is carried out. It is shown that they must be: 1°) able to evolve in the same direction as water management practices; 2°) adaptable to the specific needs of their users; 3°) deployable on various computer systems configurations; 4°) able to interact with third party software systems; 5°) built and modified easily ; 6°) user friendly; and 7°) affordable.

This research, taking into account these requirements, proposes to build software systems by assembling business components. Business components are reusable autonomous groups of software components that allow the handling of business objects, which are themselves high granularity objects that represent things active in the business domain. Indeed, such components allow the construction of software systems that exactly match users' requirements while taking advantage of the integration of preexisting parts : rapid development processes, cost reduction and increased quality.

From a theoretical point of view, the proposal is built on three main principles :

- the complexity inherent to the domain of urban water management is tackled by segmentation. Each business component allows the handling of a coherent, limited and clearly defined set of concepts. A software tool can be built by integrating such business components and then be customized, thus recreating the complexity of the domain, but in a controlled manner;
- "real world" entities act as points of articulation between the businesses of the stakeholders : in the field of urban water management, the different actors handle, in the context of their business processes, abstractions of the same "real world" entities. Specific business components support the handling of these abstractions, submitting them to each user in a form that fits their needs. In this way, a business component, implementing a well defined set of concepts, will be used by the group of users concerned by these concepts, allowing an integrated approach of the different business processes;
- business components must be conceived in order to maximise their reusability potential in different software applications. Indeed, if this potential is high it is more worthwhile to develop the concerned

business component.

From a technical perspective, the proposal is based on :

- object orientation, which allows a direct mapping from the real world to the software system, and facilitates the development of systems that are modular, evolutive and easy to understand;
- software components, which allow the grouping of logically correlated functions and concepts into autonomous units. Such components are, with regards to their reusability, of an adequate level of granularity;
- interoperability among components realised through a clear distinction between interfaces (the definition of the services offered) and implementations (the way of carrying out these services), associated with a standardisation of these interfaces.

This research presents a way to identify, characterize and develop a set of business components specifically for urban water management software systems. The structure of such components is described. Two main kinds of business components are distinguished : entity components, which allow the handling of real world entities; and process components, which are designated to support specific business processes, and offer for this aim the adequate functions.

Designated to be reused, business components must be constructed very meticulously, and the use of a software development method is for this purpose unavoidable. A method, dedicated for the development of business components in the domain of urban water management, is presented, whose characteristics are : 1°) to be based on the analysis of the business processes of the stakeholders; 2°) to allow rapid, iterative cycles, each of them leading to a set of operational business components; 3°) to allow the stepwise definition of a global reference model of the urban water management domain that supports the development of coherent and non redundant business components; and 4°) to offer a technique that maximises the reusability potential of these components.

The presented concepts, as well as the development method, are validated through prototyping. It is shown that a business component based approach of software systems offers a real opportunity to improve the business processes in the field of urban water management. Such components also offer possibilities to easily implement software systems that will help the monitoring and management of the urban water system. The enhancement of the value of the available data and the information exchange between stakeholders is facilitated. Thus, the development and use of business component based approaches must be encouraged. For this reason, this research concludes with the proposal of an organisational process aimed to stimulate and coordinate the setting up of a "business component market" for urban water management systems.

Remerciements

Une thèse de doctorat est un travail de longue haleine qui nécessite non seulement un investissement personnel important, mais aussi le concours et le soutien de nombreuses personnes. Je tiens ici à remercier celles et ceux qui ont contribué à la réussite de mon entreprise : le professeur André Musy, qui, après m'avoir accueilli dans l'institut qu'il dirige, m'a fait confiance et m'a accordé son soutien durant les différentes phases de cette recherche; le professeur Willi Gujer, qui a suivi de très près mon travail et m'a consacré beaucoup de temps. Ses remarques pertinentes et sa manière de mettre à l'épreuve mes idées n'ont certes pas toujours été faciles à gérer, mais ont contribué de manière décisive à la qualité de mon travail; le professeur François Golay, pour sa disponibilité et les nombreuses discussions que nous avons eues lors des différentes étapes de ma recherche.

Mon travail s'appuie en grande partie sur des entretiens et des discussions menés avec des spécialistes tant en gestion des eaux qu'en informatique. À ce titre, différentes personnes m'ont généreusement donné de leur temps pour répondre à mes questions. Je tiens ici à toutes les remercier, et en particulier celles que j'ai rencontrées à plusieurs reprises : Daniel Bernasconi, Roland Cottier, Mohamed Kande, Eric Mennel, Guy Reyfer, Jacques Roduit, Luca Rossi et Claude-Alain Vuillerat.

Un grand merci également à ces autres personnes qui ont relu tout ou partie de mon travail et se sont penchées tant sur le fond que sur la forme de celui-ci, soit Alain Buogo, Didier Buchs, Marc Riedo, Luca Rossi, Christina Benagli, ainsi que mon épouse Annick et mon père Claude-André Mottier.

Certains encouragements m'ont été particulièrement précieux pour mener à terme ce travail. Je pense ici à ceux prodigués par Guy Reyfer, Marc Riedo et Luca Rossi, déjà cités ci-dessus. L'intérêt qu'ils ont manifesté pour ma recherche a constitué pour moi un surcroît de motivation bienvenue.

J'adresse une pensée toute particulière à Annick, qui m'a toujours encouragé à mener à terme ce travail. Je la remercie pour la patience dont elle a fait preuve ces dernières années, puisque la réalisation de la présente recherche a nécessité passablement de sacrifices au niveau familial, et ne m'a pas permis de m'occuper de mon fils Grégoire autant que je l'aurais souhaité. Un grand merci également à mes parents, qui ont facilité mes études et m'ont soutenu tout au long de ce travail.

Enfin, toute ma reconnaissance va à l'Institut Fédéral pour l'Aménagement, l'Épuration et la Protection des Eaux (EAWAG), qui a assuré la majeure partie du financement de cette recherche.

Table des matières

Chapitre 1	Introduction	1
1.1	Contexte	1
1.2	Problématique	3
1.3	Démarche	4
1.4	Plan du rapport	5
Chapitre 2	Gestion intégrée des eaux en milieu urbain	7
2.1	Le système "Eaux en milieu urbain"	7
2.1.1	Origines des concepts actuels de gestion des eaux en milieu urbain	7
2.1.2	Définition de la gestion des eaux	9
2.1.2.1	Les missions	9
2.1.2.2	Systémique et gestion des eaux en milieu urbain	10
2.1.3	La gestion des eaux et le développement durable	16
2.1.3.1	La notion de développement durable	16
2.1.3.2	Les causes de "non-durabilité" des pratiques actuelles	18
2.1.3.3	Une nouvelle approche de la gestion des eaux en milieu urbain ?	20
2.2	L'organisation de la gestion des eaux en Suisse	21
2.2.1	Le système politique suisse et la gestion des eaux	21
2.2.2	Les bases législatives de la gestion des eaux	23
2.2.3	Les intervenants dans le contexte de la gestion des eaux	25
2.3	Analyse des pratiques actuelles en matière de gestion des eaux en milieu urbain	26
2.3.1	Procédure	26
2.3.2	Objectifs généraux, processus et opérations	28
2.3.2.1	Définition des concepts	28
2.3.2.2	Les objectifs généraux	30
2.3.2.3	Les processus métier	31
2.4	Les technologies informatiques et la gestion des eaux en milieu urbain	32
2.4.1	Une typologie des outils logiciels	32
2.4.1.1	Programmes, logiciels et outils d'aide à la gestion	32
2.4.1.2	La notion de système d'information (SI)	34
2.4.1.3	Les systèmes d'information à référence spatiale (SIRS)	36
2.4.2	Les outils logiciels utilisés dans le contexte de la gestion des eaux	38
2.4.2.1	Généralités	38
2.4.2.2	Les logiciels de conception assistée par ordinateur	39
2.4.2.3	Les bases de données et les systèmes de gestion de bases de données	40
2.4.2.4	Les systèmes d'information géographique	42
2.4.2.5	Les outils de simulation de comportement	44
2.4.2.6	Les tableurs	45
2.4.2.7	Les outils de manipulation de séries temporelles	45
2.4.2.8	Les solutions spécifiques	46
2.5	Synthèse du chapitre	48

Chapitre 3	Caractéristiques requises d'un outil d'aide à la gestion des eaux.....	49
3.1	Les missions.....	49
3.1.1	Assurer la gestion des données	49
3.1.2	Fournir des informations de contrôle et de pilotage	53
3.1.3	Soutenir les processus métier	56
3.2	Les contraintes.....	57
3.2.1	Une grande diversité d'utilisateurs et d'utilisations.....	57
3.2.2	Des organisations de taille variable	59
3.2.3	Des outils informatiques qu'il s'agit de prendre en compte.....	61
3.2.4	Un contexte en pleine évolution.....	62
3.2.5	Des besoins de personnalisation	63
3.2.6	Des modèles de données complexes	64
3.2.7	La nécessité de pouvoir s'appuyer sur des processus de développement rapides	66
3.2.8	Des moyens financiers limités	67
3.3	Les objectifs.....	68
Chapitre 4	Une conception modulaire des outils d'aide à la gestion des eaux en milieu urbain	71
4.1	Principe général.....	71
4.2	Paradigme orienté objet.....	73
4.2.1	Principe	73
4.2.2	Formalismes	76
4.3	Approche par composants	77
4.3.1	Motivations et principes.....	77
4.3.2	Définition du concept de composant.....	80
4.3.3	SIAGEM et composants logiciels.....	84
4.4	Prise en compte du territoire et de ses composantes	87
4.5	Les composants métier	91
4.5.1	Concept général.....	91
4.5.2	Définitions.....	92
4.5.3	Structure interne d'un composant métier.....	94
4.5.3.1	Introduction.....	94
4.5.3.2	Les différentes constituantes d'un composant métier	95
4.5.3.3	Composants métier et bases de données	99
4.5.3.4	Communication interne au composant métier	100
4.5.4	Les catégories de composants métier	102
4.5.4.1	Les catégories	102
4.5.4.2	La notion de médiateur	103
4.5.4.3	La notion de façade.....	104
4.5.5	Sémantique des composants métier	107
4.5.6	Interactions entre composants métier.....	108
4.5.6.1	Principes	108
4.5.6.2	Dépendances fonctionnelles entre composants	109
4.5.6.3	Communication par événements.....	110
4.5.6.4	Relations entre composants métier de même niveau hiérarchique	110

4.6	Synthèse du chapitre et conclusion	112
Chapitre 5	Proposition d'une méthode de développement.....	115
5.1	Motivations.....	115
5.2	Méthodes de conception de systèmes d'information	116
5.2.1	Introduction.....	116
5.2.2	Les processus de développement	118
5.2.3	Les méthodes de développement	120
5.3	Description de la méthode COWIS.....	121
5.3.1	Processus de développement.....	121
5.3.2	Méthode de développement	124
5.3.2.1	Introduction.....	124
5.3.2.2	Spécifications des caractéristiques des composants métier	126
5.3.2.3	Mise en place d'une infrastructure pour composants métier.....	127
5.3.2.4	Analyse des besoins	129
5.3.2.5	Analyse et segmentation de domaine	132
5.3.2.6	Analyse orientée composants	136
5.3.2.7	Conception orientée composants	144
5.3.2.8	Implémentation de composants métier	146
5.3.2.9	Test et documentation de composants métier	146
5.4	Synthèse.....	147
Chapitre 6	Mise en oeuvre du concept de composant métier	149
6.1	Contexte.....	149
6.2	Réalisation d'un prototype de composant métier	150
6.2.1	Spécification des caractéristiques et mise en place de l'infrastructure.....	150
6.2.2	Analyse des besoins	151
6.2.3	Analyse et segmentation de domaine.....	152
6.2.4	Analyse orientée composant	153
6.2.5	Conception orientée composant.....	155
6.2.6	Implémentation, test et documentation	160
6.3	Intégration du composant métier dans une application	161
6.3.1	Objectifs et architecture générale.....	161
6.3.2	Présentation de l'application	163
6.4	Synthèse du chapitre et conclusion	171
Chapitre 7	Perspectives et conclusion	175
7.1	Synthèse.....	175
7.2	Vers un "marché" de composants métier ?	179
7.3	Conclusion	182
Bibliographie		185

Annexes	197
Annexe 1 Description des processus métier	198
Annexe 2 Les formalismes UML	205
Annexe 3 COM et ses évolutions	211
Annexe 4 Systèmes d'information et architectures en couches	214
Annexe 5 Principaux types de processus de développement	217
Annexe 6 Notions d'acteur et de cas d'utilisation	222
Annexe 7 Infrastructures de persistance	225
Annexe 8 La vision de l'Open GIS Consortium (OGC)	227
Annexe 9 Le composant logiciel SIG MapX.....	231
Annexe 10 Exemples de cas d'utilisation	235

Liste des abréviations

AGL	Atelier de Génie Logiciel
BD	Base de Données
CAO	Conception Assistée par Ordinateur
CASE	Computer Assisted Software Engineering
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COWIS	COmponents for Water Information Systems
DAO	Dessin Assisté par Ordinateur
IDL	Interface Definition Language
MCD	Modèle Conceptuel de Données
MIDL	Microsoft Interface Definition Language
MLD	Modèle Logique de Données
MPD	Modèle Physique de Données
MCT	Modèle Conceptuel de Traitements
MOT	Modèle Organisationnel de Traitement
MPT	Modèle Physique de Traitement
OCL	Object Constraint Language
PGEE	Plan Général d'Evacuation des Eaux
PREE	Plan Régional d'Evacuation des Eaux
SGBD	Système de Gestion de Bases de Données
SI	Système d'Information
SIAGEM	Système Informatisé d'Aide à la Gestion des Eaux en Milieu urbain
SIG	Système d'Information Géographique
SIRS	Système d'Information à Référence Spatiale
SIT	Système d'Information du Territoire
UML	Unified Modeling Language

Introduction

1.1 Contexte

La gestion des eaux dans les agglomérations doit de nos jours se concevoir selon une approche globale et intégrée, tenant compte des nombreuses interactions qui existent entre les différents maillons du système hydrologique urbain (infrastructures d'approvisionnement en eau, ménages et entreprises, infrastructures d'assainissement, stations d'épuration, eaux de surface, eaux souterraines, précipitations et territoire). Ces sous-systèmes sont en effet interdépendants, et la moindre intervention au niveau de l'un d'entre eux a des répercussions sur les autres sous-systèmes. Cette transition d'approches sectorielles à une considération globale de la gestion des eaux s'est par exemple traduite dans le domaine de l'assainissement par le passage d'une vision qui préconisait d'évacuer les eaux usées hors des agglomérations (sans prise en compte de leur qualité) le plus rapidement et de la manière la plus complète possible (Hörler, 1966), à une nouvelle conception, matérialisée en Suisse par le "Plan général d'évacuation des eaux" ou PGEE (VSA, 1989), selon laquelle les cycles hydrologiques naturels doivent être aussi peu que possible perturbés, et les eaux usées peu ou non polluées infiltrées sur place lorsque la situation le permet.

Une telle approche implique pour les intervenants concernés la nécessité de tenir compte de grandes quantités d'informations. De nombreuses données (séries de mesures, informations relatives aux infrastructures ou à différentes catégories d'événements, données cadastrales, etc.) sont collectées par diverses institutions et organismes. Toutes ces données constituent un patrimoine de grande valeur, dont il faut à tout prix assurer la pérennité. De plus, les données disponibles ne sont pas à l'heure actuelle suffisamment exploitées. Leur utilisation et leur valorisation plus intensives conduirait pourtant à une amélioration des processus de gestion des eaux, car il serait possible de fournir aux gestionnaires une

information plus riche et mieux adaptée à leurs besoins. Dans ce contexte, les outils informatiques, de par leur qualité intrinsèque, sont d'une grande utilité. Ils ne donnent cependant pas encore entière satisfaction.

L'attention s'est en fait jusqu'à présent principalement focalisée sur les aspects relatifs au stockage et à la gestion des données. A cet effet, il a été proposé en Suisse un certain nombre de modèles de données qui touchent de près ou de loin au domaine de l'eau : citons la norme SIA 405 (SIA, 1999), qui s'attache à la description, à la représentation et à l'échange de données relatives aux différents types de réseaux souterrains (eaux, eaux usées, gaz, électricité, télécommunications), le modèle VSA-DSS, proposé par l'Association suisse des professionnels de la protection des eaux, relatif au domaine de l'assainissement (VSA, 1999), ou le projet GESREAU de l'administration cantonale vaudoise, qui intègre un modèle de données relatif à la gestion des eaux de surface (Crausaz et Musy, 1997).

De tels modèles sont extrêmement utiles dans la mesure où ils fournissent aux différents intervenants un canevas permettant l'organisation et facilitant les échanges de données. Ils définissent de façon univoque les objets qui doivent être considérés, les attributs qui les caractérisent ainsi que les relations qui les lient. Des outils informatiques basés sur ces modèles existent qui permettent l'acquisition, la gestion et la visualisation des données. Cependant, la valorisation de ces dernières implique également la génération d'informations de plus haut niveau (indicateurs par exemple), ainsi que leur utilisation dans le but de soutenir les processus métier des différents intervenants. Or ces problématiques n'ont que peu été prises en compte jusqu'à présent. L'informatique doit ici également être mise à contribution : des outils logiciels interagissant avec ces modèles de données et proposant les fonctions évoluées dont il vient d'être question doivent être proposés. Sans de tels outils, à même d'appuyer la gestion et l'exploitation des énormes volumes de données concernés, la perspective de pouvoir mener une réelle gestion intégrée des eaux en zones urbaines est illusoire.

La mise en place de ces outils logiciels se heurte cependant à de nombreuses difficultés : les modèles de données dont il est question ici, conçus pour répondre aux besoins de multiples utilisateurs, sont extrêmement complexes et difficiles à maîtriser, et la construction d'outils informatiques intégrant de tels modèles nécessite du temps et l'engagement de moyens financiers trop importants pour de nombreux organismes actifs dans le domaine. Le fait que les différents intervenants, dont les responsabilités professionnelles diffèrent, exigent des outils logiciels spécifiquement adaptés à leurs besoins, complique encore ce problème. En effet, si les modèles de données cités plus haut permettent l'intégration des données, les utilisateurs souhaitent à présent disposer d'outils leur apportant un appui dans le contexte de leurs processus de gestion, et leur présentant l'information de manière adéquate en regard de l'utilisation qu'ils ont l'intention d'en faire. Les outils logiciels demandés doivent de plus pouvoir être adaptés à des configurations très variables : les institutions et organisations en charge de la gestion des eaux sont en effet de tailles diverses. A ces différentes contraintes s'ajoute encore le fait que les différentes facettes de la gestion des eaux ne font généralement pas l'objet d'une gestion centralisée : si l'on veut mener une gestion véritablement intégrée, il faut pouvoir accéder à des informations disséminées dans de multiples institutions, gérées par des systèmes informatiques différents, ce qui a des implications sur la conception des outils logiciels chargés d'appuyer cette gestion intégrée des eaux.

De plus, la manière de considérer la gestion des eaux est à l'heure actuelle remise en question. En effet, force est de constater que les pratiques actuelles dans ce contexte sont inadéquates si l'on prend en

compte les questions de développement durable. Un changement des paradigmes qui gouvernent la gestion des eaux semble donc nécessaire. Ceci doit être pris en compte lors de la conception d'outils logiciels, ces derniers devant être à même de s'adapter facilement et d'évoluer rapidement sous peine de très vite devenir obsolètes et inutilisables.

1.2 Problématique

Le développement d'outils logiciels adaptés au contexte de la gestion des eaux en milieu urbain semble donc très difficile. Il faut en effet que de tels outils assistent les gestionnaires de l'eau dans leurs tâches tout en répondant aux critères suivants :

- pouvoir être adaptés à différentes configurations informatiques;
- offrir des services spécifiques à chaque catégorie d'intervenants, sous la forme attendue;
- pouvoir rapidement et facilement être mis en place et modifiés;
- être peu coûteux à développer et à maintenir;
- faciliter l'exploitation de modèles de données complexes, inévitables dans un contexte de gestion intégrée des eaux.

Si de telles exigences paraissent, de prime abord, être difficiles à concilier, les technologies informatiques actuelles offrent des opportunités qui rendent possible la mise en place de solutions adéquates. Une réflexion est cependant nécessaire quant à la manière de mettre en oeuvre ces technologies dans le contexte de la gestion des eaux.

Le présent travail a précisément pour objectif de **proposer des concepts permettant la mise en place de systèmes informatisés d'aide à la gestion intégrée des eaux en milieu urbain**. Les résultats issus de cette recherche passent par :

- une identification et une analyse des processus métier relatifs à la gestion des eaux en milieu urbain;
- un inventaire des besoins et des contraintes à prendre en compte en ce qui concerne la mise en place d'outils informatisés;
- le développement d'un ensemble de concepts et de propositions quant à la manière de créer de tels outils;
- la description d'une méthode de développement d'outils logiciels adaptée aux concepts proposés.

L'idée n'est ainsi pas, dans ce travail, de se concentrer sur un processus ou un aspect particulier de l'hydrologie urbaine, mais de mettre sur pied des concepts à même de faciliter la valorisation des données, l'utilisation des outils disponibles et la collaboration des intervenants dans ce contexte. Il n'est par conséquent pas porté de jugement sur tel ou tel modèle de comportement ou processus métier.

Précisons enfin que les concepts et la méthode proposés peuvent être adaptés à tout contexte 1°) dans lequel de nombreuses informations relatives au territoire et à ses composantes sont manipulées; 2°) où les besoins de nombreuses catégories d'intervenants doivent être pris en compte; et 3°) qui est soumis à une évolution rapide des technologies et philosophies de gestion. Nous nous sommes cependant, dans le cadre de cette recherche, plus particulièrement consacrés à la problématique des eaux en milieu urbain.

1.3 Démarche

La démarche adoptée dans cette recherche (figure 1) peut être placée dans la perspective d'une recherche-action, au sens où la définit Resweber (1995; cité dans Pornon, 1998) : à partir de l'observation des processus métier du domaine de la gestion des eaux et d'interactions avec les acteurs concernés, des concepts et propositions sont développés, puis leur pertinence est vérifiée. La figure 1 présente les différentes étapes de la démarche adoptée :

- dans un premier temps, une analyse de la situation est effectuée. Les besoins des intervenants du domaine de la gestion des eaux sont identifiés, sur la base d'entretiens et de l'examen de la littérature disponible (littérature scientifique, de vulgarisation et documents de travail). Leurs méthodes de travail sont analysées, et les interactions de ces intervenants avec les outils informatiques qu'ils utilisent habituellement sont discutées.

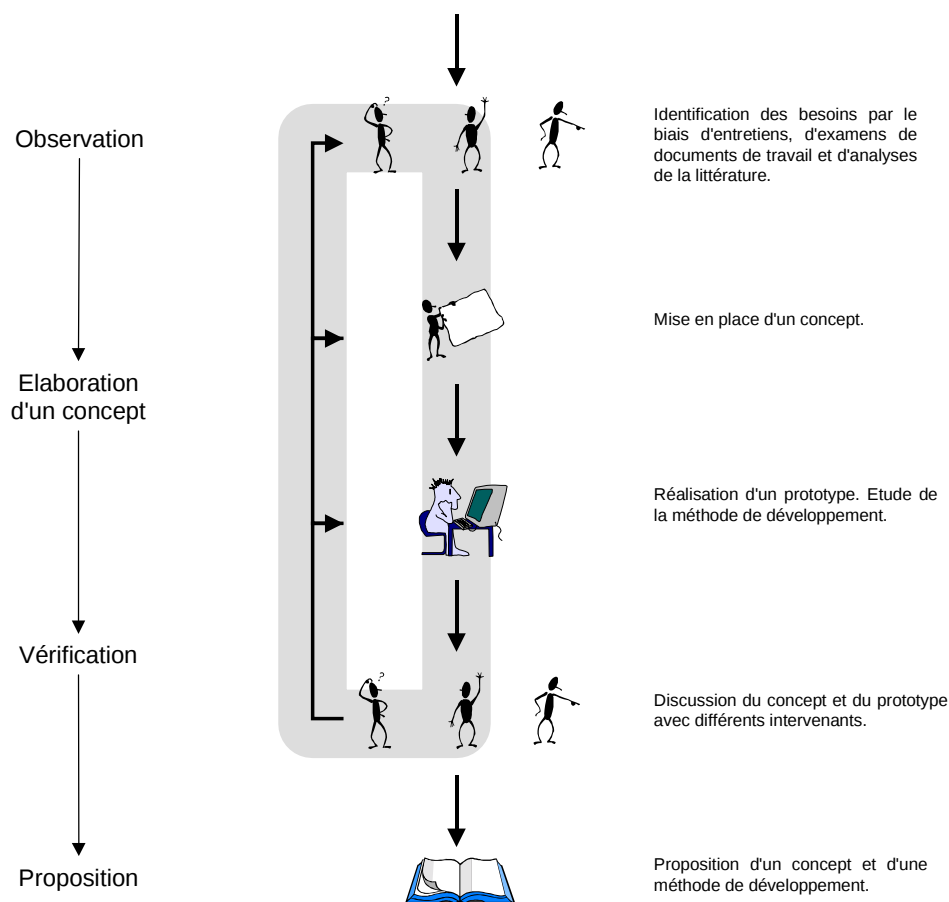


Figure 1. Démarche suivie dans le cadre du présent travail.

- une étude des méthodologies et technologies informatiques disponibles est ensuite menée, avec pour objectif l'identification de celles qui pourraient être utiles dans le contexte de la présente recherche.
- à partir de ces investigations, tant dans les domaines de l'informatique que de la gestion des eaux, des concepts quant à la manière de structurer et de développer des **systèmes informatisés d'aide à la gestion intégrée des eaux en milieu urbain**, que nous appelons **SIAGEM** dans la suite de ce rapport, sont mis sur pied.
- des prototypes sont développés, avec pour objectif de tester et de valider les différentes propositions,

ainsi que de servir de base de discussion avec les intervenants tant des domaines de l'informatique que de la gestion des eaux.

- enfin, une fois les différents concepts vérifiés et validés, une proposition est rédigée.

S'agissant d'un projet exploratoire, la démarche proposée est itérative : les discussions avec les intervenants, les études de la littérature disponible et les conclusions auxquelles conduisent les prototypes impliquent de fréquents va-et-viens entre les étapes qui viennent d'être décrites.

1.4 Plan du rapport

Ce rapport s'attache successivement aux aspects suivants : analyse de la situation, élaboration et proposition d'une solution, description d'une méthode puis d'un cas d'application des concepts proposés, de manière à en démontrer la validité.

Le **chapitre 2** définit précisément la notion de gestion des eaux en milieu urbain, et montre quelles en sont les évolutions actuelles. La manière dont cette gestion est organisée en Suisse, ainsi que les acteurs impliqués dans cette problématique, font l'objet d'une analyse. Enfin, une typologie des outils logiciels utilisés est établie, et ces derniers, leurs utilisations, avantages et inconvénients, font l'objet d'une discussion.

Se basant sur la littérature disponible ainsi qu'une série d'entretiens menés avec des intervenants du domaine de la gestion des eaux, le **chapitre 3** s'attache à décrire les responsabilités qui doivent être assumées par un SIAGEM, ainsi que les contraintes qui doivent être prises en compte lors de la conception d'un tel outil.

Dans le **chapitre 4**, la solution proposée - à savoir baser le développement de SIAGEM sur une approche modulaire - est ébauchée, puis les technologies logicielles disponibles actuellement et présentant une utilité dans ce contexte sont discutées : paradigme orienté objet, approche par composants logiciels, objets et composants métier. Les particularités liées au domaine de la gestion des eaux sont intégrées à la réflexion. La solution proposée est décrite de manière détaillée. Enfin, il est montré comment les contraintes identifiées au chapitre 3 peuvent, grâce à l'approche proposée, être aisément prises en compte. Précisons dès à présent que nous proposons de concevoir les SIAGEM par assemblage de composants logiciels pré-programmés et réutilisables.

Le **chapitre 5** propose, après analyse des pratiques actuelles dans le domaine du génie logiciel, une méthode de développement de composants logiciels réutilisables adaptée au contexte de la gestion des eaux.

Le **chapitre 6** présente enfin le prototype d'un tel composant logiciel réutilisable. Ce dernier, développé dans le cadre de ce travail, a servi de support à la réflexion menée. La mise en oeuvre de ce composant au sein d'applications opérationnelles est présentée et discutée, et les conclusions qui en sont tirées permettent de valider l'approche proposée.

Une synthèse de ce travail et la présentation de quelques perspectives dans ce domaine clôturent ce rapport.

Gestion intégrée des eaux en milieu urbain

Ce premier chapitre s'attache à décrire les différents aspects de la gestion des eaux en zones urbaines : comment définit-on cette notion et quelles en sont les évolutions actuelles. Les intervenants dans ce contexte, leurs responsabilités, et les outils, en particulier informatiques, qu'ils utilisent dans le cadre de leur activité professionnelle, sont présentés. L'objectif est ici d'apporter au lecteur une vue d'ensemble du contexte de cette recherche.

2.1 Le système "Eaux en milieu urbain"

2.1.1 Origines des concepts actuels de gestion des eaux en milieu urbain

La manière de gérer les eaux dans les zones urbanisées de nos contrées remonte à l'épidémie de choléra qui décima l'Europe dans la première moitié du 19^{ème} siècle. En effet, bien que la manière dont se transmettait le choléra n'ait pas été à l'époque bien comprise, on pensait que cette maladie était due aux médiocres conditions d'hygiène qui régnaient alors. La mise à disposition des populations d'une eau de bonne qualité ainsi que l'assainissement des zones urbaines furent reconnus comme une nécessité, et dès lors des investissements considérables ont été consentis, d'une part pour la construction d'infrastructures et de réseaux d'approvisionnement en eau, et d'autre part pour la construction de réseaux d'assainissement chargés d'évacuer les eaux usées des centres urbains le plus rapidement possible.

Trois principes fondamentaux gouvernent depuis cette époque la gestion des eaux dans les agglomérations:

- l'eau destinée à l'approvisionnement des populations est prélevée sur l'environnement, soit sur les réserves d'eau souterraines ou superficielles, puis purifiée de manière centralisée dans des stations de traitement des eaux;
- la distribution de l'eau se fait par le biais de réseaux sous pression, de manière à prévenir sa contamination par des infiltrations du milieu extérieur;
- une fois utilisée pour les tâches domestiques et d'hygiène personnelle, l'eau est évacuée par un réseau

séparé. Elle est donc à cette occasion utilisée comme moyen de transport pour différents déchets. Ces eaux usées sont autant que possible traitées par des installations centralisées, les stations d'épuration, puis rejetées dans les eaux superficielles.

A noter que les réseaux d'assainissement sont généralement également utilisés pour évacuer les eaux de ruissellement ainsi que, souvent, des eaux claires (eaux de refroidissement, de fontaines, eaux superficielles canalisées, etc). On distingue, en assainissement urbain, les réseaux unitaires des réseaux séparatifs (figure 2). Dans les premiers, toutes les eaux sont conduites indistinctement par le biais d'un même réseau d'assainissement vers les stations d'épuration, tandis que les seconds ont pour objectif une séparation des eaux usées domestiques et industrielles des eaux de ruissellement et des eaux claires, ces dernières étant directement rejetées dans le milieu naturel sans traitement. Une approche de l'assainissement selon le principe du séparatif implique la coexistence de deux réseaux : un réseau pour les eaux usées et un réseau pour les eaux claires. Les réseaux séparatifs ont pour avantage d'éviter le mélange entre des eaux de relativement bonne qualité, soit les eaux de ruissellement et les eaux claires, avec les eaux fortement polluées que sont les eaux usées domestiques et industrielles. Le système séparatif conduit de plus à une diminution des rejets d'eaux usées non traitées dans les eaux superficielles par le biais des déversoirs d'orage, ainsi qu'à un meilleur fonctionnement des stations d'épuration, dont le fonctionnement est moins perturbé par des apports d'eaux peu polluées.

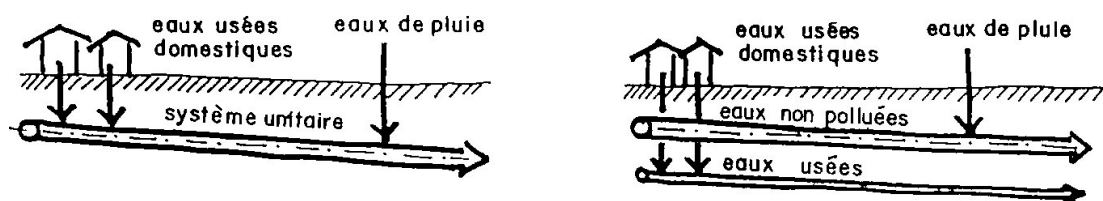


Figure 2. Réseaux unitaire, à gauche, et séparatif, à droite (VSA, 1992).

Les principes fondamentaux présentés ci-dessus, s'ils peuvent paraître évidents, ne sont pas toujours applicables, d'une part parce que la construction et la gestion d'infrastructures de traitement, de distribution et d'évacuation sont extrêmement coûteuses, et que des défaillances du système ne sont pas exclues, d'autre part parce que le fait d'utiliser les eaux comme moyen de transport des déchets implique une relative abondance de cette ressource, or celle-ci n'est, dans beaucoup de pays, pas disponible en quantité suffisante. Ainsi, on assiste ça et là à l'éclosion de foyers de maladies infectieuses transmises par l'eau, souvent dans des pays à économies faibles, mais aussi dans nos régions (Maystre, 1984).

Cette mission historique de la gestion des eaux en zones urbaines, qui est donc d'assurer de bonnes conditions d'hygiène aux populations, s'accompagne évidemment d'un objectif de protection de ces mêmes populations contre les risques d'inondations : les infrastructures urbaines doivent être conçues de manière à ce que les événements pluviaux exceptionnels n'entraînent ni pertes humaines ni dégâts matériels importants.

Si ces objectifs - assurer l'hygiène des centres urbains ainsi que l'approvisionnement en eau et la protection contre les crues des populations - ne sont à l'heure actuelle pas remis en question, la manière de les atteindre fait l'objet de nombreuses discussions, en particulier dans la perspective, à terme, d'un développement durable. Ces questions seront abordées dans les chapitres qui suivent. Ainsi sera

clairement défini ce que sous-tend l'expression "gestion des eaux en zones urbaines", puis seront discutés les problèmes auxquels se heurtent les praticiens, ainsi que les outils informatiques qui les aident dans leurs tâches.

2.1.2 Définition de la gestion des eaux

2.1.2.1 Les missions

La gestion de "quelque chose" implique une administration de ce "quelque chose" en fonction d'objectifs qui fournissent une ligne de conduite à l'action de gestion. Ces objectifs sont en général relativement faciles à définir, mais il n'en demeure pas moins qu'il est indispensable de les connaître. Les paragraphes qui suivent ont donc pour but de comprendre quelles sont les considérations qui devraient guider les actions de gestion des eaux en zones urbaines.

Celles-ci englobent de multiples activités dont il n'est pas aisé d'avoir une vue d'ensemble. Une première approche permettant d'en définir les contours serait de spécifier les missions qui doivent être assurées dans ce contexte (Larsen et Gujer, 1997; Niemczynowicz, 1999). A noter que la notion de gestion des eaux en zones urbaines doit généralement être considérée dans un cadre dépassant celui de l'agglomération, les milieux urbains, péri-urbains et ruraux étant intimement liés. Il suffit, pour s'en convaincre, de penser aux épandages de boues d'épuration, issues de l'assainissement des agglomérations, sur les terres agricoles, à l'utilisation potentielle d'eaux usées pour l'irrigation, ou encore aux conséquences d'accidents industriels sur la qualité des cours d'eau. De ce fait, dans la suite de ce travail, nous parlerons parfois simplement de "gestion des eaux", bien que notre intérêt soit plutôt focalisé sur le contexte et les problématiques spécifiques aux zones urbanisées. Ces missions attendues de la gestion des eaux en zones urbaines peuvent donc se définir comme il suit :

- ***Assurer l'approvisionnement en eau de boisson et destinée à l'hygiène personnelle***

La fourniture aux ménages d'une eau de qualité irréprochable leur permettant d'assurer leurs besoins en eau de boisson et de cuisine ainsi que leur hygiène personnelle constitue une mission primordiale de la gestion des eaux. Cette mission implique que des mesures doivent être prises pour préserver la quantité et la qualité des ressources en eau, entraînant donc la nécessité d'assurer la protection à long terme de ces dernières et de suivre leur évolution.

- ***Assurer l'hygiène des communautés urbaines***

La mission traditionnelle de l'assainissement urbain est l'évacuation la plus rapide possible des déchets et donc des eaux usées hors des centres urbains, ceci afin d'empêcher toute dissémination d'agents infectieux. Ceci implique que soit fournie l'eau servant au transport des matières polluantes vers l'extérieur des agglomérations. De l'eau destinée au nettoyage, que ce soit pour les ménages, l'industrie, les commerces, ainsi qu'à l'entretien des infrastructures urbaines, doit également être fournie. Il peut être envisagé, pour ceci, d'utiliser une eau de qualité moindre que celle utilisée pour la boisson ou dans des buts d'hygiène personnelle, raison pour laquelle cette mission fait l'objet d'une discussion distincte.

- ***Assurer la gestion des eaux pluviales et la protection contre les inondations***

Le développement des agglomérations conduit à une modification des cycles hydrologiques naturels, car de fortes proportions du territoire sont imperméabilisées, induisant de gros volumes de ruissellement lors de précipitations et empêchant l'infiltration des eaux pluviales au lieu de leur production. De plus, les concentrations de personnes et de biens que constituent les agglomérations font que même de petites inondations peuvent avoir des conséquences humaines et matérielles très graves. Une mission fondamentale de la gestion des eaux consiste donc à diminuer au maximum le risque d'inondation, par le biais de la maîtrise du ruissellement par temps de pluie (menée à bien, traditionnellement, par l'évacuation des volumes ruisselés par les réseaux d'assainissement). Notons que dans certaines agglomérations il est également nécessaire de drainer les eaux souterraines.

- ***Permettre à l'eau d'assurer un rôle récréatif au sein des agglomérations***

L'eau et les milieux aquatiques ont de tout temps été des éléments importants des paysages urbains. Les berges de rivières et de lacs servent fréquemment de lieu de récréation et de baignade, les fontaines et autres bassins contribuent à l'esthétique des espaces publics, et les piscines font partie des infrastructures sportives souhaitables. L'intégration de l'eau dans les villes contribue à une amélioration de l'habitat et constitue de ce fait une facette importante de la gestion des eaux. Le rôle de l'eau en tant qu'élément de loisir et de plaisir doit être pris en compte.

- ***Permettre la pratique de l'agriculture urbaine***

L'agriculture urbaine, bien que peu pratiquée à l'heure actuelle dans le monde occidental, est un phénomène très ancien, les populations urbaines ayant de tout temps cherché à améliorer leur quotidien en cultivant des fruits et légumes dans leur jardin. Avec le développement des agglomérations, en particulier dans le tiers-monde, cette pratique prend de plus en plus d'importance, et doit être favorisée car elle permet, en plus d'une augmentation de la qualité de vie, le recyclage des nutriments dans le périmètre de la ville. Elle doit cependant faire l'objet d'une grande attention car les eaux disponibles pour l'irrigation de telles cultures ne possède pas toujours les qualités adéquates et la consommation des produits issus de ce type d'agriculture peuvent présenter des risques pour la santé.

- ***Assurer la protection des milieux naturels***

L'impératif de protection des milieux naturels et des écosystèmes est implicitement contenu dans les missions présentées ci-dessus : la nécessité de préserver les ressources en eau, tant en terme de quantité que de qualité, va de pair avec une préservation des écosystèmes. De manière peut-être plus concrète, les activités humaines en général doivent permettre de maintenir les eaux superficielles et souterraines dans un état conforme à ce qu'exige la législation (Krejci et Schilling, 1992). L'une des missions de la gestion des eaux est alors de s'assurer que cet équilibre entre qualité des écosystèmes et bonne marche des activités humaines ne soit pas rompu, ou qu'il soit rétabli.

2.1.2.2 Systémique et gestion des eaux en milieu urbain

La diffusion de la notion de systémique (de Rosnay, 1977; Le Moigne, 1984) date du milieu du vingtième siècle. La systémique s'oppose aux principes analytiques de la démarche cartésienne, qui

postule "qu'il faut diviser les difficultés en autant de parcelles qu'il se peut pour mieux les résoudre" (Descartes, 1637; cité dans Gayte et al., 1996). En effet, la démarche systémique part du principe que les multiples interactions qui existent au sein de systèmes complexes conduisent à des effets qui ne sont pas prévisibles à partir de l'analyse des parties de ces systèmes prises isolément (Stephens et Hess, 1999), et postule de ce fait une analyse des systèmes dans leur ensemble, s'attachant notamment à bien définir la nature des liens qui existent entre les différents sous-systèmes.

De Rosnay propose la définition suivante de la notion de système :

"Un système est un ensemble d'éléments en interaction dynamique, organisé en fonction d'un but." (de Rosnay, 1977)

Cette définition est aisément compréhensible si l'on considère des entreprises ou d'autres types d'organisations humaines, qui constituent une catégorie de systèmes :

"Les organisations comme objet d'étude sont des entités sociales délibérément construites et reconstruites pour poursuivre des buts spécifiques. " (Noy, 1990; cité dans Gayte et al., 1996)

Le but d'une organisation est en principe clairement identifiable : une école a pour objectif de former des élèves, une entreprise privée a pour but de produire tels biens ou tels services et de les commercialiser de manière profitable, etc. Mais est-ce que les systèmes naturels ont un but ? Sont-ils des systèmes au sens systémique du terme ? Il peut être répondu par l'affirmative à cette question : en effet, deux comportements peuvent être mis en oeuvre pour qu'un système atteigne son but : le pilotage et la régulation. Le pilotage est un comportement actif, tandis que la régulation est réactive, se produit pour que le système garde un certain état. L'objectif des systèmes naturels peut être vu comme une volonté de conserver un certain état qui peut être qualifié d'équilibre. Les systèmes naturels sont donc des systèmes dont le but est atteint par régulation. Une approche systémique de la gestion des eaux implique que soit considéré de manière globale le système composé des sous-systèmes naturels, des organisations, des infrastructures construites et gérées par ces dernières, ainsi que des personnes qui vivent au sein de ce système. Ainsi, chaque système peut être considéré comme faisant lui-même partie d'un système plus étendu, et étant lui-même composé de sous-systèmes (Revaz et al., 2000; Stephens et Hess, 1999). De nombreux auteurs insistent sur ce fait : seule une gestion intégrée, soit suivant une approche systémique et donc tenant compte des différents sous-systèmes et des interactions entre ces derniers, est efficace dans le contexte des eaux en milieu urbain (Lundin et al., 1999; Schütze et al., 1996). Luiten note :

"Integrated water management means that all aspects that are related to water resources are treated in their full complexity and mutual interactions." (Luiten, 1995)

Quels sous-systèmes faut-il donc plus précisément considérer dans le cadre d'une gestion intégrée des eaux en zones urbaines ? Nous appellerons dans la suite de ce travail "système Eaux en milieu urbain" le système constitué : 1°) des systèmes naturels délimités par le cycle hydrologique naturel (eaux superficielles, eaux souterraines, précipitations, etc); 2°) des organisations interagissant avec ces systèmes naturels (organisations responsables de l'approvisionnement en eau, de la protection des eaux de surface, de l'assainissement des eaux usées, consommateurs d'eau et producteurs d'eau usée, etc), incluant les infrastructures construites ou gérées par ces organisations; ainsi que 3°) les consommateurs

de la ressource en eau, qui par leur comportement et les choix de société effectués, influencent grandement le comportement du système "Eaux en milieu urbain" (figure 3). Lundin (1999) donne de ce système une description particulièrement exhaustive :

"The urban water system should be seen as consisting not only of the material technical systems, with installations such as pipes, pumps and buildings, but also integrated urban activities, as a means to fulfil basic human needs. The user of the system should thus be included as an important part of the analysis as well as the organisation and technical functions needed to build, operate and maintain the production and transport of drinking water and collection and treatment of wastewater. The whole hydrological cycle provides a natural, if somewhat large boundary, for the urban water system. The system boundaries should also include catchment areas, water reserves, receiving waters and the use of nutrients, energy or other products from the treatment processes of chemical additives used in the treatment processes." (Lundin et al., 1999)

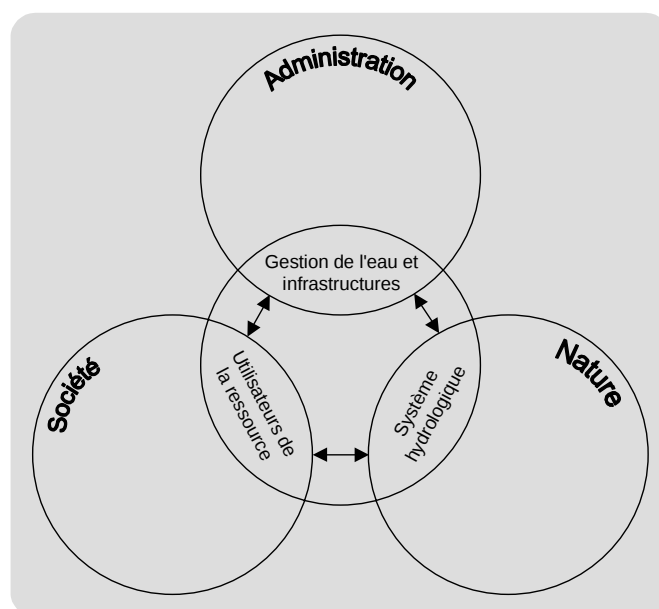


Figure 3. Eléments devant être pris en considération dans le cadre d'une gestion intégrée des eaux, d'après Wisserhof (1995).

La définition citée ci-dessus illustre bien la complexité de la tâche de gestion intégrée des eaux en zones urbaines, si tous ces aspects doivent être pris en compte. La figure 4 propose une vision schématique des interactions qui doivent être considérées pour une gestion intégrée des eaux.

Le système "Eaux en milieu urbain" est de plus un système complexe : en effet le niveau de complexité d'un système peut être appréhendé par le volume d'information nécessaire pour en assurer la description (Cohen et Steward, 1994; cités dans Geldof, 1997) ainsi que par le nombre d'interactions entre sous-systèmes, et assurément ceux-ci sont ici très élevés. Dans la mesure où une approche systémique peut être vue comme une méthode à même d'appréhender une telle complexité (Crausaz, 2000), ce paradigme peut nous servir de base. Cependant, il faut bien se rendre compte que l'approche systémique implique une bonne compréhension des sous-systèmes qui composent le système général, et qu'il est nécessaire de combiner différentes échelles de considération pour bien appréhender un système complexe (de

Rosnay, 1977).

Nous avons ici une première indication de la raison pour laquelle les outils informatiques jouent un rôle important dans le contexte de la gestion des eaux : de grandes quantités d'information doivent être collectées à un "micro-niveau" de manière à ce que les décisions adéquates puissent être prises à un "macro-niveau" (Korwing et Wiggers, 1999). En matière de gestion des eaux, cependant, une telle approche se heurte dans la pratique à différents problèmes : la quantité d'informations disponible est la plupart du temps insuffisante, et l'échange d'informations entre intervenants concernés est difficile, en raison des nombreuses structures organisationnelles qui interviennent dans ce contexte.

Les lignes qui suivent proposent une rapide description des sous-systèmes du système "Eaux en milieu urbain" et ont surtout pour but de montrer quelques exemples des nombreuses interactions qui existent entre ces sous-systèmes. Comme de nombreux ouvrages extrêmement détaillés sont disponibles relativement à ces différents sous-systèmes (Bourrier, 1985; Degrémont, 1990; Valiron et Affholder, 1994), l'objectif est ici uniquement de proposer aux lecteurs non familiarisés avec cette problématique une vue d'ensemble de cette dernière.

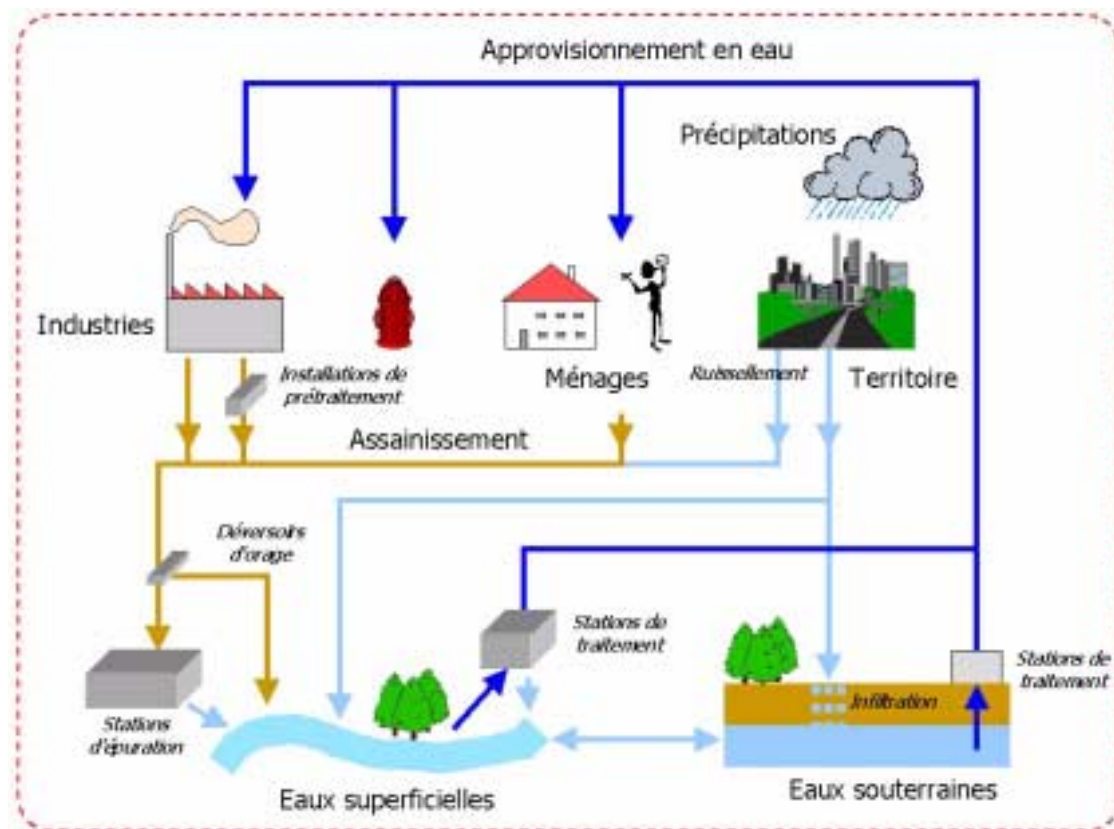


Figure 4. Vue simplifiée du système "Eaux en milieu urbain" et de ses différents sous-systèmes.

Le système hydrologique naturel

Le cycle de l'eau, abondamment décrit dans la littérature (Chow et al., 1998; Musy et Laglaine, 1992), met en contact différents sous-systèmes : l'atmosphère (nous parlerons plus spécifiquement des précipitations, dans la mesure où celles-ci constituent la composante du sous-système "atmosphère" qui

nous intéresse tout particulièrement), les eaux superficielles et les eaux souterraines. En zone urbaine, ce cycle est profondément modifié (Kaiser, 1997) : les quantités d'eau consommées dans le périmètre d'une agglomération sont nettement supérieures à celles qui y sont "produites" (figure 5); l'imperméabilisation de surfaces importantes conduit à une diminution de la recharge des nappes d'eau souterraines par le biais de l'infiltration, les importants volumes qui ruissellent conduisent à des augmentations des débits dans les eaux superficielles et les déversements d'orage par les réseaux d'assainissement conduisent à des dégradations qualitatives des cours d'eau. Mentionnons enfin que dans les zones urbaines, le type de couverture de sol influence les volumes d'eau qui ruissellent.

Le système d'approvisionnement en eau potable

Il comprend les canalisations du réseau d'approvisionnement, les diverses installations et ouvrages sur ce réseau (vannes, pompes, réservoirs, etc.), les ouvrages de prélèvement d'eau dans le milieu naturel ainsi que les stations de traitement des eaux. La figure 5 présente l'utilisation qui est faite de l'eau potable en Suisse. Des exemples d'interactions du système d'approvisionnement en eau potable avec des systèmes tiers sont : les prélèvements dans le milieu naturel (nappes phréatiques ou eaux de surface) peuvent conduire à une diminution non négligeable des ressources disponibles; la qualité des eaux prélevées conditionne le type de traitement que doivent subir les eaux (traitements légers, voire absence de traitement dans le cas de prélèvement dans des nappes souterraines de qualité (Lundin et al., 1997), traitements coûteux et sophistiqués des eaux puisées par exemple dans des lacs eutrophes ou pollués); des exfiltrations se produisent du réseau dans le sol; enfin, les trop-pleins des réservoirs se déversent généralement dans les réseaux d'assainissement.

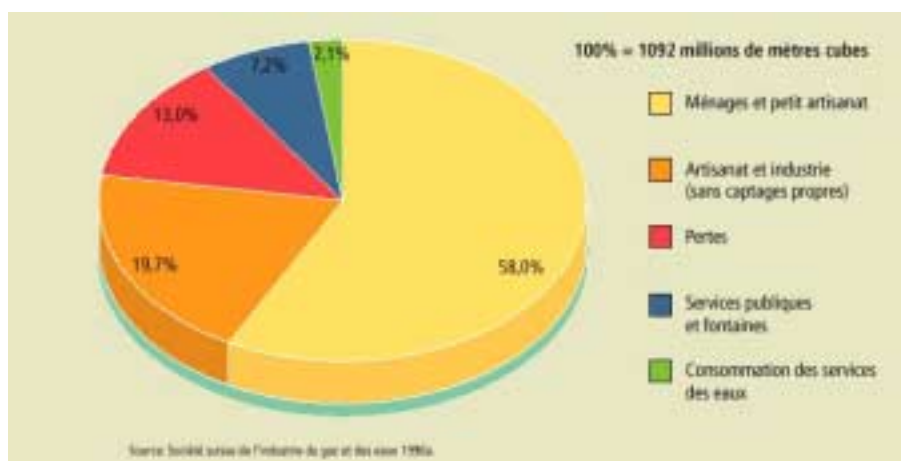


Figure 5. Consommation d'eau potable par secteurs en Suisse en 1995 (OFS/OFEFP, 1997).

Le système d'assainissement

Sa fonction est d'évacuer les eaux usées des localités afin de les rejeter dans le système hydrologique naturel (en général dans les eaux de surface), après un éventuel traitement dans des stations d'épuration des eaux usées (STEP). Les réseaux d'assainissement comportent de nombreux ouvrages (Bourrier, 1985): regards et chambres de visite, bassins de rétention, bassins de décantation, exutoires, déversoirs, canalisations, etc.). Des exemples d'interactions entre ce sous-système et les autres compartiments du système "Eaux en milieu urbain" sont : les déversements par temps sec (exutoires de STEP) et par temps

de pluie (déversoirs d'orage) dans le milieu naturel; les infiltrations d'eaux claires dans le réseau d'assainissement qui conduisent à des dilutions des eaux usées et surchargent inutilement les stations d'épuration et, à l'inverse, la contamination d'aquifères en raison d'exfiltrations d'eaux usées des canalisations (Mull, 1996); les épandages de résidus des processus d'épuration (boues d'épuration) sur des terres agricoles, etc.

Les ménages

La dimension sociologique ne peut pas être négligée lors de considérations relatives à la gestion des eaux. En effet, l'homme, de par ses habitudes et son comportement, influence énormément les volumes d'eau utilisés ainsi que la manière de les utiliser. De manière générale, les habitants des pays industrialisés sont de très gros consommateurs d'eau : à titre d'exemple, une douche nécessite 25 litres, un bain 150 à 200 litres, un lave-linge consomme 40 à 50 litres par machine (Durant-Dastes, 1992), et en Europe de l'Ouest, la consommation d'eau par les ménages variait en 1990 de 108 en Belgique à 264 litres par habitant et par jour en Suisse (SSIGE, 1992). Des modifications des habitudes des ménages peuvent avoir des conséquences importantes pour le système hydrologique. La suppression des phosphates dans les lessives par exemple est à l'origine d'une nette amélioration de la qualité des eaux de surface (figure 6).

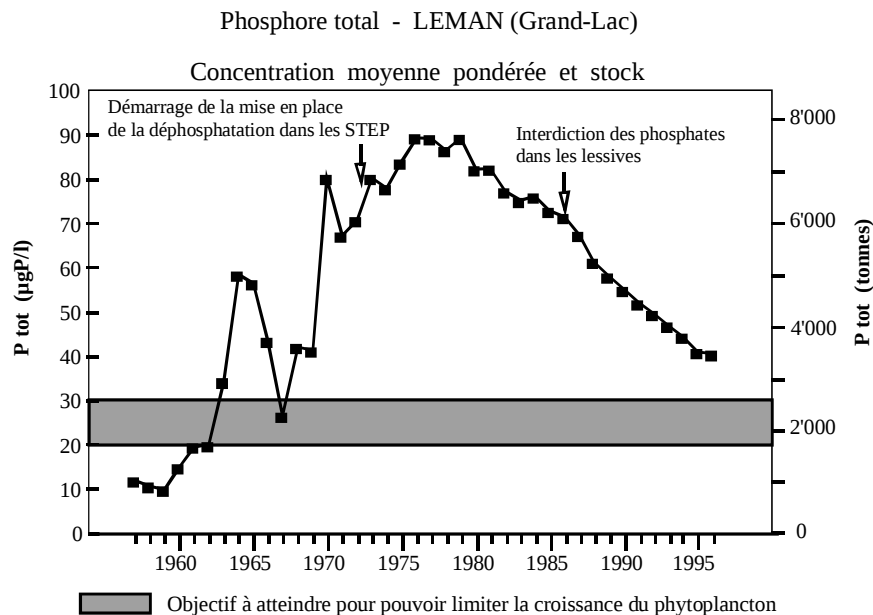


Figure 6. Evolution de la teneur en phosphore dans le lac Léman (CIPEL, 1997).

Les biens-fonds

Les biens-fonds sont les immeubles et propriétés qui composent le territoire urbain : la manière dont leur approvisionnement et leur assainissement sont conçus a un énorme impact sur le cycle de l'eau dans les agglomérations : c'est au niveau du bien-fonds que sont prises les mesures dites "à la source", qui sont destinées à diminuer les impacts négatifs des différentes pratiques actuelles en matière de gestion des eaux (réutilisation de l'eau, infiltration et rétention in-situ, etc.). C'est également au niveau du bien-fonds que se développent des pratiques d'agriculture urbaine.

Les entreprises, industries et commerces

Certains processus industriels impliquent la consommation d'énormes quantités d'eau, d'autres sont sources de risques de pollutions aiguës (zones d'entrepôt de produits toxiques) ou peuvent être responsables de pollutions chroniques (rejets réguliers de substances polluantes dans l'environnement). De manière générale, l'utilisation de l'eau pour les activités industrielles doit être considérée, et variait en 1990 en Europe de l'Ouest entre 166 en Belgique et 402 litres par habitant et par jour en Suisse (SSIGE, 1992). Des mesures peuvent être imposées aux consommateurs industriels d'eau (rétention, pré-traitements des eaux usées, recyclage, etc), mais doivent dans ce cas faire l'objet d'un suivi et d'un contrôle de la part des autorités.

Le territoire urbain et ses infrastructures

Les différentes infrastructures publiques, et en particulier les voies de communication, jouent également un rôle important dans le contexte de la gestion des eaux : en ce qui concerne les aspects négatifs, elles sont responsables de l'imperméabilisation de surfaces énormes et leur entretien nécessite la consommation de gros volumes d'eau (figure 7). Leur fonction de voies de communication les conduit cependant à être intégrées dans la réflexion relative à l'amélioration des processus de gestion.



Figure 7. Nettoyage des rues (Durant-Dastes, 1992).

2.1.3 La gestion des eaux et le développement durable

2.1.3.1 La notion de développement durable

La notion de développement durable trouve son origine dans le rapport du Club de Rome (Meadows et al., 1972), qui développe la thèse selon laquelle le développement de l'humanité ne peut se poursuivre indéfiniment, en raison d'une part des limites inhérentes aux charges que peut supporter la planète et d'autre part de l'épuisement de certaines ressources naturelles. En 1987, une commission des Nations Unies publie un rapport (World Commission on Environment and Development, 1987) communément appelé rapport Brundtland du nom de son auteur principal, qui définit clairement le concept de développement durable :

"[A sustainable development is] a development that fulfils the needs of the present generation, without compromising the ability of the future generations to fulfil their needs."

S'ensuit en 1992 une conférence internationale sur ce même thème, qui contribue fortement à la médiatisation du concept de développement durable : des travaux de cette conférence est issu le programme Action 21, plus connu sous son nom anglais : Agenda 21 (United Nations, 1992), qui présente un ensemble de préceptes ayant pour but de diriger le mode de croissance de l'humanité sur le chemin du développement durable. En ce qui concerne la gestion des eaux, l'Agenda 21 précise :

"Water resources must be planned and managed in an integrated and holistic way to prevent shortage of water, or pollution of water resources, for impeding development. Satisfaction of basic human needs and preservation of ecosystems must be the priorities."
(United Nations, 1992)

Depuis cette conférence (connue sous le nom de Conférence de Rio), ce concept de développement durable a été amplement développé et discuté (Rijsberman et Ven, 1999), et influence fortement la manière d'approcher les différentes problématiques. En effet le message délivré par ces différents événements est suffisamment clair et explicite pour être intégré par une forte proportion de la population (Varis et Somlyódy, 1997), et en particulier par les professionnels actifs dans le domaine de la gestion des eaux (Larsen et Gujer, 1997). Se référant aux ressources, la notion de développement durable implique que celles qui sont non renouvelables, comme les énergies fossiles, ne doivent pas être épuisées avant qu'une ressource de remplacement ne soit trouvée, et en ce qui concerne celles qui sont renouvelables, comme l'eau, il ne faut pas que leur consommation dépasse leur taux de renouvellement. Cependant, et de manière générale, les questions que soulève la notion de développement durable sont loin d'être résolues: les besoins des générations futures ne peuvent être identifiés à l'heure actuelle, puisqu'ils sont inconnus (Rijsberman et Ven, 1999); la notion de la charge maximale que peut supporter un écosystème est difficile à cerner; les indicateurs à prendre en considération pour évaluer la "durabilité" d'un système, ainsi que de son évolution quant à sa "durabilité", font l'objet de nombreux débats. En fait, le concept de développement durable constitue un principe éthique, difficile à mettre en oeuvre (Somlyódy, 1995).

Les objectifs de base de la gestion des eaux, soit l'éradication des maladies infectieuses transmises par l'eau, le contrôle des risques d'inondations dues au ruissellement urbain, ainsi que la maîtrise des impacts de pollutions localisées et à court terme, peuvent être considérés comme étant atteints dans le monde occidental. Cependant, le fondement même du système mis en place repose sur la disponibilité de grandes quantités d'eau qui sont utilisées pour transporter les déchets à l'extérieur des zones urbanisées. Une telle approche conduit irrémédiablement à une dissémination de substances, dont certaines sont polluantes, dans l'environnement, et donc à la dégradation de ce dernier, ainsi qu'à une diminution des ressources en eau (Butler et Parkinson, 1997). D'autre aspects (chapitre 2.1.3.2) montrent que la gestion des eaux, telle qu'elle est pratiquée à l'heure actuelle, comporte de nombreux indices de "non-durabilité". Il est donc nécessaire de reconsidérer toute l'approche de la gestion des eaux en zones urbaines, et de compléter les deux buts généraux de cette dernière - à savoir garantir la santé et la sécurité des communautés humaines, ainsi que protéger l'environnement - par un troisième : assurer la conformité de la gestion des eaux aux préceptes du développement durable.

Malheureusement, les moyens et techniques pouvant être appliqués et conformes à ces préceptes sont difficiles à déterminer (Chen et Beck, 1997; Larsen et Gujer, 1997; Luiten, 1995; Varis et Somlyódy, 1997). De fait, si la nécessité d'accorder la croissance avec le principe de développement durable a été largement acceptée par les professionnels dans ce domaine, la manière de le faire est, en raison des incertitudes qui persistent, l'objet d'âpres débats (Harremoës, 1997). Pour certains auteurs, la mise en place d'une gestion des eaux conforme aux principes du développement durable passe par une amélioration des technologies existantes, pour d'autres par la mise en oeuvre de nouvelles technologies. De manière générale cependant, l'idée d'induire un changement des paradigmes qui gouvernent la gestion des eaux fait son chemin (Langeveld et Wiggers, 1999). Les chapitres qui suivent ont pour objectif de montrer pourquoi un tel changement est nécessaire, et comment les objectifs en matière de durabilité peuvent être atteints.

2.1.3.2 Les causes de "non-durabilité" des pratiques actuelles

Préalablement à une remise en question de la gestion des eaux telle que pratiquée à l'heure actuelle, on peut s'interroger sur les facteurs qui font que cette dernière n'est pas conforme aux préceptes du développement durable. De manière générale, un mode de développement durable implique (Grottker et Otterpohl, 1996; Otterpohl et al., 1997) :

- des consommations en énergie et en matières plus faibles que celles constatées à l'heure actuelle;
- une utilisation des ressources en eau qui ne conduise ni à leur dégradation, ni à leur diminution;
- que les problèmes environnementaux rencontrés ne soient transférés ni dans le temps ni dans l'espace;
- que les activités humaines ne perturbent pas les cycles naturels;
- l'intégration de l'homme dans les réflexions relatives à la manière de tendre vers ce mode de développement durable.

Malheureusement, de nombreux auteurs ont identifié des points qui font que les pratiques actuelles en matière de gestion des eaux ne sont pas conformes à ces préceptes :

- le principe de l'utilisation de l'eau comme moyen de transport pour les déchets, sur laquelle s'appuie toute la philosophie actuelle en matière d'assainissement, est fortement remis en question, et ce pour plusieurs raisons (Butler et Parkinson, 1997). D'une part, une telle approche conduit à un gaspillage de nutriments, puisque les stations d'épuration ne parviennent à retenir, dans le meilleur des cas, que 80% de l'azote, 95% du phosphore et 90% du potassium contenu dans les eaux usées avant leur traitement (Otterpohl et al., 1997). Ces nutriments s'accumulent dans les eaux superficielles, dont les capacités d'auto-épuration sont dépassées, et s'y rendent responsables de dysfonctionnements (eutrophisation par exemple). Cet état de fait est d'autant plus dérangeant que la synthèse d'engrais azotés pour les besoins de l'agriculture nécessite de grandes quantités d'énergie et est très coûteuse (Grottker et Otterpohl, 1996), et que le phosphore constitue une ressource non renouvelable qui, au rythme actuel de consommation, sera épuisée d'ici environ 150 ans (Lundin et al., 1997). De plus, le transport des déchets par l'eau entraîne la pollution de grosses quantités d'eau de bonne qualité, alors que ces mêmes eaux font auparavant l'objet de coûteux traitements en vue de leur potabilisation (van Lier et Lettinga, 1999). De fait, le principe de l'utilisation de l'eau comme moyen de transport pour les déchets conduit à une interconnexion des cycles de l'eau et des cycles nutritifs contraire aux préceptes du développement durable.

- Les réseaux d'assainissement constituent des investissements très lourds, dont la durée de vie est certes élevée (de 50 à 100 ans), mais qui nécessitent une maintenance constante et coûteuse. Celle-ci n'est, dans de nombreux cas, pas ou mal assurée, ce qui conduit à des dysfonctionnements des dits réseaux, comme des exfiltrations d'eaux usées dans le milieu naturel, dont les effets sur la qualité des eaux souterraines ont été très bien mis en évidence par Trauth (1996), ou inversement, à des infiltrations d'eaux claires dans le réseau d'assainissement (Gustafsson et al., 1996). Enfin, la complexité des réseaux d'assainissement en font des ouvrages inflexibles, dont l'adaptation à de nouvelles conditions est très difficile (Larsen et Gujer, 2000).
- L'extraction des matières organiques et la minéralisation des composés azotés par des stations d'épuration centralisées nécessite de grosses quantités d'énergie. De plus, la présence de polluants, en particulier de métaux lourds, dans les boues résultant de l'épuration des eaux usées par les stations d'épuration, rend la valorisation agricole de ces dernières problématique. Enfin, les rejets de stations d'épuration dans les rivières, même s'ils ont été épurés, contiennent toujours une importante charge polluante (Michelbach, 1996).
- La présence de nombreux micro-polluants a été constatée dans les eaux naturelles : métaux lourds bien sûr, mais aussi micro-polluants organiques (Larsen et Gujer, 2000), dont les impacts sur l'environnement sont parfois avérés, d'autres fois inconnus.
- Les déversements par temps d'orage, ainsi que les dysfonctionnements des déversoirs (Krebs et Larsen, 1997), constituent une source de pollution des milieux naturels par les eaux usées (Langeveld et Wiggers, 1999; Valiron et Affholder, 1994) insidieuse car difficile à mettre en évidence (Merz et Gujer, 1997).
- L'évacuation des eaux de ruissellement par les réseaux d'assainissement constitue un facteur de pollution des eaux naturelles. En effet les eaux de ruissellement elles-mêmes sont fortement chargées en polluants (Mottier et Boller, 1992), et en particulier en métaux lourds et en hydrocarbures, qui sont disséminés dans l'environnement (Boller, 1997; Rossi, 1998; van der Graaf et al., 1997).
- L'infiltration artificielle des eaux de ruissellement, technique de contrôle à la source recommandée à l'heure actuelle, peut conduire à la contamination des nappes souterraines et des sols par les métaux lourds et autres polluants (Mikkelsen et al., 1996).
- La dégradation qualitative des eaux naturelles implique que des traitements de plus en plus sophistiqués, et donc de plus en plus coûteux, doivent être utilisés pour la production d'eau potable.
- Le principe de l'évacuation des eaux polluées par un réseau d'assainissement souterrain et leur traitement décentralisé, loin des producteurs d'eaux usées, ajouté à l'invisibilité des dégradations, en particulier des ressources en eau souterraines (Lawrence et al., 1999), ne facilite pas la responsabilisation des différents acteurs et la prise de conscience de la population vis-à-vis des problématiques de gestion des eaux.
- Les populations toujours plus nombreuses et le mode de vie moderne font que les quantités d'eaux consommées par les ménages sont très importantes (chapitre 2.1.2.1). Cela entraîne une baisse du niveau des nappes d'eau souterraines, y compris dans les pays à forte pluviosité comme la Suisse (Krebs et Larsen, 1997), qui elle-même peut conduire à une diminution des débits d'étiage dans les cours d'eau et donc avoir des influences négatives sur la flore et la faune de ces derniers.

Il ressort de ces quelques points que les pratiques actuelles dans le domaine de la gestion des eaux ne sont pas conformes aux préceptes du développement durable, par le fait que les différents cycles (cycles

des nutriments, cycle de l'eau) sont perturbés, que les ressources en eau diminuent en raison de la trop grande consommation et sont altérées par l'introduction de composants polluants, que les techniques actuelles sont consommatrices de grandes quantités d'énergie, et que les populations ne sont pas assez impliquées dans la problématique. Dans ces conditions, de nombreuses alternatives sont discutées, expérimentées ou mises en oeuvre de manière opérationnelle en différents endroits. De plus, des approches plus radicales, basées non plus uniquement sur des améliorations des techniques actuelles mais aussi sur un changement des paradigmes sous-jacents, font actuellement l'objet de réflexions.

2.1.3.3 Une nouvelle approche de la gestion des eaux en milieu urbain ?

Jusqu'à une époque récente, les améliorations proposées dans le contexte de la gestion des eaux concernaient essentiellement les aspects technologiques - amélioration des processus techniques et des performances dans le domaine du traitement et de l'épuration des eaux en particulier - sans remise en question des fondements mêmes des pratiques en vigueur. Constatant qu'une telle approche est, à terme, vouée à l'échec, de plus en plus de voix s'élèvent qui proposent un changement des paradigmes qui gouvernent la gestion des eaux : alors que jusqu'à récemment les solutions proposées étaient essentiellement de type "traitement à l'exutoire" ("end-of-pipe"), il est à présent préconisé de favoriser autant que faire se peut les mesures de contrôle à la source ("source control"), soit près du lieu de production ou d'émission (Geldof, 1995; Langeveld et Wiggers, 1999; Payne et Gardiner, 1996; van Lier et Lettinga, 1999). Parmi les mesures proposées, on peut citer :

- les économies d'eau potable. Les quantités d'eau utilisées par les systèmes de chasses d'eau des toilettes par exemple, qui sont proportionnellement énormes (figure 8) et se montent à 10 à 15 litres lors de chaque utilisation (Somlyódy, 1995), peuvent être ramenées à moins de 10 litres par jour et par personne avec des systèmes appropriés (Otterpohl et al., 1997).
- l'utilisation d'eaux de moindre qualité (par exemple des eaux de ruissellement de toit) pour certaines tâches ne nécessitant pas l'utilisation d'une eau potabilisée (pour le rinçage des toilettes, l'arrosage des jardins, le lavage du linge par exemple).
- la mise en oeuvre de techniques d'infiltration "in situ" et de rétention au niveau du bien-fond des eaux de ruissellement, ce qui diminue la surcharge des réseaux d'assainissement, permet une diminution des déversements d'orage et une amélioration des performances des stations d'épuration (Sieker, 1998).

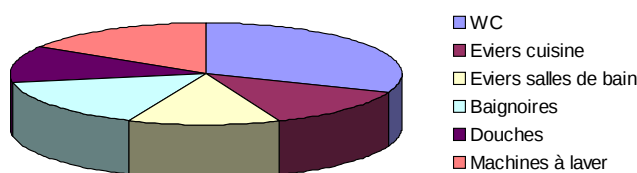


Figure 8. Répartition des volumes d'eau rejetés au réseau d'assainissement en fonction de leur utilisation. Chiffres basés sur une étude effectuée en Angleterre, relative à un quartier résidentiel. Rejets moyens totaux par habitant et par jour : 102.3 litres (Almeida et al., 1999).

- l'établissement de contraintes claires ou d'autres incitations en ce qui concerne la qualité et les quantités des eaux qui peuvent être rejetées dans le réseau d'assainissement ou les eaux

superficielles. De telles mesures responsabilisent les ménages et industries et les incitent donc à contrôler leurs rejets (Larsen et Gujer, 2000).

- il est suggéré que les polluants dont le traitement n'est pas envisageable de manière décentralisée soient déversés dans le réseau d'assainissement à des moments bien précis, de manière à harmoniser leur réception au niveau des installations centralisées, ceci en vue d'un traitement plus efficace. Une telle approche est proposée par Larsen et Gujer (1996), qui suggèrent ainsi de stocker les urines près de leur lieu de production et de profiter des périodes nocturnes, durant lesquelles les réseaux d'assainissement sont peu sollicités, pour les évacuer et par la suite les recycler (engrais azotés).

De telles approches tendent vers une décentralisation des infrastructures d'assainissement et une responsabilisation des différents intervenants. Elles ne comportent cependant pas que des avantages (Payne et Gardiner, 1996) : la décentralisation des infrastructures ne facilite ni leur maintenance, ni leur contrôle; les impacts réels de ces alternatives sur l'environnement, ainsi que leurs performances à long terme sont encore largement inconnus (installations d'infiltration des eaux pluviales par exemple). De plus, s'agissant de technologies peu matures, leurs bases de dimensionnement ne sont pas encore éprouvées.

L'adaptation de la gestion des eaux urbaines aux préceptes du développement durable, combinée à la prise en compte de la notion de systémique et donc des multiples interactions qui existent entre les différentes composantes du système "Eaux en milieu urbain", rendent la tâche des gestionnaires de l'eau beaucoup plus complexe qu'elle ne l'était auparavant, et impliquent de nombreux intervenants aux compétences et intérêts très divers. Les citations suivantes, clôturant ce chapitre, illustrent bien la situation actuelle dans le domaine de la gestion des eaux en milieu urbain :

"We begin to understand that the water future will not be so simple as it was in the past : not only pipes and treatment plants, but a great variety of technologies possibly composed of all these ingredients we know now and talk about now plus other solutions we even do not dream about yet, will be necessary to achieve present goals." (Niemczynowicz, 1999)

"The future challenges within urban water management during the next decades will be to organize cross-sectorial cooperation between several actors in order to introduce innovative water technologies, management systems and institutional arrangements which are able to meet the multiple objectives of equity, environmental integrity and economic efficiency, simultaneously maintaining or/and providing high level of water services for urban residents." (Niemczynowicz, 1999)

2.2 L'organisation de la gestion des eaux en Suisse

2.2.1 Le système politique suisse et la gestion des eaux

La Confédération suisse est une fédération de cantons et de demi-cantons (20 cantons et 6 demi-cantons en 1999) qui possèdent une grande autonomie politique et administrative. Ces cantons et demi-cantons comprennent eux-mêmes un certain nombre de communes (2912 communes au total en 1995)

relativement indépendantes. L'organisation de la gestion des eaux suit cette organisation hiérarchique à trois niveaux - Confédération, cantons, communes - et le partage des tâches entre ces différents échelons est spécifié par les différents textes légaux en vigueur. La Constitution, base de la législation fédérale, détermine les domaines qui sont du ressort direct de la Confédération - politique étrangère, affaires militaires par exemple - et ceux pour lesquels la Confédération émet des textes de loi mais en délègue l'application aux cantons - droit de l'environnement par exemple - (OFS/OFEFP, 1997). Ces derniers peuvent compléter cette législation fédérale par une législation cantonale, et, selon le même principe, attribuer aux communes des compétences dans un domaine ou l'autre. A chaque niveau politique, on retrouve donc des textes législatifs, qui peuvent être une constitution, des textes de loi ou des textes d'application (ordonnances). Ces textes peuvent préciser ceux qui sont promulgués au niveau politique supérieur, mais ne doivent pas y contrevenir. L'incitation des administrations cantonales, respectivement communales, à remplir les prérogatives de la Confédération passe par l'octroi de subventions et d'échéanciers (Crausaz, 2000).

Cette hiérarchie de niveaux décisionnels influence donc logiquement l'approche qui prévaut en matière de gestion des eaux. La répartition des tâches sur trois niveaux hiérarchiques complique les processus de gestion. De plus, les entités responsables de la gestion des eaux couvrent des périmètres qui ne cadrent généralement pas avec la réalité hydrologique : le territoire d'une commune ne correspond généralement pas à un bassin versant hydrologique, un réseau d'assainissement dessert parfois plusieurs communes, peut même franchir dans certains cas la frontière nationale (dans le Canton de Genève, par exemple, les eaux usées de certains villages situés en France voisine sont traitées par les stations d'épuration genevoises). Des entités organisationnelles transversales sont donc parfois mises en place dans le contexte de la gestion des eaux : ainsi, les syndicats d'assainissement regroupent des communes qui souhaitent exploiter en commun une station d'épuration. Les tâches et prérogatives en matière de gestion des eaux sont donc disséminées entre cinq niveaux hiérarchiques complémentaires (figure 9).

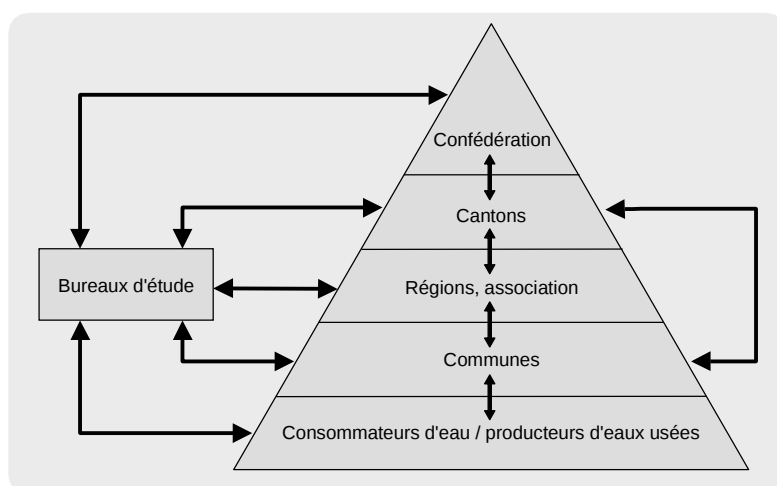


Figure 9. Les niveaux organisationnels intervenant dans la gestion des eaux. Les flèches représentent les flux d'information. Adapté de AGW (1997).

Cette situation occasionne des flux d'information entre ces différents niveaux. L'échelon régional, qui est le plus proche de la réalité hydrologique, n'est pas systématiquement présent : à l'heure actuelle, le flux d'information a encore tendance à circuler directement de commune à canton, bien que l'interlocuteur

régional s'impose peu à peu (AGW, 1997). De nombreuses tâches sont déléguées par les niveaux hiérarchiques précédemment cités à des organismes privés (bureaux d'ingénieurs, d'aménagement du territoire, d'écologie,...), impliquant à nouveau des transferts d'information des organisations politiques à ces acteurs. Enfin, il ne faut pas oublier qu'à un même niveau organisationnel, les catégories d'intervenants sont nombreuses, car les différentes facettes relatives à la gestion des eaux (approvisionnement, traitement, épuration, transport, etc) ont longtemps été considérées indépendamment les unes des autres, et placées sous la responsabilité de groupes et d'organisations différentes.

2.2.2 Les bases législatives de la gestion des eaux

L'article 76 de la Constitution fédérale constitue la base de la législation relative à la gestion des eaux :

"[...] la Confédération pourvoit à l'utilisation rationnelle des ressources en eau, à leur protection et à la lutte contre l'action dommageable de l'eau [...] Elle légifère sur la protection des eaux, sur le maintien de débits résiduels appropriés, sur l'aménagement des cours d'eau [...] Les cantons disposent des ressources en eau [...]" (Const. Féd., art. 76)

Cet article présente l'eau comme une ressource qu'il s'agit de protéger, mais qui peut aussi être source de danger (inondations par exemple) dont il faut se prévenir. Deux autres concepts sont introduits qui revêtent un grand intérêt relativement à la gestion des eaux : l'article 73, qui introduit la notion d'équilibre durable, et l'article 74, qui instaure le principe du pollueur-payeur dans le contexte général de la protection de l'environnement.

La loi fédérale sur la protection des eaux du 24 janvier 1991 (LEaux) traite des différents aspects légaux qui doivent être pris en compte dans le cadre d'une gestion des eaux en zones urbaines. Son premier article en présente les objectifs :

"La présente loi a pour but de protéger les eaux contre toute atteinte nuisible. Elle vise notamment à :

- a. préserver la santé des êtres humains, des animaux et des plantes;*
- b. garantir l'approvisionnement en eau potable et en eau d'usage industriel et promouvoir un usage ménager de l'eau;*
- c. sauvegarder les eaux piscicoles;*
- d. sauvegarder les biotopes naturels abritant la faune et la flore indigènes;*
- e. sauvegarder les eaux en tant qu'élément du paysage;*
- f. assurer l'irrigation des terres agricoles;*
- g. permettre l'utilisation des eaux pour les loisirs;*
- h. assurer le fonctionnement naturel du régime hydrologique." (LEaux, art. 1)*

Cette loi a pour objectif de préserver quantitativement les ressources en eau, en limitant et en contrôlant les prélèvements dans les eaux de surfaces et les eaux souterraines. Elle s'attache également à préserver leur qualité, en demandant aux cantons la délimitation de différentes catégories de surfaces (secteurs, zones et périmètres de protection des eaux) dans lesquelles des mesures particulières doivent être appliquées, et édicte des réglementations relativement à l'aménagement du territoire de ces surfaces. Elle oblige au traitement des eaux polluées, et à l'évacuation, si possible par infiltration, sinon dans les eaux

de surfaces, des eaux non polluées; elle contraint au raccordement au réseau d'assainissement tous les bâtiments pour lesquels un tel raccordement est possible, les autres devant assurer le traitement de leurs eaux usées par des installations de traitement autonomes. Elle légifère en ce qui concerne le stockage, l'évacuation et le traitement des liquides présentant un risque de pollution des eaux. Enfin, elle limite les interventions sur les cours d'eau.

Deux points doivent être relevés dans la LEaux : premièrement, elle cherche à minimiser les perturbations sur les cycles hydrologiques naturels, par exemple en favorisant l'infiltration des eaux non polluées, en demandant que soient régulés les écoulements en cas de forts débits, en limitant les prélèvements dans les eaux de surface et souterraines. Deuxièmement, elle cherche à refermer les cycles de nutriments (par exemple *"Toute exploitation pratiquant la garde d'animaux de rente s'efforce d'équilibrer le bilan des engrais"*, LEaux art. 14), ce qui va dans le sens d'une gestion durable des ressources.

La loi fédérale du 21 juin 1991 relative à l'aménagement des cours d'eau doit ici également être citée : elle traite de la protection contre l'action dommageable de l'eau, et donc de la protection contre les crues, qui constitue également un élément de la gestion des eaux en zones urbaines. Enfin, la loi fédérale sur la protection de l'environnement (LPE) du 7 octobre 1983 traite, entre autres, des substances dangereuses pour l'environnement, et donc pour les eaux.

Ces lois sont appuyées par des ordonnances, qui en précisent les contours et en règlent les détails techniques (Tercier, 1989) :

- L'ordonnance sur la protection des eaux (OEaux) du 28 octobre 1998 précise les points abordés par la LEaux (évacuation des eaux polluées, construction et exploitation d'installations d'évacuation et d'épuration des eaux, élimination des boues d'épuration, ...) et introduit les notions, très importantes dans le contexte de ce travail, de plan régional d'évacuation des eaux (PREE) et de plan général d'évacuation des eaux (PGEE). Il s'agit de deux instruments de planification de l'évacuation des eaux, le PGEE *"garantissant dans les communes une protection efficace des eaux et une évacuation adéquate des eaux en provenance des zones habitées"*, et le PREE ayant pour objectif d'assurer *"une protection efficace des eaux dans une région limitée formant une unité hydrologique"*. Ces deux instruments, qui se complètent, abordent le problème à deux échelles différentes, le PGEE à celle de la commune et le PREE à celle du bassin versant. Le PREE doit permettre d'harmoniser les mesures prises au niveau des PGEE. PGEE et PREE constituent un pas important vers une gestion intégrée des eaux.
- L'ordonnance sur l'aménagement des cours d'eau (OACE) du 2 novembre 1994 traite principalement des indemnités et aides financières de la Confédération en ce qui concerne les aménagements de cours d'eau, ainsi que de la répartition des tâches entre Confédération et cantons dans ce contexte.
- L'ordonnance sur la protection des eaux contre les liquides pouvant les polluer (OPEL) du 1^{er} juillet 1998 s'applique aux installations d'entreposage, aux places de transvasement, aux installations d'exploitation et aux *"circuits qui prélèvent ou rejettent de la chaleur dans les eaux, le sol ou le sous-sol"*. Son article 12 précise que les cantons doivent tenir un registre des installations et son article 17 précise qu'ils doivent assurer une surveillance des travaux de révision menés par les entreprises.
- Enfin, bien que les aspects relatifs à la pollution des eaux soient principalement traités dans l'OEau, certains aspects de l'ordonnance sur les substances dangereuses pour l'environnement (OSubst) du 9

juin 1986 peuvent avoir un impact dans le contexte de la gestion des eaux et de la mise en place de systèmes d'information s'y rapportant.

Les différents textes de lois contiennent des précisions quant aux responsabilités des différents intervenants et au partage des tâches entre ces derniers. De manière générale, dans le contexte de la gestion des eaux, la Confédération édicte les lois, fixe des objectifs (par exemple sur les exigences auxquelles doivent satisfaire les eaux superficielles, les eaux usées introduites dans les réseaux publics d'assainissement, sur les conditions nécessaires à l'obtention de permis de construire, etc.), les cantons étant responsables de leur application. La Confédération surveille l'application de la loi par les cantons, et en encourage la mise en oeuvre. Les cantons sont tenus de mettre en place un service de protection des eaux, ainsi qu'une police de protection des eaux et un service d'intervention en cas d'accident (LEau art. 49). Ils sont également chargés d'effectuer les différents relevés et autres inventaires nécessaires à l'application de la loi. Les différentes lois et ordonnances cantonales reprennent la législation fédérale, et définissent à leur tour leurs rapports et le partage des tâches avec les communes. Les cantons disposent également de moyens financiers d'incitation des communes à appliquer la législation relative à la gestion et à la protection des eaux.

Ces contraintes légales associées aux différents niveaux de gestion des eaux d'une part impliquent la gestion de gros volumes de données, et d'autre part entraînent des échanges d'informations entre les différents intervenants. C'est précisément dans ce contexte que les technologies de l'information ont un rôle à jouer et peuvent contribuer de manière significative à une amélioration des processus de gestion qui ont cours à l'heure actuelle.

2.2.3 Les intervenants dans le contexte de la gestion des eaux

L'organisation politique suisse et la législation relative à la gestion des eaux déterminent les acteurs institutionnels destinés à jouer un rôle dans ce contexte (chapitre 2.2.1) : la Confédération, les cantons, les régions, associations ou syndicats d'assainissement, ainsi que les communes.

D'autres catégories d'institutions jouent également un rôle dans le contexte de la gestion des eaux. Les organismes non gouvernementaux, de type association de protection de la nature, ont pour objectifs de sensibiliser les populations aux différents problèmes de dégradation de l'environnement en général, et des ressources en eau en particulier, ainsi que de prévenir et de dénoncer les atteintes au milieu naturel. Les associations d'usagers ont pour but la défense d'intérêts particuliers (associations de pêcheurs, de riverains de cours d'eau,...). Les associations professionnelles actives dans le domaine de l'eau, comme l'Association suisse des professionnels de la protection des eaux (VSA) ou la Société Suisse de l'industrie, du gaz et des eaux (SSIGE) ont pour but la formation continue de leurs membres, l'amélioration des techniques utilisées dans leurs domaines respectifs, et la vulgarisation des connaissances issues de la recherche. Enfin, différentes institutions de recherche mènent des programmes dans le but d'améliorer les pratiques et technologies dans le domaine de la gestion des eaux. Ces institutions sont soit subordonnées à l'échelon cantonal, comme les universités, soit dépendantes de la Confédération, comme les Ecoles polytechniques fédérales et les centres de recherche fédéraux tel l'Institut fédéral pour l'aménagement, l'épuration et la protection des eaux (EAWAG).

Dans le secteur privé, de nombreuses catégories professionnelles sont concernées par la gestion des

eaux: d'une part des intervenants sont chargés par les institutionnels de mener à bien des tâches en relation avec la gestion des eaux (bureau d'ingénieurs chargé de la réalisation d'un PGEE, entreprise chargée de la renaturation d'un cours d'eau, etc). D'autre part, certains intervenants, de par leurs activités professionnelles, sont responsables d'impacts sur le système des eaux en milieu urbain (industriel produisant des eaux usées ou gros consommateur d'eau, supermarché construisant des parkings de vaste capacité et par là imperméabilisant de grandes surfaces de terrain, etc.). Enfin, les particuliers, de par leur comportement et leur mode de vie, font partie intégrante du système à prendre en considération.

2.3 Analyse des pratiques actuelles en matière de gestion des eaux en milieu urbain

2.3.1 Procédure

L'objectif de la présente recherche est de proposer des concepts permettant de construire des systèmes informatisés d'aide à la gestion des eaux en zones urbaines. Nous avons vu qu'une telle gestion, pour être efficace, doit adopter une approche intégrée, encouragée par la législation suisse, et s'orienter dans une perspective de développement durable. Les outils informatiques développés doivent faciliter cette approche intégrée, et donc constituer un champ d'interaction entre les différentes catégories d'intervenants au niveau de leurs besoins. Afin de fonder les concepts développés sur une base solide, il est nécessaire d'appréhender non seulement qui sont les intervenants, mais aussi quels sont leurs objectifs généraux, quels sont leurs rôles, quels sont les processus de gestion qu'ils poursuivent, et quelles sont leurs motivations. C'est pourquoi des entretiens ont été menés avec des représentants de différentes catégories d'intervenants susceptibles d'utiliser un tel outil dans le cadre de leur activité professionnelle. Les objectifs de ces entretiens étaient les suivants :

- identifier les objectifs généraux de la structure organisationnelle représentée par l'interviewé, et identifier et comprendre les responsabilités de ce dernier;
- identifier et comprendre les différentes facettes de la gestion des eaux en zones urbaines;
- connaître les moyens informatiques mis en oeuvre par les interviewés dans le cadre de leur activité professionnelle, en cerner les points forts et les faiblesses;
- connaître les aspirations et vœux des interviewés relativement aux outils informatiques.

En tout, trente entretiens ont été menés, principalement par l'auteur de ce rapport lui-même, pour partie dans le cadre d'un travail de recherche effectué sous sa direction (Benagli, 1997). Ces entretiens ont porté sur vingt-quatre intervenants, certains d'entre eux ayant participé à plusieurs entretiens. Le tableau qui suit présente les différentes catégories d'intervenants rencontrés. Les personnes interrogées sont issues de différentes structures (cantons, communes, bureaux d'étude) de tailles différentes, ceci afin que les conclusions de la recherche soient affranchies de particularités locales. De ce fait, des redondances entre les responsabilités de certaines catégories d'intervenants ne sont pas exclues, puisque ces responsabilités font l'objet de répartitions parfois différentes suivant le canton, la commune, en fonction des caractéristiques de ces derniers (taille et nombre d'habitants principalement).

Le tableau 1 donne un premier aperçu des intervenants ayant été interviewés dans le cadre de cette

recherche. Une approche beaucoup plus formelle pour caractériser ces intervenants figure au chapitre 5.

Intervenant	Niveau organisationnel	Description des responsabilités
Chercheur dans une institution de recherche	fédéral	Mène des recherches sur des sujets trop complexes pour être prises en charge par des entreprises privées. Etudie le comportement d'infrastructures d'assainissement novatrices. Effectue pour cela des simulations de comportement des écoulements et du transport de composés polluants.
Directeur du contrôle de l'assainissement	cantonal	Dirige les services cantonaux qui ont pour mission de contrôler la conformité des différentes infrastructures avec la législation relative à la protection des eaux. Est responsable du cadastre des industries, du contrôle des raccordements, des citernes, des installations d'assainissement autonomes, de l'attribution des autorisations de raccordement et du maintien du service d'intervention en cas d'urgence. Examine les différentes demandes d'autorisation (raccordement au réseau, construction) sous l'angle de la protection des eaux.
Responsable du contrôle de la qualité des eaux de surface	cantonal	Organise le contrôle de la qualité des eaux de surface et du fonctionnement des stations d'épuration. En cas de problèmes, incite les communes à prendre les mesures nécessaires.
Responsable des travaux de l'assainissement	cantonal	Assure la gestion des interventions sur le réseau d'assainissement cantonal, suit les chantiers et s'occupe du financement et du contrôle de la bonne exécution des travaux. Contrôle les raccordements au réseau d'assainissement.
Hydrologue	cantonal	Assure la gestion des eaux de surface. En contrôle les débits et niveaux. Examine les demandes de prélèvement et gère les prélèvements autorisés. Procède à l'acquisition de données hydrologiques et assure leur pérennité.
Géologue	cantonal	Assure la gestion des nappes d'eau souterraines. Gère les demandes de prélèvement dans les eaux souterraines. Contrôle le niveau et la qualité des eaux souterraines. Procède à l'acquisition et à la gestion de données géologiques.
Gestionnaire des eaux	cantonal	Met en place la politique de gestion des eaux, et planifie les évolutions à long terme, selon une approche intégrée.
Responsable métrologie de l'assainissement	cantonal	Organise des campagnes de mesure et effectue des mesures de débit et de qualité aux points névralgiques du réseau d'assainissement dans le but de quantifier les rejets dans le milieu naturel. Assure la gestion des données acquises.
Ingénieur communal	communal	Est responsable de la gestion et de la planification des différentes infrastructures communales, et en particulier des aspects relatifs à l'assainissement. Examine les demandes de permis de construire, gère le cadastre des infrastructures communales.
Responsable du cadastre de l'assainissement	communal	Est responsable de la tenue du cadastre du réseau d'assainissement. Gère les chantiers d'interventions sur le réseau d'assainissement.

Tableau 1. Responsabilités et niveau d'action des différents intervenants interrogés.

Intervenant	Niveau organisationnel	Description des responsabilités
Directeur de la distribution d'eau potable	communal	Est responsable du transport de l'eau de son captage aux installations de traitement, puis de ces dernières aux consommateurs. Egalement responsable des réservoirs et de leur gestion.
Responsable administratif de l'approvisionnement en eau potable	communal	Gère les aspects administratifs de l'approvisionnement en eau potable en rapport avec les abonnés, les traitements, l'adduction, les stations de pompage, le réseau de distribution et les réservoirs).
Responsable de la qualité de l'eau	communal	Surveille la qualité des eaux au niveau des différentes infrastructures du réseau d'approvisionnement en eau potable, dans les eaux naturelles. Surveille également la qualité des eaux des piscines et des zones de baignades. Organise des tournées de prélèvement et des séries d'analyses.
Responsable des installations de potabilisation	communal	Est chargé de l'exploitation et de la maintenance des usines de production d'eau potable ainsi que des stations de pompage.
Employé d'une association de protection de la nature	association à but non lucratif	Identifie des sources de pollution des eaux superficielles. Etablit un cadastre des points de rejet dans les eaux de surface. Sensibilise les populations aux problématiques de pollution des eaux et de protection de l'environnement.
Ingénieur dans un bureau d'étude	organisme privé	Mène à bien les mandats obtenus. Planifie et dimensionne des infrastructures, organise et gère des chantiers, procède à des expertises, élabore ou coordonne des PGEE et des PREE, prépare la planification financière et élabore des clés de répartition des dépenses entre différents intervenants.

Tableau 1. Responsabilités et niveau d'action des différents intervenants interrogés.

2.3.2 Objectifs généraux, processus et opérations

2.3.2.1 Définition des concepts

Le terme de "gestion des eaux en milieu urbain" a été défini (chapitre 2.1.2). Les propos tenus par certains intervenants dans ce domaine ayant aussi été recueillis (chapitre 2.2.3), l'analyse de ces propos, ainsi que de différents documents de travail, a permis l'identification des objectifs généraux de ces intervenants et des processus mis en oeuvre pour satisfaire à ces objectifs généraux. Les processus peuvent être vus comme des enchaînements d'opérations qui concourent à un but précis. Prises isolément, ces opérations présentent peu d'intérêt, alors que l'accomplissement d'un processus apporte une satisfaction certaine aux intéressés. Les processus sont en fait les tâches que l'on retrouve dans le "cahier des charges" des différents intervenants. Kettani (1998) donne la définition suivante du concept de "processus métier" :

"Un processus métier est l'ensemble des activités internes d'un métier dont l'objectif est de fournir un résultat observable et mesurable pour un utilisateur individuel du métier."

Les objectifs généraux apportent une réponse relativement à la finalité des processus et leur donnent un

sens plus général.

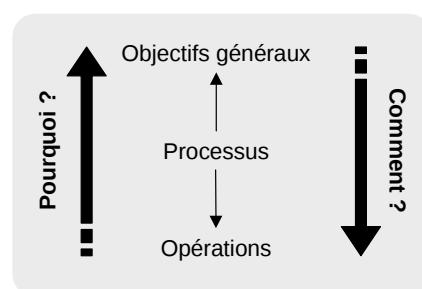


Figure 10. Hiérarchie des objectifs généraux, processus et opérations. En allant vers le haut, on répond à la question "Pourquoi" et en allant vers le bas à la question "Comment".

On retrouve cette hiérarchisation des activités dans toutes les organisations, décrite en termes plus ou moins équivalents. Ainsi, Kotler et Dubois (1997) expliquent que la définition de la mission d'une entreprise est sans doute la tâche la plus importante dont doit s'acquitter la direction générale, car elle donne une image claire des activités de l'entreprise ainsi qu'une justification aux efforts et aux réflexions des employés. Grâce à une telle mission, les objectifs généraux poursuivis peuvent être situés dans un cadre clair. Il en va de même dans le domaine de la gestion des eaux. Cependant, un aspect distingue ce domaine de celui des entreprises à but lucratif (dans lequel chacune cherche à se démarquer de ses concurrentes) : la mission et les objectifs généraux qui guident les activités des intervenants dans le domaine de la gestion des eaux sont les mêmes pour tous, donnés par la législation. Les différentes catégories d'intervenants ont comme objet d'attention un même système, le système "Eaux en milieu urbain", et ont vis-à-vis de ce système des objectifs généraux identiques : on retrouve donc que les processus métier accomplis par les intervenants d'une même catégorie sont similaires, et une analyse de ces processus métier peut fournir de ce fait une base stable pour la mise en place d'outils informatisés d'aide à la gestion.

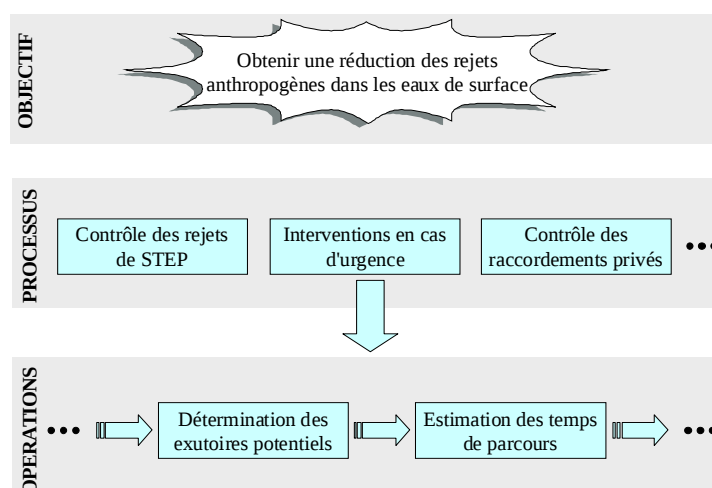


Figure 11. Un exemple de décomposition d'un objectif en processus puis en opérations : pour obtenir une réduction des rejets anthropogènes dans les eaux de surface, les autorités cantonales sont tenues de maintenir un service d'interventions en cas d'urgence qui, dans le cadre d'une intervention, peut être emmené à évaluer les temps de parcours de l'eau dans le réseau d'assainissement.

Les entretiens ont donc été analysés afin d'identifier les processus métier qui caractérisent et constituent la gestion des eaux en zones urbaines en Suisse. Les résultats de ces analyses ont été complétés par une recherche sur Internet concernant différentes administrations fédérales et cantonales. Ont été en particulier examinés les cas des villes de Berne, Zurich, Lausanne, ainsi que du demi-canton de Bâle-Ville. Les résultats de ces investigations font l'objet des chapitres qui suivent.

2.3.2.2 Les objectifs généraux

La mission de la gestion des eaux, présentée de manière détaillée au chapitre 2.1.2, peut être résumée de la manière suivante : protéger les ressources en eau tout en protégeant les populations des dangers que peuvent représenter les eaux, et ceci en respectant les préceptes du développement durable. Dans ce but, la Confédération a mis en place un arsenal législatif dont elle soutient l'application par le biais d'incitations financières (subventions), de campagnes de sensibilisation et d'information, de soutiens techniques et scientifiques; l'acquisition et la diffusion d'informations d'intérêt général dans ce contexte. Elle contrôle l'application de la loi par le biais de statistiques, par l'examen des projets susceptibles d'être subventionnés ou soumis à son agrément, par des contrôles directs. Etant chargés de faire appliquer la loi, les cantons transcrivent les objectifs fixés par la Confédération en termes plus concrets. A titre d'exemple, quelques-uns des objectifs généraux que s'est fixés le canton de Fribourg ¹ sont présentés ci-dessous:

"Garantir à long terme l'approvisionnement en eau potable par l'utilisation sans traitement préalable des eaux souterraines."

"Assurer le maintien de la qualité des réserves en eau potable non encore contaminées."

"S'occuper de la prévention (examen des demandes), dans laquelle il faut compter la maîtrise des risques (protection contre les accidents majeurs), ainsi que de l'assainissement des entreprises industrielles."

"Vérifier l'effet des mesures d'assainissement prises sur la qualité de l'eau et orienter l'action supplémentaire d'assainissement si nécessaire."

"Equiper les communes des infrastructures nécessaires à l'évacuation et au traitement correct des eaux usées [...]."

Ces exemples montrent bien que les objectifs généraux constituent la finalité, et offrent donc un cadre précis, aux actions entreprises par les intervenants concernés. Ces objectifs généraux s'affinent lorsque l'on s'intéresse à des structures organisationnelles de plus petite taille. Ainsi, ils deviennent au niveau communal, par exemple pour la Ville de Lausanne ² :

"Entretenir, étendre et rénover le réseau d'assainissement."

Au niveau des entreprises privées qui sont mandatées par les administrations pour réaliser des travaux, on peut considérer que la mission de ces entreprises consiste en la réalisation de manière correcte et

1. <http://www.fr.ch/open/fr/eaux/>

2. <http://www.lausanne.ch/assainissement/>

rentable des mandats qui leurs sont confiés : ce sont les acteurs institutionnels qui fixent les objectifs en matière de gestion des eaux et assurent le pilotage du système "Eaux en milieu urbain".

Les objectifs généraux permettent de bien appréhender les problématiques et les finalités de la gestion des eaux en zones urbaines telles qu'elles sont perçues par les différents intervenants. Ne contenant que peu d'indications sur la manière dont ils peuvent être atteints, les objectifs ne permettent pas de comprendre précisément quelles sont les tâches quotidiennes que doivent accomplir les intervenants. Ce sont pourtant ces tâches qui nous intéressent dans le contexte de cette recherche puisque ce sont elles qui pourraient faire l'objet d'un soutien par des outils informatiques. Il est donc nécessaire d'affiner l'analyse, de se demander comment ces objectifs généraux sont atteints, et donc d'identifier les processus métier.

2.3.2.3 Les processus métier

L'identification des processus métier a été menée à l'aide des entretiens, des renseignements trouvés sur Internet, ainsi que de documents écrits utilisés par les praticiens ou produits par leurs associations professionnelles. Les différents processus identifiés, résumés dans la figure 12, sont détaillés à l'annexe 1. La liste qui y est présentée, qui est issue d'une analyse exploratoire, se veut descriptive et non exhaustive. Ces informations seront exploitées lors de la présentation de la méthode de conception de systèmes d'information pour la gestion des eaux en zones urbaines (chapitre 5).

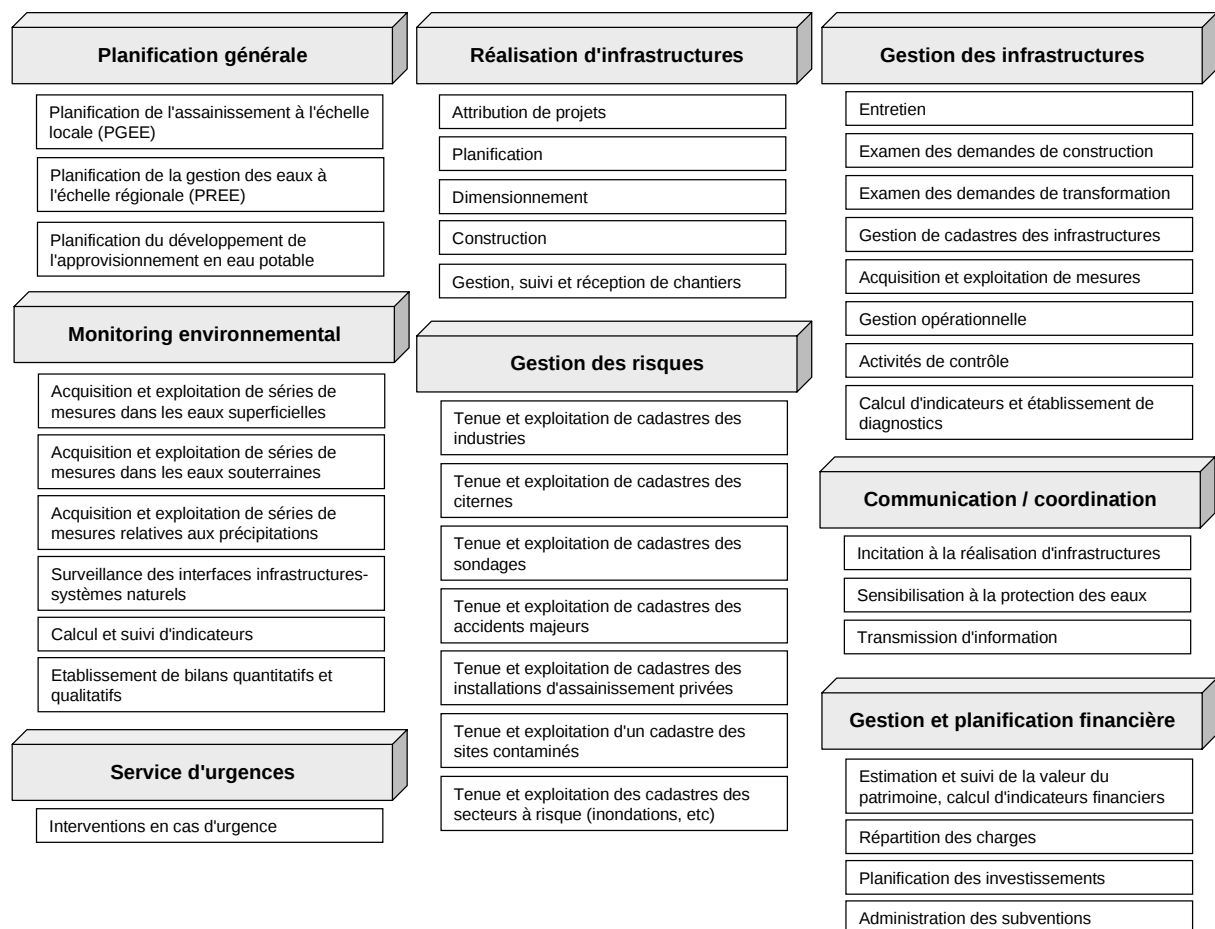


Figure 12. Classification des familles de processus identifiés, par catégorie d'objectifs généraux.

Deux conclusions peuvent être tirées des entretiens qui ont été menés ainsi que des processus métier identifiés:

- les intervenants travaillent pour des organisations qui répondent toutes à des objectifs généraux identiques, fixés par la législation. De plus, les interventions et processus métier menés par ces intervenants ont pour support les différentes composantes d'un même système, le système "Eaux en milieu urbain". Pour ces différentes raisons, on se retrouve, en toute logique, face à un nombre fini de processus métier. De plus, la manière de mener à bien un même processus métier ne diffère pas fondamentalement d'un intervenant à l'autre : tous les ingénieurs doivent parfois mener à bien des simulations de comportement, et le font de manière relativement similaire, et dans tous les cantons on retrouve un service d'intervention en cas d'urgence, dont les opérations sont guidées par une logique identique.
- la finalité d'un grand nombre de ces processus métier est de gérer de l'information : tenue de cadastres, acquisition de séries temporelles relatives à différents phénomènes, etc. Une autre catégorie de processus métier est plutôt consommatrice de cette information, et donc s'appuie sur la première catégorie citée : les activités de dimensionnement, de planification ou de calcul d'indicateurs en sont des exemples. Cette similarité et récurrence de processus métier entre organisations indique que des outils communs devraient pouvoir être développés. De tels outils seraient appelés à manipuler de gros volumes de données et d'informations afin que puisse être menée une gestion réellement intégrée des eaux. Ces exigences rendent par conséquent nécessaire la mise en oeuvre de technologies informatiques. Ces techniques constituent le sujet des chapitres qui suivent.

2.4 Les technologies informatiques et la gestion des eaux en milieu urbain

2.4.1 Une typologie des outils logiciels

2.4.1.1 Programmes, logiciels et outils d'aide à la gestion

Une gestion intégrée des eaux implique donc de nombreuses activités, de nombreux intervenants et d'importants flux d'information entre ces derniers, qu'ils appartiennent à la même catégorie ou à des catégories/niveaux organisationnels différents. On retrouve, dans le cadre des activités professionnelles des intervenants appartenant à une même catégorie, des processus métier identiques. La mise en oeuvre de moyens informatiques trouve dans ce contexte tout son sens, compte tenu de la signification même de l'informatique :

"Informatique : [...] science du traitement de l'information; ensemble des techniques de la collecte, du tri, de la mise en mémoire, du stockage, de la transmission et de l'utilisation des informations traitées automatiquement à l'aide de programmes mis en oeuvre sur ordinateurs." (Petit Robert, 1996)

Notons que l'intérêt de pouvoir s'appuyer sur des outils informatiques dans un but de gestion intégrée des problématiques urbaines a été mis en évidence par différents auteurs et a conduit à la notion d'urbistique

[...]. Ce concept, qui s'appuie sur la notion de systémique, *"vise à acquérir, maîtriser et exploiter l'information urbaine à des fins de gestion avertie et intelligente des flux qui parcourent la cité, dans une optique de développement durable"* (Revaz et al., 2000).

Ainsi, nous avons des processus métier qui se retrouvent au sein de différentes organisations et qui conduisent à la manipulation de grosses quantités d'information. Il vaut donc la peine de développer des outils informatiques dans ce contexte. Avant d'en arriver là, il est toutefois nécessaire de définir les quelques concepts utilisés par la suite dans ce rapport.

Un système informatique est, selon Schach (1996), l'ensemble constitué par le matériel informatique (hardware) et le logiciel (software). Le présent travail se concentre sur la partie logicielle de l'outil informatique, avec pour objectif de proposer des concepts aptes à supporter les évolutions futures du matériel informatique. Le Petit Robert (1996) donne du terme logiciel la définition suivante : *"Ensemble des programmes et des procédures nécessaires au fonctionnement d'un système informatique"*. Plusieurs auteurs (Schach, 1996; Sommerville, 1995) insistent sur le fait que le terme "logiciel" recouvre non seulement les programmes, mais également la documentation relative à ces programmes, permettant en particulier d'installer et d'utiliser l'application logicielle (software product). Nous employerons désormais indifféremment les termes d'application logicielle, de logiciel et d'application.

Pressman (1997) distingue différentes catégories d'applications logicielles (tableau 2). L'examen de ces dernières montre cependant qu'un outil informatisé d'aide à la gestion des eaux peut posséder des caractéristiques appartenant à l'une ou l'autre de ces catégories :

Logiciels-système	Collections de programmes écrits pour apporter un support à d'autres programmes. Ces logiciels sont caractérisés par un fort degré d'interaction avec le matériel informatique. On classe dans cette catégorie les compilateurs, les systèmes de gestion de fichiers, les pilotes de matériel.
Logiciels temps-réel	Logiciels qui suivent, analysent et contrôlent des événements du monde réel au fur et à mesure de leur développement.
Logiciels de gestion	Logiciels assurant le traitement d'informations relatives à la gestion. Il s'agit de la catégorie la plus large de logiciels. Assurant au départ le traitement d'une catégorie d'information (gestion des stocks, comptabilité), ces applications logicielles ont évolué pour devenir des applications de gestion de l'information accédant à de vastes bases de données. Ces applications restructurent et manipulent les données disponibles de manière à faciliter les opérations de gestion ou la prise de décision.
Logiciels scientifiques	Logiciels intégrant des algorithmes de calcul souvent sophistiqués. On retrouve cette catégorie de logiciels dans de très nombreux domaines.
Logiciels embarqués	Logiciels résidant dans la mémoire morte de composants de produits grand-public ou d'installations industrielles, et dont les fonctions sont très spécialisées (logiciel de contrôle des freins dans une voiture, de contrôle du thermostat d'un four, ...).
Logiciels de bureautique	En raison de l'explosion du marché des ordinateurs personnels (PC), de nombreux produits ont été conçus spécifiquement pour ces derniers (traitements de textes, tableurs, etc).
Logiciels d'intelligence artificielle	Logiciels faisant appel à des algorithmes non-numériques pour résoudre des problèmes complexes. On classe dans cette catégorie les systèmes experts, les systèmes à base de connaissances, les réseaux de neurones, etc.

Tableau 2. Les différentes catégories de logiciels (Pressman, 1997)

La classification présentée au tableau 2 n'est dans la pratique pas aussi nette, et, à l'heure actuelle en particulier, les grosses applications de gestion empruntent certaines caractéristiques à l'une ou l'autre des catégories décrites. Ainsi, Bouzeghoub (1997) montre qu'aux applications de gestion relativement simples qui avaient cours jusqu'à la fin des années 80 ont succédé des applications beaucoup plus complexes, dans lesquelles *"les frontières entre des domaines aussi variés que la gestion, la bureautique, les systèmes d'aide à la décision et les systèmes experts se sont estompées pour laisser place à des activités groupées dans un objectif global de représentation et de gestion du système d'information de l'entreprise"*. Il note également que depuis cette époque sont apparus de nouveaux besoins, comme la simulation de systèmes à comportement complexe et la cartographie. Les caractéristiques de ces applications sont *"une forte interactivité avec l'utilisateur, la représentation et la manipulation d'objets complexes et volumineux, des temps de réponse courts et des interfaces homme-machine conviviales"*. Ainsi, les systèmes de gestion du tableau 2 sont devenus des systèmes d'information, applications complexes dont la discussion figure au chapitre 2.4.1.2.

2.4.1.2 La notion de système d'information (SI)

La notion de SI a fait l'objet de nombreuses définitions ne recouvrant pas forcément des concepts équivalents (Pantazis et Donnay, 1996). Deux orientations principales peuvent être décelées : la première considère le SI comme appartenant à une certaine catégorie de système informatique conformément à la définition qui en a été donnée au chapitre 2.4.1.1. La seconde, qui découle de l'approche systémique (Le Moigne, 1990), considère que le SI est étroitement lié à un système plus vaste, ce dernier correspondant généralement à l'entreprise (Collongues et al., 1989; Matheron, 1991) .

La définition apportée par Mylopoulos (1995; cité dans Pantazis et Donnay, 1996), est représentative de la première orientation :

"Information systems are computer-based systems which facilitate the storage, retrieval and management of large amounts of information."

Cette définition, qui se rapporte spécifiquement à ce que nous appellerons un SI informatisé, nous paraît trop restrictive, en ce sens qu'elle ne tient pas compte du fait que, comme le postule l'approche systémique, le SI est une composante d'un système plus vaste. La définition donnée par O'Brien (1995), cité dans Pantazis (1996), va dans cette direction puisqu'il définit un SI comme *"[...] un ensemble de personnes, de procédures et de ressources qui recueillent de l'information, la transforment et la distribuent au sein d'une organisation"*. Il manque toutefois dans cette dernière définition une référence à la finalité du SI. C'est la définition proposée par Rolland (1986), cité dans Pantazis (1996), que nous adopterons :

"Un système d'information est un artefact, un objet artificiel greffé sur un objet naturel qui peut être une organisation. Il est conçu pour mémoriser un ensemble d'images de l'objet réel à différents moments de sa vie; ces images doivent être accessibles par les partenaires de l'organisation qui s'en servent pour décider des actions à entreprendre dans les meilleures conditions. Un système d'information est, en quelque sorte, une extension de la mémoire humaine qui amplifie le pouvoir de mémorisation des acteurs de l'organisation et facilite leur prise de décision."

Cette définition appelle un certain nombre de commentaires : premièrement, il est mentionné qu'un SI fait partie d'un système plus vaste, puisqu'il est greffé sur un objet naturel. Deuxièmement, on parle d'"objet réel" et d'"objet naturel", qui peut être autre chose que l'organisation : cet objet réel peut être un système naturel, soit, dans le cas qui nous occupe, le système "Eaux en milieu urbain" dans son ensemble. Enfin, il est attribué au SI un rôle d'aide à la décision.

L'approche systémique, présentée au chapitre 2.1.2.2, permet de positionner précisément le rôle du SI (figure 13). Elle postule que tout système est constitué de trois sous-systèmes qui échangent des informations entre eux, le système de pilotage, le système d'information et le système opérant :

- Le système opérant, qui incarne le comportement du système complexe (Le Moigne, 1990) et est à l'origine de l'activité et de la dynamique de ce dernier (Gayte et al., 1996). Il exécute les tâches demandées par le système de pilotage et est affecté par les décisions de celui-ci.
- Le système de pilotage, qui définit les objectifs du système, prend les décisions et transmet ses instructions au système opérant.
- Le SI, qui assure le lien entre le système opérant et le système de pilotage. Le SI enregistre sous forme symbolique les opérations du système opérant (le comportement du système complexe), les mémorise et les met à disposition du système de décision. Ce dernier, après avoir élaboré ses décisions d'action, les fait enregistrer et mémoriser par le SI, en les transmettant "pour action" (Le Moigne, 1990). Ce même auteur affirme que ce modèle à trois niveaux (opération - information - décision) permet en tout temps d'entreprendre la modélisation d'un système complexe. De plus, il ne faut pas oublier que tout système complexe est en interaction constante avec son environnement, qui est défini comme "l'ensemble des éléments n'appartenant pas au système, et dont l'état est susceptible d'affecter ou d'être affecté par le système" (Le Moigne, 1984).

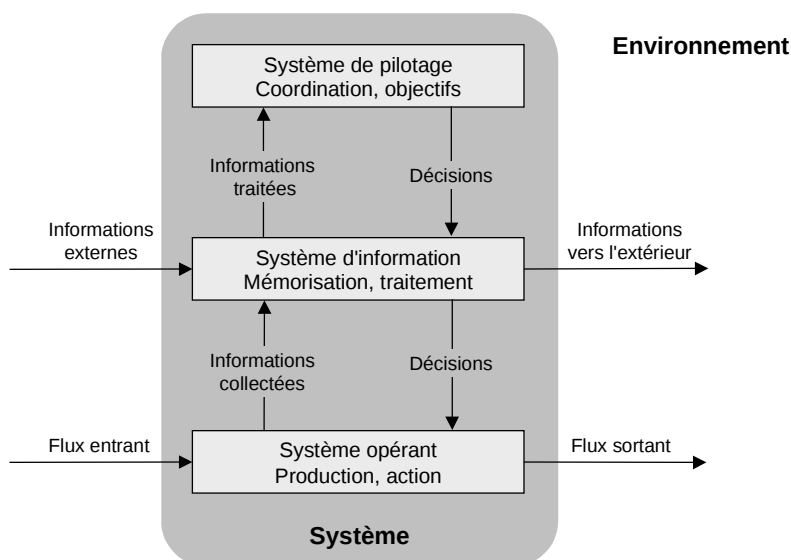


Figure 13. Le schéma systémique de l'entreprise, d'après Lemoigne (1984), en tenant compte des modifications apportées par Gayte (1996) et Collongues (1989).

Les quelques paragraphes qui précèdent ont permis de situer la notion de système d'information selon une perspective systémique. Il en est ressorti que le système d'information a pour mission d'emmagasiner de l'information et de la rediffuser, sous une forme permettant son utilisation "pour exécution" par le

système opérant ou pour permettre la décision et le pilotage par le système de pilotage. Matheron (1991), en se référant au SI d'une entreprise, précise que "*le système d'information d'une entreprise est [...] un système dont les flux entrants et sortants sont exclusivement constitués d'information*". Il devient dès lors évident qu'à partir du moment où ces flux deviennent nombreux et compliqués (ce qui est le cas des systèmes complexes), l'informatique peut apporter son soutien. Notons à ce stade que les mots "système d'information" n'impliquent pas nécessairement le recours à des moyens informatiques, un système informatique n'étant qu'un outil permettant à un SI d'atteindre son objectif (Gayte et al., 1996). Cependant, puisque nous cherchons à soutenir les fonctions du système d'information (au sens systémique du terme) à l'aide d'outils informatiques, nous nous attachons à proposer des concepts pour la mise en place de SI informatisés, et ne ferons donc plus la distinction entre SI et SI informatisés.

Pantazis (1996) met en garde contre le risque de confondre les SI et d'autres types de systèmes informatisés, à savoir les systèmes d'aide à la décision (SAD) et les systèmes experts, qui sont destinés au support des tâches de pilotage, et les systèmes informatisés de type conception assistée par ordinateur (CAO) ou fabrication assistée par ordinateur (FAO), qui eux sont destinés au support du système opérant. Il cite en particulier les définitions apportées par Holtzman (1989) :

Decision support system (D.S.S) : "[...] and information-processing system designed specifically to address the information needs of decision-makers. Decision support systems evolved from data base and management information systems."

Expert systems : "[...] a computersystem designed to capture the skills and factual knowledge of one or more individuals. To achieve high levels of performance, expert systems typically combine traditional software engineering methods with techniques from artificial intelligence research."

Si la distinction entre SI et systèmes-experts est relativement évidente, un système-expert faisant partie intégrante du système de pilotage, il n'en va pas de même en ce qui concerne la distinction entre SI et système d'aide à la décision. En effet, l'un des rôles du SI est de permettre au système de pilotage de prendre des décisions, et il doit dans ce but mettre à la disposition de celui-ci des informations traitées de manière à permettre la prise de décision, ce qui constitue bien évidemment une forme d'aide à la décision. Nous ne pouvons de ce fait pas être d'accord avec la distinction établie par Pantazis (1996), l'aide à la décision constituant pour nous une des caractéristiques et des raisons d'être des SI.

2.4.1.3 Les systèmes d'information à référence spatiale (SIRS)

Les SIRS constituent un sous-ensemble des systèmes d'information (Bédard, 1987; Pantazis et Donnay, 1996; cités dans Thériault, 1996), leur particularité étant d'intégrer une dimension territoriale (Pornon, 1998). C'est pour cette raison que les SIRS font l'objet d'une attention particulière dans ce travail : la gestion des eaux nécessite la manipulation d'informations localisées, et un système d'aide à la gestion des eaux peut être assimilé à un SIRS mis en place spécifiquement pour cette problématique.

Le concept d'information à référence spatiale, ou d'information localisée, se rapporte aux propriétés géométriques et topologiques des entités manipulées. Les propriétés géométriques concernent les références à la position, à la forme et à la taille de ces entités, tandis que les propriétés topologiques

concernent les relations de voisinage, les composantes structurales, les jonctions et les connexions entre les entités. La partie informatisée d'un SIRS comprend nécessairement un mécanisme (généralement une base de données) à même d'assurer la pérennité des données manipulées, ainsi que des moyens de manipuler et de gérer ces données.

Une certaine confusion règne cependant dans la littérature quant aux différentes dénominations et à la définition exacte des termes utilisés dans le domaine des SIRS. Les auteurs anglo-saxons utilisent généralement le terme de GIS (geographical information system) pour désigner à la fois les outils informatiques mis en oeuvre pour le traitement et la manipulation d'informations localisées, et l'ensemble plus vaste intégrant non seulement les outils informatiques, mais aussi les données manipulées ainsi que tout le contexte organisationnel mettant en oeuvre ces outils (Pantazis et Donnay, 1996).

Cette dernière approche est également relayée par Bédard (1982; cité dans Pornon, 1998), pour qui un SIRS *"est un ensemble organisé globalement comprenant des éléments (données, équipements, procédures, ressources humaines) qui se coordonnent, à partir d'une référence commune, pour concourir à un résultat."* Pornon (1998) relève qu'en France le terme de SIG (système d'information géographique) est couramment utilisé pour désigner à la fois la base de données, le logiciel, le système informatique et le système d'information. Thériault (1996) considère que parmi les SIRS, qui sont pour lui des systèmes d'aide à la décision intégrant des données localisées, il est possible de distinguer les SIG, qui sont *"conçus pour l'étude synthétique des milieux et des activités distribués sur le territoire"* et *"qui poursuivent des objectifs de recherche"*, des SIT (systèmes d'information sur le territoire) dont la finalité est principalement le *"traitement administratif des données d'échelle cadastrale"*.

Nous basant sur une approche systémique de la gestion des eaux, nous nous proposons dans ce travail de définir la manière de concevoir un outil logiciel dont le rôle est d'appuyer le sous-système d'information d'un système global, le système "Eaux en milieu urbain". La définition suivante est celle qui convient de ce fait le mieux à l'outil faisant l'objet de cette étude :

"[...] a geographical information system may be defined as a computer-based information system which attempts to capture, store, manipulate, analyse and display spatially referenced and associated tabular data, for solving complex research, planning and management problems" (Fischer et Nijkamp, 1993; cités dans Pouliot, 1999).

Mais pourquoi accorder à la discussion relative aux SIRS une place particulière par rapport à celle accordée aux SI ? Parce que la manipulation par les SIRS d'informations localisées constitue un point de différenciation essentiel par rapport aux SI d'entreprise habituels : du point de vue technique, la gestion d'entité spatiales présente un degré de complexité certain, et, du point de vue organisationnel, la dimension territoriale des informations gérées induit un certain nombre de particularités.

La dimension intégratrice du territoire est un fait mis en évidence par différents auteurs (Crausaz, 2000; Prélaz-Droux, 1995a), et constitue à la fois un avantage et un inconvénient pour les concepteurs de SIRS. Le problème, dans la mise en oeuvre d'un SIRS, est *"avant tout un problème de mise en cohérence et d'intégration d'un vaste ensemble de données hétérogènes et d'applications diverses"* (GERMINAL, 1995). Heureusement, la localisation spatiale joue le rôle d'un dénominateur commun pour les données

et informations collectées de manière indépendante par les différents acteurs (géomètres, géologues, urbanistes, gestionnaires de réseaux d'assainissement, etc). Le fait que ces derniers aient tous affaire à un support commun, le territoire, permet *"l'articulation entre elles (les données et informations) et l'intégration des visions différenciées des différents acteurs"*, et, donc, l'intégration de l'information, soit *"la mise en commun, au sein d'un même système [...] d'information (à caractère spatial ou non spatial) de sources multiples et de formes de représentations diverses"* (Buogo, 1995). Cette intégration de l'information offre différents avantages :

- l'utilisation de données nombreuses et d'informations variées pour la gestion et la prise de décision peut en améliorer la qualité (en particulier dans le domaine de l'eau);
- une approche interdisciplinaire, qui en général est favorable à la résolution de problèmes;
- un accès facilité aux différentes informations, le référentiel commun que constitue le territoire permettant un accès plus intuitif aux données, même lorsque ces dernières sont issues de sources différentes.

Par contre, l'intégration de données localisées apporte également son lot de difficultés supplémentaires puisque *"les données localisées sont [...] soumises à un double impératif d'intégration : avec les données des entreprises concernées d'une part, et avec les autres données relatives à une même portion du territoire d'autre part"* (Golay, 1992). De plus, les différents métiers sont à l'origine de différentes modélisations conceptuelles des informations localisées, le développement des SIRS étant basé sur ces modélisations conceptuelles. La dimension intégratrice du territoire permet de ce fait à différents modèles d'une même réalité de coexister au sein d'un même système d'information, mais il peut devenir difficile d'éviter des redondances, ces dernières conduisant à des difficultés de maintenance des bases de données. Enfin, cette coexistence d'abstractions différentes d'une même réalité débouche sur des problèmes de représentations multiples (Vangenot, 1998) qui, si elles ne sont qu'une conséquence de la complexité du monde réel, n'en constituent pas moins une source de difficulté pour la mise en place d'outils informatiques (Scholl et al., 1996).

Enfin, d'un point de vue plus technique, la gestion et la manipulation des différents types spatiaux (points, lignes, polygones, etc.) présente une certaine complexité, qui implique une adaptation des outils informatiques, notamment des bases de données (Laurini et Thompson, 1992). Les structures de données usuelles telles que proposées par les bases de données classiques pour assurer la pérennité des informations habituellement rencontrées dans les entreprises (textuelles ou numériques) ne sont pas adaptées au stockage de données spatiales. Au modèle relationnel, sur lequel se basent la plupart des systèmes d'information actuels, doivent être intégrées de nouvelles structures de données relatives à l'information spatiale.

2.4.2 Les outils logiciels utilisés dans le contexte de la gestion des eaux

2.4.2.1 Généralités

Le recours à des outils informatiques dans le contexte de la gestion des eaux en zones urbaines n'est pas récent : les importants volumes de données qu'il faut manipuler, que ce soit pour alimenter des modèles de simulation, ou pour gérer des données cadastrales, analyser et visualiser les résultats, etc., ont fait que très tôt des outils logiciels, tant spécifiques que génériques, ont été utilisés. Les chapitres qui suivent

présentent succinctement les différentes catégories de logiciels utilisés dans des domaines qui touchent au système "Eaux en milieu urbain", et en exposent les avantages et inconvénients. À noter que les outils de type gestion financière et opérationnelle (gestion des raccordements au réseau d'approvisionnement en eau potable, outils de facturation, etc) ne sont pas discutés ci-dessous, dans la mesure où ils n'ont pas été examinés.

2.4.2.2 Les logiciels de conception assistée par ordinateur

Les logiciels de CAO (conception assistée par ordinateur) sont conçus comme des métaphores de l'environnement de travail de dessinateurs techniques. L'utilisateur d'un tel logiciel dispose de palettes d'outils analogues à ceux dont il disposerait dans un environnement non informatisé (crayons, règles, gommes, etc.) et développe ses dessins techniques en couches successives rappelant le principe des calques. Bien sûr l'informatisation apporte des améliorations, telles les bibliothèques graphiques, les outils de placage de textures et de rendu en trois dimensions, la gestion des cotes et des échelles, la sauvegarde électronique des documents, etc. Dans le contexte de la gestion des eaux, de tels outils sont plus particulièrement utilisés lors des travaux de génie civil pour la conception des plans des ouvrages, ainsi que pour l'élaboration et la gestion de cadastres (parcellaire, réseau souterrain, etc.). AutoCAD¹ (société Autodesk) ou MicroStation² (société Bentley) sont des exemples de tels logiciels.

Les logiciels de ce genre conviennent très bien pour le dessin vectoriel. Ils ne permettent cependant pas de gérer l'information thématique de manière satisfaisante, celle-ci ne constituant qu'un élément graphique supplémentaire (label) du document graphique, et n'étant en conséquent absolument pas structurée : quel ingénieur en hydraulique urbaine ne s'est-il pas battu avec des plans de canalisations livrés sous forme de fichiers de type "dxf" (AutoCAD) pour en extraire de l'information thématique (Baumann et Keller, 1998)?

Différents constructeurs contournent ce problème en associant le logiciel de CAO à un système de gestion de bases de données (SGBD), les données thématiques associées aux éléments graphiques des documents étant gérées par le SGBD. La séparation physique entre données vectorielles et données thématiques, induite par une telle approche, ne facilite pas les manipulations de données et les échanges d'informations entre logiciels. Les aspects topologiques, lorsqu'ils sont gérés, le sont par des mécanismes propriétaires du logiciel de CAO, et la récupération de cette information topologique, lorsque l'on souhaite exporter des données, est complexe. Des langages standardisés de description de données géoréférencées existent cependant, comme Interlis³ (Keller et Thalmann, 1999) ou Edigéo⁴, qui constituent un moyen de transférer des données géographiques entre les applications, pour autant que l'on dispose des traducteurs adéquats.

Ainsi, les logiciels de CAO, s'ils sont parfaitement adaptés à l'élaboration de dessins techniques et autres plans cadastraux, ne sont pas conçus pour la gestion de données thématiques. L'association à ceux-ci de SGBD permet de pallier cette faiblesse, mais le fait que la composante géographique soit gérée séparément de la composante thématique rend leur utilisation compliquée. Cela rend de plus difficile le

1. <http://www.autodesk.com/adsk/section/0,,284288,00.html>

2. <http://www.bentley.com/products/geproducts/index.htm>

3. <http://www.interlis.ch/>

4. <http://www.cnig.fr/cnig/infogeo/france/Normalisation/Norme.html>

transfert de ces données de manière à les exploiter avec d'autres outils. Enfin, les capacités analytiques de tels outils sont très faibles, voire inexistantes, ce qui rend leurs capacités insuffisantes dans un contexte de gestion des eaux.

2.4.2.3 Les bases de données et les systèmes de gestion de bases de données

Une base de données (BD) est une collection partagée de données logiquement liées entre elles, qui est conçue de manière à répondre aux besoins des différents membres d'une organisation en matière d'informations (McFadden et al., 1993). Un système de gestion de bases de données (SGBD) est un outil logiciel qui permet de manipuler les données contenues dans une BD, soit, entre autres, d'interroger et de mettre à jour cette BD, de gérer les accès aux données et d'en maintenir la cohérence et l'intégrité. Différents modèles logiques permettent d'organiser les données dans une BD, les plus connus étant (dans l'ordre chronologique de leur apparition) : les modèles hiérarchiques, en réseau, relationnel et objet, auxquels on peut ajouter le modèle relationnel-objet, qui est une évolution du modèle relationnel à même d'assurer la persistance d'objets.

A l'heure actuelle, dans le domaine de la gestion des eaux, lorsqu'une BD est conçue, il est fait appel aux modèles relationnel ou relationnel-objet. Une abondante littérature est disponible sur les thèmes des BD et SGBD (Date, 1999; McFadden et al., 1993; Spaccapietra, 1996). Comme il sera, dans ce travail, fait appel au modèle relationnel, nous faisons tout de même de celui-ci une brève description. Dans le modèle relationnel, les objets et associations du monde réel sont représentés par un même concept, la relation, qui est un tableau à deux dimensions souvent appelé table. Les en-têtes des tables sont les attributs de la relation, et les lignes de la table en constituent les occurrences. Chaque occurrence est identifiée de manière univoque par un identifiant (Spaccapietra, 1996) et, ensemble, ces identifiants permettent l'établissement de liens entre les relations (figure 14).

Parmi les SGBD qui sont utilisés dans les domaines en relation avec la gestion des eaux, on peut distinguer :

- les SGBD d'entreprise, généralement mis en place suivant une architecture client-serveur (voire distribuée), que l'on peut retrouver dans les administrations d'une certaine importance. Il s'agit de produits complexes, dont l'installation, la configuration et la maintenance nécessitent des spécialistes informatiques, les intervenants dans le domaine de la gestion des eaux n'en étant que des utilisateurs. Les SGBD de la société Oracle¹, ou SQL Server² de Microsoft, appartiennent à cette catégorie.
- Les SGBD de bureau, nettement moins coûteux et moins complexes, mais également moins performants que les outils précédemment cités. Les SGBD Access de Microsoft³ ou Paradox de Corel⁴ appartiennent à cette catégorie. Généralement utilisés en mode mono-utilisateur, ces outils, très simples d'accès, peuvent être adaptés à des problématiques particulières de façon très aisée. De ce fait, parmi les intervenants rencontrés, il s'en est trouvé plusieurs qui avaient développé eux-mêmes des applications pour utilisation dans le cadre de leur activité professionnelle : citons le cas

1. <http://www.oracle.com/>

2. <http://www.microsoft.com/sql/>

3. <http://www.microsoft.com/office/access/default.htm>

4. <http://www.corel.com/paradox9/index.htm>

d'un SGBD utilisé au sein d'une administration cantonale, relatif aux analyses de contrôle effectuées dans les eaux de surface et dans les stations d'épuration. De nombreux bureaux d'étude ou d'informatique développent, pour le domaine de la gestion des eaux et dans un but de commercialisation, des solutions spécifiques basées sur ce type de SGBD : gestion de stations d'épuration, gestion des données relatives aux sondages géologiques, etc. À noter que la recommandation de l'Association Suisse des Professionnels de la Protection des Eaux, concernant la structuration des données dans le domaine de l'assainissement (VSA, 1999), a fait l'objet d'une implémentation relationnelle sous forme d'une base de données Access¹. Nous la mentionnons puisqu'elle a été utilisée lors du développement du prototype réalisé dans le contexte de ce travail.

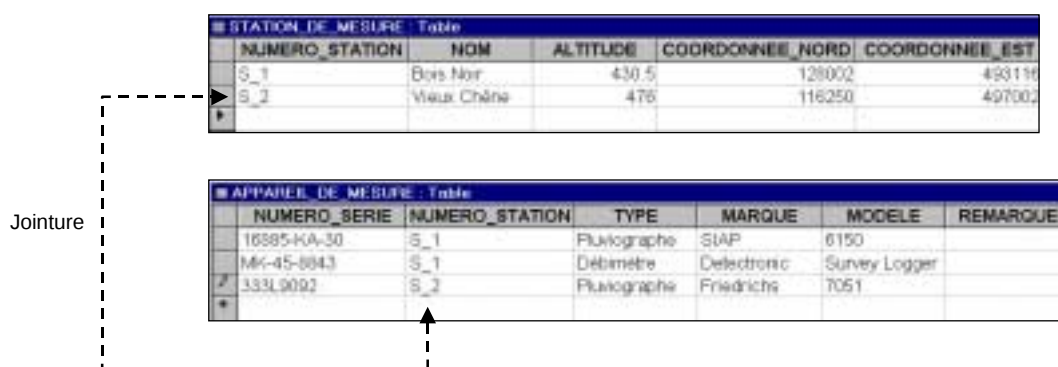


Figure 14. Illustration du modèle relationnel. La clé permettant d'établir le lien entre la station de mesure et l'appareil de mesure est l'attribut "NUMERO_STATION".

Ainsi, de nombreux SGBD sont à l'heure actuelle basés sur le paradigme relationnel, et il s'agit d'une technologie qui est à présent mature et dont les performances sont excellentes. Pendant longtemps toutefois, la gestion des données spatiales en a constitué un point faible, les structures de données spatiales, compliquées, ne pouvant, pour des raisons de performances, faire l'objet d'une adaptation au modèle relationnel. Différents SGBD d'entreprise ont à présent été étendus de manière à pouvoir gérer des types de données plus complexes, dont les types de données spatiaux comme les points, lignes, polygones, etc. En revanche, les SGBD de bureau ne sont pas à même de gérer de telles données, et doivent donc, si l'on veut les intégrer dans des solutions informatiques manipulant de l'information spatiale, être couplés avec d'autres outils, les systèmes d'information géographique (SIG).

Enfin, la diversité des modèles de données structurant les BD dans le contexte de la gestion des eaux présente l'inconvénient de rendre les échanges de données difficiles, et rend quasiment impossible la mise en place d'outils informatiques destinés à des traitements plus sophistiqués de ces données. Le problème de la standardisation des modèles de données est cependant bien connu, et de nombreux organismes et associations professionnelles s'attachent à l'heure actuelle à proposer des structures de données plus mûrement réfléchies concernant les différents domaines de la gestion des eaux (Blanc et Clabaut, 1997; Crausaz et Musy, 1997; SIA, 1999; VSA, 1999).

1. Cette base de données peut être téléchargée à partir de l'adresse <http://www.imanagement.org>.

2.4.2.4 Les systèmes d'information géographique

Nous considérons dans ce travail que la notion de système d'information géographique (SIG), à la différence de celle de SIRS (chapitre 2.4.1.3), ne représente qu'une catégorie de logiciels utilisés pour saisir, enregistrer, manipuler et analyser des données à référence spatiale. Il existe à l'heure actuelle plus d'une centaine de logiciels SIG commerciaux, et l'on assiste à la segmentation de ce marché en différentes catégories de produits allant du serveur de données spatiales aux outils de consultation (tableau 3)

Les SIG s'appuient généralement à la fois sur un système de gestion de fichiers pour les données spatiales et un système de gestion de tables relationnelles pour les données thématiques, la correspondance entre les données thématiques et spatiales étant assurée par un identifiant unique.

Catégorie	Solutions	Caractéristiques
Serveur de données	Oracle Spatial Cartridge, SDE, SpatialWare, Bentley Continuum Server, Geotask Server	Gestion des données spatiales dans un SGBD Applications de grande envergure Gestion de la sécurité des accès Gestion d'accès concurrents Langages de manipulation de données pour le spatial Accès aux données par différents clients
Outil SIG professionnel, outils de gestion et d'analyse	ArcInfo, Intergraph MGE, Vision, Argis, Apic, SmallWorld, Lamps, ...	Gestion d'accès multi-utilisateurs Applications spécialisées Analyses complexes Bases de données réparties Gestion de volumes de données importants Structuration de la base de données Contrôles de qualité Gestion de la topologie Prix élevé Complexes à maîtriser
SIG de bureau	MapInfo, AtlasGis, ArcView, Idrisi, Vertical Mapper, Geomedia, Geoconcept ...	Analyses Traitements simples Cartographie thématique Flexibles et conviviaux Applications à budget limité Structure de données simple (pas de topologie) Analyse rapide Aide à la prise de décision Intégration à des logiciels de bureautique
Composant SIG	MapObject, MapX, GeoObjects, ...	Outil de développement Conception d'application personnalisée Traitements simples
Outil de consultation	ArcExplorer, GeoExplorer, MapXSite, MapObjects IMS, ArcIMS, MapGuide, ...	Visualisation Localisation Information sur un objet

Tableau 3. Les différentes catégories de SIG, principales caractéristiques et exemples de produits commerciaux (Riedo et Mottier, à paraître).

Cependant, la tendance actuelle est de vouloir gérer l'ensemble des données (spatiales et thématiques) dans un même SGBD, et ce grâce à la capacité que commencent à avoir les SGBD de gérer les types de données spatiaux (Longley et al., 1999). Une telle approche permet de séparer les aspects relatifs à la gestion des données, assurée par le SGBD, de ceux relatifs au traitement et à l'exploitation de ces données, qui peuvent ainsi être effectués non seulement par différents SIG, mais aussi par d'autres outils.

Cette approche permet à l'utilisateur de choisir les outils les mieux à même de satisfaire ses besoins.

Dans le cadre de la présente recherche, il a été constaté que les SIG de bureau sont très courants dans le contexte de la gestion des eaux, et que ces outils sont familiers aux différents intervenants : de nombreuses applications développées à partir des SIG MapInfo¹, ArcView² (Esri) et GeoConcept³ ont été rencontrées (figure 15).

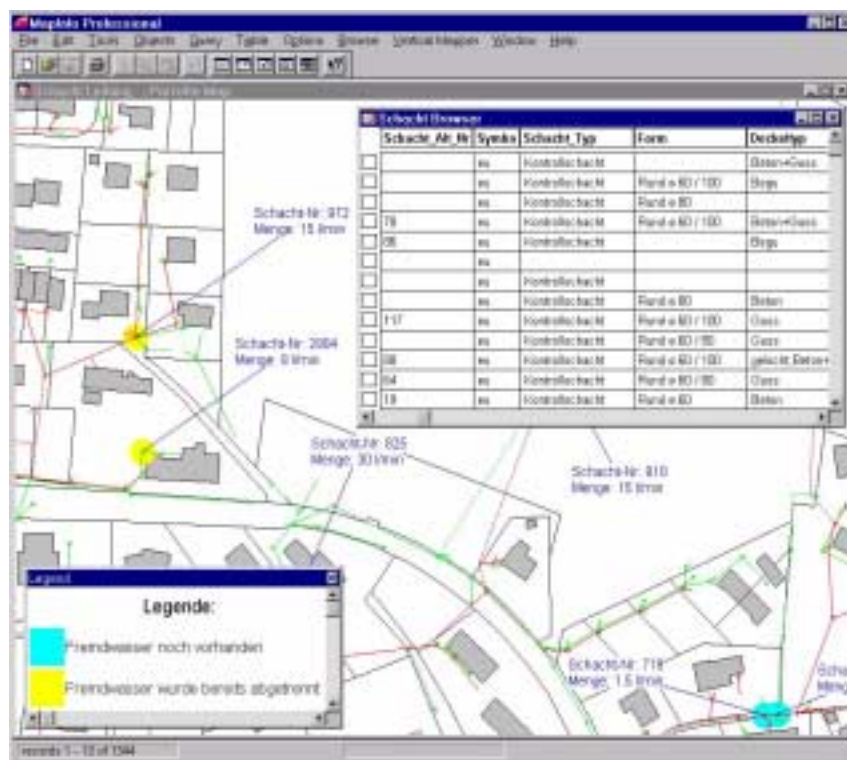


Figure 15. Un PGEE réalisé à l'aide du SIG MapInfo (bureau d'étude Ernst Fuchs, St-Ursen).

Dans quelques administrations ont été vus des SIG "professionnels", basés sur ARGIS de Unisys, comme par exemple l'application GESREAU dédiée à la gestion des eaux de surface du Canton de Vaud (Crausaz et Musy, 1996). Les utilisateurs de tels outils semblent les trouver trop complexes, et la tendance citée plus haut, à savoir une gestion des données au moyen d'un SGBD, l'exploitation de ces données par le biais de différents SIG de bureau ou la visualisation de celles-ci au moyen d'outils de consultation, se confirme.

Les utilisateurs de tels outils se retrouvent confrontés à deux problèmes :

- Comme dans le cas des SGBD, de nombreux modèles de données sont utilisés et rendent les échanges d'information difficiles et le développement de fonctionnalités spécifiques peu aisé. Dans ce contexte également, l'emploi de modèles de données standardisés constitue une solution.
- Le manque de fonctions élaborées propres à la gestion des eaux (par exemple le calcul d'indicateurs, l'extraction de données vers des modèles de simulation de comportement ou le couplage avec de tels modèles, ou encore le soutien de certains processus comme le suivi du cheminement des

1. <http://www.mapinfo.com/>

2. <http://www.esri.com/>

3. <http://www.geo-concept.com/fr/index.html>

écoulements dans les réseaux d'assainissement, etc.). Bien que les logiciels de SIG offrent des possibilités de personnalisation, peu de fonctions de ce type sont développées, et ce pour deux raisons principales : de tels développements sont généralement trop coûteux, et certaines fonctionnalités, comme le calcul d'indicateurs, sont certes jugées intéressantes, mais, n'étant pas demandées par les mandataires et ne s'insérant pas dans les processus métier menés habituellement, ne se justifient pas.

2.4.2.5 Les outils de simulation de comportement

Les activités de planification, de gestion opérationnelle et d'établissement de bilans et de diagnostic requièrent dans le domaine de la gestion des eaux l'usage fréquent d'outils de simulation des comportements : simulation des écoulements dans les réseaux d'assainissement, des flux dans les réseaux d'approvisionnement en eau potable ou dans les cours d'eau, de la diffusion de polluants dans les milieux aquatiques, du fonctionnement d'installations techniques, etc. Citons comme exemples, dans le domaine de l'assainissement, les logiciels Mouse (DHI)¹, Hydroworks (Wallingford Software)², Visual Otthymo³ (figure 16), qui permettent d'effectuer les simulations des écoulements dans les réseaux.

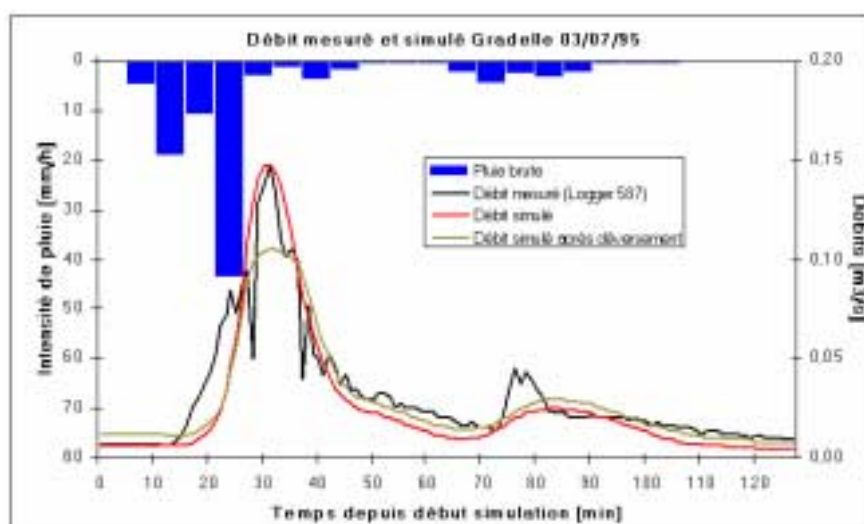


Figure 16. Visualisation des résultats d'une simulation des débits dans une canalisation (Rossi, 1999).

Si les outils qui ont été décrits dans les paragraphes précédents (BD et SIG essentiellement) "*visent principalement à présenter un état du système*", les modèles de simulation cherchent avant tout à décrire les "*mécanismes d'échanges et de transferts entre les composantes du système*" (Pouliot, 1999). Les finalités de ces deux catégories d'outils diffèrent, les premiers s'attachant à décrire les aspects statiques du système (le "quoi ?" et le "où ?"), et les seconds les aspects dynamiques (le "comment ?"). Dans la mesure où ces finalités sont complémentaires, les gestionnaires des ressources en eau vont généralement faire appel à ces deux catégories d'outils (Crausaz, 2000), et les recherches relatives aux possibilités de couplages entre SIG, SGBD et modèles de simulation sont très nombreuses (Prins et al., 1998). En effet, même si les modèles de simulation disposent de leur propre système de mémorisation de données,

1. <http://www.dhi.dk>
2. <http://www.wallingford-software.co.uk/>
3. <http://www.grnland.com/sftware/VisOTT/visott.htm>

l'avantage de tels couplages est double : d'une part les SIG et SGBD offrent des fonctionnalités bien plus évoluées que les modèles de simulation en matière de gestion, de représentation et de préparation des données (Lam et al., 1996), et d'autre part une gestion distincte des données offre au gestionnaire une certaine indépendance, puisque les mêmes données peuvent être exploitées à l'aide de différents logiciels de simulation (Schuurmans et Nelen, 1996). A noter que le degré de couplage entre outil de gestion des données et modèle de simulation peut être plus ou moins fort, et qu'une étude fouillée a été menée à ce sujet par Pouliot (1999). Le principal problème auquel se heurte l'intégration de tels outils dans un système d'aide à la gestion des eaux en zones urbaines est la spécificité des données sur lesquelles se basent les algorithmes de calcul de ces outils de simulation : les entités du monde "réel" y sont décrites selon des structures et à l'aide de données et de paramètres propres aux modèles. Il n'est de ce fait pas aisé de définir une base de données commune à différents modèles de simulation.

2.4.2.6 Les tableurs

Il peut sembler surprenant d'intégrer les tableurs dans une discussion relative aux outils utilisés dans le contexte de la gestion des eaux. Il s'avère pourtant que ces outils sont parmi les plus utilisés dans ce domaine. Un tableur permet *"la création, la manipulation et l'édition de données sous forme de tableaux"* (Petit Robert, 1996). Le tableur Excel, de Microsoft, a été cité par la plupart des intervenants comme un outil utilisé très fréquemment, que ce soit pour l'analyse de données, le traitement de séries temporelles, les calculs de simulation basés sur des formules simples (méthode rationnelle pour le calcul des débits dans les réseaux de simulation), ou l'élaboration de graphiques.

Plusieurs intervenants ont fait ressortir qu'un outil d'aide à la gestion des eaux, pour être accepté et utilisé, doit être ouvert sur les applications existantes en général, et les tableurs en particulier. On retiendra à cet égard deux points, à savoir :

- que les intervenants souhaitent pouvoir utiliser Excel conjointement à d'autres outils;
- que le format Excel est souvent utilisé comme "format d'échange de données", car on peut être sûr de retrouver le logiciel Excel sur quasiment chaque ordinateur de bureau.

Mentionnons enfin que dans l'esprit de certains intervenants, Excel est une BD, ce qui n'est bien évidemment pas le cas : les données dans les tableurs ne sont pas structurées, il n'existe que peu de moyens d'éviter les incohérences et les redondances, ou de faire des requêtes. Une fois que leurs projets sont terminés, de nombreux ingénieurs archivent pourtant les données uniquement sous forme de fichiers Excel, et il est souvent bien difficile, après quelques années, d'y retrouver les informations dont on a besoin (Bernasconi, 1999).

2.4.2.7 Les outils de manipulation de séries temporelles

L'acquisition, la gestion, la manipulation et la diffusion de séries temporelles sont des activités récurrentes dans le contexte de la gestion des eaux : séries hydrométéorologiques, mesures en continu de paramètres physico-chimiques sont acquises par le biais de réseaux et de stations de mesures. Des séries temporelles de mesures (acquises ou générées) alimentent les modèles de simulation de comportement cités plus haut, qui à leur tour livrent les résultats de leurs calculs sous forme de telles données.

Les séries temporelles constituent une catégorie particulière de données, car elles peuvent contenir un très grand nombre de mesures, et les SGBD et les SIG ne sont généralement pas adaptés à la gestion de données de ce type (Larsen, 1999). Des structures de données propres aux séries temporelles et des applications spécialisées dans leur manipulation sont donc utilisées dans le domaine de l'hydrologie et de la gestion des eaux (Da Costa et al., 1996; de Heer et Wyss, 1995; de Souza et al., 1994; Grauer et al., 1996; Wolf-Schumann et Vaillant, 1996).

Les outils qui permettent la manipulation de ces données, comme le logiciel CODEAU, dont une copie d'écran est montrée à la figure 17 (CODEAU, 1993) sont extrêmement spécialisés. Si celui-ci assure la gestion de séries temporelles de manière très performante, il n'est pas en mesure d'assurer celle des données thématiques, relatives aux stations de mesures par exemple : la localisation des stations de mesure, sur la base de critères géographiques par exemple, n'est que rarement possible avec ce type de logiciel. Pourtant, le souhait a été exprimé par plusieurs intervenants, et est rapporté dans la littérature (Grauer et al., 1996), de pouvoir coupler ce genre d'application avec des SIG.

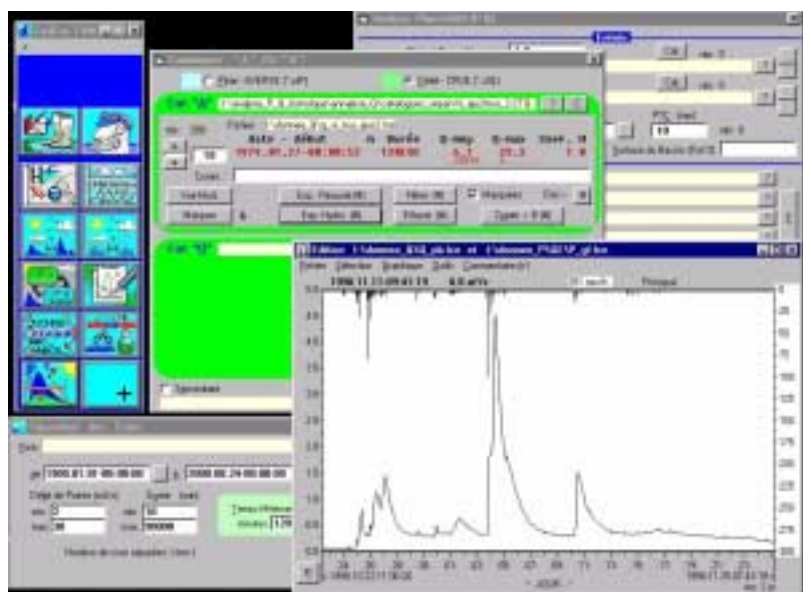


Figure 17. Le logiciel CODEAU, spécialisé dans le traitement, la manipulation, la qualification et la validation de données hydrométéorologiques (CODEAU, 1993).

Notons que l'on observe dans le contexte des SGBD d'entreprise également la même évolution que celle constatée en ce qui concerne les types de données spatiaux : ces systèmes intègrent des types de données propres au domaine du temporel.

2.4.2.8 Les solutions spécifiques

Nous appelons solutions spécifiques des applications logicielles basées sur des assemblages de logiciels génériques ayant été étendus et configurés de manière à répondre aux besoins de secteurs d'activité et de métiers particuliers. Ces solutions spécifiques sont dans certains cas commercialisées par les mêmes sociétés que les applications génériques, dans d'autres cas ce sont le fruit de développements menés par des sociétés tierces.

Dans le domaine de la gestion des eaux en milieu urbain, des exemples de tels produits utilisés en Suisse

sont LIDS¹, de la société Berit, TopoBase, de la société C-Plan², Magellano, de la société InterCAD ou encore ArcFM for Water and Wastewater, de la société Esri³. Les trois premiers associent un logiciel de CAO (MicroStation pour le premier et AutoCAD pour les deux suivants) à un SGBD, des modules spécifiques à la composante-métier étant développés et intégrés à l'édifice. L'illustration 18 présente l'application Magellano, basée sur le SGBD Oracle, spécialement conçue pour la gestion des cadastres souterrains. Les différentes extensions ArcFM sont basées sur le SIG ArcInfo et spécifiquement destinées à la gestion de différentes catégories de réseaux urbains.

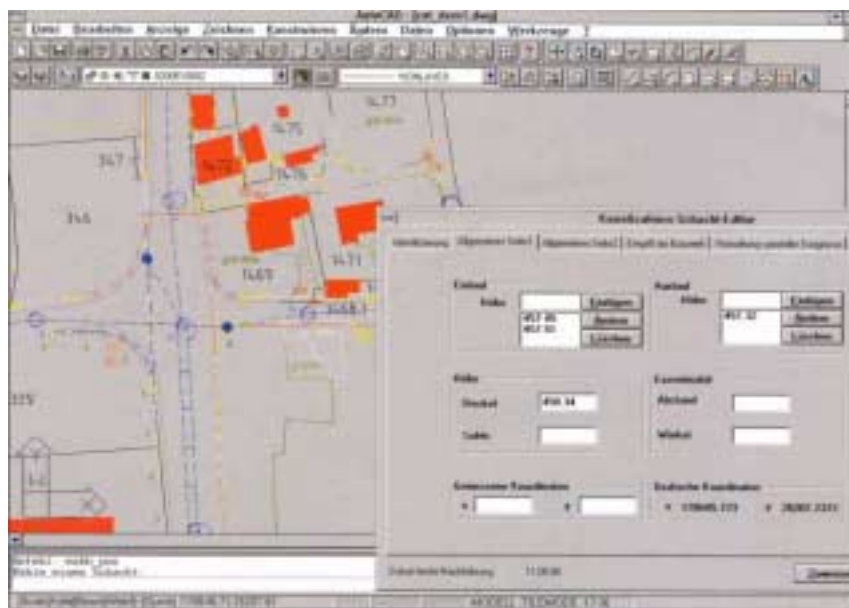


Figure 18. L'application Magellano, développée par la firme InterCAD SA pour la gestion des données du cadastre souterrain.

De telles solutions sont extrêmement intéressantes car elles combinent les avantages de plusieurs catégories d'outils : s'appuyant sur des SGBD d'entreprise, elles offrent des possibilités en terme de gestion des données extrêmement satisfaisantes; elles profitent des fonctions de manipulation de l'information à caractère spatial propres au SIG ou aux outils de CAO; enfin, elles mettent à disposition des utilisateurs des fonctionnalités spécifiquement adaptées à leurs métiers. Ces solutions présentent cependant certains inconvénients :

- elles sont complexes et exigent une période de formation importante;
- elles sont généralement extrêmement coûteuses, puisqu'elles intègrent de nombreux outils et requièrent en plus des développements spécifiques, et sont ainsi au dessus des moyens de nombreuses organisations actives dans le domaine de la gestion des eaux;
- certaines de ces solutions sont peu ouvertes sur des applications tierces et difficiles à personnaliser;
- les organismes disposant des moyens nécessaires pour acquérir de tels outils ont souvent des exigences particulières et veulent personnaliser ces solutions, ce qui conduit à augmenter encore leur coût.

1. <http://www.berit.ch/>

2. <http://www.c-plan.com/>

3. <http://arconline.esri.com/>

2.5 Synthèse du chapitre

Le système "Eaux en milieu urbain" a été décrit, et les différentes activités humaines en rapport avec la gestion des eaux ont été présentées. Il a également été montré que ce domaine traverse à l'heure actuelle une période de changements, car les paradigmes qui sous-tendent la vision actuelle de la gestion des eaux sont remis en question : si les infrastructures actuelles d'approvisionnement, de traitement et d'évacuation des eaux, en raison des investissements considérables qu'elles représentent, continueront à être utilisées et entretenues pendant de nombreuses années, de nouvelles technologies et méthodes de gestion des eaux, dont une grande partie sont à l'heure actuelle en phase de test ou n'existent tout simplement pas encore, feront bientôt leur apparition.

Le système "Eaux en milieu urbain" est un système complexe, qui doit faire l'objet d'une gestion intégrée, c'est-à-dire que les interactions entre les sous-systèmes qui le composent doivent être considérées au cours des différents processus de gestion. Une telle approche est cependant très difficile à mettre en oeuvre, et ce pour différentes raisons :

- le système étant complexe, les interactions entre sous-systèmes sont extrêmement nombreuses. De plus, les évolutions qui marquent à l'heure actuelle la philosophie de la gestion des eaux en zones urbaines conduisent à une complexification de ce système. En effet, on passe progressivement d'une approche très centralisée de cette gestion, à des modèles plus divers et plus décentralisés, et donc aussi plus difficiles à gérer.
- Les intervenants concernés par cette gestion, soit ceux dont les activités impliquent des actions et décisions susceptibles d'influencer l'une ou l'autre des composantes du système "Eaux en milieu urbain", sont très nombreux, sont responsables de quantité de processus métier et sont disséminés dans des structures organisationnelles différentes.
- Une gestion intégrée des eaux en zones urbaines nécessite ainsi la maîtrise de très nombreux flux d'information entre de multiples intervenants, ainsi que la manipulation d'énormes volumes de données.

De nombreux outils informatiques sont à l'heure actuelle utilisés dans le contexte de la gestion des eaux, mais ils ne donnent pas entière satisfaction. Les échanges d'information entre les intervenants restent compliqués, le manque de fonctionnalités propres à la gestion des eaux, à même de soutenir les processus métiers, est fréquemment évoqué par ces intervenants. Enfin, la complexité des données qui doivent être manipulées (données spatiales et temporelles) font que la mise sur pied d'outils informatiques dans ce domaine reste une tâche malaisée. Le rôle intégrateur que joue le territoire offre des possibilités de construire des outils informatiques basés sur les technologies des SI et des SIRS et pouvant appuyer les processus nécessaires à une gestion intégrée des eaux en zones urbaines. Une réflexion doit toutefois être menée quant à la manière de concevoir de tels outils. Elle fait l'objet de la suite de ce rapport.

Caractéristiques requises d'un outil d'aide à la gestion des eaux

Les attentes des intervenants envers un système informatisé d'aide à la gestion intégrée des eaux (non pas du point de vue fonctionnalités spécifiques, mais du point de vue caractéristiques générales), ainsi que les contraintes dont il faut tenir compte lors de la réalisation d'un tel outil, sont discutées. Enfin, les objectifs du présent travail sont reformulés, conformément aux attentes ayant été identifiées.

3.1 Les missions

3.1.1 Assurer la gestion des données

Les différentes organisations et institutions actives dans le domaine de la gestion des eaux collectent et stockent de grandes quantités de données, qui peuvent en fait se rapporter à l'ensemble des éléments du système "Eaux en milieu urbain" (figure 3). La gestion de ces données constitue à l'heure actuelle l'une de leurs préoccupations majeures. Les problèmes qui résultent d'une absence ou d'une mauvaise gestion des données sont relevés par Bernasconi et Gujer (1998) : travail identique effectué plusieurs fois, pertes de savoir par les organisations, interruptions des processus-métier induites par les absences des personnes détentrices de l'information, pertes de temps en raison de problèmes d'incompatibilité de formats de données et mauvaises prises de décisions. Le problème devient de plus en plus aigu, car les données nécessaires pour pouvoir mener à bien les tâches relatives à la gestion des eaux sont de plus en plus nombreuses et de types variés. Enfin, les données ayant été collectées représentent une valeur importante et reconnue. Par contre, on peut constater que la valorisation de ces données, dans le but d'en extraire différentes informations, ne constitue pas encore un thème de réflexion. Il est à cet égard intéressant de noter que la plupart des outils du domaine de la gestion des eaux, baptisés "systèmes d'information", servent en réalité essentiellement à la visualisation et à la gestion de données (Spring et Kaufmann, 1998).

L'action de "gérer des données" recouvre quatre notions complémentaires : assurer la pérennité et l'intégrité des données, permettre la numérisation des données collectées, gérer l'accès aux données et en

assurer l'indépendance.

Assurer la pérennité et l'intégrité des données

De nombreuses études montrent que la valeur des données qui ont été collectées par les différents services dans le domaine de la gestion des eaux est énorme. A titre d'exemple, la valeur des données relatives au réseau d'assainissement de la Ville de Berne représentait, en 2000, un montant compris entre 10 et 20 millions de francs suisses¹. Il est par conséquent indispensable qu'un système informatisé d'aide à la gestion intégrée des eaux en zones urbaines (SIAGEM) assure la pérennité, l'intégrité et la cohérence des données relatives au système "Eaux en milieu urbain", qui constituent de fait un précieux patrimoine. Un SIAGEM doit donc intégrer un SGBD à même de gérer de gros volumes de données localisées et temporelles. Dans la mesure où la littérature relative aux SGBD est particulièrement riche, nous ne développerons pas plus les aspects techniques liés à cette partie d'un SIAGEM. Le tableau 4 présente la quantité d'éléments que comprend le réseau d'assainissement d'un centre urbain de grande taille, et donne au lecteur une idée des volumes de données qui doivent pouvoir être gérés dans ce contexte.

Longueur des canalisations du domaine public :	850 km
Nombre de raccordements de bien-fonds :	70'000
Nombre de canalisations :	29'500
Nombre de chambres normées :	27'400
Nombre de dépotoirs de chaussées :	56'800
Nombre d'ouvrages de rétention :	20'000
Nombre de déversoirs d'orage :	200
Nombre de stations de pompage :	10

Tableau 4. Quelques chiffres quant à la quantité d'infrastructures d'assainissement gérées par la Ville de Zurich (Mallaun et Burgermeister, 1994).

Permettre la numérisation des données collectées

Plusieurs interviewés ont mentionné la grande quantité de données et d'informations qui ne sont présentes que dans la mémoire des collaborateurs de l'organisation. Un ingénieur cite l'exemple des dépôts dans les réseaux d'assainissement : la présence de dépôts indique des vitesses d'écoulements trop faibles, et donc des mauvaises conditions hydrauliques dans le réseau d'assainissement. Or, cette information n'est connue que du personnel chargé de l'entretien du réseau. La transmission de cette information aux responsables de la gestion du réseau permettrait de mieux cibler les interventions. Cela montre que les informations détenues par les différents intervenants, de manière souvent informelle et déstructurée, doivent être rendues accessibles à des personnes tierces, qui peuvent les mettre à profit. Ce processus d'acquisition d'information ne doit cependant pas représenter un effort supérieur aux bénéfices qui peuvent en être retirés : se référant à la mise en place d'une base de données de résultats d'analyses d'échantillons d'eau prélevés au niveau des stations d'épuration et dans les eaux de surface, un

1. Données collectées depuis une vingtaine d'années, relatives au cadastre et aux attributs thématiques des ouvrages du réseau, à son état et à sa gestion, aux surfaces connectées et aux écoulements (séries temporelles). Le réseau en question comptait en 2000 311 km de canalisations, 8'500 chambres, 28 stations de pompage et 91 stations de mesure en continu (Bruno Widmer, Ville de Berne).

intervenant explique que les laborants devaient à l'origine eux-mêmes introduire les résultats des analyses dans un système informatique. Aussi, ayant constaté que cela leur prenait trop de temps, il n'ont pas continué cette tâche.

Les connaissances doivent donc être transférées des personnes qui possèdent cette information à un système centralisé de gestion des connaissances. Pour assurer ce transfert de connaissances de la mémoire collective "humaine" à un système d'information informatisé, il faut mettre en place des outils qui s'insèrent dans les processus métier habituels, les soutiennent, et profitent de l'occasion pour, de manière transparente, acquérir et structurer l'information (figure 19) qui leur est communiquée. Par exemple, dans le contexte de l'inspection des réseaux d'assainissement, les faits observés par les inspecteurs doivent être reportés sur une fiche. Pour faciliter ce processus, un outil informatique pourrait mettre à disposition de l'inspecteur une version informatisée d'une telle fiche, et en extraire lui-même l'information pour la structurer et la mémoriser de manière transparente pour l'inspecteur.

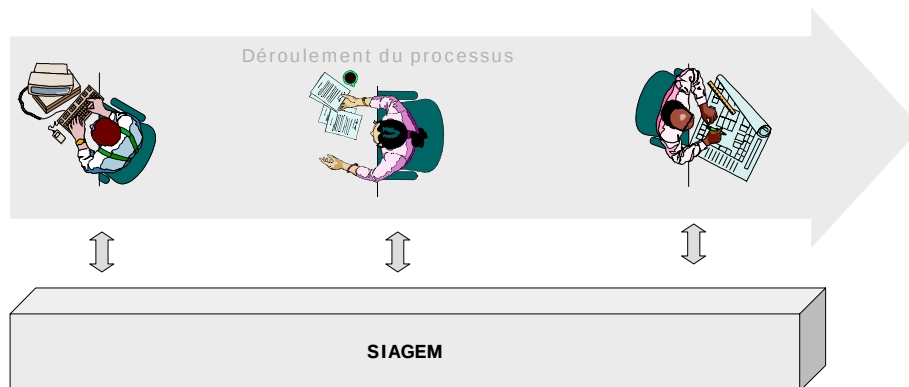


Figure 19. Un SIAGEM doit accompagner les différentes étapes des processus-métier, en faciliter l'accomplissement et en profiter pour acquérir de l'information.

Gérer l'accès aux données

L'accès aux données n'est pas toujours aisé dans la pratique : il est mentionné que, souvent, au sein d'une même organisation administrative, on ne sait parfois pas quelle est la personne qui possède ou peut fournir les données dont on a besoin. Il n'est souvent pas possible d'obtenir une vue d'ensemble des données existantes. Un ingénieur interrogé explique par exemple que, dans le cas du bureau d'étude pour lequel il travaille, les données sont manipulées à l'aide de différents logiciels, en fonction des besoins, et ne sont de ce fait accessibles que par le biais de ces mêmes logiciels : ainsi les données relatives au réseau d'assainissement sont gérées à l'aide de deux logiciels de CAO différents, les données relatives aux ouvrages du réseau le sont par le biais d'un SGBD "bureautique", et les séries temporelles sont stockées directement soit au format du logiciel de simulation utilisé, soit sous la forme de fichiers-texte. De plus, les différentes données qui seraient nécessaires à la réalisation d'un projet sont souvent collectées par des organisations différentes, et donc gérées de manières très diverses. C'est-à-dire qu'avoir une vue d'ensemble des données existantes, ce qui serait souvent nécessaire en début de projet lorsque l'ingénieur cherche à savoir quelles données sont disponibles et quelles données doivent être acquises, est très difficile, et demande beaucoup de temps. Bernasconi (1999) note que 30% du temps de travail des ingénieurs est consacré à la recherche d'informations.

Un SIAGEM doit donc être à même de proposer à ses utilisateurs une vision d'ensemble et un accès aisé aux données dont il assure la gestion. Notons qu'une gestion intégrée implique la possibilité de pouvoir accéder à des données relatives aux différentes composantes du système "Eaux en milieu urbain", ces dernières pouvant faire l'objet d'une gestion par des organisations tierces. Un SIAGEM doit être affranchi des barrières transorganisationnelles, et ce de manière transparente pour ses utilisateurs (figure 20).

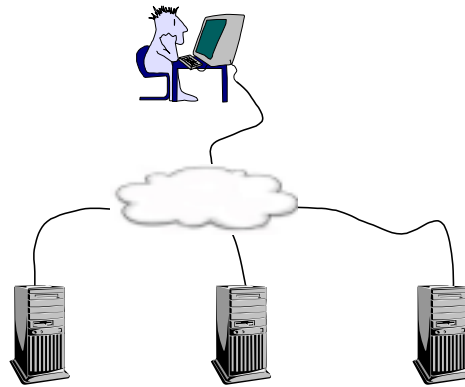


Figure 20. Un utilisateur doit pouvoir accéder à différentes sources de données de manière transparente.

A partir du moment où un SIAGEM doit permettre à des intervenants d'accéder à des données placées sous la responsabilité d'intervenants tiers se pose le problème de la confidentialité des données. Un SIAGEM ne peut en effet pas fonctionner comme un système ouvert, offrant à chacun la possibilité d'accéder à toutes les données disponibles : que l'on pense en effet, par exemple, aux cadastres des industries ou des sites contaminés, qui peuvent contenir des informations confidentielles. De plus, l'intérêt de publier certaines données est tout relatif, et peut même occasionner une certaine confusion : des descriptions précises de la manière dont des données ont été acquises, ou des personnes responsables de leur acquisition, ne doivent pas forcément être diffusées partout : un responsable de la métrologie dans les réseaux d'assainissement note par exemple qu'il peut être extrêmement intéressant, pour le propriétaire des données, de connaître la manière dont des analyses d'échantillons d'eau ont été réalisées, mais que cette information ne doit pas être mise à disposition de toute personne ayant accès aux résultats des analyses. De même, la gestion du parc d'appareils de mesure nécessite la manipulation de données comme les numéros de séries des sondes ou des appareils de mesure : ces données, si elles sont très utiles au gestionnaire de ces appareils (qui peut par là déceler un défaut dans un appareil et corriger rétroactivement les erreurs commises), ne présentent aucun intérêt pour des personnes extérieures.

Un SIAGEM doit en conséquence permettre un contrôle des différents utilisateurs, et assurer un accès différencié aux données, suivant les droits spécifiques de chaque intervenant.

Assurer l'indépendance des données

Les notions de donnée et d'information sont distinctes : une donnée est un fait concernant une personne, un objet ou un événement, tandis qu'une information résulte du traitement de données et est présentée de manière à être interprétée par une personne particulière (McFadden et al., 1993). Les données qui sont gérées par un SIAGEM doivent souvent être mises à disposition d'intervenants qui génèrent à partir de là des informations très variées : des données relatives au réseau d'assainissement par exemple peuvent

servir à la création d'un plan cadastral, d'un plan synoptique, d'une série de profils en long, elles peuvent être utilisées pour le calcul d'indicateurs ou pour alimenter un modèle de simulation de comportement. Bref, ces données doivent pouvoir d'une part servir à générer des informations très variées, et d'autre part être valorisées à l'aide de logiciels divers (figure 21). Un SIAGEM doit donc assurer une gestion des données indépendante de leur utilisation dans un contexte particulier (Baumann et Keller, 1998).

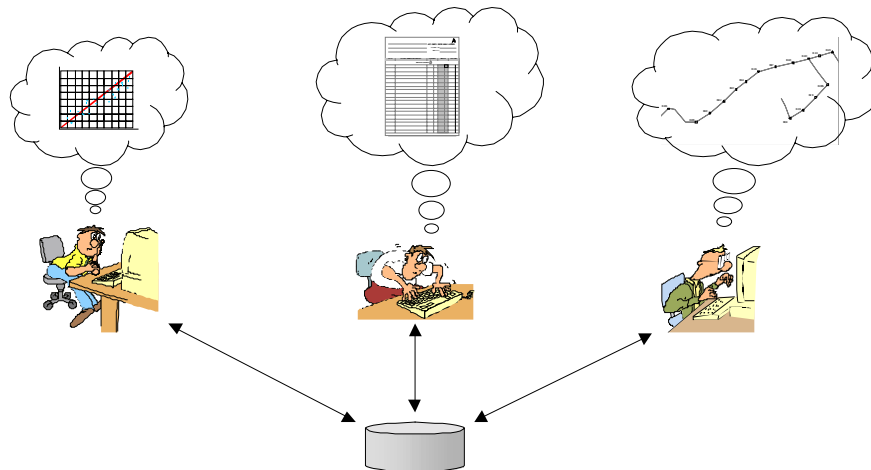


Figure 21. Les mêmes données utilisées pour produire des informations différentes.

3.1.2 Fournir des informations de contrôle et de pilotage

Les différentes activités de la gestion du système "Eaux en milieu urbain" ont pour but d'assurer la santé, la sécurité et le confort des populations, ainsi que de protéger les ressources en eau, et ce de manière économique et en conformité avec les préceptes du développement durable (chapitre 2.1). Or le système "Eaux en milieu urbain" est un système complexe, qui subit d'innombrables interventions, de très nombreuses modifications, et dont les évolutions sont difficiles à prévoir et à contrôler.

Un SIAGEM a cependant précisément pour objectif, en tant que système d'information, de fournir des informations aux gestionnaires de manière à ce que ceux-ci soient à même de prendre des décisions pour "piloter" le système "Eaux en milieu urbain" dans la direction voulue. Une analogie entre la gestion des eaux en milieu urbain et le pilotage d'une entreprise peut ici être établie : *"Piloter une entreprise, c'est choisir un objectif, définir la meilleure stratégie, [...] puis corriger en permanence les écarts par rapport à la ligne prévue"* (Apothéloz et Stettler, 1995). Pour qu'un pilotage du système "Eaux en milieu urbain" soit possible, il faut non seulement que des moyens d'agir existent, mais aussi que des informations qui permettent d'évaluer l'état du système et de ses orientations soient disponibles. Ces informations, que nous appelons indicateurs¹, offrent en tout temps aux gestionnaires un "tableau de bord" du système dont ils ont la charge. Un SIAGEM doit être à même de proposer des tableaux de bord du système "Eaux en milieu urbain" aux gestionnaires de ce dernier, et donc être capable, à partir des données dont il assure la gestion, de calculer des indicateurs. Notons que l'usage d'indicateurs est peu répandu dans le domaine de l'assainissement (et probablement dans celui de la gestion des eaux), alors que ce sont des outils fréquemment utilisés dans l'industrie (Cardoso et al., 1999).

1. Un indicateur est une variable ayant pour objet de mesurer ou d'apprécier un état (Petit Robert, 1996).

Les indicateurs peuvent être répartis en quatre catégories : indicateurs relatifs aux infrastructures, indicateurs économiques, indicateurs environnementaux et indicateurs socio-culturels. Ils doivent d'une part permettre l'évaluation de l'état existant, et d'autre part permettre l'identification des tendances à court et à long termes. A noter qu'un indicateur peut être classé dans différentes catégories à la fois.

- Les indicateurs relatifs aux infrastructures permettent le suivi de ces dernières et de leurs performances: taux de rendement de processus d'épuration, taux de raccordement aux réseaux d'approvisionnement en eau potable, état moyen des canalisations par secteur, combinaisons de la durée de vie estimée et de l'année de pose de canalisations d'assainissement en vue de la planification de leur remplacement (Sawatzki, 1996), etc.
- Les indicateurs économiques permettent de planifier le financement à court et à long termes, et de surveiller l'évolution du système du point de vue financier : estimation de la valeur réelle d'ouvrages en vue de calculs de clés des dépenses (VSA, 1994); estimations de la valeur globale d'un certain type d'infrastructure, dont on veillera à ce qu'elle ne baisse pas; montant total ou par secteur des taxes d'assainissement encaissées; etc. Le VSA est à l'heure actuelle en train d'élaborer une liste d'indicateurs financiers relatifs à la problématique de l'assainissement, destinée aux gestionnaires et décideurs politiques (tableau 6).
- Les indicateurs environnementaux permettent de contrôler l'évolution du système "Eaux en milieu urbain" en regard des préceptes du développement durable. Depuis la publication du rapport Brundtland (chapitre 2.1.3.1), de nombreux indicateurs environnementaux ont été proposés (Azar et al., 1996). Les travaux de Lundin et al (1997; 1999) proposent plus spécifiquement des indicateurs relatifs à la gestion des eaux en milieu urbain (tableau 5).
- Les indicateurs socio-culturels permettent d'appréhender la relation entre les populations et le système dont il font partie intégrante. De tels indicateurs sont par exemple la consommation d'eau, les statistiques sur les usages de l'eau, le degré d'acceptation de technologies alternatives, etc.

Compartiment du système	Dimension	Exemple d'indicateur
Ressources	Prélèvements	Prélèvements annuels par rapport aux ressources totales
	Qualité des eaux	Teneur en phosphore
	Protection des eaux	Débit annuel des sources protégées
Eaux potables	Consommation d'eau	Consommation par jour et par personne
	Traitement	Degré de traitement nécessaire
	Distribution	Pertes des réseaux (volumes exfiltrés / volumes délivrés aux consommateurs)
	Distance	Distance moyenne des sources ou installations de traitement aux consommateurs
	Qualité	Comptage des coliformes
Eaux usées	Réutilisation de l'eau	Volumes réutilisés / volumes consommés
	Production	Production quotidienne d'eaux usées
	Performance des traitements	Taux d'élimination DBO ₅ , P, N
	Charge des eaux superficielles	Rejets en DBO ₅ , P, N

Tableau 5. Proposition d'indicateurs environnementaux, adapté de Lundin (1999)

Compartiment du système	Dimension	Exemple d'indicateur
Produits résiduels	Consommation de ressources	Consommation de composés chimiques par unité de P éliminée
	Consommation d'énergie	Energie consommée par unité de DBO ₅ ou de N éliminée
	Affectation des boues d'épuration	Pourcentage de boues d'épuration revalorisées par rapport aux boues produites
	Recyclage des nutriments	P et N recyclés
	Qualité des boues d'épuration	Contenance en cadmium
	Récupération d'énergie	Energie de chauffage récupérée des boues
	Transport	Distances parcourues pour la valorisation des boues

Tableau 5. Proposition d'indicateurs environnementaux, adapté de Lundin (1999)

Différents critères doivent être pris en compte lors du choix d'indicateurs : leur capacité à mettre en évidence des tendances, leur reproductibilité, l'information qu'ils donnent, la disponibilité des données permettant leur génération, la capacité des gens à les comprendre, et enfin leur coût d'acquisition.

De toute évidence, la palette d'indicateurs qui pourraient être considérés est immense. De plus, un ensemble fini d'indicateurs ne peut être choisi une bonne fois pour toutes, puisque les indicateurs utilisés s'attachent à décrire des phénomènes connus : on mesure par exemple uniquement les concentrations de polluants dont on connaît les effets néfastes, or de nombreux composés dont les effets sur les écosystèmes ne sont pas connus sont utilisés par l'industrie ou vont être créés dans les années à venir. De ce fait, la palette d'indicateurs proposée par un SIAGEM doit pouvoir évoluer en fonction des besoins et du développement des connaissances du système "Eaux en milieu urbain".

Station d'épuration

Charges en personnel, en consommables, intérêts payés, charges annuelles d'entretien, investissements annuels pour le développement de la capacité, calculés par m³ d'eau potable délivré, par m³ d'eau arrivant à la STEP, par habitant.

Pourcentage du budget de fonctionnement alloué au traitement de l'eau / au traitement des boues.

Coûts de traitement des boues par tonne de substance sèche.

Coûts d'évacuation des boues par tonne de substance sèche.

Réseau d'assainissement

Charges en personnel, en consommables, intérêts payés, charges annuelles d'entretien, investissements annuels pour le développement de la capacité, calculés par m³ d'eau potable délivré, par m³ d'eau arrivant à la STEP, par mètre linéaire de canalisation. Peuvent être calculés de manière séparée pour les canalisations et les ouvrages du réseau.

Pourcentage du budget relatif à l'assainissement alloué aux canalisations / ouvrages du réseau.

Tâches d'inspection

Charges en personnel, en consommables, intérêts payés calculés par m³ d'eau potable délivré et par habitant.

Inspections spécifiques aux bien-fonds : charges en personnel, en consommables, intérêts payés, calculés par m³ d'eau potable délivré, par habitant, par biens-fonds.

Inspections spécifiques aux entreprises (contrôle des eaux usées) : charges en personnel, en consommables, intérêts payés, calculés par m³ d'eau potable délivré, par habitant, par entreprise.

Taxes d'assainissement

Tableau 6. Exemples d'indicateurs financiers relatifs au domaine de l'assainissement.

Taxes d'assainissement encaissées, par m³ d'eau potable délivrée, par habitant.

Part des taxes d'assainissement relatives aux eaux usées

Part des taxes d'assainissement relatives aux eaux pluviales

Tableau 6. Exemples d'indicateurs financiers relatifs au domaine de l'assainissement.

3.1.3 Soutenir les processus métier

Quant à la question de savoir ce qu'elles attendent d'un système informatisé d'aide à la gestion des eaux, les personnes interrogées ont fréquemment cité le besoin d'être appuyées dans le cadre de leurs activités professionnelles usuelles. Par exemple :

- un ingénieur, dans le contexte de la simulation hydrologique des écoulements dans les réseaux d'assainissement, se déclare insatisfait des possibilités offertes par les outils dont il dispose pour ce qui est de la préparation des données en vue de la simulation. Utilisant d'une part un SIG pour la gestion des données et d'autre part un logiciel de simulation pour le calcul des débits, il souhaite un outil facilitant le transfert des données du SIG au logiciel de simulation.
- un chercheur d'une institution fédérale, dans le même contexte, va encore plus loin puisqu'il émet le souhait d'un outil qui l'assiste dans la tâche fastidieuse de création, à partir de données "réelles", d'un réseau fictif simplifié destiné à la simulation.
- un autre ingénieur, dans le cadre de son activité professionnelle, a pour tâche de répartir les dépenses entre les différents membres d'un syndicat d'assainissement. Un outil informatique qui mette à sa disposition les données nécessaires et l'assiste dans son calcul serait pour lui extrêmement précieux.
- deux intervenants (ingénieur et gestionnaire de l'eau) mentionnent l'intérêt d'un outil qui les aide à estimer la valeur de remplacement de différents ouvrages de réseau d'assainissement. Ce souhait s'insère lui aussi dans un processus particulier, ces deux intervenants ayant besoin d'informations leur permettant de planifier les investissements à prévoir pour la réfection des réseaux.
- deux intervenants du niveau cantonal éprouvent un intérêt pour un outil qui permette de déterminer, à la suite de l'introduction de substances dans le réseau d'assainissement, les cheminements suivis par ces substances jusqu'à leur exutoire dans le milieu naturel ou à une station d'épuration. Là encore, il ne s'agit pas d'un intérêt isolé pour une telle fonction, mais bien du soutien d'un processus métier, l'intervention en cas d'accidents majeurs, dans le cadre duquel les techniques employées révèlent leurs limites : *"Où part l'eau d'extinction, impossible de répondre avec les feuilles du cadastre"*.

D'autres exemples peuvent être brièvement cités, comme l'organisation des tournées d'inspection des canalisations par caméra vidéo, l'organisation des tournées de prélèvement d'échantillons d'eau aux stations d'épuration et dans les eaux superficielles, le calcul de la consommation annuelle en eau des ménages, le soutien à l'examen des demandes de permis de construire ou de rénovation. Ces demandes se rapportent aux différents processus métier qui font l'objet d'une description en annexe 1. Ainsi, pour être utile et utilisé, un SIAGEM doit soutenir les processus qui ont cours à l'heure actuelle. Un outil dont les fonctionnalités ne correspondent pas aux besoins actuels des utilisateurs n'a aucune chance d'être utilisé, même si ses fonctionnalités pouvaient conduire, d'un point de vue théorique, à une meilleure gestion du système "Eaux en milieu urbain".

Un SIAGEM doit faciliter le déroulement des processus métiers et ne doit pas imposer une manière spécifique de travailler. Au contraire, un SIAGEM doit être suffisamment souple pour que ses fonctionnalités puissent être combinées de différentes façons et s'adaptent aux manières de procéder des divers intervenants. En s'insérant ainsi dans les processus métier, un SIAGEM est accepté et utilisé par les intervenants. Un SIAGEM doit aussi, à la faveur du soutien qu'il apporte aux processus métier, enrichir sa base de connaissances relatives au système "Eaux en milieu urbain" (chapitre 2.1), et être assez souple pour intégrer progressivement d'autres fonctionnalités.

3.2 Les contraintes

3.2.1 Une grande diversité d'utilisateurs et d'utilisations

Le chapitre 2.2.3 a présenté les différentes catégories d'intervenants concernés par la problématique de la gestion des eaux sous l'angle du niveau organisationnel (particuliers, communes, associations, cantons, Confédération, bureaux d'étude, entreprises). Cette classification des différentes catégories d'intervenants peut être complétée par une classification selon le domaine d'activité : des intervenants aux compétences professionnelles différentes interagissent dans le cadre d'activités en relation avec la gestion des eaux : géomètres, écologues, hydrogéologues, hydrologues, responsables de stations d'épuration, architectes, entrepreneurs, pompiers, ingénieurs en charge de projets divers, propriétaires d'installations, etc. (Bernasconi, 1999). Les différents intervenants agissent donc à des niveaux de gestion différents, certains se préoccupant de planification à long terme, d'autres de gestion opérationnelle, etc., et ont des objectifs professionnels différents. De ce fait, ils ont besoin d'informations qui, bien que se rapportant au même système "Eaux en milieu urbain", sont différentes.

- Ces différences peuvent concerner la précision de l'information nécessaire : un ingénieur chargé d'effectuer une modélisation des écoulements dans le réseau d'assainissement n'a pas besoin d'une information très précise quant à l'emplacement et l'emprise des différents ouvrages du réseau d'assainissement, et sa vision de ce dernier consistera généralement en une version simplifiée de celle proposée par le cadastre des canalisations (Hermann, 1998). Il aura cependant besoin d'une information exacte quant à la topologie du réseau. Les personnes chargées de la planification d'un chantier par contre doivent pouvoir disposer d'une information très précise quant à l'emprise des ouvrages pour pouvoir intervenir à bon escient.
- Suivant son activité professionnelle, l'intervenant peut s'intéresser à des données différentes quant au même objet : une personne chargée de la planification de l'entretien des ouvrages du réseau d'assainissement éprouve un intérêt pour une information relative à la date de dernière intervention sur ces derniers, tandis qu'une personne en charge de la maintenance du cadastre de l'assainissement n'est pas intéressée par une telle information.
- Un même intervenant peut vouloir des informations totalement différentes sur un même objet, en fonction de ses besoins : ainsi, un ingénieur utilisera tantôt une vue en plan du réseau, afin d'y visualiser les cheminements des écoulements, tantôt des profils en long de manière à identifier les endroits du réseau présentant des sous-capacités du point de vue hydraulique. La position dans la hiérarchie organisationnelle influence le niveau d'agrégation de l'information désirée: au niveau communal un gestionnaire du réseau a besoin d'une information très détaillée quant au réseau

d'assainissement. Au niveau cantonal, les services en charge de la protection des eaux n'ont généralement besoin que d'une connaissance des infrastructures principales (réseau d'assainissement primaire), voire de statistiques agrégeant l'information au niveau de la commune, comme par exemple le pourcentage d'habitants raccordés au réseau d'assainissement ou la proportion de ce dernier conçue selon le mode séparatif. A un niveau supérieur (Confédération), des données détaillées et géoréférencées relatives aux systèmes d'assainissement ne sont pas demandées, seuls quelques indicateurs à caractère global étant utilisés (le pourcentage d'habitants non raccordés au réseau d'assainissement ou les proportions de réseau d'assainissement unitaire et séparatif par exemple). Ainsi, entre intervenants dont les activités professionnelles peuvent être situées sur un niveau organisationnel équivalent, une gestion intégrée se rapporte plutôt à une intégration de données (axe horizontal de la figure 22), tandis qu'entre niveaux organisationnels les flux d'information sont caractérisés par une modification de la granularité de l'information (axe vertical de la figure 22). Le degré d'agrégation de l'information utilisée a tendance à s'élever avec le niveau organisationnel, et inversement.

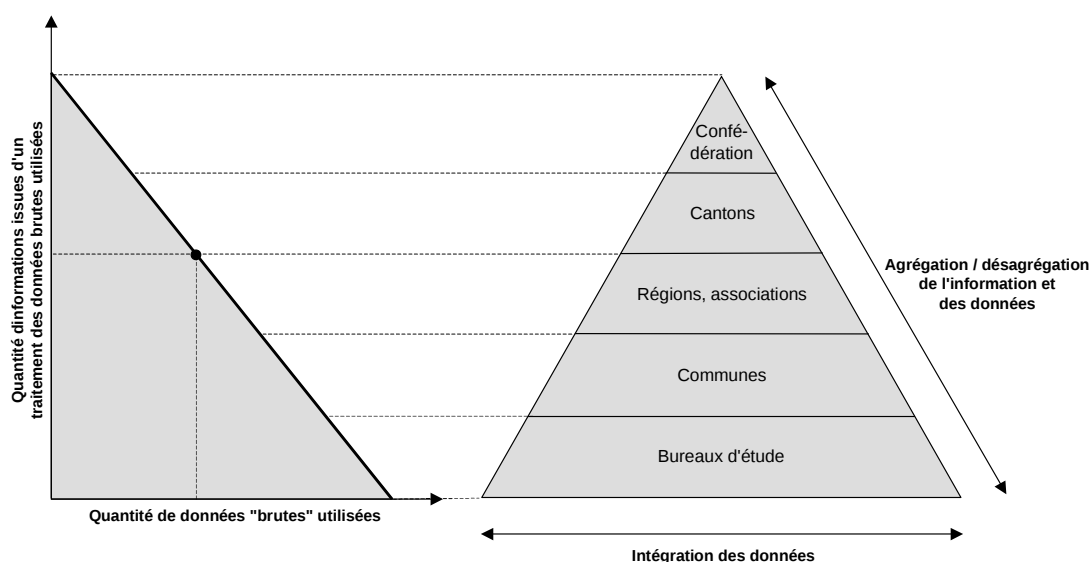


Figure 22. Les acteurs professionnels interagissant dans le contexte de la gestion des eaux : classification en niveaux organisationnels et mode d'utilisation du SIAGEM (adapté de AGW, 1997).

Ces quelques exemples montrent que de très nombreuses catégories d'intervenants sont impliqués dans la gestion des eaux en zones urbaines, et que les informations utilisées ou générées par les intervenants diffèrent quant à leur nature, à leur degré de détail et à leur mode de représentation. Ainsi :

- la structuration des données dans le mécanisme de gestion des données du SIAGEM (soit la base de données, ou BD) doit faire l'objet de soins attentifs de façon à pouvoir générer différents types d'informations;
- un SIAGEM ne peut offrir à la totalité des utilisateurs le même ensemble d'outils logiciels, mais doit proposer aux utilisateurs exactement ce dont ils ont besoin. Un SIAGEM doit donc pouvoir être construit de manière modulaire, certains de ces modules étant communs aux différents utilisateurs, et d'autres spécifiques à certains intervenants.

3.2.2 Des organisations de taille variable

Les intervenants dans le contexte de la gestion des eaux peuvent appartenir à des organisations ayant une taille et des moyens financiers très variés. Au niveau d'une petite commune, une seule personne peut être en charge de la maintenance d'un cadastre des canalisations tandis que, dans le cas d'une grande commune, quelques dizaines d'intervenants doivent pouvoir accéder à un tel cadastre (le système informatisé du cadastre des canalisations d'assainissement de la ville de Berne, par exemple, compte une vingtaine de postes de travail). Dès lors que des approches intégrées de la gestion des eaux seront effectivement mises en oeuvre avec le soutien d'un SIAGEM, on peut estimer que le nombre d'intervenants souhaitant interagir avec un tel outil augmentera. Certaines sociétés privées ou semi-privées, en charge de l'approvisionnement en eau potable ou de l'assainissement sur des portions de territoire débordant des frontières administratives, disposent déjà à l'heure actuelle de systèmes d'information comprenant un nombre de postes de travail très élevé. La société Thames Water (Grande-Bretagne) par exemple, l'une des plus grandes sociétés au monde s'occupant d'approvisionnement en eau potable et d'assainissement, dispose d'un système informatique consacré à la gestion de ses infrastructures auquel 1500 postes de travail peuvent être connectés simultanément (GEOEurope, 2000). A l'inverse de ce gigantisme, dans les bureaux d'étude, les systèmes d'information utilisés pour manipuler des données relatives aux eaux et infrastructures urbaines sont généralement des applications mono-utilisateur, qui quelquefois partagent des bases de données communes. Enfin, on cherche à favoriser, à l'heure actuelle, l'accès des particuliers à l'information par le biais d'Internet : il faut donc permettre que des entreprises ou des particuliers accèdent ponctuellement à des SIAGEM dans un but informatif (Brashear et al., 1999).

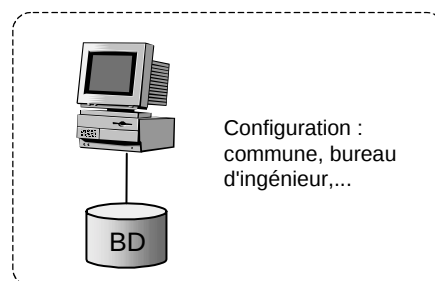


Figure 23. Configuration monoposte.

Ainsi, un SIAGEM doit pouvoir être adapté à de nombreuses configurations, en fonction de l'organisation au sein de laquelle il est implanté. Les illustrations 23, 24 et 25 présentent quelques types de configurations qui doivent pouvoir être envisagées. La figure 23 présente le cas de la configuration monoposte : on trouve sur un même ordinateur les applications et la BD. Les transferts de données vers "l'extérieur" se font par le biais de fichiers (par Internet ou par supports physiques). Ce type de configuration est adapté aux petits bureaux d'étude ou aux communes comptant peu d'habitants.

La figure 24 illustre une configuration que l'on peut trouver dans un bureau d'étude ou une institution (administration communale d'une agglomération relativement importante) d'assez grande taille, au sein de laquelle les responsabilités en matière de gestion des eaux sont réparties entre différents collaborateurs et les données réparties entre différentes BD. Dans l'exemple présenté, les applicatifs sont localisés sur les différents ordinateurs qui communiquent avec les BD.

Enfin, la figure 25 présente un cas plus complexe : à nouveau, les données sont réparties entre différentes bases de données, mais un serveur d'applications supporte la partie logicielle à même de retrouver les données disséminées dans les différentes bases de données, et contient une partie de la logique-métier de l'application. De plus, le système informatique est ouvert sur l'extérieur, des organisations tierces peuvent accéder au SIAGEM par le biais d'Internet, un serveur web se chargeant, entre autres, et avec l'aide du serveur applicatif, de générer l'information demandée par les utilisateurs extérieurs. En ce qui concerne ces derniers, plusieurs configurations sont également envisageables en fonction de leurs besoins. Certaines utilisations permettent un accès par le biais d'un browser Internet (client léger). La logique applicative est, elle, en majeure partie localisée du côté serveur, le côté client ayant pour tâche d'assurer la visualisation des informations. On peut également imaginer des applications informatiques plus conséquentes localisées du côté du client. Des calculs de simulation, par exemple, sont effectués par le bureau d'ingénieur, à partir de données qui sont "prélevées" dans le système d'information d'une administration cantonale ou d'une association d'assainissement,

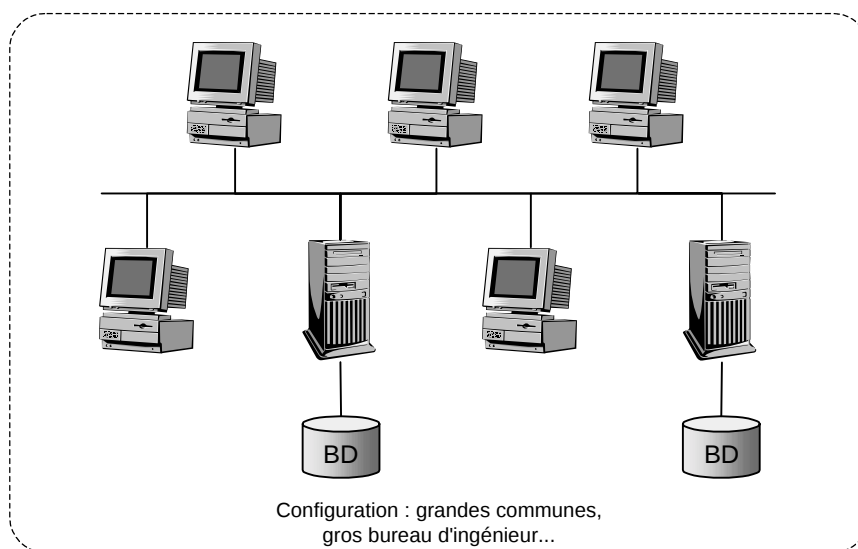


Figure 24. Configuration de type intranet.

Un SIAGEM doit ainsi être constitué suivant une architecture modulaire. Ses modules doivent pouvoir être répartis physiquement en différents noeuds des réseaux informatiques (intranets ou Internet) et être à même de communiquer entre eux. Des capacités de communication doivent donc être intégrées aux différents modules, et une réflexion à ce sujet doit être menée dès la conception de ces derniers.

Les utilisateurs d'un SIAGEM doivent pouvoir disposer de toute l'information dont ils ont besoin, accéder à toutes les fonctionnalités nécessaires, depuis leur poste de travail. La configuration des modules doit dissimuler à l'utilisateur toute la complexité relative à l'accès à des fonctionnalités s'exécutant sur des ordinateurs distants ou dont les données sont réparties entre différentes BD.

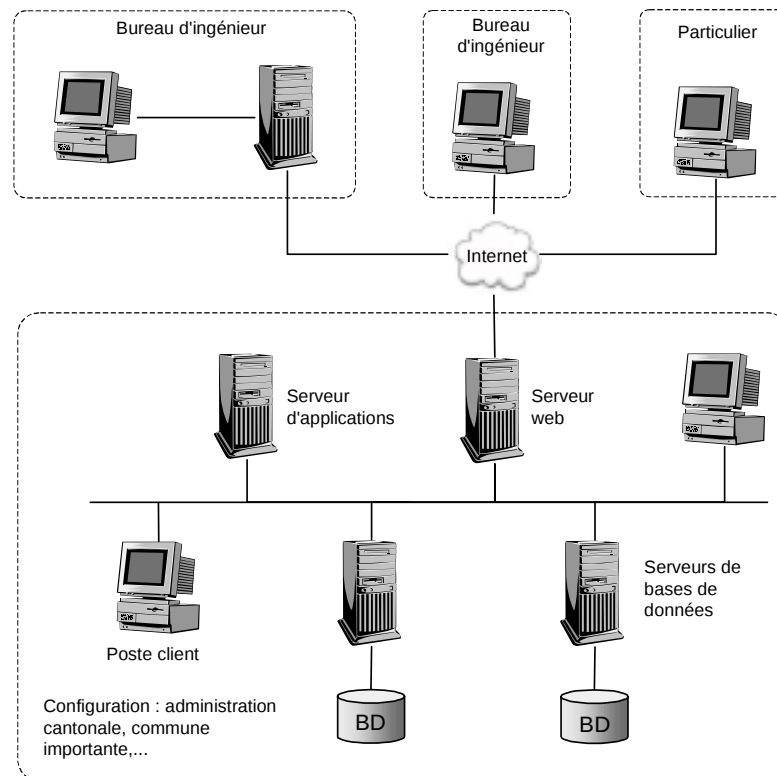


Figure 25. Configuration intranet avec possibilités d'accès par le biais d'Internet.

3.2.3 Des outils informatiques qu'il s'agit de prendre en compte

Comme cela a été montré au chapitre 2.4.2, de nombreux outils sont à l'heure actuelle utilisés par les différentes catégories d'intervenants dans le domaine de la gestion des eaux. Ces outils vont des applications généralistes, de type tableur, aux modèles de simulation les plus sophistiqués. Les entretiens qui ont été menés ont clairement montré qu'un SIAGEM doit être un système ouvert, laissant à ses utilisateurs la possibilité de combiner les outils de leur choix, et ce pour différentes raisons :

- Certains logiciels très spécifiques, par exemple des logiciels de simulation des écoulements, nécessitent un long apprentissage. Les utilisateurs ayant investi du temps dans cet apprentissage souhaitent pouvoir continuer à utiliser ces logiciels. De plus, il serait illusoire de vouloir "redévelopper" de tels outils : certains d'entre eux sont issus d'années de développement et d'améliorations successives, et un redéveloppement de tels produits dans le but de les intégrer dans un SIAGEM serait, pour des raisons économiques, impossible. Un SIAGEM doit donc disposer de mécanismes de couplage avec les outils utilisés habituellement, permettant ainsi, par exemple, de mener plusieurs simulations de comportement avec différents outils, tout en exploitant les mêmes données.
- D'autres logiciels, par exemple le tableur Excel, font partie du "référentiel commun" des différents intervenants. Ces derniers utilisent quotidiennement cet outil, qui s'adapte à des contextes différents. Un SIAGEM doit pouvoir échanger de l'information avec ce type d'outil, ou même pouvoir être couplé avec celui-ci.

Ainsi, un SIAGEM ne doit pas être conçu de manière à remplacer les outils qui sont utilisés à l'heure actuelle, mais être ouvert sur ces derniers, voire faciliter leur utilisation (figure 26). Un couplage de degré plus ou moins fort, et pouvant aller jusqu'à une intégration totale, entre le SIAGEM et différents logiciels de simulation, doit être possible. Le SIAGEM doit tenir le rôle de plate-forme d'échange d'information entre différents logiciels. Une telle possibilité permettrait par exemple de mener un grand nombre de simulations au moyen de différents logiciels, sur la base des mêmes données, et ce très rapidement.

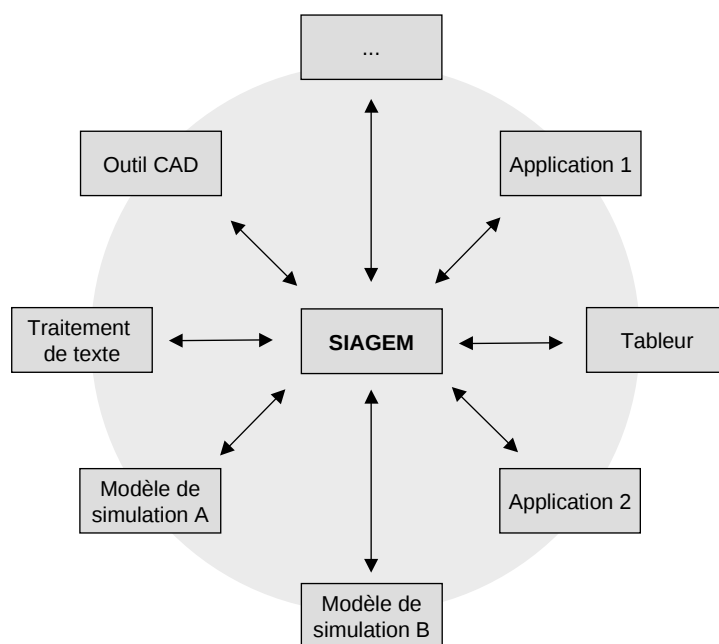


Figure 26. Un SIAGEM doit être ouvert sur les applications existantes et jouer le rôle de plate-forme d'échange d'information.

3.2.4 Un contexte en pleine évolution

A l'heure actuelle, dans le domaine de la gestion des eaux et des infrastructures urbaines, on trouve de nombreux outils informatiques qui ont pour fonction essentielle d'assurer la pérennité des données disponibles : les intervenants des différentes administrations ont pris conscience de l'utilité de disposer de données de qualité, et de la valeur patrimoniale que représentent les données acquises jusqu'à présent. Les outils informatiques existant actuellement, et dont l'objectif affiché est d'apporter un soutien à la gestion, servent donc essentiellement à l'acquisition, à la numérisation et à la visualisation de données. De fait, peu de réflexions ont été menées quant à la manière d'exploiter et de valoriser les données acquises. L'usage d'indicateurs par exemple, pourtant abondamment discuté dans la littérature scientifique (Azar et al., 1996; Lundin et al., 1997; Lundin et al., 1999; Oestereich, 1998), ou l'utilisation de modules fonctionnels ayant pour objectif l'amélioration des processus de gestion (optimisation des tournées de curage des canalisations d'assainissement par exemple), est rare. Pour ces différentes raisons, les intervenants dans le domaine de la gestion des eaux n'ont pas une vision claire de leurs attentes envers ce type d'outil informatique, ce qui rend l'identification des besoins très difficile. De plus, les attentes vis-à-vis de tels outils évoluent très vite, de nouveaux besoins sont suscités par l'utilisation des outils existants et des fonctionnalités disponibles. Enfin, nous avons vu que la gestion des eaux en zones

urbaines recouvre un ensemble d'activités qui sont à l'heure actuelle en pleine évolution, ce qui rend les acquis instables et entraîne des modifications des besoins et attentes vis-à-vis des outils informatiques.

Ces différents points font qu'une analyse des besoins et la mise en place de spécifications précises relatives à un SIAGEM sont, dans le contexte actuel, extrêmement difficiles. En particulier, l'élaboration d'un cahier des charges, incluant un dictionnaire des données à gérer et un catalogue des fonctionnalités à implémenter, est totalement impossible.

Un SIAGEM doit donc être un outil informatique évolutif, pouvant être adapté à différentes catégories d'intervenants et à même d'évoluer au rythme des changements des principes et méthodes qui sous-tendent la gestion des eaux en zones urbaines.

3.2.5 Des besoins de personnalisation

Trois points auxquels il convient d'être attentif lors de la conception d'un SIAGEM sont abordés ici :

- un SIAGEM doit pouvoir être adapté spécifiquement aux besoins de chaque intervenant;
- un SIAGEM doit pouvoir être personnalisé par les intervenants qui le souhaitent;
- un SIAGEM ne doit pas imposer une manière unique de travailler.

Un SIAGEM est appelé à soutenir différentes catégories d'intervenants dans le cadre de leur activité professionnelle. Il a été constaté cependant qu'au sein d'une même catégorie d'intervenants sont formulées des exigences quant à la représentation de l'information utilisée, le fond de cette dernière n'étant pas généralement mis en question. Seule une personnalisation des SIAGEM, selon les vœux spécifiques des intervenants, rendra possible l'appropriation par ces derniers d'un tel outil. Un intervenant mentionne le fait que certain de ses collègues, plutôt que d'utiliser les logiciels existants, investissent beaucoup de temps pour développer eux-mêmes des outils spécifiquement adaptés à leurs besoins. Des concepteurs de systèmes d'information ont, quant à eux, rapporté la difficulté qu'il y a à concevoir des systèmes d'information pour la gestion des infrastructures urbaines : bien que se déclarant ouvert à une standardisation au niveau de la manière de présenter l'information, les intervenants réclament tellement d'adaptations spécifiques qu'une réduction des coûts de développement par le biais de la standardisation des outils d'aide à la gestion devient dans les faits impossible. Enfin, mentionnons un dernier exemple de la difficulté que représente le développement d'un "modèle unique" d'outil d'aide à la gestion : ayant pour objectif l'amélioration de la gestion de l'assainissement, de nombreuses administrations élaborent à l'heure actuelle des modèles de documents graphiques pour représenter l'information utilisée dans ce contexte. Il a été constaté que, même si elles ont des tâches similaires, les différentes institutions s'inspirent des modèles existants mais ne les reprennent pas tels quels : elles les adaptent à leurs besoins et habitudes.

Deux catégories d'utilisateurs potentiels de SIAGEM ont été rencontrés : d'une part ceux qui souhaitent pouvoir disposer d'outils informatiques immédiatement fonctionnels et ne veulent pas avoir à se préoccuper des aspects informatiques (*"Mon métier, c'est l'hydrologie et pas l'informatique"*), et d'autre part ceux qui souhaitent disposer d'un certain pouvoir sur l'outil logiciel, et donc disposer de la possibilité d'en personnaliser les fonctionnalités. Un SIAGEM doit aussi permettre à cette catégorie d'utilisateurs d'intervenir.

Enfin, un SIAGEM ne doit pas imposer un déroulement rigide des processus métiers : une administration cantonale en charge de la protection des eaux a cité l'exemple d'une tentative d'informatisation ayant échoué à cause de la rigidification des processus-métier qu'une telle informatisation aurait entraînée. La démarche du consultant avait été interprétée comme une volonté d'imposer une certaine manière de travailler, ce qui avait conduit à l'échec du projet.

Ces quelques lignes montrent qu'un SIAGEM ne peut être conçu comme un outil vendu "clé en main", mais doit être un outil personnalisable, soit par les personnes en charge de sa mise en place, soit, dans une certaine mesure, par ses utilisateurs eux-mêmes.

3.2.6 Des modèles de données complexes

Un SIAGEM, comme tout SIRS, comporte nécessairement une BD ayant été structurée selon un modèle conceptuel précis. Comme nous l'avons vu (chapitre 2.4.2.4), il existe à l'heure actuelle différents modèles de données se rapportant à différents compartiments du système "Eaux en milieu urbain". Des modèles de données standardisés, communs à différents intervenants, facilitent grandement les échanges de données entre ces derniers. Ce contexte permet d'évoquer la notion d'interopérabilité des données : il existe des langages de description de données structurées, qui peuvent être mis en oeuvre pour l'échange de données entre applications. En ce qui concerne les données localisées, on citera Edigéo, Interlis ou encore les extensions de la norme XML prenant en compte la dimension spatiale (GML¹ par exemple). Aborder le problème de l'informatisation des systèmes d'information pour la gestion des eaux uniquement par le biais de la standardisation des modèles de données rencontre cependant certaines limites. Celles-ci sont discutées dans les lignes qui suivent à partir des observations faites quant à l'utilisation du modèle de données VSA-DSS proposé par l'Association Suisse des Professionnels de la Protection des Eaux VSA (VSA, 1999) :

- un modèle de données destiné à être utilisé par différentes catégories d'acteurs, et élaboré après une analyse des besoins de ces derniers, intègre différentes visions. Un tel modèle est de ce fait certes extrêmement puissant, dans la mesure où il permet la gestion de toutes les informations nécessaires, mais il est également très complexe (figure 27). Un utilisateur souhaitant extraire des données relativement à un élément bien précis de la base de données aura beaucoup de mal à y trouver l'information dont il a besoin, cette dernière étant disséminée dans une BD intégrant de nombreuses entités dont il ne connaît pas la signification : une implémentation en relationnel d'un modèle conceptuel de données conçu pour satisfaire aux besoins de multiples utilisateurs comprend un grand nombre de tables, et l'extraction d'information d'une telle BD devient vite compliquée. De plus, si la théorie veut qu'un modèle conceptuel de données représente la réalité de l'utilisateur, en fait on constate que le modélisateur dissèque cette réalité pour en extraire un ensemble cohérent et non-redondant d'entités et de relations dans lequel l'utilisateur ne reconnaîtra pas forcément ses construits. Par exemple, pour une canalisation, l'utilisateur d'une base de données structurée selon les propositions du VSA doit répartir les données entre au moins six tables relationnelles (illustration 28). On conçoit aisément que la gestion de données relatives à l'ensemble d'un réseau d'assainissement conduise à la manipulation d'un nombre beaucoup plus considérable de tables. Ainsi, l'implémentation relationnelle du modèle proposé par le VSA comporte 86 tables principales,

1. <http://www.opengis.org/techno/specs.htm>

ce qui rend son utilisation par un non-spécialiste difficile.

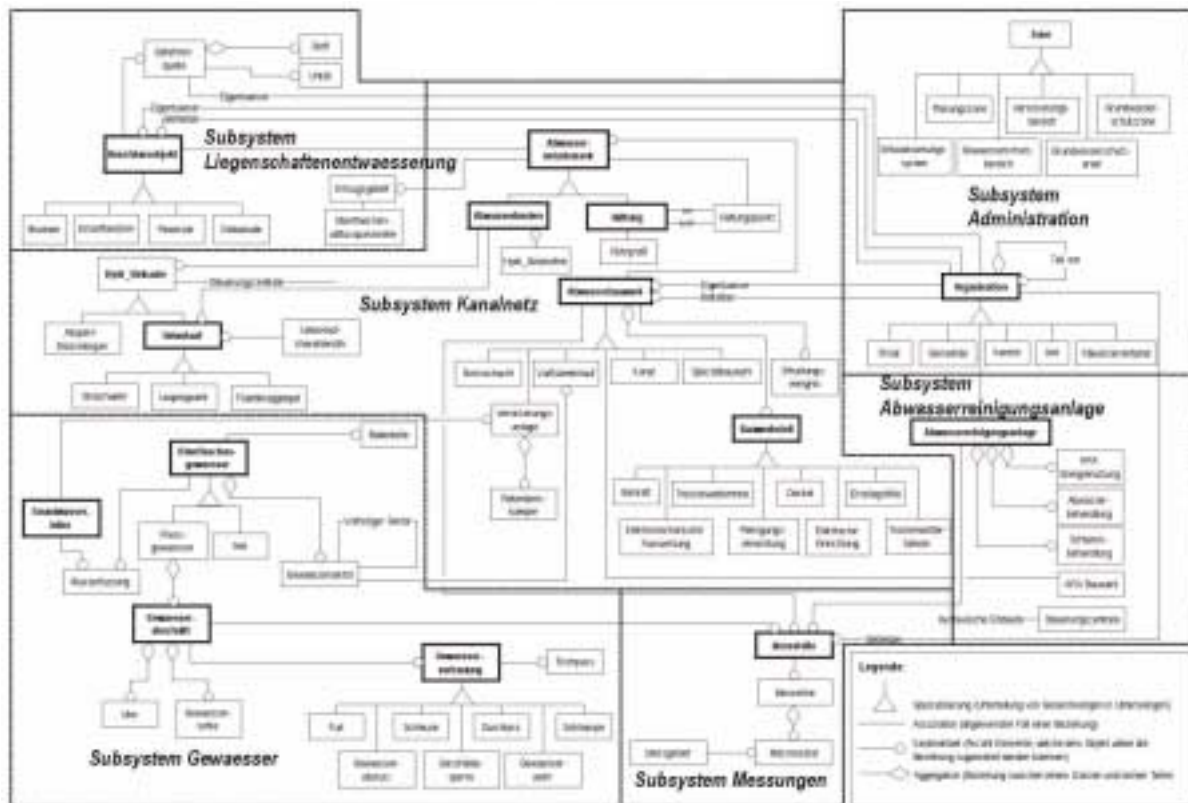


Figure 27. Vue complète sur le modèle de données proposé par l'ASPEE (VSA, 1999).

- Devant une telle complexité, de nombreux utilisateurs préfèrent n'employer qu'une partie de la structure de données, quitte à, dans certains cas, "dénaturer" la modélisation de la réalité sur laquelle se base le modèle de données. Pour la structure de données VSA-DSS par exemple, un ouvrage de réseau d'assainissement est un agrégat comportant, entre autres, des noeuds et des tronçons de réseau permettant de représenter la topologie réseau, mais n'ayant pas d'existence propre. Or certains utilisateurs de la structure de données n'utilisent, pour la gestion des données relatives au réseau d'assainissement, que les entités noeuds et tronçons de réseau, les informations liées à ces dernières suffisant à leurs besoins. Cependant, une telle approche conduit inmanquablement à des problèmes lors de l'échange d'informations, puisque le fait de n'utiliser qu'une partie de la structure de données correspond à une interprétation personnelle de la modélisation, interprétation qui probablement ne sera pas la même que celle d'un autre utilisateur de la base de données.
- De plus, la complexité de la structure de données rend le développement d'outils logiciels exploitant ces données très difficile. Il a été constaté dans le contexte de ce travail que le simple développement d'un formulaire informatisé pour l'acquisition et l'enregistrement de données conformément au modèle VSA-DSS devient, en raison de la complexité de ce dernier, une tâche de longue haleine si elle est abordée de manière conventionnelle.
- Enfin, dans un domaine comme celui de la gestion des eaux, il existe déjà de nombreuses applications informatiques qui intègrent une base de données structurée selon un modèle spécifique. La diffusion d'un modèle commun comme VSA-DSS se heurte à ces structures, qui constituent déjà le noyau d'applications opérationnelles dont le développement a nécessité parfois de gros

investissements et auxquelles les intervenants se sont habitués. Les utilisateurs de ces structures spécifiques sont très réticents à l'idée de migrer leurs données vers une structure de données commune.

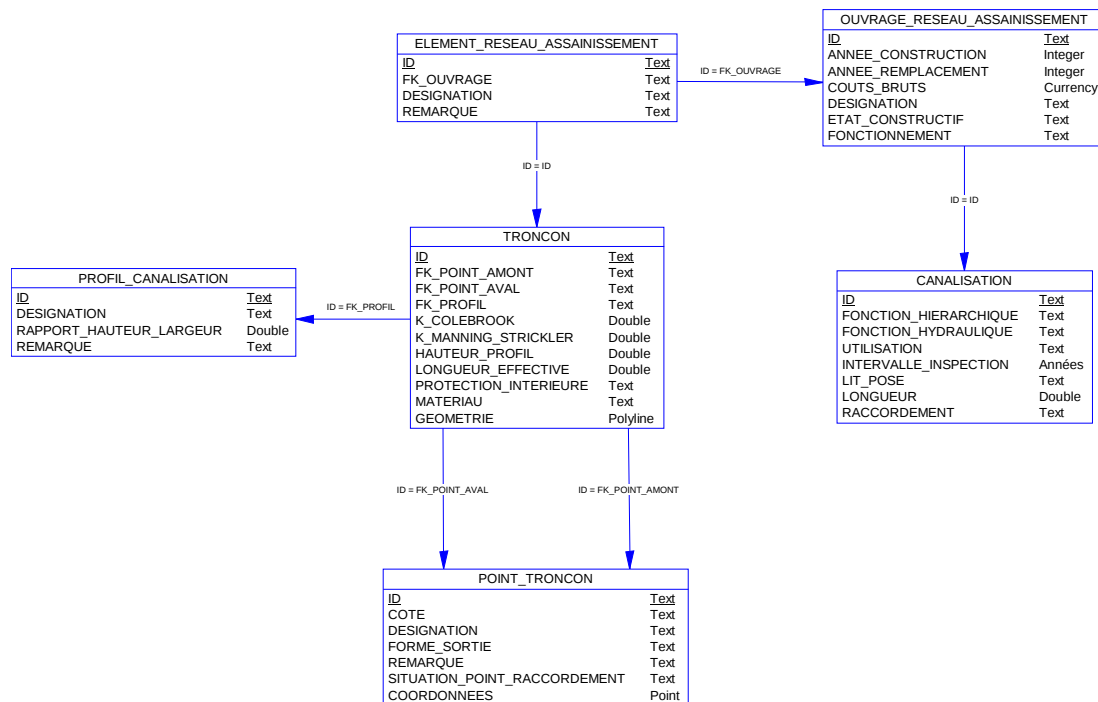


Figure 28. Le modèle relationnel nécessaire à la gestion des canalisations, conformément à la structure de données VSA-DSS (VSA, 1999).

De ces observations peuvent être tirées trois conclusions :

- Un SIAGEM doit offrir à ses utilisateurs la possibilité de manipuler des concepts conformes à la perception qu'ils en ont. Parallèlement à cela, un SIAGEM doit s'assurer que la décomposition en données des concepts et informations dont l'enregistrement est demandé se fasse de manière cohérente et conforme au modèle structurant la base de données.
- Nous avons vu que les besoins en matière de fonctionnalités évoluent, et donc également les besoins en termes de données dont le SIAGEM doit être en mesure d'assurer la gestion. De ce fait, un SIAGEM doit être conçu de manière à ce que les modèles de données sur lesquels il se base puissent eux aussi évoluer, et ceci sans que soient affectées les fonctionnalités destinées aux utilisateurs non concernés par ces changements.
- Enfin, nous avons vu que de nombreux modèles de données existent dans le contexte de la gestion des eaux. Certains sont promus par des associations professionnelles, intégrés dans des applications existantes, et ont fait l'objet de réflexions approfondies. Un SIAGEM doit pouvoir être adapté de manière à gérer ces données de manière conforme aux propositions de modèles de données préexistants, qui constituent de ce fait une contrainte à considérer.

3.2.7 La nécessité de pouvoir s'appuyer sur des processus de développement rapides

Une telle évolution est générale dans le domaine des projets informatiques, et est mentionnée par de

nombreux auteurs : il n'est généralement plus envisageable, à l'heure actuelle, d'initier des projets informatiques dont les premiers résultats ne sont visibles qu'au bout de plusieurs années. En raison de différents échecs relatifs à des projets de longue durée qui avaient nécessité des investissements considérables, les commanditaires de tels développements exigent d'avoir une vision claire de l'avancée des travaux, voire que des versions incomplètes, mais opérationnelles, des outils informatiques développés, leur soient livrées à intervalles réguliers. De plus (en tout cas en ce qui concerne le secteur privé), la pression concurrentielle ainsi que les évolutions technologiques rendent nécessaire une accélération des cycles de développement d'outils informatiques. Plus spécifiquement, dans le domaine de la gestion des eaux, nous avons vu que les besoins évoluent vite, poussés par les avancées technologiques et les modifications des paradigmes et de la législation. Des cycles de développement de longue durée présentent ainsi le risque de déboucher sur des outils informatiques inadaptés (illustration 29), les besoins des utilisateurs ayant changé depuis leur définition initiale. Il est donc indispensable de mettre en oeuvre des démarches qui permettent des livraisons très rapides.

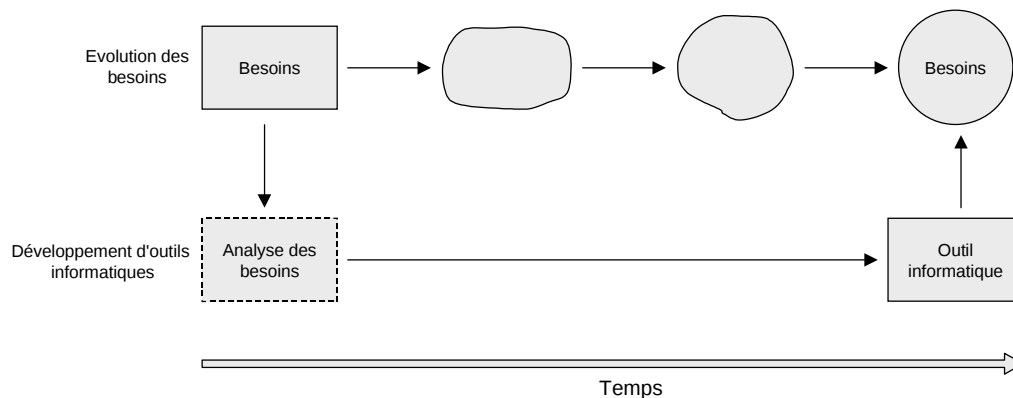


Figure 29. Cycle de développement "classique" d'un système d'information : l'outil développé est déjà caduque au moment de sa mise en service car il ne répond plus aux besoins des utilisateurs (adapté de Taylor, 1995).

3.2.8 Des moyens financiers limités

Les utilisateurs potentiels de systèmes informatisés dans le domaine de la gestion des eaux sont principalement des institutions publiques (administrations cantonales ou communales) ou des bureaux d'étude travaillant dans le domaine du génie civil, du génie rural, de l'environnement ou de la mensuration. Les premiers sont à l'heure actuelle soumis à un régime d'austérité, les dépenses publiques ayant tendance à diminuer, et les seconds sont généralement des structures de petite taille, dont les moyens financiers sont relativement limités. De ce fait, les intervenants sont très sensibles aux coûts de développement des logiciels informatiques, et ce bien que ceux-ci ne constituent qu'une faible part des dépenses à consentir par rapport à l'ensemble des dépenses occasionnées par la mise en oeuvre d'un système d'information (tableau 7). De plus, comme cela a été mentionné plus haut, les activités actuelles en ce qui concerne les technologies de l'information dans le contexte de la gestion des eaux sont essentiellement des activités d'acquisition et de numérisation des données, et peu de réflexions ont été menées quant à l'utilisation de ces informations. Les intervenants ne voient de ce fait pas la nécessité de financer le développement de fonctionnalités qui, certes, leur apportent des informations intéressantes, mais dont ils ne mesurent pas pleinement les bénéfices : ils sont davantage intéressés par des produits informatiques qui leurs permettent d'une part de mener facilement à bien les tâches qui sont les leurs, et,

d'autre part, de remplir les obligations légales.

Matériel	7 %
Logiciel	12 %
Données	23 %
Adaptation	30 %
Formation	28 %

Tableau 7. Coûts relatifs pour une configuration "importante" (un serveur avec trente-cinq stations-client), d'après Maguire (1999).

Le tableau 7 présente la répartition des dépenses consécutives à la mise en place d'un SIRS. Ce tableau montre que les coûts d'adaptation du logiciel sont énormes, et que les coûts d'introduction de données au sein du SIRS sont également relativement importants. Par contre, les coûts du logiciel sont relativement faibles. De plus, la part des coûts attribuée aux données est ici probablement sous-estimée, l'auteur n'ayant pas considéré l'acquisition des données proprement dite. Donc, d'une part il faut bien se rendre compte qu'en comparaison de la valeur que représentent les données les coûts du logiciel sont faibles, et que d'autre part des économies considérables pourraient être réalisées si les systèmes proposés pouvaient facilement s'adapter aux besoins des intervenants.

En conclusion, un SIAGEM est, de par l'envergure de la problématique à laquelle il s'attelle, un système extrêmement complexe dont les coûts de développement peuvent être très élevés. Et comme tous les intervenants ne sont pas prêts à investir de gros montants dans des développements logiciels, il conviendrait de mettre en place des concepts permettant le développement de SIAGEM adaptés aux moyens de chacun.

3.3 Les objectifs

Au vu des différents points ayant été développés dans les chapitres précédents, la mise en place de concepts généraux pour le développement de SIAGEM peut sembler délicate. Nous sommes en effet arrivés aux conclusions suivantes :

- en raison de l'évolution des paradigmes sous-tendant la gestion des eaux et des méthodes de travail, un tel outil doit être évolutif, c'est-à-dire que de nouvelles fonctionnalités doivent pouvoir lui être adjointes, et que les fonctionnalités existantes doivent pouvoir être modifiées;
- un SIAGEM doit être personnalisable, c'est-à-dire qu'il doit pouvoir être configuré de manière à présenter à ses utilisateurs des fonctionnalités répondant à leurs souhaits, sous la forme désirée;
- un SIAGEM doit pouvoir être adapté à différentes configurations informatiques (mono-poste, client-serveur, applications distribuées);
- l'interopérabilité avec les applications informatiques utilisées par les différents acteurs doit être aisée: par conséquent un SIAGEM doit être ouvert;
- un SIAGEM doit faciliter l'interaction avec des bases de données structurées selon des modèles de données complexes, et rendre simple et intuitive la manipulation des concepts du domaine de la gestion des eaux; un SIAGEM doit supporter des évolutions de structures de données sans que cela

gêne les utilisateurs. Une structure de données particulière doit pouvoir être imposée lors de la construction d'un SIAGEM, sans que l'utilisateur et les outils qu'il a l'habitude d'utiliser en soient affectés;

- des applications opérationnelles doivent pouvoir être rapidement mises sur pied, et elles doivent aussi pouvoir être rapidement modifiées.
- les coûts de développement doivent être faibles.

L'objectif que se fixe la présente recherche est la proposition de concepts permettant la mise en place de SIAGEM respectant les critères qui viennent d'être énoncés. Une fois ces concepts mis en place, il sera proposé une méthode qui permettra de développer de tels outils.

Ce chapitre a montré quels sont les critères auxquels il faut être attentif lors du développement des concepts relatifs à un SIAGEM. Le chapitre qui suit fait des propositions quant à l'architecture générale de tels outils, explore les possibilités qu'offrent les technologies logicielles actuelles, pour finalement présenter une proposition concrète.

Une conception modulaire des outils d'aide à la gestion des eaux en milieu urbain

Le principe d'une conception des SIAGEM par assemblage de composants préexistants est proposé. Les différentes technologies logicielles supposées être utiles dans ce contexte sont explorées. Enfin, un concept précis est élaboré : baser le développement de SIAGEM sur des assemblages de composants métier. Les caractéristiques qui doivent être celles de tels composants sont présentées et il est montré comment les exigences et contraintes présentées au chapitre 3 sont respectées.

4.1 Principe général

Des exigences en regard des SIAGEM ont été formulées au chapitre précédent. Ces exigences paraissent impossible à concilier :

- il est demandé une approche globale de la gestion des eaux, alors que cette dernière recouvre une multitude de processus, généralement menés de façon indépendante les uns des autres;
- les besoins des utilisateurs et l'environnement technologique changeant rapidement, des outils disponibles rapidement et évolutifs sont nécessaires;
- les outils logiciels, de par les exigences formulées, seront certainement relativement complexes (capacité à être répartis de plusieurs manières sur les différentes machines des réseaux informatiques, BD structurées selon des modèles compliqués), mais leur mise en oeuvre doit rester simple;
- enfin, à cela s'ajoute le fait que les produits proposés doivent rester abordables car les organisations susceptibles de les mettre en oeuvre disposent de moyens limités. Malgré cela, un SIAGEM doit être personnalisable.

Malgré ces contraintes, une approche qui permette de concilier ces différentes exigences peut être envisagée. Le principe en est tout d'abord présenté de manière succincte, puis les différents points seront détaillés dans les chapitres qui suivent.

- Un SIAGEM doit être conçu de manière modulaire, être constitué d'un assemblage de composants. Chacun de ces composants doit être pleinement opérationnel et à même d'accomplir un ensemble de fonctionnalités cohérentes qui soient utiles à un intervenant particulier.
- En raison de l'immensité de la tâche considérée, tous les composants nécessaires à la mise sur pied de

SIAGEM ne peuvent être conçus simultanément. Ces composants doivent pouvoir être développés successivement, et chacun d'entre eux doit permettre la mise en place d'une "portion" de SIAGEM traitant d'un aspect particulier, soutenant un processus particulier du domaine de la gestion des eaux en zones urbaines.

- Afin que son développement en vaille la peine, un tel composant doit être réutilisable : ses fonctionnalités doivent donc faire l'objet d'une réflexion approfondie ayant pour but de vérifier qu'elles présentent un intérêt pour un ensemble raisonnable d'intervenants. Un composant doit donc posséder un degré certain de standardisation.
- Chaque composant doit être autonome. Nous avons vu qu'un SIAGEM peut devoir être réparti sur différents noeuds d'un réseau informatique (chapitre 3.2.2). Un composant doit donc lui-même pouvoir être réparti en fonction des besoins et de la configuration du SIAGEM, et ainsi être constitué de "parties" investies de différentes missions - présentation de l'information et interactions avec les utilisateurs, traitement de l'information, archivage des données, etc. - à même de communiquer entre elles.
- Les différents composants standardisés doivent pouvoir être facilement assemblés, en fonction des souhaits et besoins des utilisateurs, de manière à constituer des applications logicielles parfaitement adaptées à leurs besoins : de l'assemblage de ces composants standardisés naissent des SIAGEM opérationnels et spécifiques aux différents utilisateurs.
- Des modifications doivent pouvoir être apportées à un SIAGEM de manière simple et rapide : un composant doit donc pouvoir être facilement remplacé dans un SIAGEM. De même, de nouveaux composants doivent pouvoir être ajoutés à un SIAGEM existant. Une approche par composants, respectant ce souhait, offre la possibilité de construire des outils capables d'évoluer en fonction de la technologie ainsi que des besoins et desiderata de chaque intervenant.

Notre propos peut être illustré par un parallèle avec l'industrie automobile : les acheteurs d'une automobile ont tous un objectif général commun, à savoir acquérir une certaine autonomie, pouvoir se déplacer. Leurs critères de choix vis-à-vis d'un véhicule et les utilisations qu'ils en feront peuvent être très variés. Pourtant, il ne viendrait pas à l'idée d'un constructeur automobile (mis à part pour certains marchés spécifiques) de construire pour chacun de ses clients un véhicule sur mesure, en partant de zéro. Les différentes pièces qui composent une automobile sont des éléments standardisés (châssis, moteurs, pneus, carrosseries, pièces mécaniques, etc.), qui sont assemblés de manière à produire un véhicule conforme aux attentes du client. À titre d'exemple, un seul modèle de la marque Citroën peut être décliné en 260'000 combinaisons différentes, et ce bien qu'étant constitué d'éléments standardisés¹. Dans le domaine des applications logicielles, on a eu longtemps tendance à partir de zéro pour chaque projet informatique, l'application étant intégralement développée au moyen d'un langage de programmation particulier. Petit à petit, des éléments standardisés, bibliothèques de fonctions, modules de code et autres composants logiciels, ont été intégrés dans les développements effectués, et ont contribué à simplifier la conception de logiciels.

Nous proposons de pousser cette logique plus loin et de développer un certain nombre de composants logiciels standardisés spécifiquement destinés à la conception de SIAGEM. Une telle démarche aurait pour avantage de diminuer les coûts de construction de ces outils par un effet d'économies d'échelle, de réduire les temps nécessaires à leur développement, un assemblage de composants préexistants

1. Source : prof. Marino Widmer, Ecole des HEC, Lausanne

nécessitant logiquement moins de temps que la construction intégrale d'un système d'information, et enfin de produire des applications de qualité, induisant des coûts de maintenance largement diminués, puisque des composants standardisés, destinés à être réutilisés, seraient conçus avec beaucoup de soin et testés de manière approfondie.

Une telle approche ne peut cependant se justifier que s'il est possible de déterminer des "portions" de SIAGEM susceptibles de faire l'objet d'une standardisation, et si le nombre de SIAGEM dans lesquels de tels composants seraient susceptibles d'être intégrés est suffisamment grand pour justifier leur existence.

Un concept a ci-dessus été proposé. Les chapitres qui suivent explorent les technologies informatiques disponibles en regard des propositions ayant été faites, et montrent comment une utilisation adéquate de ces technologies ouvre la voie à la mise en place de composants qui répondent aux exigences formulées.

4.2 Paradigme orienté objet

4.2.1 Principe

Deux approches existent dans le contexte du développement d'applications logicielles : l'approche fonctionnelle et l'approche orientée objet. Une approche fonctionnelle implique l'identification et la description des traitements que doivent subir les données en vue de produire l'information désirée, et la segmentation de ces traitements en une succession de parties suffisamment petites pour être maîtrisables. Une approche fonctionnelle conduit ainsi à une nette séparation des données et des traitements. Un exemple de méthode de développement basée sur une approche fonctionnelle est celle proposée par Gane et Sarson (1979). L'implémentation d'un logiciel conçu sur la base d'une telle méthode se fait généralement à l'aide d'un langage de programmation structurée (Fortran, Cobol, Basic, Pascal, etc). Différents auteurs s'accordent à dire que l'approche fonctionnelle conduit à des outils logiciels dont la maintenance est difficile : un système logiciel qui résulte d'une telle approche est difficile à comprendre, et lorsque des modifications doivent être faites, elles entraînent des interventions en de nombreux points.

Dans l'approche orientée objet, tout système est considéré comme *"une société d'objets qui coopèrent"* (Kettani et al., 1998). Cette coopération se fait par le biais de messages que les objets s'échangent (figure 32). Tout système peut faire l'objet d'une modélisation orientée objet, qu'il s'agisse d'un système naturel ou d'une entreprise. Un objet est une abstraction de quelque chose d'identifiable, ce quelque chose pouvant être un élément du monde réel (une canalisation, une personne, un bassin versant), un concept comme un modèle de calcul, ou un élément propre au monde du logiciel comme une fenêtre ou un bouton d'interface graphique. Au cours du déroulement d'une méthode de développement suivant une approche orientée objet, le modélisateur commencera par établir des modèles du système "réel" (entreprise ou système naturel) qui l'intéresse, et définira donc des objets "abstraction du monde réel", puis, dans la suite du processus de modélisation, définira, à partir de ceux-ci, des objets logiciel, qui seront des "images logicielles" des objets "abstractions du monde réel" (figure 30).

Le développement d'une application selon une approche orientée objet conduit donc à une

correspondance très étroite entre le système "réel" et le système "logiciel" : on manipule dans le système "logiciel" des objets logiciels qui sont des images d'objets du monde réel. Cette correspondance entre réalité et application logicielle rend la structure interne de ces dernières plus intuitive, et donc plus facile à comprendre (pour le développeur). Cette raison, associée à d'autres caractéristiques de l'approche orientée objet qui seront discutées plus loin, fait que celle-ci facilite d'une part la construction et la maintenance d'applications logicielles complexes, et d'autre part rend la réutilisation de portions du logiciel plus aisée.



Figure 30. Deux niveaux d'abstraction des concepts du monde réel.

Les techniques de modélisation et de programmation orientées objet sont le sujet de très nombreux ouvrages (Booch et al., 1999; Bouzeghoub et al., 1997; Jacobson et al., 1992; Kurata, 1998; Rumbaugh, 1994). Ce sujet ne sera de ce fait pas approfondi ici. Seuls quelques concepts de base, importants pour la suite de ce travail, sont brièvement rappelés.

Une approche peut être qualifiée d'orientée objet pour autant qu'elle réunisse quatre caractéristiques : l'identité, la classification, la généralisation et le polymorphisme (Rumbaugh, 1994) :

- Un objet est une chose ou un concept qui doit être identifiable. Un objet est caractérisé par une identité, un état et un comportement. Son identité constitue le moyen de l'identifier par rapport aux autres objets, son état est défini par l'ensemble des valeurs que prennent ses attributs à un instant donné, et son comportement précise *"la manière dont l'objet agit et réagit aux divers messages qui lui parviennent de son environnement"* (Kettani et al., 1998).
- Les objets de même nature peuvent être "classifiés" (Bouzeghoub et al., 1997), c'est-à-dire qu'une classe factorisant les caractéristiques communes de ces objets peut être définie. Une classe est une abstraction décrivant un ensemble d'objets aux caractéristiques communes. Elle constitue une sorte de modèle à partir duquel des objets peuvent être instanciés, ces derniers étant quelquefois appelés instances de classe (figure 31). Une classe est caractérisée par ses attributs et ses opérations, qui ensemble constituent ses propriétés. Les opérations d'une classe sont celles applicables à tout objet instancié à partir de cette classe.
- Une relation de généralisation est une relation taxonomique entre une classe spécialisée et une classe plus générale. Les caractéristiques des classes plus générales sont héritées par les classes spécialisées.
- Le polymorphisme implique que la même opération peut prendre différentes formes selon les classes qui la proposent (Rumbaugh, 1994).

Ainsi, à la différence de l'approche fonctionnelle, une approche orientée objet conduit à une liaison très forte entre les données et les traitements qui se rapportent à un même objet. Une telle cohérence favorise, comme nous l'avons dit, la compréhension de systèmes orientés objet, et en facilite la modularité et la

maintenance.

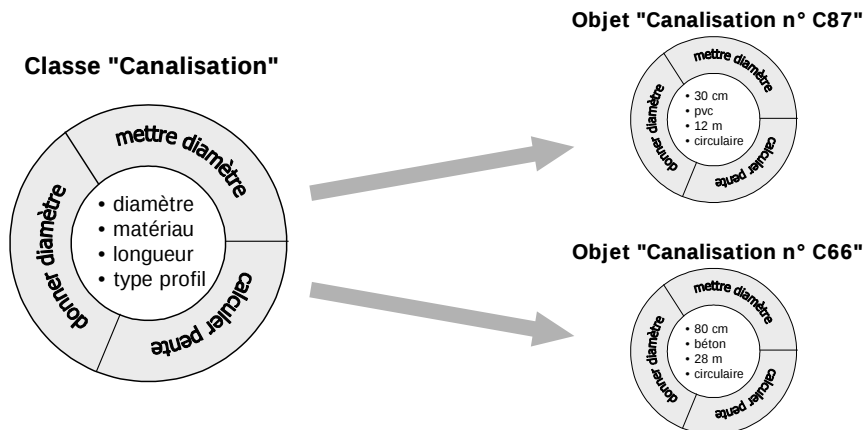


Figure 31. La classe "Canalisation" donne lieu à deux instanciations : les objets "canalisation C87" et "canalisation C66".

Enfin, Rumbaugh (1994) note que "les systèmes informatiques bâtis sur la structure de l'objet sont plus stables sur le long terme, car à mesure que les besoins évoluent, les caractéristiques fournies par un objet sont beaucoup plus stables que les façons dont il est employé".

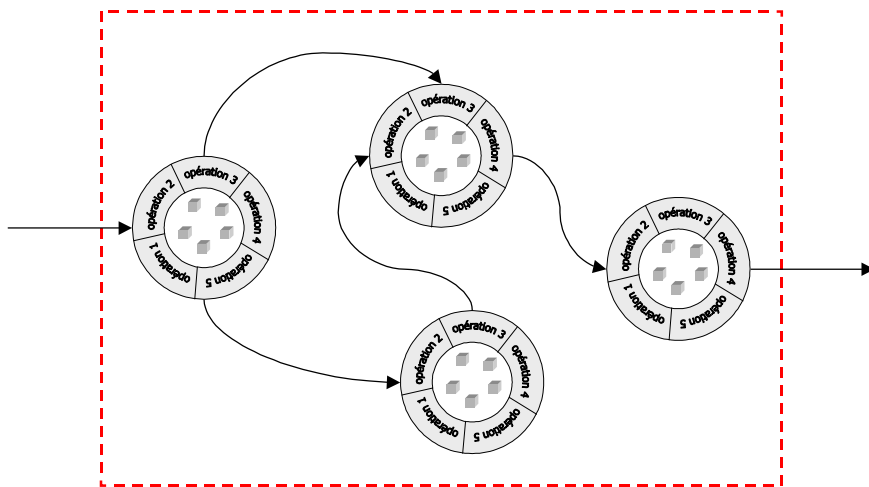


Figure 32. Un système orienté objet : les objets communiquent entre eux par le biais de messages. Une demande à l'un des objet induit si nécessaire une collaboration entre objets jusqu'à obtention du résultat désiré.

Au niveau programmation, l'implémentation d'un logiciel conçu sur la base d'une méthode orientée objet se fait généralement (mais pas obligatoirement) à l'aide d'un langage de programmation orienté objet (Java, C++, Delphi, Smalltalk, Eiffel, etc.), et l'encapsulation favorise la réutilisation logicielle (Brooks, 1996), car elle permet "de décrire une abstraction sous forme d'une partie visible, appelée interface, et d'une partie cachée, appelée implémentation" (Kettani et al., 1998). Ainsi, un objet présente à chacun de ses utilisateurs les services qu'il est à même de fournir (l'interface) et garde cachée la manière de rendre ces services. L'interface est constituée de la liste des opérations que l'objet propose à ses utilisateurs, chaque opération étant implémentée par une méthode (Rumbaugh, 1994). A noter que pour pouvoir modifier la valeur d'un attribut d'un objet, il faut accéder à cet objet par le biais de l'une des opérations proposées par l'interface de l'objet. Comme le montre la figure 31, les attributs sont localisés au centre

de l'objet, et donc protégés de toute manipulation directe : ils ne peuvent l'être que par le biais des méthodes auxquelles on accède en appelant les opérations qui leur correspondent. L'encapsulation offre deux avantages majeurs: la modularité, le code source de chaque classe pouvant faire l'objet d'une maintenance de manière indépendante des autres classes, et la dissimulation d'information, la modification de cette dernière étant soumise au contrôle de l'objet qui en est propriétaire.

L'approche orientée objet apporte ainsi de nombreux avantages en regard des objectifs qui ont été fixés, tant aux niveaux modélisation que programmation : elle facilite la modularité, par le biais du regroupement des données et des traitements dans des mêmes unités, les objets; elle rend l'évolutivité des systèmes logiciels plus aisée, grâce au principe d'encapsulation; la séparation entre interface et implémentation conduit à produire des systèmes de plus en plus robustes, les implémentations étant petit à petit améliorées sans que cela n'influence les utilisateurs des systèmes objets, qui n'en voient que les interfaces. Enfin, le principe même de cette approche permet la manipulation de modèles logiciels qui offrent une correspondance directe avec les systèmes réels qui sont considérés, et sont de ce fait plus faciles à comprendre. Pour ces différentes raisons, **nous adopterons le paradigme orienté objet** pour la suite de notre réflexion.

4.2.2 Formalismes

Le développement d'une application logicielle implique différentes étapes de modélisation¹, les modèles finaux faisant l'objet d'une implémentation, c'est-à-dire d'une traduction dans un langage de programmation qui constitue l'application informatique ou le logiciel proprement dit. Les modèles développés dans le cadre du processus de création d'une application logicielle peuvent se rapporter au système "réel" auquel se rapporte l'application en cours de création, ou peuvent concerner l'application logicielle elle-même. Comme les "éléments jugés essentiels" diffèrent d'une application à l'autre, on se retrouvera avec différents modèles d'une même réalité ou d'un même système logiciel, en fonction des besoins des modélisateurs.

De tels modèles doivent pouvoir être discutés, confrontés, communiqués. Il est nécessaire pour le modélisateur de s'appuyer sur un langage de modélisation formalisé, c'est-à-dire dont *"la précision et la netteté excluent toute méprise, toute équivoque"* (Petit Robert, 1996). Ce langage permet ainsi au modélisateur de mieux comprendre les systèmes qu'il modélise, de les spécifier, c'est-à-dire de les décrire complètement et sans ambiguïté, de partager cette compréhension du système avec d'autres intéressés, de documenter le système modélisé.

De nombreux formalismes ont été proposés dans le contexte de la conception de systèmes d'information. Entrent en ligne de compte dans ce travail les formalismes orientés objet, puisque nous avons opté pour une telle approche. Ces formalismes sont généralement proposés dans le cadre de méthodes de développement de systèmes logiciels (chapitre 5.2). Citons ici les formalismes qui accompagnent les méthodes Booch, OMT (Rumbaugh, 1994) et FUSION (Coleman et al., 1994). Il n'y a, à vrai dire, pas de différence majeure entre les différents formalismes (Pouliot, 1999), bien que certains prennent mieux

1. La modélisation désigne l'action de représenter un système au moyen d'un modèle qui est une abstraction de ce système mettant en évidence les éléments jugés essentiels par rapport au problème abordé (Pouliot, 1999).

en compte certains aspects que d'autres. **Nous adopterons pour ce travail le langage UML (Unified Modeling Language).** Ce formalisme, ou plutôt cet ensemble de formalismes, a été conçu pour permettre *"la modélisation de tous les phénomènes de l'activité de l'entreprise (processus métier, systèmes d'information, systèmes informatiques, composants logiciels,...) indépendamment des techniques d'implémentation mises en oeuvre par la suite."* (Kettani et al., 1998). UML, dont la première diffusion date de 1995 (version 0.8 draft), résulte de la volonté de trois auteurs majeurs de méthodes orientées objet de conception d'applications logicielles, à savoir Booch, Jacobson (1992) et Rumbaugh (1994), de proposer un langage de modélisation unifié. Chacune des méthodes proposées auparavant par ces trois auteurs comporte ses points forts et ses points faibles. UML, qui reprend des éléments présents dans chacune d'entre elles, est extrêmement complet. De plus, en 1997, ce formalisme a été adopté comme langage de modélisation de référence suite à un appel d'offre de l'Object Management Group¹ (OMG) relatif à la standardisation des langages de modélisation. De ce fait, de nombreux auteurs se sont ralliés à UML, qui est ainsi le résultat d'un large consensus et le fruit du travail d'experts reconnus. De multiples ouvrages décrivent les différentes facettes de UML (Booch et al., 1999; Kettani et al., 1998; Larman, 1997; Rumbaugh et al., 1999; Sturm, 1999). Nous nous bornerons donc à en faire une description succincte à l'annexe 2.

A noter que d'autres formalismes existent qui sont spécialement adaptés à la modélisation des données et utilisés dans le contexte de la structuration de BD et de SIG. Un modèle conceptuel de données expose de manière formelle et indépendamment de tout contexte d'implémentation les données qui sont retenues pour décrire un système. L'un des formalismes les plus connus dans ce contexte est le formalisme entité-relation (Chen, 1976; cité dans McFadden et al., 1993), pour lequel de multiples évolutions ont été proposées (Spaccapieta et Parent, 1992). Dans le domaine des SIG plus particulièrement, différentes recherches ont été menées quant à la prise en compte de la dimension spatiale (et parfois temporelle) par de tels modèles de données (Bédard, 1999; Caron, 1991; Pantazis et Donnay, 1996; Parent et al., 1999a; Parent et al., 1999b; Shekhar et al., 1999). Les formalismes de modélisation des données concernent ainsi essentiellement des aspects structurels, statiques, du système d'information, et non ses aspects dynamiques, et il doit être fait appel à des formalismes supplémentaires lorsque l'on souhaite modéliser l'ensemble des aspects relatifs à un système d'information.

4.3 Approche par composants

4.3.1 Motivations et principes

Indubitablement, l'approche orientée objet facilite le développement et la maintenance d'applications complexes. Cependant, force est de constater que si l'industrie du logiciel plaçait de grands espoirs dans le paradigme orienté objet, les gains de productivité qui en découlent ne dépassent en fait pas 10 à 30% par rapport aux techniques traditionnelles (META, 1995; cité dans Sutherland, 1999). Trois facteurs peuvent expliquer ces modestes résultats :

- l'approche orientée objet facilite le travail du (ou des) développeur(s) dans le cadre du

1. <http://www.omg.org>

développement d'une même application, et encourage la réutilisation des unités logicielles que sont les classes. Elle facilite donc la réutilisation logicielle au cours du processus de développement d'une application particulière, mais n'apporte que de maigres avantages en ce qui concerne la réutilisation logicielle d'une application à l'autre (bien que tout développeur se construise, au cours des différents projets auxquels il participe, un ensemble de classes qu'il réutilise). Il est certes envisageable de mettre en place des infrastructures de classes non compilées¹, qui sont en fait des applications non terminées, comprenant un ensemble de classes de base devant être adaptées à un contexte particulier - et de les proposer à des développeurs tiers. Cependant, la mise en oeuvre d'une telle infrastructure représente une tâche complexe. De plus, cette approche permet un accès à l'implémentation des classes qui constituent cette infrastructure, ce qui viole le principe d'encapsulation (au niveau programmation) et donc annule les avantages qui en découlent.

- Comme le notent Edwards et al (1998), *"Software objects in standard object-oriented applications exist within a single program. Only the language compiler knows of their existence, and all their beneficial properties are lost when they become a piece of compiled object code"*. Il a été expliqué au chapitre précédent que l'un des grands avantages de l'approche objet est précisément de permettre la manipulation d'objets informatiques correspondant à des concepts du monde réel : une fois l'application compilée, cet avantage est donc perdu (figure 33), car une approche orientée objet traditionnelle n'apporte d'avantages que durant le développement même de l'application.
- enfin, le degré de granularité de l'objet est trop faible pour que s'institue un "marché" d'objets (ou plutôt de classes), à partir desquels les différentes applications seraient construites. En effet, un développeur qui désirerait construire un logiciel devrait en premier lieu chercher sur ce "marché" les différentes classes susceptibles de lui rendre service. Or la faible granularité de celles-ci l'obligerait à en rechercher un grand nombre, ce qui constituerait ainsi une perte de temps.

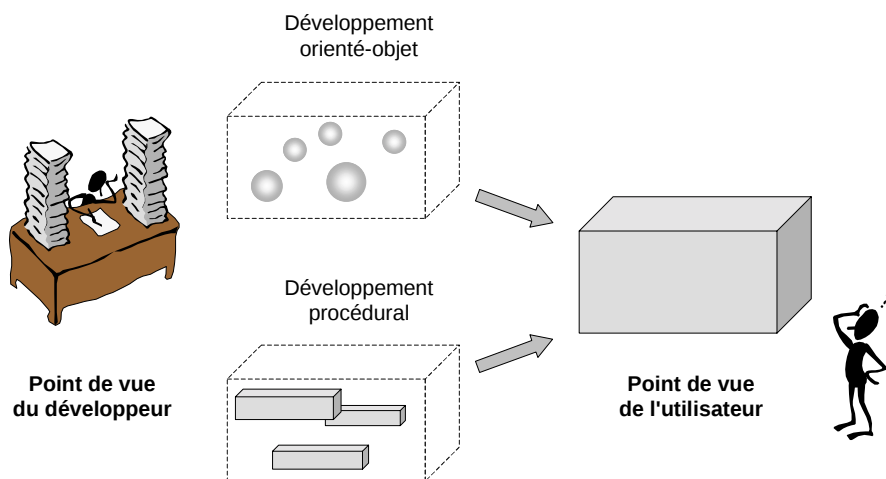


Figure 33. Du point de vue de l'utilisateur, rien ne permet de distinguer une application développée au moyen d'un langage orienté objet d'une application développée avec un langage traditionnel.

L'approche orientée objet n'est donc pas la panacée. Indépendamment de cette approche, l'industrie du

1. Un programme écrit en langage évolué est, une fois terminé, traduit en langage machine afin de pouvoir être exécuté sur un ordinateur : c'est la compilation, et le logiciel assurant cette traduction est un compilateur.

logiciel est depuis longtemps préoccupée par le problème de la réutilisation. En effet, le génie logiciel se trouve depuis longtemps confrontée à un certain nombre de problèmes (dépassements budgétaires et de délais, produits peu fiables) et cherche à surmonter ces difficultés entre autres en mettant en oeuvre des politiques de réutilisation des différents artefacts produits au cours du processus de développement logiciel (Hislop, 1998; Lim, 1998; Rine et Sonnemann, 1998). Développer à chaque fois intégralement une application n'est plus envisageable (Szyperski, 1997) : les coûts de développement sont très élevés, puisque l'on part de rien. Les temps de développement, très longs, sont non seulement en partie responsables de ces surcoûts, mais en plus ne permettent souvent plus de suivre les évolutions de la demande : au moment où le logiciel est terminé, les besoins des utilisateurs ont changé (figure 29). La maintenance de logiciels développés ex nihilo est également très complexe, puisque ceux-ci ne sont pas standardisés, et donc que des compétences très spécifiques doivent être conservées afin de pouvoir assurer cette maintenance. Une approche diamétralement opposée, à savoir prendre un logiciel existant et l'adapter à des besoins spécifiques, présente également un certain nombre de désavantages : de telles applications ne peuvent souvent pas être exactement adaptées aux besoins des utilisateurs. De plus, pour des sociétés soumises à la concurrence, le fait d'utiliser des logiciels standard comme base ne leur permet pas d'acquérir par ce biais d'avantage concurrentiel, les autres sociétés ayant la possibilité d'utiliser ces mêmes logiciels de base. Enfin, ces derniers, issus de développements externes, ne peuvent être adaptés rapidement aux changements d'environnement et à l'évolution des besoins des utilisateurs.

Il est cependant possible d'emprunter une voie intermédiaire : la conception d'applications logicielles par assemblage de composants. Un composant logiciel est un élément de base utilisé pour la construction d'applications logicielles plus complexes. Jacobson en donne la définition suivante :

"By components we mean already implemented units that we use to enhance the programming language constructs. These are used during programming and correspond to the components in the building industry." (Jacobson, 1992)

Les composants sont des produits standard, aux caractéristiques et fonctionnalités bien définies, mais qui peuvent être assemblés de manière à proposer à l'utilisateur final un produit qui réponde exactement à ses besoins (figure 34). Les objectifs d'une telle approche sont de favoriser la réutilisation de codes dans la construction d'applications modulaires tout en réduisant le coût. Cette manière de faire combine les avantages des deux approches "extrêmes" présentées précédemment, à savoir permettre un développement relativement rapide et bon marché d'applications correspondant exactement aux besoins des utilisateurs finaux. En fait, une telle approche permet au développeur de ne se concentrer que sur les

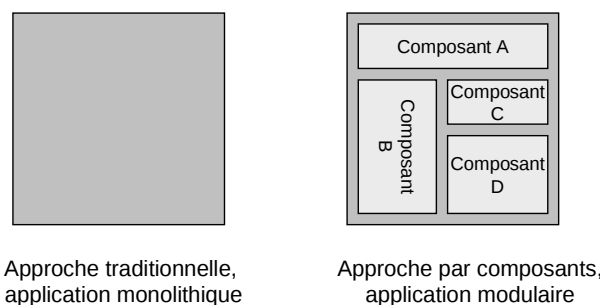


Figure 34. Illustration des approches traditionnelles et modulaires dans le contexte du développement d'applications logicielles

caractéristiques de l'application logicielle qui la distinguent des produits destinés à d'autres utilisateurs. Makrygiannis (1998) parle de *"mass-customized information systems"*, une approche par composants permettant de trouver le juste milieu entre les développements spécifiques d'applications (*"specific developments"*) et la production en série (*"mass-production"*) de logiciels.

D'un point de vue conceptuel, une approche par composants permet de profiter des avantages de l'approche objet, mais en éliminant les inconvénients et en introduisant la notion de regroupement : les différentes classes qui doivent être développées peuvent être réparties en un certain nombre de composants (figure 35). Les objets instanciés à partir de classes appartenant à un même composant doivent être étroitement liés les uns aux autres, constituer un sous-système cohérent et opérationnel. Par contre, la répartition des classes entre les différents composants doit être bien réfléchie de manière à ce que le degré de couplage entre composants soit le plus faible possible, ceci afin que ces composants puissent, en fonction des besoins, être intégrés dans des applications indépendamment les uns des autres.

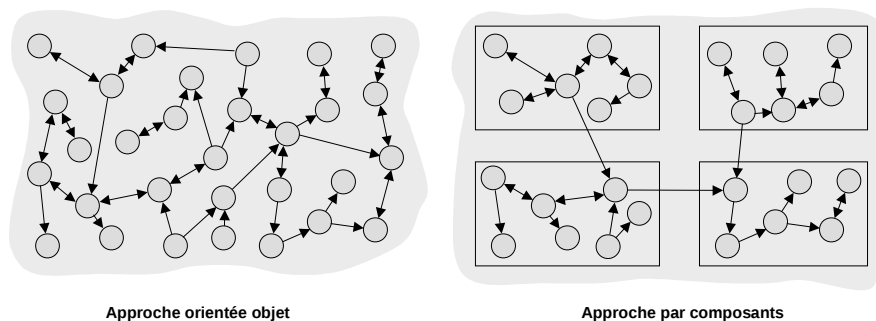


Figure 35. Les objets identifiés doivent être regroupés en ensembles cohérents. Les composants correspondant à ces ensembles implémentent les classes qui permettent l'instanciation de ces objets.

4.3.2 Définition du concept de composant

La littérature propose de nombreuses définitions du concept de composant logiciel, dont quelques-unes sont reportées ci-dessous afin de nous permettre d'en découvrir les différentes facettes.

"A component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces." (Booch et al., 1999)

"A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties." (Szyperski, 1997)

"[A software component is] a self-contained piece of software with a well-defined interface or set of interfaces [...], has interfaces that are accessible at run-time, and at some point in its development lifecycle, the component can be independently delivered and installed." (Herzum et Sims, 2000)

"[Components are] precompiled units of code that expose one or more interfaces for use by clients programs." Définition de Microsoft, rapportée par Lhotka (1998).

Il ressort de ces définitions les points suivants :

- un composant est un élément physique, une unité de code compilée. Un composant n'est donc pas un objet, puisqu'un objet résulte de l'instanciation d'une classe. Un composant doit toutefois être instancié, tout au moins partiellement, lorsqu'il est fait appel à lui par l'application cliente. On parlera alors d'instance de composant. Une différence essentielle entre un objet et une instance de composant découle du fait que l'instance de composant ne possède pas d'état, contrairement à l'objet. La confusion entre objet et composant, qui est induite par le fait que dans les deux notions il est question d'interfaces et d'encapsulation, est extrêmement fréquente, comme le relèvent Lhotka (1998) ou Szyperski (1997);
- un composant, pour pouvoir être utilisé, doit proposer un ensemble de services clairement spécifiés. Ces services sont regroupés de manière cohérente dans des interfaces qui sont, techniquement parlant, des ensembles d'opérations à la sémantique clairement définie et qui peuvent être invoquées par un client (Szyperski, 1997). Les interfaces proposées par un composant (qui peut proposer une ou plusieurs interfaces) peuvent être vues comme des contrats entre le composant et ses clients qui engagent le composant à réaliser les opérations proposées par l'interface (figure 36);

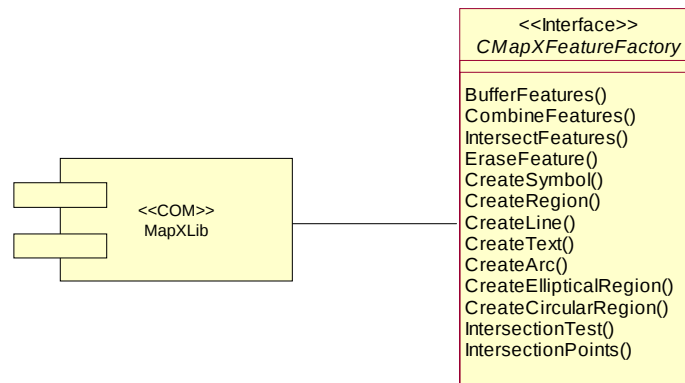


Figure 36. Le composant technique MapX (MapInfo Corporation, 1999b), décrit à l'annexe 9, implémente entre autres une interface "FeatureFactory" permettant de créer des objets spatiaux. Cette interface spécifie un certain nombre d'opérations (dont la signature n'est ici pas intégralement visible), mais ne dit rien quant à la manière de réaliser ces dernières.

- les interfaces sont clairement distinctes des implémentations des opérations qu'elles proposent : un composant s'engage à fournir des services, à effectuer des opérations, mais ne promet rien quant à la manière d'implémenter des opérations. On dit qu'il encapsule l'implémentation des opérations. Le composant est, pour son utilisateur, une boîte noire, que celui-ci utilise par le biais de ses interfaces. De ce fait, l'implémentation d'un composant peut évoluer même lorsque ce dernier est déjà intégré dans certaines applications, pour autant bien sûr que ses interfaces ne soient pas modifiées. Cette séparation claire entre interfaces et implémentations permet d'assurer l'interopérabilité entre composants, par le biais de ces interfaces, au sein d'un même environnement pour composants;
- un composant est une entité logicielle indépendante, ou tout au moins dont les dépendances sont explicites : un composant peut avoir besoin d'autres composants, dont il devient le client, pour réaliser certaines opérations (figure 37). Grâce à cette indépendance (totale ou partielle), un composant peut être distribué, commercialisé, et ce de manière autonome;
- les composants sont destinés à être assemblés et à communiquer par le biais de leurs interfaces. Ils peuvent être vus comme des briques logicielles servant à la construction d'applications.

Cette notion de composant réutilisable n'est pas nouvelle : il est depuis longtemps possible d'appeler des bibliothèques de procédures, compilées au moyen d'un compilateur particulier, à partir de programmes¹ compilés par un compilateur différent (Pfister et Szyperski, 1998). Les bibliothèques de fonctions mathématiques, abondamment utilisées, constituent un exemple de tels composants.

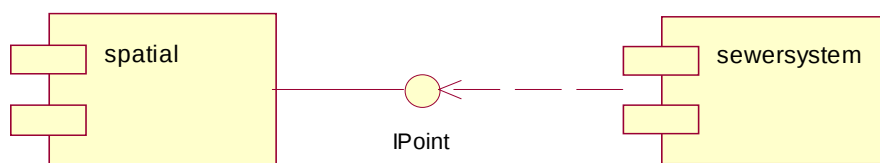


Figure 37. Exemple de relation entre deux composants : le composant "spatial" réalise une interface "IPoint", permettant de manipuler des objets de type Point. Le composant "sewersystem" importe cette interface, et peut donc manipuler des objets de type Point.

Il a cependant été montré dans le chapitre précédent que l'approche objet facilite grandement le développement d'applications complexes, en raison de la correspondance qui peut être établie entre les "objets du système réel" et les "objets du système logiciel", des relations sémantiques qui caractérisent l'approche orientée objet, et enfin du concept d'encapsulation. Or, jusqu'à récemment, les technologies informatiques ne permettaient pas le développement de composants logiciels qui, une fois compilés, mettent à disposition de leurs utilisateurs un ensemble d'interfaces d'objets. Tout le bénéfice de l'approche orientée objet se perdait une fois que le composant, pourtant développé et implémenté suivant une telle approche, était compilé (Pfister et Szyperski, 1998).

Des technologies sont toutefois actuellement disponibles et arrivent à maturité. Elles permettent la construction de composants proposant à leurs utilisateurs de telles interfaces orientées objet. Autrement dit, des composants conçus sur la base de ces technologies permettent aux personnes qui les mettent en oeuvre de manipuler les objets implémentés par le composant. Les objets sont dans ce cas instanciés "dans" le composant, et ce dernier permet à ses clients d'accéder aux objets ayant été instanciés, de les manipuler par le biais des interfaces qui leur sont fournies, ces dernières jouant alors le rôle de références vers ces objets. On retrouve cette notion dans la définition donnée par Lhotka (1998) : "*Components are binary entities that contain one or more objects that can be used by applications.*" Un composant de ce type "exporte" des interfaces d'objets, ce qui le distingue des composants classiques de type bibliothèque de fonctions. Les avantages qui résultent d'une approche objet ont été présentés au chapitre précédent : afin d'en tirer profit, nous nous intéressons dans la suite de ce travail uniquement à cette catégorie de composants, que nous appellerons composants objet.

Les principales technologies permettant la construction de composants objets sont : COM (Component Object Model) de Microsoft et ses différentes déclinaisons (DCOM, ActiveX, COM+)², JavaBeans et Enterprise JavaBeans (EJB), basées sur la technologie Java de Sun³, et CORBA (Component Object Request Broker Architecture) pour composants, de l'OMG (Object Management Group)⁴. Elles

1. Programme : "Ensemble des instructions, rédigées dans un langage de programmation, permettant à un système informatique d'exécuter une tâche donnée." (Petit Robert, 1996)
2. <http://www.microsoft.com/com/>
3. <http://www.javasoft.com/>
4. <http://www.omg.org/>

fournissent un environnement logiciel qui rend possible l'intégration de composants dans une même application et la collaboration de ces composants entre eux. Par analogie avec l'industrie du matériel informatique, les composants logiciels peuvent être vus comme les puces, et l'environnement pour composants comme les cartes sur lesquelles ces puces peuvent être fixées et grâce auxquelles elles peuvent communiquer. Ces technologies normalisent entre autres la manière de décrire les interfaces que proposent les composants.

Un environnement logiciel pour composants (figure 38) peut rendre possible la communication entre composants installés sur un même ordinateur (COM par exemple) ou alors entre plusieurs ordinateurs, par le biais de réseaux informatiques (DCOM ou COM+, EJB, Corba par exemple). On parle alors de composants distribués.

De nombreux ouvrages traitant des technologies relatives aux composants sont disponibles (Serain, 1997; Szyperski, 1997). D'ailleurs, ce travail n'a pas pour objet des questions techniques, mais la manière de mettre en oeuvre une approche par composants dans le contexte de la gestion des eaux. Nous ne faisons donc que définir le concept de composant. Cependant, étant donné que COM est la technologie qui a été utilisée pour les prototypes réalisés dans le cadre de cette recherche, nous en présentons brièvement, à titre d'exemple, les principes à l'annexe 3.

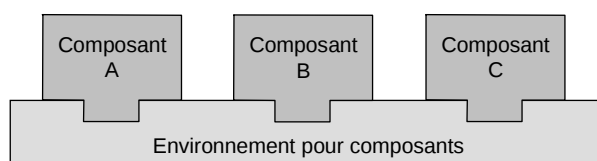


Figure 38. Pour que des composants puissent être assemblés et puissent communiquer, un environnement pour composants (appelé middleware) doit être disponible.

Les environnements pour composants, qui permettent l'interopérabilité entre ces derniers, offrent ainsi à l'intégrateur de composants la possibilité de continuer à bénéficier des avantages de l'approche orientée objet : les composants mettent à disposition de l'utilisateur des interfaces lui donnant l'illusion de manipuler des objets appartenant à un système caractérisé par toute la richesse sémantique de l'approche orientée objet (figure 39). Nous parlons ici d'illusion car le composant protège l'implémentation des fonctionnalités qu'il offre, se comporte en boîte noire : rien ne prouve que son implémentation suive les préceptes de l'approche objet.

Une approche par composants permet ainsi de bénéficier des avantages de l'approche orientée objet, tout en évitant les inconvénients. En effet, le problème de la trop faible granularité des objets se trouve résolu: un composant peut être vu comme une collection d'objets qui coopèrent entre eux et présentent une frontière bien définie vis-à-vis d'autres objets ou composants. De plus, un composant est une unité de code compilée et autonome, accessible uniquement par le biais de ses interfaces. Un composant constitue de ce fait un élément ayant les spécificités requises pour être réutilisé facilement. De plus, la notion d'interface ainsi que la séparation nette entre interface et implémentation permettent l'interopérabilité entre composants et rendent la maintenance d'applications constituées d'assemblages de composants très aisée, tout composant pouvant sans problème être remplacé par un autre composant proposant les interfaces adéquates.

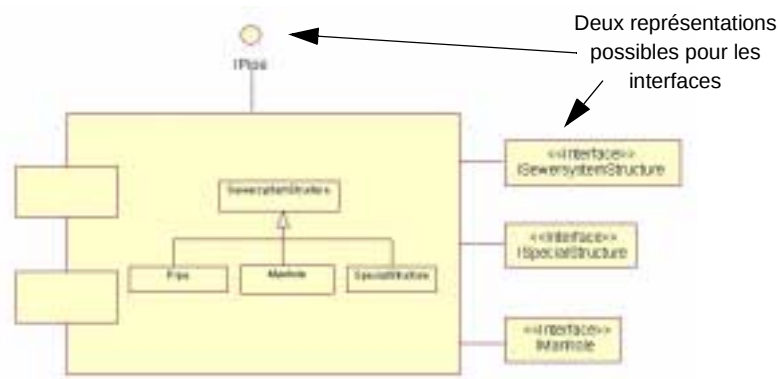


Figure 39. Un composant objet propose à ses utilisateurs (clients) un ensemble d'interfaces. Les implémentations de ces interfaces peuvent être le fait de classes codées à l'aide d'un langage orienté objet. Les interfaces proposées par le composant sont sémantiquement liées par un modèle objet cohérent.

4.3.3 SIAGEM et composants logiciels

Une approche par composants semble donc constituer une voie à examiner de plus près dans le contexte des SIAGEM : le présent chapitre montre quels sont les avantages d'une telle approche en regard des contraintes ayant été présentées au chapitre 3.2.

Une approche par composants : 1°) permet de procéder au développement des différents composants par étapes, ce qui raccourci la durée du cycle de développement de chaque composant; 2°) favorise la réutilisation des composants et donc abaisse les coûts spécifiques des SIAGEM mettant en oeuvre de tels composants; 3°) apporte une réponse au problème de trop faible granularité des objets, en favorisant la constitution d'unités pouvant être facilement distribuées et commercialisées. Nous avons vu que des SIAGEM doivent pouvoir être adaptés spécifiquement aux besoins de chaque intervenant (chapitre 3.2.5). Ici peut être apporté un premier élément de solution : un SIAGEM peut être construit à partir d'un nombre quelconque de composants sélectionnés en fonction des besoins et vœux exprimés par les utilisateurs, l'intégrateur de ces composants n'ayant plus qu'à se préoccuper de sa personnalisation (interfaces graphiques, ergonomie, fonctionnalités très spécifiques).

Une approche par composants apporte également une solution aux problèmes posés par la rapide évolution des besoins des intervenants pour ce qui est des fonctionnalités, ainsi que par la maintenance des SIAGEM existants. La figure 40 présente deux interfaces accompagnées de leur implémentation, qui se rapportent au domaine de la mesure, ainsi que la description de ces interfaces au moyen du langage IDL¹. Deux points peuvent être vérifiés ici : d'une part, une interface ne décrit que ce que la classe (ou le composant) va pouvoir effectuer, mais ne dit absolument rien de la manière dont cette fonction va être remplie. D'autre part, comme une interface est totalement indépendante de son implémentation, il est possible de changer une implémentation sans que les utilisateurs de l'interface n'en soient gênés. A la figure 40, la série temporelle est liée sémantiquement à la mesure. Cette liaison est définie au niveau des

1. Interface Definition Language, langage de description d'interfaces proposé par l'OMG (Objet Management Group)

interfaces. Si l'implémentation de l'une de ces interfaces doit être modifiée (par exemple pour améliorer les performances des opérations effectuées sur les séries temporelles), il suffit de remplacer l'ancienne version de l'implémentation par sa nouvelle version, et les clients qui s'adressent à l'interface proposée par le composant continueront sans problème à fonctionner tout en bénéficiant des améliorations apportées. Ainsi, interchanger, au sein d'un SIAGEM, deux versions d'un même composant, ne pose aucun problème pour autant que chacun propose les mêmes interfaces.

À première vue, une telle condition - à savoir qu'une fois une interface mise à disposition d'intégrateurs, elle ne doit plus être modifiée car cela équivaudrait à une rupture du contrat que constitue l'interface, qui ne serait dès lors plus reconnu par les applications - peut paraître contraignante : nous avons vu que le domaine de la gestion des eaux évolue très vite, et que les besoins en termes de fonctionnalités sont extrêmement difficiles à identifier.

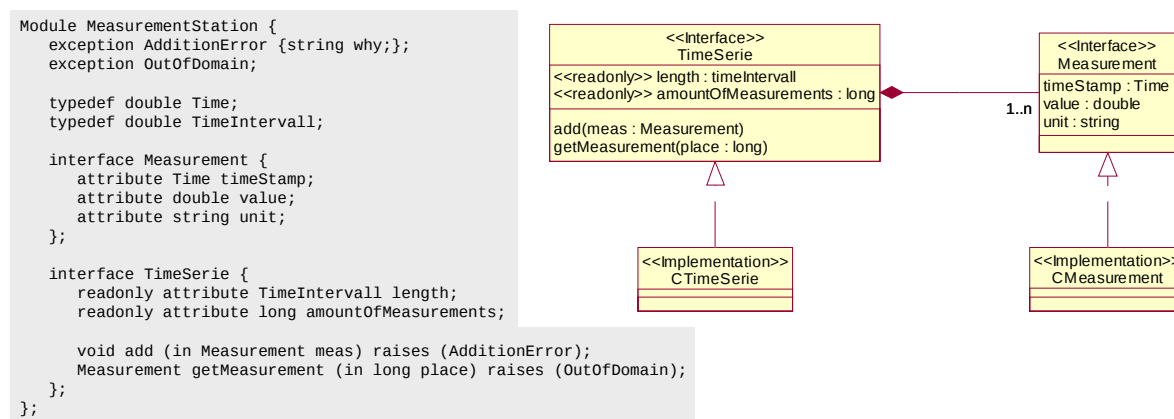


Figure 40. À droite, exemple d'implémentations et d'interfaces pour les classes "Série temporelle" et "Mesure". À gauche, les mêmes interfaces décrites en IDL (interface definition language). Ces classes peuvent être intégrées à un composant, qui en proposera les interfaces à divers clients.

En fait, si une interface, une fois publiée, ne doit plus faire l'objet de modifications, rien n'empêche d'ajouter de nouvelles interfaces à un composant. La figure 41 montre le cas d'un composant ayant fait l'objet de deux versions successives. La version la plus récente du composant implémente toujours l'interface "Bâtiment" proposée par sa première version. Elle implémente cependant également une nouvelle interface, relative à ce même concept de "Bâtiment", mais dotée de fonctionnalités plus riches (permettant par exemple de gérer les installations de récupération d'eaux pluviales, dont on ne pouvait pas prévoir, au moment de la conception du premier composant, qu'une telle information présenterait un jour un intérêt). De plus, la nouvelle version du composant implémente des interfaces se rapportant à de nouveaux concepts (en l'occurrence au concept d'installation de récupération des eaux pluviales). Ainsi, à partir du moment où il continue de supporter toutes les interfaces qui sont utilisées par ses clients, un composant peut évoluer en proposant de nouvelles interfaces. Lorsqu'un concept déjà implémenté par le composant, et donc disponible sous forme d'une interface, doit être modifié, une nouvelle interface doit être créée, le composant supportant alors deux interfaces se rapportant à un concept similaire.

Cet exemple montre donc que les composants peuvent changer, proposer de nouvelles versions des interfaces existantes et des interfaces supplémentaires relatives à de nouveaux concepts. Une telle approche, menée de manière non réfléchie, conduirait cependant rapidement à une profusion d'interfaces, ce qui ne faciliterait pas la compréhension des fonctionnalités du composant. La nette

séparation qui existe entre interface et implémentation dans l'approche par composants permet donc bien de résoudre les problèmes que posent les difficultés qu'il y a à identifier les besoins des intervenants ainsi que la dynamique des besoins dans le domaine de la gestion des eaux, mais pour cela la spécification d'interfaces doit se faire de manière extrêmement soignée et être l'objet de réflexions approfondies.

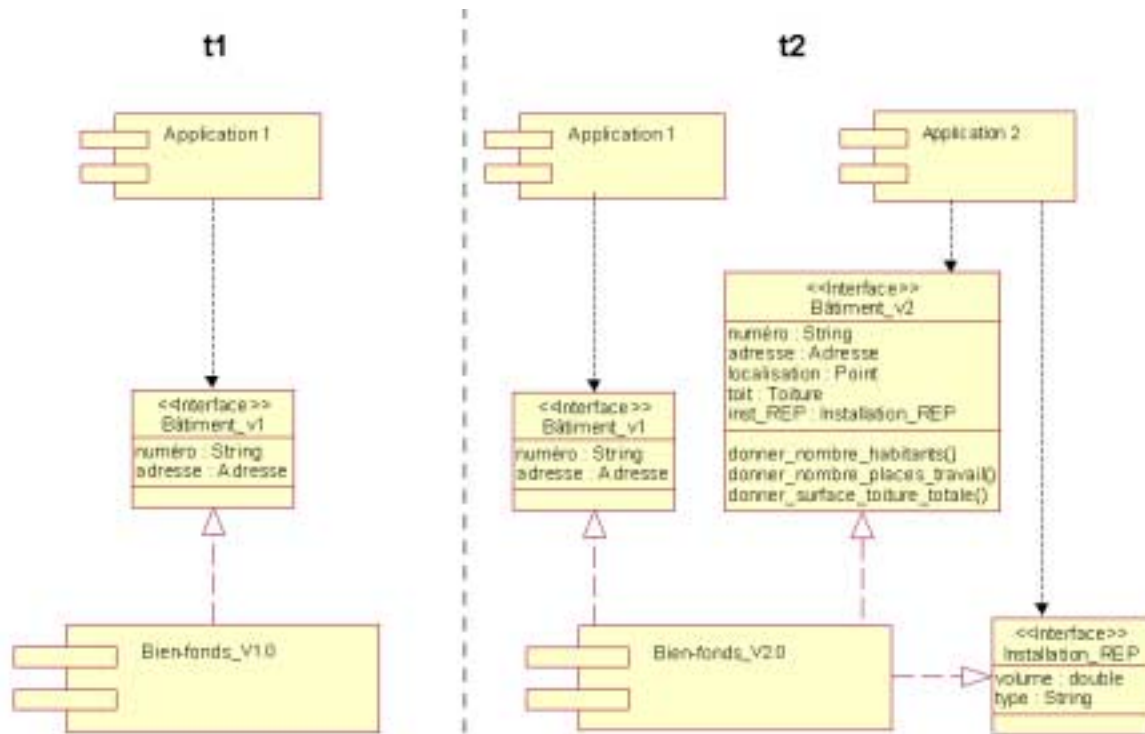


Figure 41. Evolution d'un composant de t_1 à t_2 : en t_1 , le composant ne réalise qu'une interface, qui est utilisée par l'application 1. En t_2 , le composant continue de proposer l'interface bâtiment_v1, afin que l'application 1 puisse toujours fonctionner avec la nouvelle version du composant (dont les performances ont été améliorées), mais réalise de nouvelles interfaces qui peuvent être exploitées par de nouvelles applications.

Enfin, il est à noter qu'interfaces et implémentations doivent, dans le cadre d'un processus de développement, être l'oeuvre d'équipes différentes; des experts du domaine de la gestion des eaux définissent des interfaces (ou au moins contribuent à le faire), en publient les spécifications et les caractéristiques, et ces interfaces sont ensuite implémentées par des développeurs. À partir du moment où les spécifications d'interfaces sont publiques et reconnues comme standard, on peut imaginer que, en tout cas en ce qui concerne les composants dont le développement présente un potentiel économique, différentes équipes de développeurs s'attellent à leur implémentation, et en proposent différentes versions qui, disposant des mêmes interfaces, peuvent à loisir être interchangeables dans les applications clientes.

Nous proposons donc ici de définir des ensembles cohérents d'interfaces à même de faire l'objet d'implémentations dans des composants objet réutilisables. Il se pose à présent la question de savoir comment regrouper ces interfaces, qui correspondent à des concepts du domaine de la gestion des eaux, de manière à faciliter la réutilisation et la compréhension desdits composants. Cette problématique fait l'objet du chapitre qui suit.

4.4 Prise en compte du territoire et de ses composantes

On peut donc à partir de ce qui a été présenté précédemment imaginer que des composants spécifiques au domaine de la gestion des eaux soient construits par différentes institutions et référencés dans une véritable bibliothèque. Les concepteurs de SIAGEM pourraient prélever dans une telle bibliothèque les composants nécessaires à la construction de chaque SIAGEM. Chaque composant doit permettre la manipulation d'un certain nombre de concepts. Sur quels critères appuyer la décision de regrouper ces concepts dans un même composant ? De combien de concepts un composant réutilisable doit-il permettre la manipulation ? Les réponses à ces questions doivent être guidées par l'objectif de concevoir des composants qui puissent être réutilisés, donc qui présentent un intérêt pour un maximum d'utilisateurs.

Notons que le problème de la segmentation de l'informatisation de systèmes d'information ou d'autres applications logicielles d'une certaine importance est abordé par différents auteurs (Golay, 1992; Rumbaugh, 1994; Vu et Dar-Biau, 1997). Parmi les solutions proposées, deux présentent ici un intérêt particulier:

- Tabourier (1986), cité dans Golay (1992), dans le contexte de la méthode Merise, propose de procéder sur la base d'un découpage initial du système opérant (chapitre 2.4.1.2) fondé sur un inventaire des activités de l'entreprise, celles-ci étant regroupées en domaines de gestion répondant à des objectifs homogènes. Cette approche, si elle est particulièrement adaptée au contexte de la conception d'un SI d'entreprise, doit être considérée avec prudence dans le contexte des composants réutilisables : Makrygiannis (1998) note "[...] *the design of components must be done with no regard to organizational structure. Components must not embed any structural aspects of the organizational arrangements*". Est-ce que les activités (ou processus métier) des intervenants dans le domaine de la gestion des eaux peuvent être considérés de manière indépendante des structures organisationnelles auxquelles ces intervenants appartiennent ? Nous avons constaté que les activités du domaine de la gestion des eaux sont très semblables d'une institution à l'autre, d'une commune à l'autre, d'un canton à l'autre (chapitre 2.3.2.3), ce qui donne à penser qu'elles peuvent servir de base à la conception de composants.
- Dès lors que l'information manipulée se rapporte au territoire et aux éléments qui le composent, différents auteurs proposent de baser la conception de SI s'y rapportant sur une modélisation du territoire et de ses éléments, et non sur une modélisation des institutions qui sont en charge de la gestion de ce territoire et de leurs activités. En effet, "*dans le domaine des SIRS [...] l'ensemble des partenaires se retrouvent à travailler sur un support commun, le territoire.*" (GERMINAL, 1995). Le territoire et ses composantes constituent donc le dénominateur commun aux différents intervenants (le concept de dimension intégratrice du territoire a fait l'objet d'une discussion au chapitre 2.4.1.3). En se basant sur cette constatation, on pourrait imaginer une identification des différentes composantes du territoire ayant un rôle à jouer dans le contexte de la gestion des eaux, une classification de ces composantes par familles (les éléments relatifs au cours d'eau, au réseau routier, etc.), et la considération de chacune de ces familles dans le contexte de la construction d'un composant spécifique. Cette approche n'est, à elle seule, pas envisageable : d'une part "*le territoire n'existe pas en termes absolus : chaque acteur a sa propre perception de l'espace géographique et en déduit son propre modèle*" (Crausaz, 2000), et de ce fait, sans la prise en compte des perceptions des différents acteurs, l'identification de composantes du territoire présentant un intérêt pour la

gestion des eaux est vouée à l'échec; d'autre part, le principe même de gestion intégrée implique que soient pris en considération de manière simultanée, pour le traitement d'une même problématique, des éléments du territoire appartenant à différentes familles. Par conséquent, sans connaître les problématiques en jeu, comment regrouper les composantes du territoire en familles cohérentes ?

Nous proposons donc de combiner ces deux approches : d'une part doivent être conçus, à partir d'une modélisation du territoire et de ses composantes, des composants qui permettent la manipulation d'objets "du monde réel". D'autre part, des composants doivent être disponibles qui soutiennent certains processus métier. Les composants appartenant à la première de ces catégories, que nous appellerons par la suite composants-entité (Herzum et Sims, 2000), constituent le référentiel commun aux différents intervenants. L'on reprend ainsi l'approche proposée par les auteurs avocats de l'approche intégratrice du territoire. Les composants spécifiquement destinés à soutenir certains processus métier (par exemple un composant destiné à organiser et gérer les contrôles vidéo des canalisations d'assainissement ou un composant dédié à l'administration du cadastre de l'assainissement) s'appuient sur les composants-entité (par exemple un composant-entité "Système d'assainissement" permettant de manipuler les objets de type "Chambre normée", "Canalisation", "Déversoir", etc.). Un composant-entité peut ainsi être nécessaire au fonctionnement de plusieurs composants-processus (figure 42). Cette distinction entre composants-entité et composants-processus permet de constituer un ensemble de composants-entité destinés à former un référentiel commun aux différents intervenants, les composants-processus étant spécifiquement conçus pour certains groupes d'utilisateurs.



Figure 42. Les composants-processus (modélisation MOUSE, gestion cadastre d'assainissement) s'appuient sur les composants-entité.

La proposition d'utiliser le territoire et ses composantes comme élément d'articulation entre les différents intervenants, comme moyen d'intégrer les différentes visions de ces derniers, n'est pas nouvelle. Elle a été cependant jusqu'à présent considérée uniquement dans l'optique de la conception de modèles de données (Golay, 1992) destinés à être implémentés dans des BD intégrant de l'information localisée. Ces BD ont selon une telle optique pour mission de servir de plate-forme d'intégration entre les différents utilisateurs, le territoire constituant un référentiel commun à ces derniers.

L'approche en question soulève dans la pratique un certain nombre de problèmes, qui découlent de la nécessité d'harmoniser les sens des concepts manipulés. Différents auteurs relèvent que les approches basées sur une modélisation du territoire et débouchant sur l'élaboration de dictionnaires de données communs aux différents acteurs impliqués (Bernasconi, 1999; Prélaz-Droux, 1995b) conduisent à une perte en densité et richesse de l'information. Ainsi, dans le contexte des SIRS, Pornon (1998) note que

"dans tous les cas une telle démarche suggère une unification des sens (les notions de regard d'assainissement, de tronçon de voie, de cours d'eau ayant le même sens pour tous), ce qui est contraire aux objectifs de diversité, d'enrichissement par le sens, de circulation du sens. Bien qu'étant des objets physiquement clairement identifiables, la route ou la parcelle peuvent être vues de façons très différentes". Une approche "classique" basée sur une modélisation du territoire est donc menacée par deux dérives : soit le processus de modélisation débouche sur un modèle de données résultant d'un compromis entre les différents intervenants, et qui dans la pratique ne satisfera ni les uns ni les autres puisque la BD ne sera pas à même de gérer des concepts suffisamment riches pour leurs besoins; ou alors le modélisateur s'attachera à faire en sorte que chacun retrouve dans le modèle de données toute l'information dont il aura besoin, à intégrer différentes visions dans un même modèle, ce qui le conduira généralement, en particulier lorsque les intervenants sont nombreux, à proposer des modèles de données certes complets mais extrêmement compliqués, et donc difficiles à maîtriser.

La reprise du concept d'intégration par le territoire au niveau des composants logiciels qui intègrent la logique métier, apporte aux problèmes qui viennent d'être détaillés - complexité de la BD et coexistence de différentes visions - une solution élégante :

- les différents objets dont les composants permettent la manipulation gèrent eux-mêmes leurs interactions avec le mécanisme chargé d'assurer leur persistance. Nous avons vu que l'approche orientée objet permet une correspondance directe et intuitive entre les objets du "monde réel" et les objets du "monde logiciel". Nous avons aussi vu que les BD les plus employées à l'heure actuelle obéissent au modèle relationnel, dont la compréhension est nettement moins intuitive que celle d'un modèle orienté objet. Ainsi, une approche par composants permet d'encapsuler au sein du composant tout ce qui a trait au stockage des données. Lorsque des BD relationnelles sont chargées d'assurer la pérennité des données, les objets disposent de l'information leur permettant d'assurer la correspondance entre eux-mêmes et les tables relationnelles dans la BD. La personne qui manipule ces objets ne doit donc pas se préoccuper des modèles de données car ces objets constituent une couche logicielle lui dissimulant la BD et sa structure. Ainsi, le concepteur d'une structure de données peut intégrer dans cette dernière les différentes visions des utilisateurs du SI sans se préoccuper de sa complexité : celle-ci sera prise en charge et dissimulée aux intégrateurs et utilisateurs par les composants logiciels. Seuls les développeurs de composants logiciels et les responsables de ces structures de données sont confrontés à leur complexité. Une abondante littérature est disponible relativement à la problématique de la correspondance entre modèles-objet et modèles de données, en particulier relationnels (Ambler, 1998b; Keller, 1997; Rational, 2000). Notons également que d'autres modèles ou mécanismes peuvent être mis en oeuvre pour assurer la persistance des données (comme par exemple dans le cas de séries temporelles qui peuvent être stockées sous forme de fichiers binaires), et ceci de manière totalement transparente pour l'utilisateur.
- les différents intervenants peuvent avoir, suivant leurs activités respectives, des visions différentes relativement à des concepts identiques du monde réel. Cela pose un problème dans le contexte de la structuration de BD, car les redondances doivent être évitées, en tout cas au niveau conceptuel, et nous avons vu que cette exigence, couplée avec le vœu de faire coexister au sein du même modèle des visions différentes, conduit à des structures de données difficiles à comprendre. Nous avons discuté de la notion d'interface (chapitre 4.2). Un objet comme un composant sont à même d'implémenter différentes interfaces. Ainsi, un même concept peut être matérialisé par un même

objet, mais qui se présentera aux différents utilisateurs sous des interfaces différentes, adaptées à leurs besoins respectifs. L'approche par composants objet permet ainsi de mettre à disposition des utilisateurs un ensemble d'objets dotés de toute la richesse sémantique souhaitée, les composants se chargeant, de manière transparente pour l'utilisateur, des problèmes de redondance ou de structuration des données. La figure 43 représente une classe qui implémente différentes interfaces : deux d'entre elles permettent la manipulation d'objets de type Chambre conformément au modèle proposé par l'ASPEE (VSA, 1999), une interface regroupe les opérations et attributs spécifiques à la gestion financière, une interface propose les opérations et attributs qui permettent d'interagir avec le mécanisme de persistance, et enfin une interface présente une version simplifiée de la chambre (telle que la plupart des utilisateurs qui s'intéressent aux aspects de gestion du cadastre de l'assainissement se représentent ce type d'ouvrage). On peut noter sur cette figure (qui ne constitue qu'un exemple)

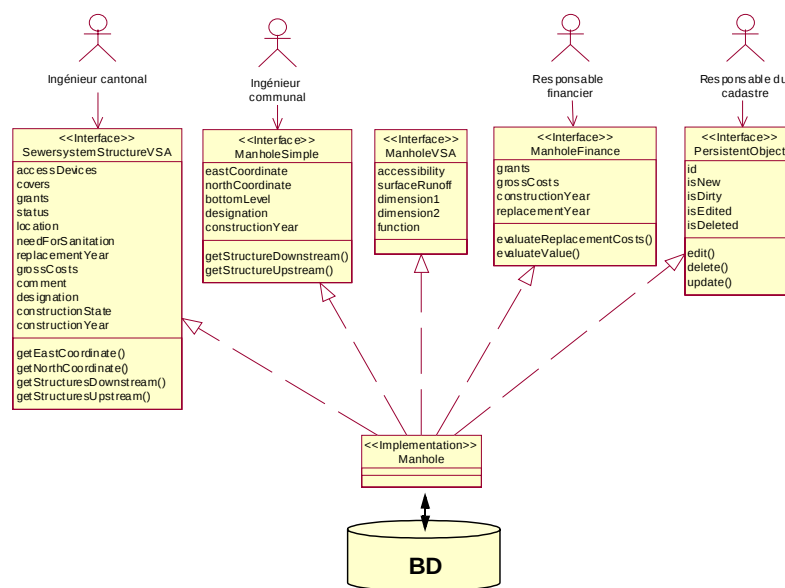


Figure 43. La classe "chambre d'assainissement" (Manhole) propose (réalise) différentes interfaces, adaptées aux besoins spécifiques des différents acteurs. Les mécanismes d'interaction avec la BD sont encapsulés.

que des redondances ne sont pas exclues d'une interface à l'autre : ceci n'a aucune importance, puisque c'est au niveau de la structure de données sous-jacente que des redondances ne sont plus tolérées. Les interfaces sont conçues pour satisfaire aux besoins des utilisateurs. De plus, on s'aperçoit que l'utilisateur des interfaces n'interagit absolument pas avec la BD, n'ayant à ce sujet accès qu'aux fonctionnalités proposées par l'interface "Objet persistant". Du fait de ce découplage, la structure de données peut évoluer sans que cela n'affecte les applications mettant en oeuvre les interfaces qui sont proposées par les composants réutilisables. Leurs implémentations par contre doivent pouvoir être adaptées lors d'évolutions des structures de données.

Les composants objet proposés permettent ainsi aux utilisateurs de manipuler, en fonction de leurs besoins, des objets représentant des concepts réels sous différents aspects, appelés interfaces. Cette notion d'interfaces multiples a pour avantage d'une part de ne présenter aux utilisateurs que les fonctionnalités dont ils ont besoin, et d'autre part de regrouper en unités logiques les aspects sémantiquement corrélés qui se rapportent à un même concept. De plus, les utilisateurs n'accèdent pas directement aux BD, mais manipulent les données par le biais des interfaces précédemment citées. Cette

intermédiation entre utilisateurs et sources de données autorise l'existence de structures de données complexes, à partir du moment où la correspondance entre objet et structures de données est assurée par l'objet lui-même, de manière totalement transparente pour l'utilisateur. De plus, l'organisation des données peut être modifiée sans que cela affecte les SIAGEM construits à partir de composants, et sans que les utilisateurs de ces derniers en subissent des conséquences négatives. Les structures de données se retrouvent ainsi libres d'évoluer en fonction des besoins, et souvent des préférences, des gestionnaires de BD.

Les chapitres qui suivent ont pour objectif de préciser l'architecture de ces composants à partir desquels nous proposons de bâtir des SIAGEM.

4.5 Les composants métier

4.5.1 Concept général

Les premiers composants logiciels réutilisables qui ont été développés l'ont été pour faciliter le travail des programmeurs. De tels composants, qui sont destinés à la simplification de problèmes spécifiquement informatiques, sont appelés composants techniques (Bouzeghoub et al., 1997). Sont compris dans cette catégorie les composants qui facilitent la construction d'interfaces-utilisateurs (boutons, fenêtres et autres menus déroulants), ceux qui permettent l'accès aux diverses BD, ceux qui facilitent la manipulation de fichiers, les bibliothèques de fonctions mathématiques, etc. Ces composants techniques simplifient grandement la construction d'applications logicielles, chacun d'entre eux étant réutilisé dans le contexte de très nombreux développements. À partir de là, on peut imaginer que des composants logiciels réutilisables, non plus destinés à la solution de problèmes techniques purement informatiques, mais propres à un domaine d'activité particulier, pourraient également rendre plus facile, plus rapide et moins coûteuse la mise en place de systèmes d'information dans le domaine auquel ils se rapportent. Une telle approche est intuitive à partir du moment où l'on se rend compte que certains domaines d'activité conduisent à la manipulation fréquente de concepts identiques : on pense alors immédiatement à analyser ces concepts et à créer des abstractions informatiques à partir de ceux-ci pouvant être réutilisées lors de nouveaux développements informatiques, et ce sans que l'on ait besoin de se préoccuper de la manière dont les services rendus par ces abstractions informatiques sont réalisés.

Dans le domaine qui nous occupe ici, à savoir celui de la gestion des eaux en milieu urbain, il a été montré que d'une part, les activités des différents intervenants se rapportent à un même système réel, le système "Eaux en milieu urbain" (chapitre 2.1.2.2), et d'autre part, que ces activités, que nous avons appelées processus métier, sont menées de manière très similaire par les intervenants chargés de responsabilités identiques (chapitre 2.3.2.3). De plus, l'importance des outils informatiques dans le domaine de la gestion des eaux a été mise en évidence (chapitre 2.4.2). Aussi, ce domaine offre-t-il un contexte adéquat pour le développement de composants logiciels réutilisables.

De tels composants sont destinés aux intervenants chargés de mettre en place des solutions logicielles dans le domaine de la gestion des eaux. Nous appelons ces intervenants des intégrateurs, car ils sont, dans le contexte de cette approche, responsables de l'intégration de composants réutilisables construits par des

développeurs tiers dans des solutions opérationnelles.

La mise en place de composants réutilisables destinés à des tâches très spécifiques se heurte cependant à certaines barrières : des composants spécialement conçus pour un domaine d'activité s'adressent à un public plus restreint que des composants techniques qui peuvent être utilisés dans toutes les catégories d'applications. Plus un composant est spécialisé, et plus il devient difficile de le réutiliser. Sur la figure 44, des composants techniques tels que ceux servant à la construction d'interfaces graphiques conviennent à un nombre élevé de développeurs, puisque quasiment toutes les applications logicielles à l'heure actuelle disposent d'une interface-utilisateur graphique. Des composants plus spécialisés, relatifs à des domaines comme la comptabilité par exemple, s'adressent déjà à un public plus restreint, et cela bien que presque chaque entreprise pratique l'activité de comptabilité. Des composants qui s'adressent à un secteur d'activité particulier comme celui de la gestion des eaux intéressent un public beaucoup plus restreint : est-il envisageable de concevoir des composants pour un tel secteur d'activité ? Ces composants trouveraient-ils suffisamment de débouchés pour être rentabilisés ?

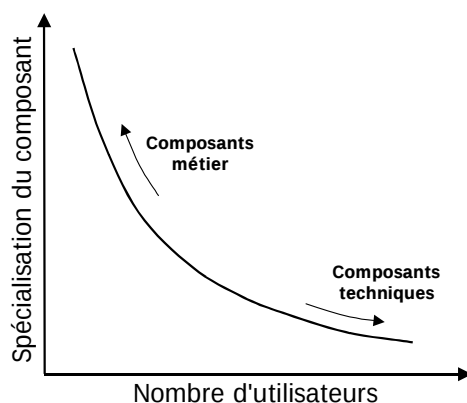


Figure 44. Le nombre d'utilisateurs potentiels d'un composant diminue avec la spécificité de ce dernier.

Ce concept de composants réutilisables concernant un domaine particulier suscite depuis peu un grand intérêt dans différents secteurs d'activité. De tels composants sont ainsi déjà utilisés dans les domaines de la médecine, de la finance ou de la comptabilité (Andro et Chauvet, 1998). Les chapitres suivants ont pour objectif de présenter plus précisément ces concepts de composants logiciels réutilisables spécifiquement liés à un domaine - que nous appellerons composants métier - et de montrer quelles sont nos propositions en ce qui concerne la conception de tels composants pour le domaine de la gestion des eaux.

4.5.2 Définitions

Différents termes sont utilisés dans la littérature lorsqu'il est question de composants logiciels réutilisables destinés au soutien d'un métier particulier : les termes d'objets métier (business objects), de composants métier (business components), de composants spécifiquement liés à un métier (business specific components) sont évoqués (Andro et Chauvet, 1998; Bouzeghoub et al., 1997; Herzum et Sims, 2000; Jacobson et al., 1997; OMG, 1995; OMG, 1997; Schmid, 1999; Sutherland, 1999). Ils ne recouvrent pas, suivant l'auteur qui les utilise, la même réalité. Il est donc nécessaire, avant d'aller plus avant dans ce travail, de définir clairement les termes que nous utiliserons, et ce au travers de la

discussion de quelques définitions. Selon l'Object Management Group (OMG) :

"A business object is a representation of a thing active in the business domain, including at least its business name and definition, attributes, behaviour, relationships, rules, policies and constraints. A business object may represent, for example, a person, place, event, business process or concept." (OMG, 1997)

Un objet métier est donc un modèle, une abstraction d'un concept réel relatif à un domaine d'activité, qui doit être décrit de manière détaillée. Le terme d'objet métier est ici dénué de toute connotation informatique. De tels modèles peuvent être utilisés par exemple pour mieux comprendre et réorganiser les processus de gestion d'une organisation, et être suffisants pour de tels objectifs. Cependant, à l'heure actuelle, la plupart des applications logicielles sont conçues suivant une approche orientée objet. La notion d'objet métier peut avec profit être mise à contribution dans ce contexte : des objets au sens informatique du terme, mais qui représentent ces mêmes concepts réels relatifs à un domaine d'activité, peuvent être instanciés et manipulés dans un système d'information. L'OMG propose d'ailleurs une seconde définition du terme "objet métier" qui le suggère :

"A business object captures information about a real world (business) concept, operations on that concept, constraints on those operations, and relationships between that concept and other business concepts. The business concept can then be transformed into a software design and implementation. And a business application can be specified in terms of interactions among a configuration of implemented business objects." (OMG, 1997)

Cette seconde définition introduit une deuxième facette du terme "objet métier" : un objet métier y est défini d'une part, conformément à la définition précédente, comme une représentation d'un concept du monde réel, sans connotation informatique aucune, mais aussi comme une abstraction informatique de ce concept. Le lien est établi entre ces deux notions : l'objet métier "abstraction d'un concept réel" peut être transformé en une abstraction logicielle et être manipulé dans le contexte d'une application. Il faudrait donc, pour être précis, distinguer les notions d'objet métier et d'objet métier informatisé. Cette distinction ne sera toutefois faite que lorsque ce sera nécessaire.

Quelle différence y a-t-il alors entre un objet informatisé (chapitre 4.2) et un objet métier informatisé ? A vrai dire aucune, si ce n'est que les objets métier informatisés constituent un sous-ensemble des objets informatisés (Eeles et Sims, 1998). Un objet métier représente un concept du monde réel utile dans un contexte particulier. Un objet est instancié à partir d'une classe. Il est peu probable qu'il suffise d'une seule classe pour implémenter un type d'objet métier : seront nécessaires une ou plusieurs classes-métier, qui implémentent le concept-métier en question, des classes "utilitaires" (Quantité, Montant, Adresse, Mesure par exemple) ainsi que des classes "techniques", à connotation purement informatique (Collection, Tableau, Chaîne de caractères, etc). Ainsi, un objet métier pourra être instancié et manipulé grâce à la collaboration de multiples objets instanciés à partir des nombreuses classes correspondantes.

L'objectif du présent travail est de proposer des concepts facilitant la réutilisation. Distribuer les différentes classes qui constituent l'implémentation d'un type d'objet métier ne constitue pas une solution adéquate, car d'une part il est difficile de réutiliser un ensemble de classes, et d'autre part, les développeurs à l'origine de ces classes ne souhaitent pas nécessairement rendre public le code

implémentant lesdites classes. Les composants (chapitre 4.3.2) constituent par contre un moyen aisé de regrouper et de distribuer les différentes classes permettant l'instanciation d'un objet métier, et offrent à des développeurs tiers la possibilité de mettre en oeuvre ces objets métier. Des composants qui permettent l'instanciation et la manipulation d'objets métier sont appelés composants métier. La littérature propose également ici un certain nombre de définitions :

"A business component is the software implementation of an autonomous business concept or business process. It consists of all the software artifacts necessary to represent, implement, and deploy a given business concept as an autonomous, reusable element of a larger distributed information system." (Herzum et Sims, 2000)

"Les composants métier sont des composants de grande taille, des classes composées de nombreuses classes, et conçus pour faciliter la réutilisation au niveau de la conception." (Bouzeghoub et al., 1997)

Nous retrouvons dans ces définitions les points venant de faire l'objet d'une discussion. Nous avons distingué, au chapitre 4.2, les objets, qui sont des instanciations de classes, des composants, qui ont une réalité physique puisqu'il s'agit de code compilé. Cette distinction se retrouve bien sûr au niveau des composants métier : un objet métier est instancié à partir d'un composant métier, mais n'est pas une instance de composant métier (figure 45).

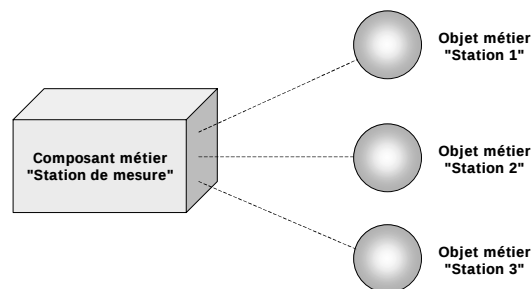


Figure 45. Les objets métier sont instanciés à partir de composants métier (adapté de Eeles et Sims, 1998).

Un composant métier exporte les interfaces d'objets métier, permettant ainsi à l'utilisateur du composant métier, appelé client, de manipuler les objets métier concernés. Les composants métier implémentent les objets métier à l'aide d'un ensemble de classes. Un composant métier est cependant plus que cela : comme le montre la définition donnée par Herzum, il doit être autonome. Cette exigence a des implications quant à la structure qui doit être la sienne. Cette problématique est détaillée au chapitre qui suit.

4.5.3 Structure interne d'un composant métier

4.5.3.1 Introduction

Parmi les exigences formulées en regard des SIAGEM, il était demandé que ceux-ci puissent être adaptés à différentes configurations informatiques (chapitre 3.2.2). Nous avons décidé de concevoir les SIAGEM comme des assemblages de composants logiciels réutilisables. Ceci implique que ce sont ces

composants qui doivent pouvoir être adaptés à différentes configurations informatiques. De plus, chaque composant métier doit être autonome (Herzum et Sims, 2000), et donc être à même de remplir toutes les tâches nécessaires à l'accomplissement des fonctions dont il est responsable : cela implique qu'un composant métier est lui-même composé de différentes parties assurant la persistance des données et la représentation graphique, intégrant les différentes logiques propres au composant, etc., qui doivent pouvoir s'adapter à différentes configurations et former un tout opérationnel. Un composant métier est donc composé de différentes parties, qui ont des responsabilités différentes, mais qui toutes concourent à un même but : permettre au composant métier d'assurer son rôle. Ces différentes responsabilités conduisent à définir une architecture interne pour chaque composant métier.

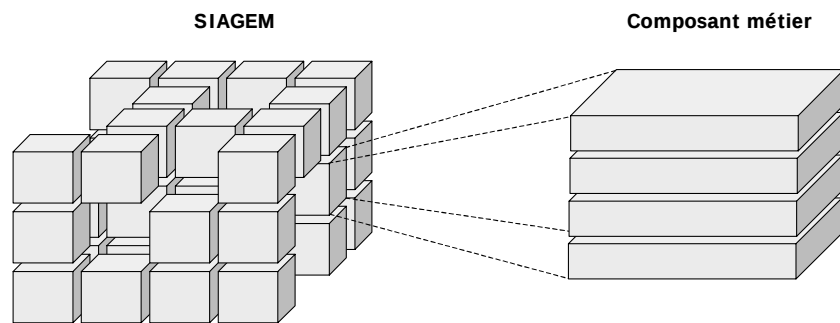


Figure 46. Un SIAGEM est constitué d'un assemblage de composants métier sélectionnés en fonction des besoins. Chaque composant métier est lui-même constitué d'un ensemble de composants logiciels.

Il ne faut pas oublier qu'un composant métier est un type de composant et nécessite, pour pouvoir être mis en oeuvre, un environnement pour composants. Ainsi, un composant métier n'est opérationnel que dans l'environnement pour lequel il a été conçu. Les environnements pour composants ayant été présentés succinctement au chapitre 4.3, ce sujet ne sera pas discuté ici.

4.5.3.2 Les différentes constituantes d'un composant métier

Un composant métier constitue une entité autonome : un concepteur d'applications travaillant par intégration de composants métier doit pouvoir, s'il le souhaite, utiliser toutes les fonctions promises par le dit composant métier, en ne se préoccupant que de leur intégration dans l'application qu'il développe.

Un composant métier peut donc être vu comme une application autonome et très spécialisée, qui doit pouvoir s'adapter à différentes configurations, permettre la manipulation d'objets métier par le biais d'interfaces graphiques pré-programmées, et offrir ses services au travers d'interfaces bien définies. L'annexe 4 montre que les systèmes logiciels contemporains peuvent être organisés en couches, et que cela facilite leur maintenance et leur adaptation à des conditions changeantes. Cet annexe détaille par ailleurs les motivations ayant conduit à une telle proposition, ainsi que les rôles qui sont dévolus aux différentes couches. Un même principe peut être appliqué aux composants métier, qui peuvent être vus comme des "tranches verticales et spécialisées" d'applications. Comme de plus il est plus facile de maintenir et de faire évoluer une application bâtie à partir de composants qu'une application monolithique, un composant métier doit être lui-même constitué d'un ensemble de composants jouant chacun un rôle spécifique, structuré selon un principe identique à celui des couches logicielles présentées en annexe 4. Après analyse des différentes propositions trouvées dans la littérature (Herzum et Sims,

2000; Schmid, 1999), nous proposons d'intégrer dans un composant métier les composants suivants : composants d'interface-utilisateur, composants de logique-utilisateur, composants domaine, composants données et composants ressources (figure 47 et 48). À noter que l'illustration 48 présente l'extension maximale d'un composant métier : un composant peut intégrer seulement certaines des couches proposées, en fonction de son type. De plus, il est également possible dans certains cas d'agréger des couches adjacentes dans un même composant.

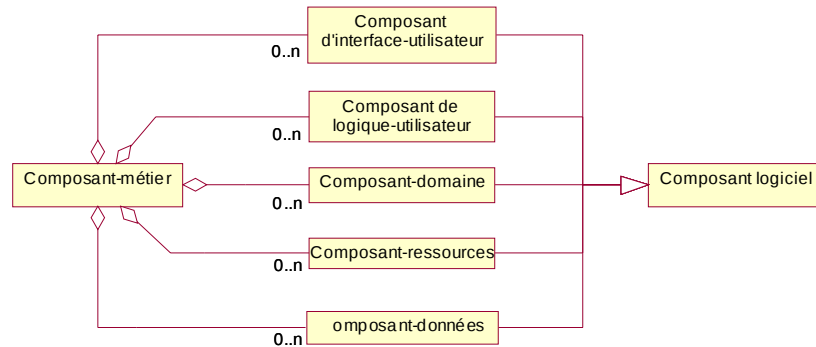


Figure 47. Méta-modèle des composants métier (pour le formalisme, voir annexe 2).

- Les composants d'interface-utilisateur permettent à l'utilisateur, en lui présentant l'information relative aux objets métier et en transmettant ses demandes à ces derniers, d'interagir avec les objets métier. Les composants d'interface-utilisateur peuvent être soit des formulaires prêts à l'emploi, soit des composants d'interfaces graphiques (composants ActiveX ou JavaBeans de type menu déroulant, carte, graphique, tableau, etc.) que l'intégrateur d'objets métier insère, en fonction de ses besoins, dans les interfaces graphiques développées pour un SIAGEM particulier (figure 49).

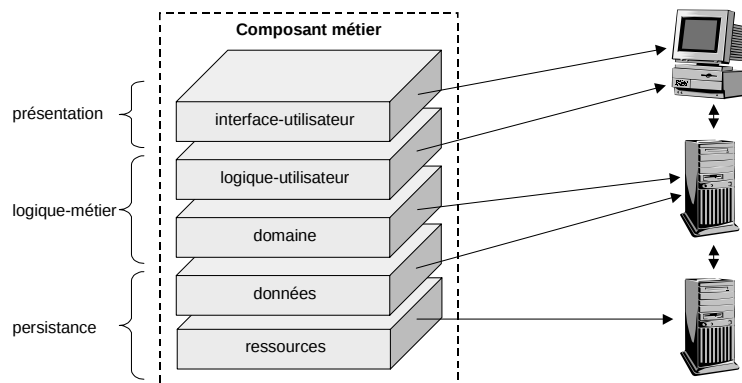


Figure 48. Les différentes constituantes que peut comprendre un composant métier, et qu'il est possible de répartir physiquement de différentes manières. Un regroupement des composants d'interface-utilisateur et de logique-utilisateur, ainsi que des composants domaine, données et ressources, conduit à une architecture classique en trois couches (la troisième couche étant constituée des BD).

- Les composants de logique-utilisateur implémentent les fonctionnalités propres à un utilisateur particulier qui doivent être exécutées sur l'ordinateur de ce dernier. Ces composants génèrent par exemple de l'information qui doit être affichée par les composants d'interface-utilisateur, à partir des informations fournies par les composants-domaines, ou transforment l'information fournie par les utilisateurs en information compréhensible par les composants-domaine. Les composants de

logique-utilisateur sont localisés sur l'ordinateur de l'utilisateur, tout comme les composants d'interface-utilisateur : ces deux catégories de composants pourraient de ce fait être intégrés dans un même composant. Ils sont toutefois distingués car c'est un principe de bonne conception logicielle que de séparer strictement la logique d'une application des fonctionnalités d'affichage et de représentation de l'information à l'écran.

- Les composants-domaine sont les composants les plus importants car ils implémentent la logique liée spécifiquement au domaine auquel se rapporte le composant métier en question, qui est commune à un certain nombre d'utilisateurs. Ce sont les composants-domaine qui implémentent les interfaces relatives aux objets métier et matérialisent la portion de modèle objet du domaine dont le composant métier assure la réalisation. Un intégrateur de composants métier manipulera fréquemment directement les interfaces des objets métier implémentés par le composant-domaine.



Figure 49. Un composant métier relatif aux ouvrages d'assainissement peut par exemple intégrer des modèles de formulaires relatifs à ces ouvrages, ou des composants d'interfaces graphiques dotés de fonctionnalités spécifiques : ici un menu déroulant, basé sur la technologie ActiveX, et présentant à l'utilisateur les domaines de valeurs des attributs du domaine de l'assainissement.

- Les composants-données assurent aux composants-domaine un certain nombre de services : ils implémentent les différentes classes qui sont nécessaires pour que la correspondance entre les objets métier dits persistants et les données qui les caractérisent dans diverses BD puisse être assurée. Ils collaborent avec les composants-ressources pour enregistrer, supprimer, modifier ou retrouver les données relatives aux composants-domaine.
- Les composants-ressources permettent d'accéder à des ressources extérieures, dans notre cas principalement d'établir la liaison physique avec des sources de données, en principe des BD relationnelles. Un composant-ressource peut cependant également établir le lien avec des systèmes de fichier, avec des sources de données accessibles par le biais d'Internet, ou des systèmes de stockage de données particuliers comme des séries temporelles, des données géographiques ou des données multi-médias. Généralement, en ce qui concerne l'accès à des BD relationnelles, on utilise comme composants-ressources les composants techniques standard faisant partie des différents environnements de programmation (dans le cadre de ce travail a été mis en oeuvre le composant ADO, c'est-à-dire ActiveX Data Objet, de Microsoft).

La structure qui doit être celle d'un composant métier a ainsi été décrite. Suivant le type de composant et le choix de son concepteur, certaines des couches proposées peuvent être absentes : un composant-entité par exemple peut ne pas intégrer de composants d'interface ou de logique-utilisateur, la conception de ces dernières étant laissée à la charge de l'intégrateur (figure 50). Il sera opté pour une telle solution en particulier lorsqu'il n'est pas possible de standardiser ne serait-ce que des portions d'interface ou de logique-utilisateur. Un tel choix a été effectué en ce qui concerne le développement du prototype

présenté au chapitre 6.2 : voulant laisser à l'intégrateur la responsabilité de concevoir l'interface-utilisateur permettant d'accéder aux objets métier, le composant métier proposé ne comprend que les composants-constituants domaine, données et ressources.

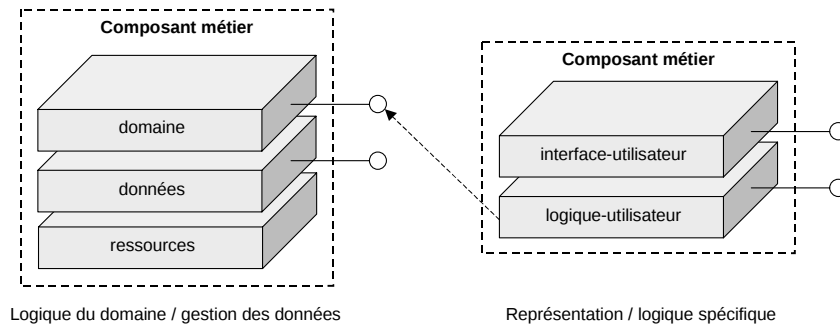


Figure 50. Les responsabilités de composants métier peuvent être réparties de différentes manières.

Un composant métier chargé de la gestion d'un processus peut ne pas avoir besoin d'assurer son état d'une session à l'autre, et donc n'intégrer ni composant-données, ni composant-ressources. De plus, une telle architecture, dans laquelle chaque couche est chargée de responsabilités bien précises, rend l'adaptation d'un même composant métier à des situations différentes beaucoup plus facile.

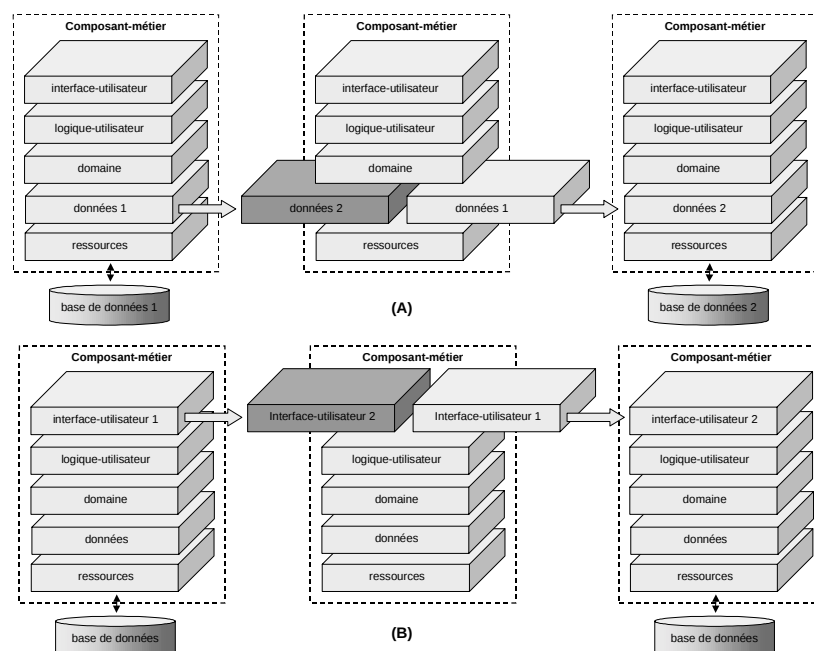


Figure 51. La structure en couches des composants métier leur permet d'être facilement adaptés.

Ce fait est illustré à la figure 51 : dans le premier cas présenté (A), on souhaite adapter un composant métier existant à une BD organisée selon un modèle différent. Il suffit dans ce cas de créer une nouvelle version du composant-données et de l'insérer dans le composant métier en lieu et place du composant-données existant. Dans la mesure où les interfaces entre composants-constituants d'un composant métier sont bien définies, cet échange se fait sans problème, et deux versions du même composant métier qui stockent leurs données dans des BD organisées différemment peuvent coexister. Dans le second cas présenté (B), une interface-utilisateur différente est souhaitée : il suffit de remplacer, au sein du

composant métier, le composant d'interface-utilisateur pour à nouveau disposer, à moindres frais, de deux versions d'un même composant métier.

La logique métier est ainsi disséminée au travers de toutes les couches : les interfaces-utilisateurs sont conçues pour permettre la visualisation d'objets métier, la couche-donnée doit intégrer la connaissance du modèle logique de données, et la BD, qui peut être considérée comme extérieure au composant métier, contient les tables relationnelles relatives aux différents objets métier. C'est pourquoi la description d'un composant métier autonome doit inclure une description de l'ensemble de ses composantes.

Cette exigence d'autonomie implique qu'une description de composant englobe également la documentation, les modèles de données, les logiciels de configuration, etc., nécessaires à sa mise en oeuvre.

4.5.3.3 Composants métier et bases de données

Le fait que les composants métiers soient des unités autonomes implique que chacun d'entre eux doit disposer de l'information nécessaire quant à la manière d'assurer sa persistance (à partir du moment où celle-ci doit être assurée). Un composant métier doit donc inclure de l'information quant à la manière dont les données qui le concernent dans la BD sont structurées. Dans le cas où ces données sont gérées par une BD relationnelle (ce qui est très souvent le cas), un composant métier doit avoir connaissance du modèle logique de données correspondant aux objets métier persistants qu'il permet de manipuler, soit les noms des tables relationnelles et des champs correspondant à ces données. Il doit donc toujours être accompagné d'une documentation qui contient, entre autres, une description du modèle de données sur lequel il s'appuie, et qui doit être implémentée dans la BD qu'il utilise. À noter que les composants-données peuvent être conçus de manière à être relativement souples quant aux structures de données sous-jacentes et doivent dans ce cas faire l'objet d'une configuration, l'information permettant d'assurer la correspondance (mapping) entre les objets métier et les tables relationnelles étant soit stockée dans la BD, soit dans un fichier spécifique, et peut être modifiée.

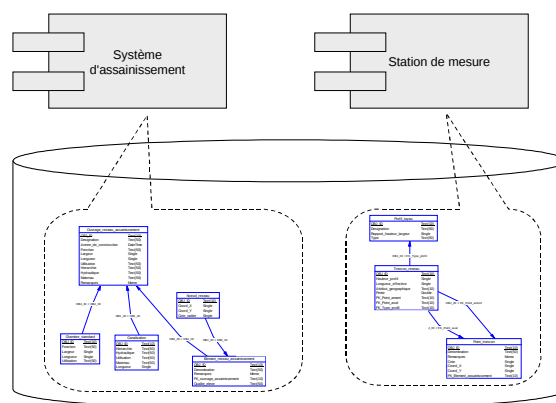


Figure 52. À deux composants métier différents correspondent des "îlots de données", constitués d'ensembles isolés et indépendants de tables relationnelles.

En théorie, chaque modèle logique de données caractérisant un composant métier peut être implémenté

dans une BD différente. Généralement cependant, les modèles de données correspondant aux différents composants métiers sont implémentés dans une même BD, et constituent ainsi des "îlots de données" (figure 52), pour reprendre l'expression de Herzum (2000), indépendants les uns des autres.

Une application logicielle conçue par intégration de composants métier doit toujours, pour modifier au sein d'une BD les données relatives à un objet métier particulier, passer par ce dernier. Ainsi, un composant métier ne doit jamais directement accéder dans une BD à des enregistrements correspondant à un composant métier tiers. Cette règle permet de construire des applications dont la maintenance est plus facile, assure l'indépendance des composants logiciels et la cohérence des données. En effet, la logique contrôlant la cohérence des données relatives à un type d'objet métier ne sera implémentée que dans le composant métier responsable de l'objet métier en question. Modifier des données relatives à un objet métier sans passer par ce dernier présente donc le risque de corrompre ces données.

4.5.3.4 Communication interne au composant métier

Les différents composants qui constituent un objet métier présentent une ou plusieurs interfaces qui leur permettent de communiquer entre eux. Ces interfaces peuvent être privées, c'est-à-dire servir uniquement à la communication entre composants au sein du composant métier, ou publiques, donc accessibles par l'utilisateur du composant. Les interfaces publiques permettent l'intégration des composants métier dans des systèmes d'information et la communication entre différents composants métier.

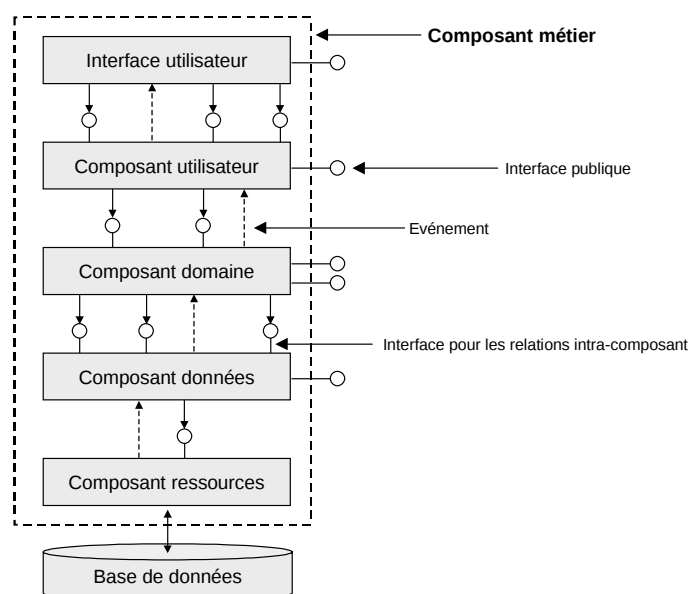


Figure 53. Les composants d'un niveau plus élevé ont une référence vers les interfaces des composants de plus bas niveaux qu'ils utilisent, et capturent les événements produits par ces derniers (adapté de Herzum et Sims, 2000).

Les différents éléments qui constituent un composant métier doivent pouvoir, comme nous l'avons vu, être répartis sur différents ordinateurs. Ils doivent donc pouvoir communiquer par l'intermédiaire d'un réseau informatique. En effet, certains de ces éléments (composants interface-utilisateur et logique-utilisateur) doivent nécessairement se trouver sur l'ordinateur de l'utilisateur du système d'information,

tandis que les autres composants peuvent être localisés physiquement soit sur ce même ordinateur, soit sur d'autres machines. C'est pourquoi les composants-constituants d'un composant métier doivent être des composants distribuables, et chacun d'entre eux doit disposer d'au moins une interface accessible depuis des ordinateurs distants (Herzum et Sims, 2000). Une abondante littérature technique relative aux composants distribués est disponible, et comme l'objectif du présent travail n'est pas de présenter des technologies informatiques, nous ne développerons pas plus la problématique des composants et applications distribuées.

Au sein d'un même composant métier, des dépendances circulaires (à savoir deux composants interdépendants) doivent être évitées (Herzum et Sims, 2000). De ce fait, les composants ne doivent communiquer directement entre eux (soit par le biais de références directes) que dans le sens allant des composants d'interface-utilisateurs aux composants-ressources, la communication dans le sens inverse se faisant par le biais d'événements. *"Un événement est un message, un signal ou un stimulus [...]. Un système [dans notre cas un composant] n'intercepte un événement que s'il possède le moyen de réagir à cet événement."* (Bouzeghoub et al., 1997). Un exemple simplifié permet d'illustrer ce propos : un composant d'interface graphique (un formulaire par exemple) permet la représentation d'objets métier de type Chambre d'assainissement. Il peut donc avoir une référence directe vers un tel objet métier. Un changement de l'état de cet objet métier (changement de la valeur d'un attribut par exemple) doit être reflété par le formulaire en question. L'objet métier n'a cependant pas de référence directe vers le formulaire - en fait, il ne sait pas combien de formulaires ou autres interfaces graphiques se chargent de son affichage -, mais génère un événement qui est capturé par le formulaire (ainsi que par tous les éléments intéressés par cette information), ce dernier modifiant alors automatiquement les informations qu'il présente. Une liaison par événements peut contribuer à diminuer le degré de couplage entre composants, car elle diminue le nombre de références directes entre ceux-ci. La figure 54 illustre ce propos.

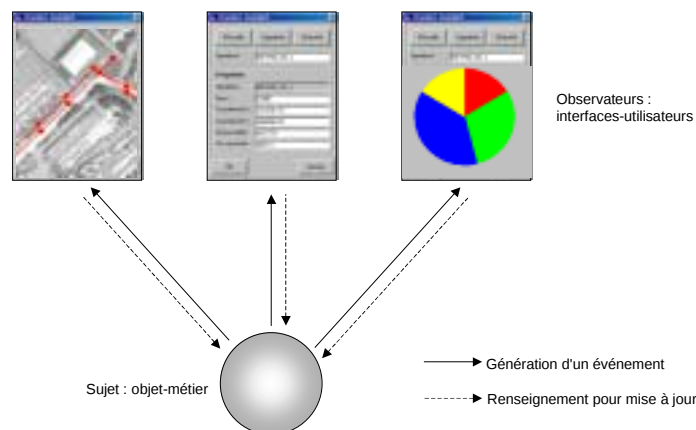


Figure 54. Une modification de l'objet métier est notifiée par le biais d'un événement aux interfaces graphiques, qui peuvent interroger l'objet métier et mettre à jour les informations qu'elles affichent (adaptation du modèle de conception Observateur, d'après Gamma et al., 1995).

4.5.4 Les catégories de composants métier

4.5.4.1 Les catégories

La littérature distingue généralement trois catégories d'objets métier : les objets métier de type entité, de type processus et de type utilitaire (Eeles et Sims, 1998; Herzum et Sims, 2000; Schmid, 1999).

Objets métier de type entité : ils représentent les éléments "statiques" et réels du domaine d'activité considéré, et correspondent généralement aux éléments identifiés lors de la conception de modèles de données. Ils possèdent d'ailleurs la plupart du temps un état dont la pérennité doit être assurée. À noter qu'un système d'information peut être amené à manipuler simultanément de très nombreux objets métier de type entité appartenant à une même catégorie (par exemple toutes les personnes habitant dans une commune). Exemples de tels types d'objets métier : Station d'épuration, Commune, Bâtiment ou Canalisation.

Objets métier de type processus : ils représentent des activités (processus métier) du domaine considéré. De tels objets métier appuient les utilisateurs dans la réalisation de leurs tâches. Pour cela, ils manipulent et utilisent des objets métier de type entité. Ils ne sont généralement pas caractérisés par un état, et souvent une seule instance d'un tel composant est mise en oeuvre dans le SI. Ces objets métier peuvent par exemple être chargés d'assister un ingénieur hydrologue dans la préparation des données en vue de simulations de comportements (simplifier le réseau d'assainissement, les séries de mesures pluviométriques, etc.), ou organiser de manière optimale des tournées de prélèvement d'échantillons d'eau en vue d'analyses.

Objet métier de type utilitaire : il s'agit là d'objets métier qui peuvent être mis en oeuvre dans des SI se rapportant à des contextes différents. Ils sont utilisés par les objets métier de type processus et de type entité. Ils sont de granularité plus faible que les deux catégories d'objets métier précédemment citées. Une mesure, une série temporelle de mesures, une adresse, une valeur monétaire, ainsi que les objets à référence spatiale comme les points, les lignes ou les polygones, sont des exemples d'objets métier de type utilitaire. Un objet métier appartenant à cette catégorie ne présente pas d'intérêt pris isolément : une adresse ou un polygone ne servent à rien s'ils ne se rapportent pas à un autre objet métier.

On peut noter que les objets métier de type utilitaire sont normalement très stables, n'évoluent quasiment pas dans le temps : une adresse reste une adresse. Ils sont de plus aisément standardisables. Les objets métier de type entité présentent également une grande stabilité, mais dans un domaine beaucoup plus spécifique. Les objets métier de type processus sont généralement présentés, dans le contexte des entreprises, comme des éléments peu stables, car appelés à évoluer pour permettre à l'entreprise de rester concurrentielle. Les organisations impliquées dans la gestion des eaux ne se trouvent pas dans un contexte de concurrence, mais cherchent au contraire à collaborer afin de satisfaire aux exigences de la législation et à assurer la gestion des infrastructures et milieux naturels de la manière la plus efficace possible. Ainsi, dans la mesure où les entités manipulées et les processus mis en oeuvre présentent de grandes similarités d'une organisation à l'autre, les objets métier de type processus recèlent dans le domaine de la gestion des eaux un potentiel de standardisation plus élevé que dans le contexte des entreprises privées.

Nous adopterons, au niveau des composants métier, les distinctions présentées ci-dessus, et parlerons de composants métier de type entité, de type processus et de type utilitaire (de manière abrégée, de composants-entité, processus et utilitaire). Les objets et composants métier peuvent parfois jouer le rôle de "façades" ou de "médiateurs". Dans la mesure où ces notions sont très importantes puisque leur prise en compte permet de maximiser le potentiel de réutilisation et la facilité d'utilisation des composants métier, nous les introduisons ci-dessous.

4.5.4.2 La notion de médiateur

L'une des tâches les plus importantes de la phase de conception d'un processus de développement orientée objet est l'attribution des différentes responsabilités à des classes précises (Larman, 1997). Ces responsabilités peuvent impliquer que les instances de ces classes soient amenées à collaborer avec de nombreux autres objets de types différents, dans le but de mener à bien les opérations invoquées. Cela implique l'existence d'un réseau de relations plus ou moins fortes entre ces objets. Lorsque l'objectif du développement est une application monolithique, ce n'est pas trop problématique, bien que certains principes doivent être respectés. Par contre, à partir du moment où l'on adopte une approche par composants, ces derniers doivent être le plus indépendants possible les uns des autres, afin de pouvoir être utilisés dans des contextes différents. Des relations directes entre objets générés à partir de composants différents doivent autant que possible être évitées. De tels liens entre les objets de type A et de type B sont par exemple les suivants :

- le type A possède un attribut de type B;
- le type A implémente une opération dont le résultat ou un argument est une instance de type B;
- le type A est une sous-classe du type B.

Pour éviter ce genre de lien, on peut mettre en oeuvre le concept de médiateur, utilisé dans de nombreux contextes. Au niveau de la conception orientée objet, Gamma et al définissent un médiateur comme *"un objet qui encapsule les modalités d'interaction d'un certain ensemble d'objets. Le médiateur favorise le couplage faible en dispensant les objets de se faire explicitement référencer"* (Gamma et al., 1995). Imaginons par exemple une opération dont l'objectif est d'obtenir le nombre d'habitants raccordés au réseau d'assainissement en amont d'un ouvrage particulier (par exemple pour estimer le débit théorique d'eaux usées par temps sec dans l'ouvrage en question). On aurait tendance, dans une application orientée objet traditionnelle, à attribuer cette responsabilité à l'ouvrage d'assainissement lui-même, suivant en cela le modèle de conception¹ Expert (Larman, 1997). A priori, l'objet de type Ouvrage d'assainissement devrait, pour assumer cette responsabilité, collaborer avec, entre autres, des objets de type Bâtiment, Adresse et Personne. Selon une approche par composants métier, la probabilité est forte que ces différents types d'objets soient implémentés par des composants métier différents. Pourtant, l'opération dont il est ici question implique une étroite collaboration entre les objets concernés, qui seront donc fortement couplés, impliquant ainsi une forte dépendance entre les composants qui les implémentent. Le concept de médiateur permet de diminuer ce couplage : la responsabilité de piloter l'opération peut être attribuée à un médiateur qui assurera la coordination de la collaboration entre les différents objets métier impliqués dans cette opération. Ainsi le couplage direct entre ces derniers peut être évité. Ce concept extrêmement important doit être gardé à l'esprit et mis en oeuvre par exemple pour créer des composants

1. Modèles de conception : *"Design patterns are reusable design techniques that you can apply to conceptual similar situations in different applications."* (Tanrikorur, 1999)

spécialisés dans le calcul d'indicateurs et s'appuyant pour ce faire sur des composants métier de type entité. Ainsi, lorsqu'une opération implique la collaboration de nombreux objets métier de types différents, pas forcément implémentés par les mêmes composants métier, il est conseillé de créer un composant jouant ce rôle de médiateur contrôlant cette opération et permettant de découpler les composants-entité qui implémentent les objets impliqués dans cette opération (figure 55, des composants métier de type processus, s'appuyant sur des composants métier de type entité, sont de fait des médiateurs). Les composants de type médiateur, en organisant la collaboration entre composants tiers, en assurent l'indépendance.

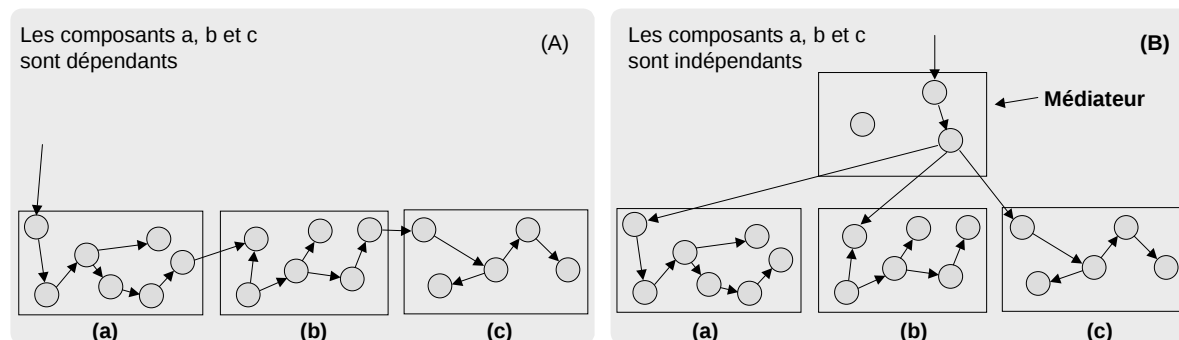


Figure 55. (A) : l'invocation de l'opération d'un objet métier implique la collaboration d'objets issus de différents composants. Ces derniers sont fortement dépendants. (B) : un objet métier joue le rôle de "médiateur", assume la responsabilité de l'opération, et assure la coordination de la collaboration entre objets issus de composants différents. Ces derniers sont indépendants.

4.5.4.3 La notion de façade

Un modèle objet relatif au domaine de la gestion des eaux est extrêmement complexe, en raison des nombreuses catégories d'intervenants auxquels il s'adresse, et qui doivent y identifier leurs "construits". En effet, un tel modèle constitue une synthèse de visions différentes d'un même système réel. C'est ici l'une des particularités des SIRS, déjà examinée aux chapitres 2.4.1.3 et 4.4. De ce fait, certains composants métier, de type entité en particulier, permettent la manipulation d'objets métier qui certes conviennent aux différents utilisateurs et s'adaptent à de nombreuses situations, mais sont difficiles à appréhender.

Ainsi, le modèle objet proposé par l'ASPEE (VSA, 1999), très complet et intégrant des éléments relatifs à la topologie du réseau d'assainissement (tronçons et noeuds de réseau) rend possible la modélisation de quantité de types d'ouvrages, même dotés d'une hydraulique complexe. Ce modèle est cependant aussi difficile à comprendre. Ainsi certains ouvrages pourtant très communs sont compliqués à modéliser. Là encore, les modèles de conception offrent une solution adaptée : le modèle de conception "façade" *"fournit une interface unifiée à l'ensemble des interfaces d'un sous-système. La façade fournit une interface de plus haut niveau, qui rend le sous-système plus facile à utiliser."* (Gamma et al., 1995). Nous exploitons ce concept à l'échelle du composant : en ce qui concerne le territoire et ses éléments, il est possible de développer un ensemble de composants-entité à granularité très fine, qui permettent de modéliser tous les cas de figure. Peuvent être construits, selon cette notion de façade, des composants métier qui offrent de ceux de type entité à fine granularité, une vision simplifiée, propre à certains acteurs.

Différents composants jouant le rôle de façade peuvent ainsi s'appuyer sur un même composant-entité ou composant-processus, en offrant une vue spécialisée, adaptée aux besoins d'un intervenant particulier. Par exemple, il a été constaté que l'objet métier "Déversoir" modélisé selon la proposition de l'ASPEE - soit un composant hydraulique se greffant sur un noeud du réseau - ne répond pas à la représentation qu'en ont communément la plupart des utilisateurs, bien que cette solution soit la plus efficace et la plus souple. Pour construire un ouvrage de type déversoir au sens où l'entendent la plupart des intervenants, tel que représenté sur la figure 56, de nombreux objets doivent être agrégés. Un composant jouant le rôle de façade, proposant par exemple une interface "Ouvrage déversoir", peut offrir d'un tel ouvrage une vue simplifiée.

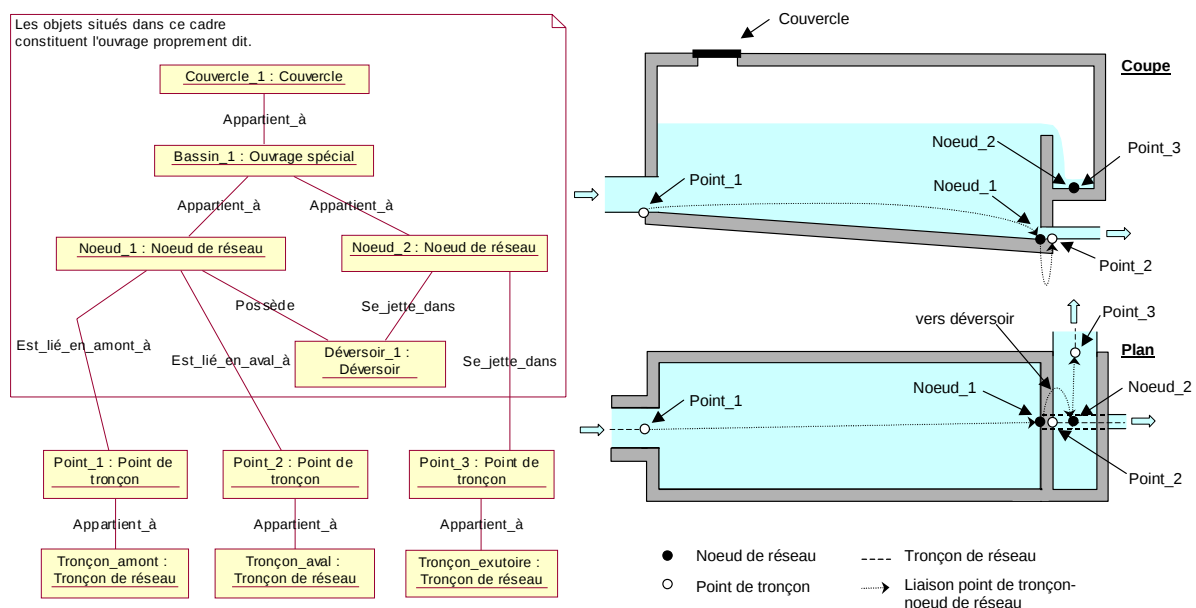


Figure 56. Déversoir d'orage avec bassin de rétention-décantation, selon les propositions de l'ASPEE (VSA, 1999), et modèle des objets correspondant (annexe 2). Un composant façade peut s'occuper de gérer le sous-système relatif à l'ouvrage proprement dit (dans le cadre) ainsi que les interactions de ce dernier avec son environnement, et en proposer une version simplifiée à l'utilisateur.

À la figure 57 est reporté le diagramme de séquence présentant la succession d'opérations qui résulte de la création d'un objet métier "façade" de type "Ouvrage déversoir simple". L'intervenant crée ainsi cet objet métier, spécifiant simultanément quelles sont les canalisations amont, aval et menant à l'exutoire. L'objet métier "Ouvrage déversoir simple" intègre des règles qui lui permettent de créer à son tour tous les objets métier de plus fine granularité, issus du composant métier de type entité "Ouvrage de réseau d'assainissement", et d'assembler ces derniers conformément à ce modèle simplifié d'un tel ouvrage. En l'absence d'un objet métier de type façade, l'utilisateur se verrait obligé de lui-même assurer les opérations assurées par la façade, et donc de connaître les règles qui président à l'assemblage des différents objets métier qui forment un ouvrage de type déversoir.

Nous pouvons ainsi nous appuyer, lors de la conception de SIAGEM, d'une part sur des composants qui ont pour mission de réaliser l'intégration entre les visions de différents acteurs et qui de ce fait proposent des objets métier relativement complexes, et d'autre part sur des composants jouant le rôle de façades, qui fournissent aux différents acteurs des visions différenciées, adaptées à leurs besoins, des concepts

manipulés.

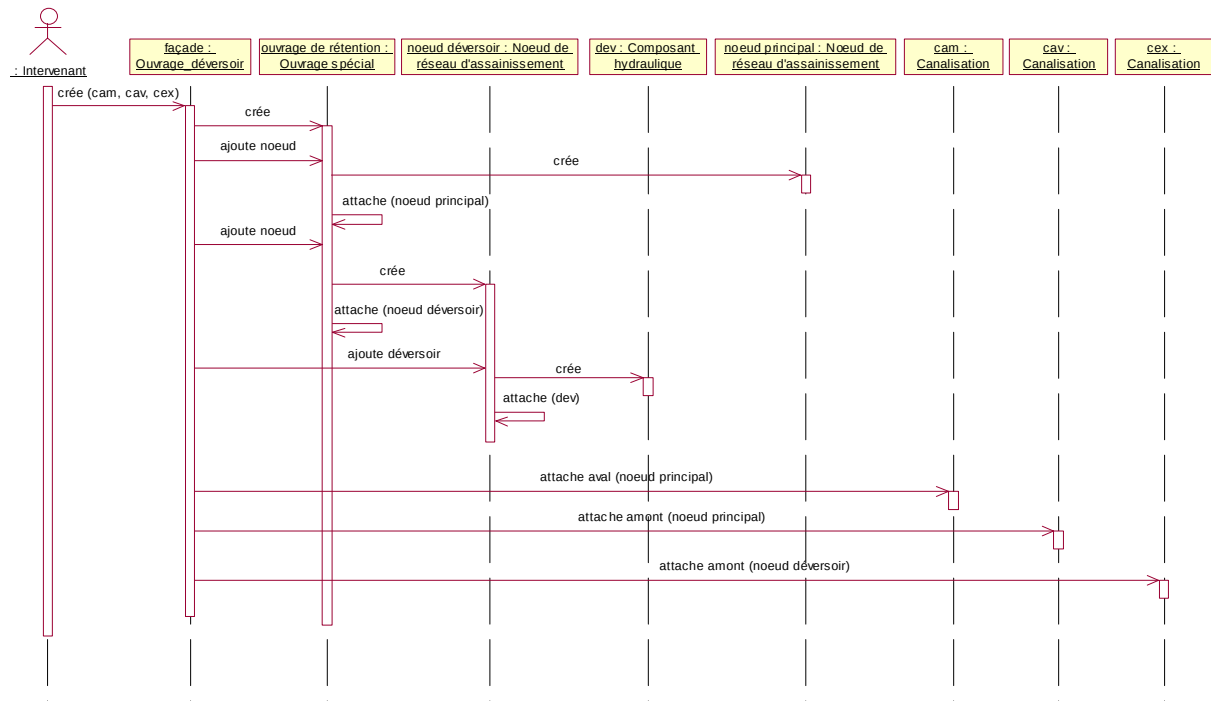


Figure 57. Diagramme de séquence (annexe 2) de la création d'un objet métier de type "Ouvrage déversoir simple". L'intervenant passe en paramètre les canalisations amont, aval et de déversement.

En résumé, trois catégories principales de composants métier peuvent être distinguées: les composants-entité, utilitaires et processus. Parmi ceux-ci, on en trouvera qui assureront des rôles de médiateur ou de façade. Ces deux notions sont très importantes car elles permettent d'une part d'assurer l'indépendance des composants métier et donc la modularité du système logiciel, et d'autre part de présenter aux différents utilisateurs des objets métier conformes à leurs besoins.

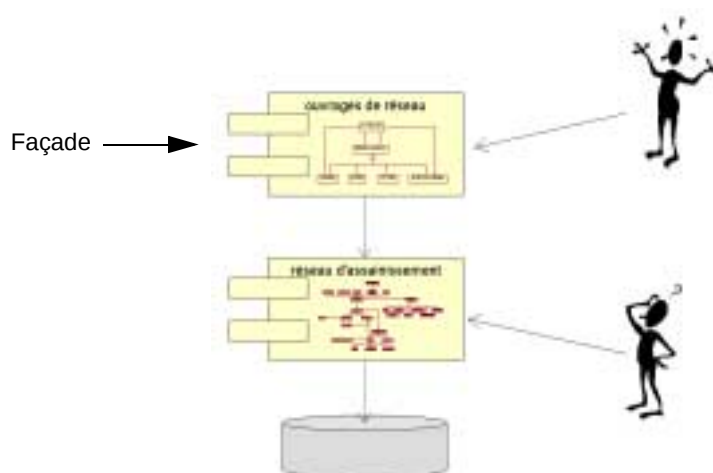


Figure 58. Les composants métier peuvent être trop compliqués pour certaines applications. Des composants jouent le rôle de façade et présentent aux utilisateurs qui en ont besoin des "vues simplifiées" des objets métiers.

4.5.5 Sémantique des composants métier

Différentes catégories de composants métier ont été décrites au chapitre précédent. Un composant métier implémente un certain nombre de concepts ou de processus métier autonomes (chapitre 4.5.2). Comment répartir ces différents concepts ou processus se rapportant à un domaine, en ce qui nous concerne celui de la gestion des eaux en milieu urbain, entre différents composants ? Quels sont les critères qui décident de quels concepts un composant métier doit permettre la manipulation ? Quelle doit être l'envergure d'un composant métier, c'est-à-dire de combien de concepts doit-il permettre la manipulation ? Il n'existe à vrai dire pas de règle précise à ce sujet, mais un certain nombre de principes de base doivent être appliqués :

- un composant métier doit intégrer un ensemble de concepts qui forment un tout cohérent. Un composant métier ne proposera donc pas des interfaces appartenant à plus d'un champ sémantique. Un composant métier intitulé "Station de mesure" par exemple n'implémentera pas d'interfaces relatives aux cours d'eau, et ce même si certaines stations de mesures sont mises en place pour mesurer des paramètres relatifs aux rivières.
- nous avons vu que ces différents concepts sont matérialisés au sein du composant par un certain nombre de classes, que nous appellerons classes-métier (concept qui peut par ailleurs être rapproché de celui de classe d'analyse dans le contexte des différentes étapes d'un cycle de développement, voir chapitre 5). Pour assurer la cohérence du composant métier, une classe-métier principale, correspondant à un concept que nous qualifierons de "concept-clé", doit être identifiée, et peut donner son nom au composant métier lui-même. Ce dernier doit proposer l'ensemble des interfaces qui ont un rapport direct avec ce concept-clé, et qui permettent sa manipulation de manière indépendante. Le composant métier "Ouvrage de réseau d'assainissement" par exemple doit proposer, entre autres, des interfaces permettant de manipuler les différentes catégories d'ouvrages d'assainissement (canalisation, chambre standard, etc.) ainsi que leurs différentes composantes : couvercle, déversoir, etc.(figure 59).

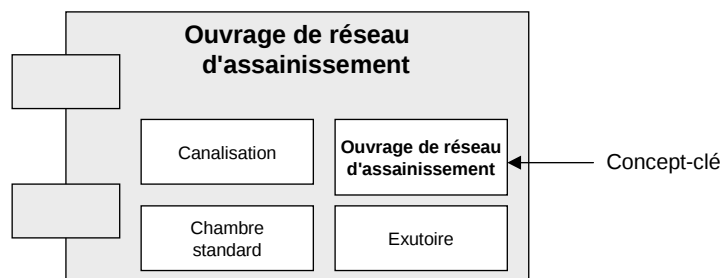


Figure 59. Un composant métier permet la manipulation d'objets métier implémentés dans le composant-domaine sous forme de classes. Les objets métier issus d'un même composant doivent former un ensemble cohérent.

- les différentes interfaces proposées par un composant métier doivent avoir une signification bien définie pour les intervenants du domaine d'activité en question en représentant pour ces derniers quelque chose de concret. Comme le dit Herzum (2000), *"good candidates for business components are those business concepts that are real and independent in the business domain."*
- la raison d'être d'un composant métier est de pouvoir être intégré dans différentes applications, d'être réutilisé. Il faut donc qu'une demande existe pour un tel composant, et donc qu'il se rapporte à des concepts récurrents, qui sont manipulés dans le cadre de différents processus métier, ou dans le

cadre d'un même processus métier mené par de nombreuses organisations différentes. Avant de se lancer dans la conception d'un composant particulier, il est donc nécessaire de déterminer si une telle entreprise en vaut la peine. La question de la durabilité du composant projeté doit également être prise en considération : s'il se rapporte à des concepts qui risquent de tomber rapidement en désuétude, sa mise sur pied ne se justifie pas nécessairement.

- il est recommandé de ne pas intégrer deux concepts susceptibles d'être manipulés dans deux applications aux finalités différentes dans le même composant.
- selon Herzum et Sims (2000), si des intervenants appartenant à des catégories différentes ont à mettre en oeuvre un même composant métier, il est probable que plusieurs composants métier différents devraient être conçus. Dans le cas qui nous intéresse ici, nous avons vu que nous utilisons la dimension intégratrice du territoire pour faciliter la collaboration entre intervenants d'horizons différents, et que cette collaboration se fait justement par le biais de la fourniture à ces différentes catégories d'intervenants de composants métiers de type entité identiques. Dans le contexte de notre travail, ce critère ne doit donc pas être appliqué.
- un composant métier, afin d'être utile, doit avoir une envergure suffisante, c'est-à-dire permettre la manipulation d'un nombre raisonnable et adéquat d'objets métiers. Les approches et techniques à mettre en oeuvre pour déterminer de quels objets métier un composant métier doit permettre la manipulation, et partant, quelles classes doivent être implémentées par les composants-domaine, font l'objet du chapitre 5.3.2. L'expérience a cependant montré qu'un composant métier implémente généralement environ 10 classes-métier, et que si ce nombre est inférieur à 5 ou supérieur à 150 cela dénote probablement un problème de conception. Il a également été constaté qu'un domaine peut généralement être couvert par 30 à 40 composants métier. Enfin, nous avons vu que le modèle de données permettant d'assurer la persistance d'un composant métier fait également partie de la définition de ce dernier : l'îlot de données (chapitre 4.5.3.3) relatif à un composant métier particulier comprend généralement entre 5 et 20 tables relationnelles (Herzum et Sims, 2000). Le composant métier développé dans le cadre de ce travail (chapitre 6.2) implémente 18 classes-métier et utilise pour assurer leur persistance une vingtaine de tables relationnelles.

4.5.6 Interactions entre composants métier

4.5.6.1 Principes

Chaque composant métier correspond ainsi à un ensemble de concepts ou de processus sémantiquement corrélés du domaine de la gestion des eaux. Les différents composants métier compartimentent ce dernier, en constituent des sous-domaines. Les différents composants métier ont de plus été catégorisés, cela afin d'en faciliter la réutilisation dans différents contextes. Ainsi, le partitionnement des différents aspects du domaine de la gestion des eaux en milieu urbain se fait d'une part en fonction du type de composant métier et d'autre part en fonction du sous-domaine considéré. Deux SIAGEM, destinés à des intervenants différents, seront en fait deux assemblages différents de composants métier en grande partie identiques, mais adaptés à des besoins spécifiques (figure 60).

Ces composants métier, à partir du moment où ils sont intégrés dans une même application logicielle, doivent être à même de collaborer, et donc de communiquer les uns avec les autres. De plus, quant à

l'autonomie des composants dont il a été question précédemment, elle n'est pas totale, en ce sens que certains composants métier s'appuient sur d'autres composants métier pour pouvoir fonctionner. Le présent chapitre traite de cette notion de relations entre composants.

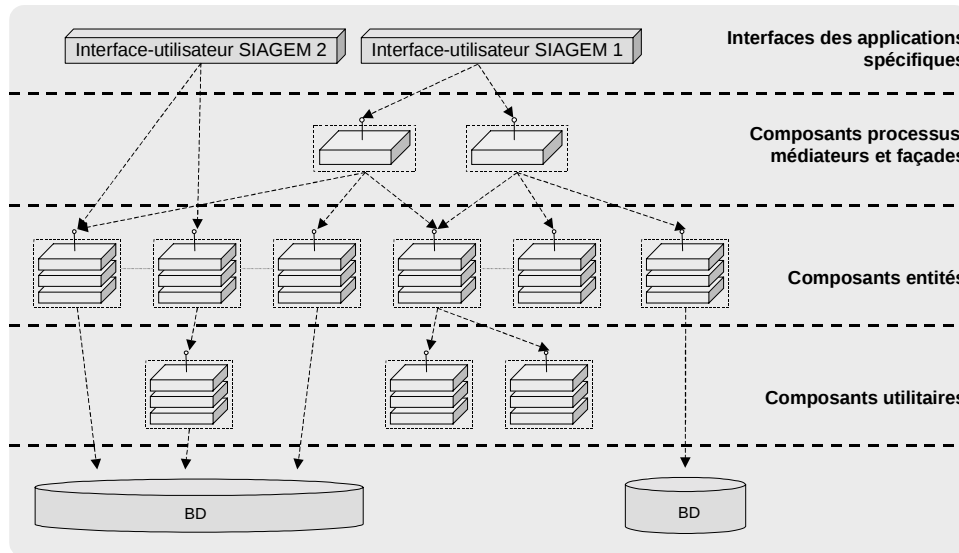


Figure 60. Deux SIAGEM construits selon une approche par composants métier réutilisables.

4.5.6.2 Dépendances fonctionnelles entre composants

On dit que le composant A dépend fonctionnellement (dépendance logicielle) du composant B lorsque A utilise un artefact logiciel de B (Herzum et Sims, 2000). Concrètement, une telle dépendance fonctionnelle existe entre composants lorsque A référence et utilise une interface proposée par B. Une relation d'un tel type a été présentée à la figure 37. Un composant métier de type processus par exemple n'est généralement pas strictement indépendant, mais dépend fonctionnellement d'autres composants pour pouvoir fonctionner (figure 61).

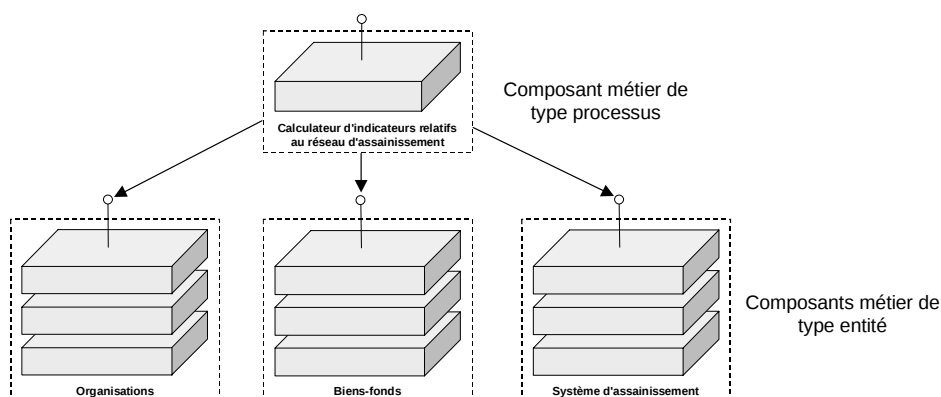


Figure 61. Un composant métier peut dépendre d'autres composants.

Afin de garantir la modularité des outils logiciels basés sur une approche par composants, il est impératif d'éviter toute dépendance circulaire, à savoir un composant A qui dépend d'un composant B, lequel dépend lui-même également de ce composant A : en effet une telle circularité impliquerait la nécessité

de simultanément mettre en oeuvre les composants A et B, qui devraient donc faire l'objet d'un développement parallèle et d'une distribution conjointe, et seraient en fait totalement couplés.

De manière générale, les composants-utilitaires sont complètement indépendants, alors que les composants-entité et les composants-processus peuvent dépendre fonctionnellement de composants utilitaires. Les composants-processus s'appuient sur des composants-entité. Les composants-façade possèdent une référence vers les composants qu'ils encapsulent, et les composants-médiateurs dépendent des composants dont ils assurent la médiation. Nous avons admis une hiérarchisation entre les différentes catégories de composants métier, et défini dans quel sens des dépendances fonctionnelles peuvent être établies (figure 60) : de telles règles permettent de diminuer le couplage entre composants métier, qui sont donc, au sein d'un SIAGEM, organisés en couches : les composants métier des niveaux hiérarchiques supérieurs s'appuient sur ceux de niveaux hiérarchiques inférieurs, et donc peuvent disposer de références fonctionnelles vers ces derniers, tandis que les composants métier des niveaux hiérarchiques inférieurs n'ont pas connaissance de ceux situés à des niveaux supérieurs. Ainsi, une telle approche permet d'envisager le développement des composants métier par étapes, en commençant par ceux des niveaux hiérarchiques inférieurs.

4.5.6.3 Communication par événements

Des dépendances fonctionnelles ne doivent ainsi être établies que dans un sens (chapitre 4.5.6.2), les composants métier des niveaux hiérarchiques inférieurs ne pouvant pas dépendre fonctionnellement de composants de niveaux hiérarchiques supérieurs. Pourtant, une communication doit pouvoir s'établir dans les deux sens pour une collaboration effective entre composants : ici, à nouveau doit être adoptée une approche par événements (la notion d'événement a déjà fait l'objet d'une description au chapitre 4.5.3.4) : les composants métier des niveaux inférieurs émettent des événements qui incluent l'information devant être communiquée, et ces événements sont capturés par les composants métier des niveaux hiérarchiques supérieurs qui sont intéressés par cette information. Les composants métier des niveaux hiérarchiques supérieurs "s'abonnent" aux événements des composants métier des niveaux hiérarchiques inférieurs. Une telle approche permet un couplage faible entre composants métier, car le composant de niveau inférieur, lorsqu'il émet des événements, ne doit pas connaître les composants métier qui y sont "abonnés".

4.5.6.4 Relations entre composants métier de même niveau hiérarchique

Les composants métier de type entité, ou tout au moins une partie d'entre eux, implémentent le modèle orienté objet du domaine de la gestion des eaux en milieu urbain. Ce sont eux qui, selon l'approche proposée, sont responsables de l'intégration des différentes visions du système "Eaux en milieu urbain" et qui permettent aux intervenants de manipuler des objets métier se rapportant au territoire et à ses constituantes (chapitre 4.4). Le modèle orienté objet du système "Eaux en milieu urbain" fait en effet l'objet d'une partition entre les différents composants-domaine des composants métier concernés, et le modèle de données relatif à ce même système fait l'objet d'une partition en "îlots de données" propres à ces mêmes composants métier.

Or nous avons vu que les relations entre les différentes composantes du système "Eaux en milieu urbain" sont extrêmement nombreuses. Des liaisons doivent subsister entre des objets métier qui n'appartiennent

pas à un même composant métier (figure 62). Comment dès lors limiter le degré de couplage entre composants métier de même niveau hiérarchique ? Les notions de "clé logique" (Schmid, 1999) et de "médiateur" peuvent être ici mises en oeuvre :

- au niveau de la BD, toute relation du modèle de données correspondant à une liaison entre des objets dont l'implémentation est le fait de composants métier différents doit être supprimée. Ainsi peuvent être formés ces "îlots de données" dont il a déjà été question (chapitre 4.5.3.3), et qu'il est possible d'administrer indépendamment les uns des autres, éventuellement au sein de BD différentes.
- au niveau des composants métier de type entité, on serait tenté d'établir des relations entre objets métier : par exemple (figure 62) une interface "Ouvrage de réseau d'assainissement" proposée par le composant métier "Système d'assainissement" devrait logiquement être associée à une interface "Organisation" - l'organisation propriétaire de l'ouvrage en question - proposée par un composant métier de même nom. Une liaison de ce type constitue toutefois une dépendance trop forte entre deux composants métier différents : le composant "Système d'assainissement" devrait pour y obéir posséder une référence fonctionnelle vers le composant "Organisation". Pour éviter une telle liaison, on remplacera donc, dans l'interface "Ouvrage d'exutoire", la référence vers un objet de type "Organisation" par un attribut référençant de manière univoque l'organisation en question grâce à une "clé logique", c'est-à-dire un identifiant. Le type de cette "clé logique" doit être commun (chaîne de caractère, nombre, etc.) et la dépendance fonctionnelle entre ces deux objets métier est ainsi éliminée.

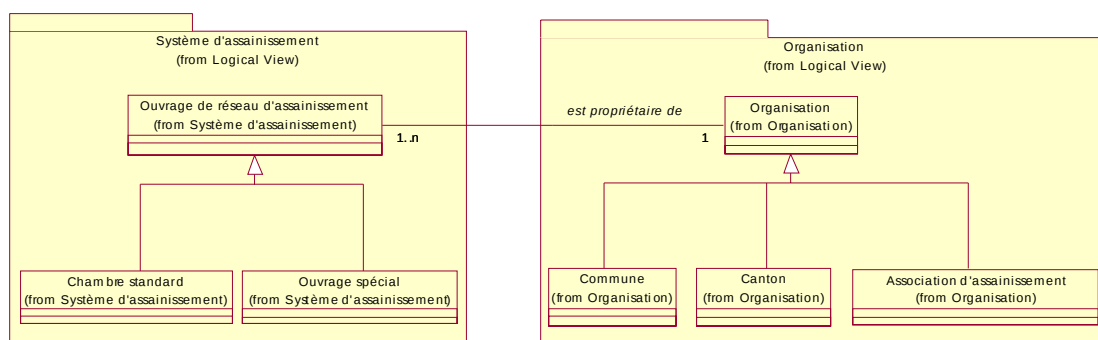


Figure 62. Regroupement des concepts en unités logiques. La liaison entre les classe-métier Organisation et Ouvrage de réseau d'assainissement doit être remplacée par un système de "clé logique" afin que soit assurée l'indépendance des composants.

Le soin de rétablir le lien entre deux objets métier instanciés à partir de composants métier différents est confié à un médiateur référençant les composants métier concernés. Dans un autre exemple, un composant-processus "Suivi de flux", apportant un soutien lors du déversement de substances dangereuses et à même de suivre le cheminement de ces dernières, peut être muni de références fonctionnelles vers deux composants métier : "Ouvrage de réseau d'assainissement" et "Eaux de surface". Une instance issue du composant "Suivi de flux" pourra obtenir d'un objet métier de type "Ouvrage d'exutoire", issu du composant "Ouvrage de réseau d'assainissement", une clé logique lui permettant ensuite d'obtenir du composant "Eaux de surface" l'instance de "Tronçon de rivière" identifiée par cette clé logique. Le lien entre ces deux objets implémentés par deux composants différents peut ainsi être établi, sans qu'il y ait relation correspondante dans la BD ni dépendances fonctionnelles entre les composants métier de type entité correspondants. Le composant processus joue le rôle de

médiateur (chapitre 4.5.4.2).

Ainsi, le respect de ces deux points permet d'assurer une indépendance entre composants métier de même niveau hiérarchique, ce qui améliore leur capacité d'être réutilisés.

4.6 Synthèse du chapitre et conclusion

Les réflexions menées nous ont conduits à proposer que les SIAGEM soient construits par assemblage d'entités logicielles indépendantes, préexistantes et réutilisables, les composants métier. De tels composants autorisent ensuite la manipulation, par le biais d'interfaces soigneusement définies, de concepts du domaine de la gestion des eaux. La structure de ces composants métier, ainsi que les mécanismes de communication tant entre les parties constituant ces composants métier qu'entre différents composants métier, ont été décrits. Enfin, une catégorisation des composants métier a été proposée.

Il a été montré que l'approche par composants métier permet de résoudre un problème crucial des SIRS actuels. Les modèles de données que ces SIRS mettent en oeuvre sont généralement soit beaucoup trop simples, soit beaucoup trop compliqués, et dans les deux cas difficiles à mettre en oeuvre :

- trop simples si la cohabitation de différentes visions du territoire au sein de la même structure de données a été obtenue par des compromis et des simplifications de cette réalité de manière à trouver le dénominateur commun entre les acteurs. Cette approche conduit à des structures de données qui finalement ne satisfont personne car elles sont trop pauvres.
- trop compliquées si la cohabitation de ces différentes visions du territoire est maintenue au sein de la BD, car toutes les spécificités des différents acteurs sont intégrées dans la même structure de données.

Avec une approche par composants, chaque concept sémantique est implémenté par un composant métier particulier. Les différentes "vues" sur un même concept peuvent être matérialisées par des interfaces différentes. Ainsi :

- chaque utilisateur manipule des interfaces spécifiquement adaptées à ses besoins, un même concept pouvant être manipulé par le biais de différentes interfaces;
- les utilisateurs savent où trouver les interfaces dont ils ont besoin; celles se rapportant à un même concept sémantique sont implémentées soit par un même composant métier, soit par un ensemble de composants métier liés par des dépendances;
- les utilisateurs n'accèdent pas directement aux BD, mais manipulent les données par le biais des interfaces proposées par les composants métier : la complexité des BD peut ainsi être gérée.

La construction de SIAGEM à partir de composants métier offre d'autres avantages. Les paragraphes qui suivent reprennent les exigences formulées au chapitre 3 et montrent comment, par une approche par composants, on peut arriver à construire des SIAGEM qui y répondent.

Dans un SIAGEM, comme dans tout SIRS, la gestion des données est assurée par une ou plusieurs BD. La valorisation des données est possible à partir des composants métier, en ce sens que ceux-ci intègrent les possibilités de générer de l'information spécifiquement adaptée aux différents intervenants. On peut

imaginer des composants métier qui, communiquant avec ceux mis en oeuvre au niveau d'un SIAGEM communal, génèrent de l'information utile à un niveau organisationnel supérieur (canton, association d'assainissement, Confédération). Le concept de composant métier apporte ainsi une solution au transfert d'informations entre différents niveaux organisationnels.

Les processus métier (chapitre 3.1.3) sont appuyés par des composants métier de type processus, qui peuvent jouer un rôle de médiateur entre des composants métier tiers. De tels composants doivent mettre à disposition des intégrateurs des fonctionnalités qui puissent être exploitées de manière à soutenir les utilisateurs finaux du SIAGEM de manière adéquate en regard de leurs besoins et habitudes.

En ce qui concerne la grande diversité d'utilisateurs (chapitre 3.2.1), il a été montré qu'un même concept peut être manipulé par le biais d'interfaces différentes, chacune étant adaptée aux besoins d'une catégorie d'utilisateurs particuliers. Un même objet métier peut ainsi se présenter sous des interfaces différentes.

Les organisations susceptibles d'utiliser des SIAGEM sont très diverses, et un SIAGEM doit être à même de s'adapter à différentes configurations (chapitre 3.2.2): la conception interne d'un composant métier, lui-même conçu comme un agrégat de composants communiquant entre eux par réseau informatique et pouvant être installés sur des machines différentes, offre une réponse à cette exigence.

Certains composants peuvent être conçus spécifiquement pour établir un couplage entre des SIAGEM et des logiciels particuliers (logiciel de simulation des comportements par exemple). Différents degrés de couplage peuvent être envisagés, allant de la génération de fichiers à partir des données gérées par le SIAGEM à une intégration totale. Dans ce dernier cas, le logiciel préexistant peut faire l'objet d'une encapsulation par un composant métier, lui permettant ainsi de communiquer avec les autres composants métier par le biais d'interfaces et donc de complètement s'intégrer au SIAGEM. Cette solution montre bien qu'une approche par composants n'implique pas le redéveloppement des outils existants, mais au contraire facilite leur exploitation (chapitre 3.2.3).

Nous avons également vu que le contexte de la gestion des eaux évolue à l'heure actuelle très rapidement (chapitre 3.2.4). Une approche par composants est parfaitement adaptée à un contexte dynamique de ce type : de nouveaux composants peuvent être mis sur le marché, de nouvelles interfaces ajoutées à des composants existants, et la structure de données peut être étendue ou modifiée de manière à répondre à de nouvelles exigences. La seule contrainte à ce niveau est qu'une interface, une fois publiée, ne doit plus être modifiée : cela implique que la définition des interfaces doit se faire de manière soignée, et ce d'autant plus que, du point de vue pratique, la prolifération d'interfaces aux sémantiques proches ne pourrait que nuire au développement d'un marché de composants métier.

Ces composants métier dont nous proposons la construction sont destinés à être réutilisés : certains d'entre eux peuvent proposer des éléments d'interfaces-utilisateur, mais cela n'est pas nécessaire. Le chapitre 3.2.5 a mis en avant la nécessité de pouvoir personnaliser les SIAGEM en fonction des besoins spécifiques des différents intervenants : ce rôle est généralement dévolu à l'intégrateur de composants métier, qui assemble ces derniers de manière à satisfaire aux besoins des utilisateurs finaux, et qui en particulier conçoit et développe des interfaces graphiques spécifiques. Des intervenants ont été rencontrés qui souhaitaient pouvoir eux-mêmes agir sur les outils logiciels qu'ils utilisent : là encore, les composants métier constituent une solution adéquate, car c'est au niveau de leur développement que les

plus grandes difficultés techniques sont rencontrées (en particulier en ce qui concerne les composants métier de type utilitaire et les composants-constituants de type ressources). L'assemblage de composants métier requiert certes des compétences en matière de développement informatique et une certaine compréhension de la notion de composants, mais ne revêt pas de difficulté particulière et peut être le fait des utilisateurs finaux eux-mêmes.

Enfin, les exigences de rapidité de développement et de coûts réduits sont des conséquences directes et bien connues d'une approche par composants : disposant de nombreuses "briques logicielles" prêtes à l'emploi, les intégrateurs, chargés de construire des SIAGEM à partir d'assemblages de composants, peuvent construire les outils requis beaucoup plus facilement et rapidement. Les coûts de développement des composants logiciels peuvent être amortis sur plusieurs SIAGEM, ce qui réduit ainsi leur coût global. Il faut être cependant conscient du fait que le coût de développement d'un composant logiciel est élevé, car un tel artefact, amené à être réutilisé, doit être l'objet d'une conception très pointilleuse, de nombreux tests et d'une documentation détaillée. On estime généralement qu'un composant métier doit être mis en oeuvre dans le cadre de trois à cinq projets au minimum pour que son développement se justifie.

Une approche par composants métier présente donc de nombreux avantages en comparaison des approches classiques. Le concept de composants existant à présent depuis plusieurs années, comment se fait-il qu'il ne soit pas plus répandu (à l'exception du concept de composant technique) ? Jacobson (1992) avance les raisons suivantes :

- les projets informatiques disposent généralement de budgets trop petits et sont planifiés avec des délais trop courts pour que soit saisie l'occasion de développer des composants réutilisables;
- les développeurs préfèrent généralement produire le code informatique eux-mêmes, d'une part pour en avoir la maîtrise, et d'autre part pour se sentir ainsi plus productifs;
- la distribution de composants soulève un certain nombre de problèmes;
- enfin, on ne disposait pas à l'époque de standards relatifs aux composants. Ce problème semble être résolu avec les technologies informatiques disponibles à l'heure actuelle (chapitre 4.3).

En raison de tous les avantages cités précédemment, une approche par composants métier doit être encouragée. Le domaine de la gestion des eaux constitue un domaine d'application idéal pour une telle démarche : tous les objectifs que nous avons fixés en ce qui concerne les SIAGEM peuvent être atteints, et les différentes contraintes respectées.

Dans ce chapitre, les différentes notions présentées l'ont été sans tenir compte de leur ordre d'apparition dans le contexte du processus de développement logiciel. Le chapitre qui suit s'attache donc à présenter la manière dont doivent être développés et documentés des composants métier spécifiques du domaine de la gestion des eaux en zones urbaines.

Proposition d'une méthode de développement

Le développement de composants métier destinés à la construction de SIAGEM nécessite une approche particulière : il s'agit de s'interroger sur la manière d'attribuer des fonctionnalités aux composants, de maximiser leur capacité à être réutilisés et de tenir compte du rôle intégrateur du territoire et de ses éléments. Après une introduction au génie logiciel et aux méthodes de développement actuelles, ce chapitre décrit de manière détaillée les étapes de la méthode COWIS que nous proposons.

5.1 Motivations

Une méthode de développement logiciel "*regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel*" (Jacobson et al., 2000). Elle décrit la démarche permettant d'organiser, de créer, de livrer des systèmes logiciels et d'en assurer la maintenance. De nombreuses méthodes de développement, relatives tant aux SI de gestion "classiques" (par exemple Coleman et al., 1994; Gane et Sarson, 1979; Jacobson et al., 1992; Rumbaugh, 1994) qu'aux SIRS (Gayte et al., 1996; Pantazis et Donnay, 1996; Prélaz-Droux, 1995a) sont disponibles à l'heure actuelle. Pourquoi alors consacrer un chapitre de ce travail à la proposition d'une méthode supplémentaire ?

Avant de répondre à cette question, une précision est nécessaire : nous proposons d'adopter une approche par composant dans le contexte de la gestion des eaux, et plus précisément de concevoir et de développer des composants métier réutilisables qui peuvent ensuite servir de base à la construction de SIAGEM. Deux processus de développement doivent ici être distingués (figure 63) : le processus de développement des composants métier eux-mêmes, et le processus de développement d'applications logicielles, les SIAGEM, à partir de ces composants. Bien que cette dernière problématique soit très importante¹, ce n'est pas à elle que nous nous intéressons ici. Nous nous attachons uniquement au processus de développement de composants métier réutilisables : notre objectif est de faciliter la fabrication et la mise à disposition de tels composants, qui doivent pouvoir ensuite être assemblés par des tiers pour former des applications aux

1. Vu (1997) constate que les modèles usuels du génie logiciel ne sont pas adaptés à la conception de logiciels basés sur une approche par composants car ils négligent les processus et les coûts associés à l'identification, l'évaluation, la sélection et l'intégration des composants logiciels dans les applications.

profils divers.

Ayant ainsi défini notre objectif avec précision, nous pouvons répondre à la question qui a été posée au début de ce chapitre. Il est prouvé qu'une méthode doit impérativement être mise en oeuvre dès le moment où le système logiciel considéré atteint une certaine envergure (chapitre 5.2), ce qui est le cas des SIAGEM. Cependant seul quelques écrits fournissent des éléments de méthodes qui sont totalement axée sur le développement d'objets et de composants métier, et ce dans le contexte de systèmes manipulant de l'information "traditionnelle", soit numériques ou textuelles (Eeles et Sims, 1998; Herzum et Sims, 2000; Schmid, 1999). Les SIAGEM, dans le contexte desquels de grandes quantités d'information à référence spatiale sont gérées, nécessitent, comme les SIRS, le recours à des méthodes spécifiques. À notre connaissance, aucune méthode n'a été jusqu'à présent publiée décrivant la manière de concevoir des composants métier dans un tel contexte. Aussi proposons-nous notre méthode, baptisée COWIS (COmponents for Water Information Systems). Notons que COWIS peut être utilisée dans l'autres cadres que celui de la gestion des eaux, pour autant que de l'information relative au territoire soit manipulée. Le chapitre 5.2 introduit différentes notions se rapportant au développement logiciel tandis que le chapitre 5.3 présente COWIS de manière détaillée.

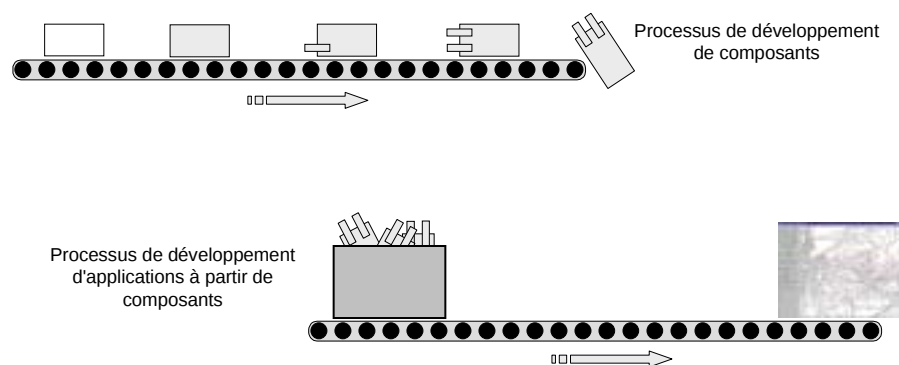


Figure 63. Deux processus de développement : l'un pour les composants logiciels, l'autre pour les applications intégrant des composants logiciels. Nous nous intéressons ici au premier d'entre eux.

5.2 Méthodes de conception de systèmes d'information

5.2.1 Introduction

L'industrie du logiciel a réalisé, vers la fin des années 60, qu'elle butait contre un certain nombre de problèmes¹, à savoir (Coleman et al., 1994; Strohmeier, 1996):

- les projets informatiques n'étaient que rarement achevés dans les délais impartis en raison de la difficulté que représentait l'estimation de la durée de développement des applications logicielles.
- les budgets alloués étaient fréquemment dépassés. Ce problème a perduré, puisque Lim (1998), citant une étude datée de 1995, rapporte que 16% seulement des projets logiciels étaient menés à temps, de manière complète et selon les budgets prévus.

1. De cette prise de conscience date l'expression "crise du software", ou "software crisis" (Schach, 1996).

- les logiciels développés étaient souvent de piètre qualité: ils ne correspondaient pas toujours aux attentes des commanditaires, et ils contenaient fréquemment des défauts conduisant à des dysfonctionnement des systèmes informatiques ainsi qu'à la perte ou à la corruption de données.
- les coûts de maintenance, liés tant à la correction d'erreurs qu'à l'ajout et la modification de fonctionnalités, étaient extrêmement élevés (figure 64).
- les projets de développement ne tenaient pas compte de la possibilité de réutiliser des éléments (portions de code, modules logiciels,...) issus de projets précédents, les applications logicielles étaient à chaque fois développées *ex nihilo*.

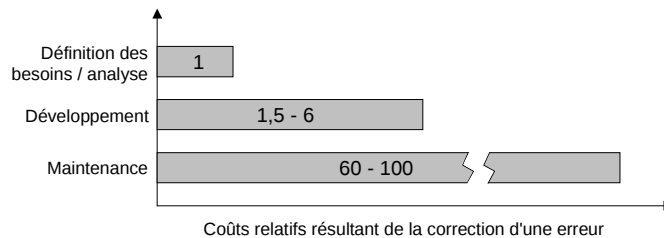


Figure 64. Coûts relatifs résultant de la correction d'une erreur suivant l'étape du cycle de vie à laquelle celle-ci est décelée (à partir de Pressman, 1997). L'intérêt d'éviter les erreurs, et donc de recourir à des méthodes de développement, est évident.

Ces différents problèmes, qui perdurent, ont été mis sur le compte d'une part d'un manque de rigueur dans la démarche de développement d'applications logicielles, et d'autre part de la complexité même qui caractérise de telles applications : selon Brooks (1996), les *"entités logicielles sont sans doute les constructions les plus complexes réalisées par l'homme, car elles n'ont pas deux parties identiques"*. De la prise de conscience des problèmes cités plus haut est issu le génie logiciel¹.

Les bénéfices de l'adoption d'une approche de type génie logiciel sont nombreux. On citera entre autres :

- une augmentation de la qualité des applications logicielles, qui recèlent moins d'erreurs et dont la maintenance est plus facile et plus efficace.
- une diminution des temps et des coûts de développement. Comme les besoins des utilisateurs évoluent vite à l'heure actuelle, les applications doivent être développées et mises sur le marché très rapidement (Jacobson et al., 2000).
- une gestion facilitée des budgets et des délais.
- une organisation plus simple du développement des applications informatiques impliquant des équipes importantes.

On distingue dans le domaine du génie logiciel le processus de développement, qui caractérise la manière dont les étapes du développement logiciel accompagnent les étapes du cycle de vie de l'application, de la méthode de développement, qui exprime de manière détaillée la manière dont les étapes du processus de développement sont abordées. Le chapitre qui suit aborde la problématique du processus lui-même, tandis que le chapitre 5.2.3 présente brièvement les différentes catégories de méthodes.

1. "[Software engineering] is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines." (Pressman, 1997)

5.2.2 Les processus de développement

Tout système d'information (SI) passe, au cours de son existence, par un certain nombre de phases qui ensemble constituent ce que l'on appelle "cycle de vie" du SI. Ce cycle de vie débute au moment où une organisation prend la décision d'examiner la possibilité d'implanter un SI et se termine lorsque cette même organisation décide que son SI est devenu obsolète et doit être remplacé : le SI est alors démantelé (Pantazis et Donnay, 1996). Le cycle, ou processus de développement, couvre l'ensemble des étapes du cycle de vie et constitue le cadre nécessaire au développement des applications logicielles. Pressman en apporte la définition suivante :

"Software process : framework for the tasks that are required to build high quality software." (Pressman, 1997)

On retrouve en général les mêmes étapes dans les différents processus de développement - analyse, conception, implémentation, test et éventuellement phase d'installation - qui sont combinées de différentes manières (Strohmeier, 1996). À noter qu'un processus de développement particulier ne peut pas être qualifié de faux ou de juste, puisque les différents processus sont plus ou moins bien adaptés à des problématiques particulières. Ces étapes sont généralement les suivantes (figure 65) :

- phase de définition : les informations devant être produites, les fonctions et performances désirées, les comportements souhaités, les contraintes à respecter sont les points à considérer. On s'attache ici à définir ce que le logiciel doit pouvoir faire.
- phase de développement : la manière de structurer les données, d'implémenter les fonctions, de concevoir l'architecture logicielle, de transférer les concepts développés dans un langage de programmation, et de conduire les tests est définie à cette étape. On s'attache donc à déterminer la manière d'implémenter les décisions prises durant la phase de définition, et l'on procède à cette implémentation.

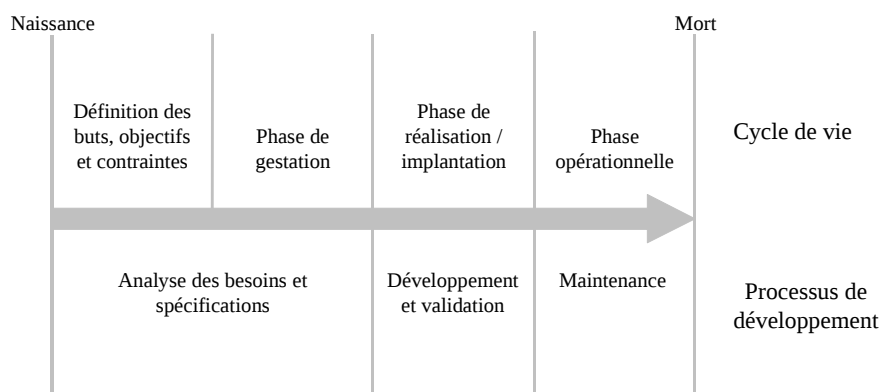


Figure 65. Aux différentes étapes du cycle de vie correspondent des phases du processus de développement d'un système d'information (adapté de Pantazis et Donnay, 1996)

- phase de maintenance : elle correspond à la phase d'exploitation de l'application logicielle. L'application subit durant cette phase des modifications qui peuvent être des corrections destinées à supprimer des erreurs, des adaptations aux évolutions de l'environnement (matériel informatique, règles de gestion,...), des améliorations répondant à des demandes de fonctionnalités supplémentaires réclamées par les utilisateurs, ou des modifications préventives, qui consistent en

des améliorations du logiciel dans le but de faciliter sa maintenance ou de prévenir d'éventuels problèmes. La fin de la maintenance est marquée par la mise hors service de l'application logicielle.

Différents auteurs ont discuté les nombreuses variantes des modèles de cycle de vie servant de cadre général à une méthode (Bouzeghoub et al., 1997; Pantazis et Donnay, 1996; Pressman, 1997; Schach, 1996; Sommerville, 1995). L'objectif des lignes qui suivent est d'examiner l'adéquation de ces différents modèles avec la problématique qui nous intéresse. L'annexe 5 présente brièvement ces différents modèles. Le cycle de développement adopté sera présenté au chapitre 5.3.

Le **modèle en cascade** est caractérisé par une succession linéaire d'étapes bien définies et formalisées : chaque étape (analyse des besoins, spécifications, planification, conception, implémentation, test) est vérifiée et documentée, l'étape suivante ne démarrant pas tant que la précédente n'est pas terminée. Dans le contexte d'applications traditionnelles, le modèle en cascade présente un certain nombre d'inconvénients. Cependant, dans celui des composants réutilisables, ce modèle présente deux caractéristiques très intéressantes : d'une part un déroulement linéaire tel que celui proposé permet de délivrer, en fin de cycle, un ensemble d'interfaces sémantiquement corrélées, qui sont à ce moment publiées et ne peuvent de ce fait plus être modifiées; d'autre part, les composants métier, destinés à être réutilisés de nombreuses fois, dont les implémentations doivent être très soigneuses et peuvent faire l'objet d'améliorations ultérieures, doivent être solidement documentés, ce que favorise le modèle en cascade.

Le but principal d'une approche par **prototypage** est la mise à disposition de l'utilisateur d'un prototype, de manière à ce que ledit utilisateur puisse estimer si l'application projetée correspond à ses besoins. Le prototype constitue un moyen de définir des besoins et mettre en place des spécifications. Nous nous intéressons dans le contexte de cette recherche au développement de composants logiciels qui doivent, pour pouvoir être mis en oeuvre, être intégrés dans un SIAGEM. La fourniture de ceux-ci à des concepteurs de SIAGEM ne présente guère d'intérêt, dans la mesure où chaque application n'exploite qu'une partie des fonctionnalités offertes par le composant-métier et que le rôle de l'intégrateur est la création d'applications opérationnelles, et non l'examen et le test de l'ensemble des fonctionnalités des composants développés. Une approche par prototypage n'est pas adaptée à notre problématique.

Le **modèle incrémental** présente une caractéristique extrêmement intéressante dans le contexte de notre projet : il favorise des cycles de développement courts et une construction par étapes des SI. La difficulté qu'il y a à cerner les besoins des utilisateurs, et donc à définir les fonctionnalités et caractéristiques des futurs composants métier, a été discutée. Un modèle incrémental peut autoriser l'identification par étapes des fonctionnalités à implémenter dans des composants, et donc faire évoluer ceux-ci par étape. Il ne faut toutefois pas oublier que les composants métier implémentent des interfaces qui, une fois publiées, ne doivent plus faire l'objet de modifications (chapitre 4.3.3). Ainsi, pour que ce modèle soit applicable à la conception de composants, chacun de ses cycles doit s'attacher au développement d'un ensemble cohérent d'interfaces.

Le **modèle en spirale** combine une approche évolutionnaire à un processus en cascade. Les différentes phases en sont précédées d'une analyse de risque. De par son approche extrêmement rigoureuse, ce modèle constitue une démarche intéressante, mais reste lourd à mettre en oeuvre et s'adresse donc en priorité à des projets dotés de moyens importants.

5.2.3 Les méthodes de développement

Une méthode de développement exprime de façon détaillée la manière dont les différentes étapes du cycle de développement du système d'information sont abordées. Dans ce chapitre, nous proposons une rapide description des groupes de méthodes, en fonction de la philosophie fondant la perception de l'application logicielle (Bouzeghoub et al., 1997).

Les méthodes cartésiennes, dont l'origine remonte aux années 70, se basent sur une décomposition hiérarchique des fonctions du système, et ce jusqu'à ce que ces dernières soient suffisamment simples à implémenter. Les problèmes principaux qui en résultent sont d'une part qu'elles conduisent à des SI difficiles à maintenir, et d'autre part que, étant basées sur des fonctions qui sont peu stables puisque les besoins des utilisateurs évoluent, elles conduisent à des applications logicielles qui sont en perpétuelle reconception.

Les méthodes systémiques sont basées sur une vision systémique des SI (chapitre 2.4.1.2). *"La modélisation du système est abordée selon deux points de vue complémentaires, la modélisation des données et la modélisation des traitements"* (Bouzeghoub et al., 1997). Citons comme exemple d'une telle méthode Merise (Collongues et al., 1989; Matheron, 1991; Tabourier, 1986). Les points forts de méthodes comme celle-ci sont 1°) une grande cohérence des données, une importance considérable étant accordée à la conception de modèles de données, et 2°) la capacité de décrire des systèmes complexes. Par contre, les modèles de traitement et de données, développés suivant des démarches différentes, sont difficiles à coordonner (figure 66).

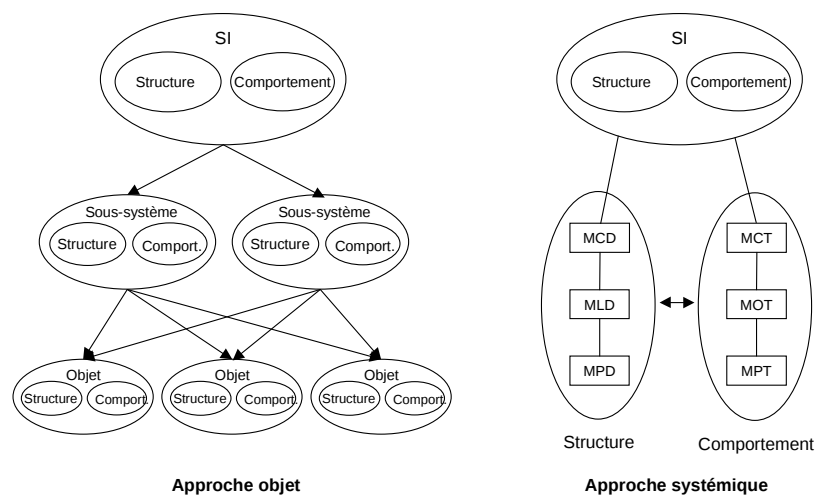


Figure 66. Les démarches systémique et objet (adapté de Bouzeghoub et al., 1997). MCD, MLD et MPD sont les modèles conceptuel, physique et logique de données, MCT, MOT et MPT sont les modèles conceptuel, organisationnel et physique de traitement.

"On peut considérer l'approche objet comme une évolution de l'approche systémique vers une plus grande cohérence entre les objets et leur dynamique" (Bouzeghoub et al., 1997) : une grande partie de la dynamique du SI est directement associée aux objets, par leurs opérations (figure 66). Les différents avantages de l'approche objet ont été discutés au chapitre 4.2, et il a été opté pour ce paradigme dans le cadre de ce travail. Nous nous appuyerons donc, pour élaborer nos propositions, sur des méthodes de développement orientées objet existantes : citons ici à titre d'exemples les méthodes FUSION (Coleman

et al., 1994), Objectory (Jacobson et al., 1992) et OMT (Rumbaugh, 1994).

En ce qui concerne le développement de SIRS, ont été rencontrées des méthodes reposant sur les paradigmes systémique (Golay, 1992; Pantazis et Donnay, 1996; Prélaz-Droux, 1995a) et orienté objet (Gayte et al., 1996). Pour ce qui est des méthodes systémiques, le point central en est la conception de structures de données communes aux différents utilisateurs et à même de servir de plate-forme d'échange entre ces derniers. L'accent est mis dans ces méthodes sur les aspects statiques (modélisation des données), les aspects dynamiques étant, comme le remarque Pouliot (1999), laissés en retrait.

5.3 Description de la méthode COWIS

5.3.1 Processus de développement

Le cycle de vie de composants métier est long comparé à celui d'applications traditionnelles : ces composants continuent en effet, après leur première mise sur le marché, à faire l'objet d'améliorations successives, et constituent ainsi des entités logicielles plus stables que les applications qui les intègrent (figure 67). Il faut en tenir compte, et le cycle de développement de ces composants métier doit répondre aux critères suivants :

- les besoins des intervenants du domaine de la gestion des eaux doivent faire l'objet d'une étude approfondie en début de processus. Ces besoins doivent être "capturés" de manière précise et détaillée, car ils ne peuvent être validés par les intervenants en raison de l'aspect "immatériel" et particulièrement abstrait des composants.
- le processus de développement doit être extrêmement soigné, accompagné d'une documentation précise, qui appuie la mise au point de composants logiciels de qualité et en facilite la maintenance. En effet, d'une part les fonctionnalités des composants sont accessibles par l'intermédiaire d'interfaces, qui doivent être définies très précisément, et d'autre part il est généralement nécessaire de modifier et améliorer des composants plusieurs mois après leur mise sur le marché, ce qui nécessite que leurs aspects internes aient été bien documentés.
- le processus de développement doit être itératif, conduisant à la livraison successive de composants métier couvrant chacun un ensemble restreint des besoins. Ces itérations doivent être à la fois suffisamment courtes pour répondre aux besoins des intégrateurs de composants dans des délais raisonnables, et suffisamment longues pour tout de même intégrer dans chaque composant un ensemble conséquent de fonctionnalités. On peut estimer que leur durée ne devrait pas dépasser six mois.

Il ressort des exigences formulées ci-dessus que les avantages apportés par deux types de processus de développement, le modèle en cascade et le modèle incrémental (chapitre 5.2.2), doivent être combinés. En effet :

- chaque composant métier propose un ensemble d'interfaces sémantiquement corrélées. Les fonctionnalités décrites par chaque interface doivent faire l'objet d'une réflexion approfondie, ces dernières ne devant plus être modifiées après publication. Ainsi, il ne peut y avoir d'itération au niveau de la définition d'une interface après sa mise à disposition (l'implémentation d'une interface

peut faire l'objet d'améliorations et de modifications après publication de l'interface elle-même). Ainsi, un processus en cascade doit être mis en oeuvre pour le développement d'ensembles cohérents d'interfaces. La rigueur de ce modèle de développement ainsi que l'abondante documentation produite contribuent à la conception de composants de qualité. Le premier reproche qui est généralement fait à ce modèle est la longue durée de développement qu'il induit : dans la mesure où il n'est ici appliqué à chaque fois qu'à un sous-ensemble bien délimité, sa durée devient tout à fait acceptable. Le second reproche à lui être adressé, à savoir de ne permettre la validation du développement effectué qu'à la fin du processus, n'est plus non plus de mise ici puisque les cycles de développement sont courts.

- nous avons vu que les composants métier doivent être développés et mis à disposition des utilisateurs rapidement. C'est un processus de développement itératif qui doit être mis en oeuvre ici, chacune des itérations de ce processus permettant le développement d'au minimum un composant métier : il est donc possible d'appliquer un processus incrémental permettant aux concepteurs de délivrer régulièrement des composants métier. Ces derniers ne proposent certes qu'un nombre restreint de services, mais sont parfaitement opérationnels. Avant chacune de ces itérations, une réflexion doit être menée quant à son bien-fondé. En particulier l'utilité et le potentiel de réutilisation des composants métier concernés doivent être évalués.
- on peut considérer qu'un deuxième niveau d'itération existe. En effet, le modèle en cascade admet, lorsque c'est nécessaire, que l'on revienne en arrière, la contrainte étant alors que la documentation de l'étape sur laquelle on revient soit remise à jour. Des itérations sont donc possibles dans le contexte du développement d'un composant métier.

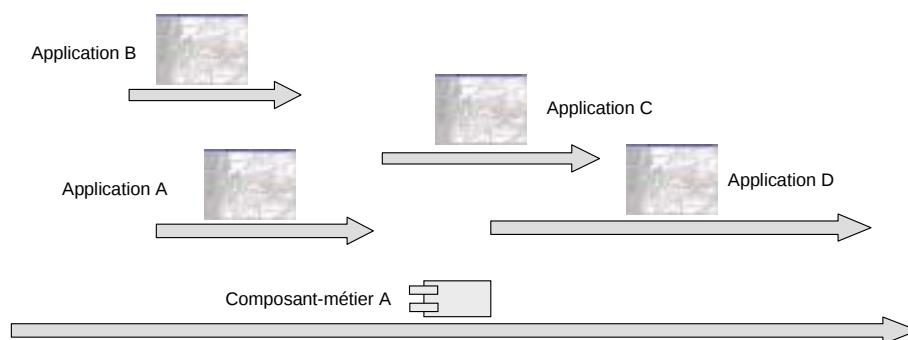


Figure 67. Le cycle de vie des composants logiciels est plus long que celui des applications qui les mettent en oeuvre. De nombreuses applications peuvent être basées sur les mêmes composants logiciels.

Le cycle de développement que nous proposons (figure 68) suit donc globalement un modèle incrémental, dont les itérations à plus petite échelle suivent un modèle en cascade. Chacun des processus en cascade permet le développement d'un ou plusieurs composants métiers. Des composants métier peuvent être développés soit séquentiellement, soit en parallèle par des équipes de développement différentes. Les étapes du processus de développement sont les suivantes :

- **Phase de spécification générale des composants métier (1).** Nous avons vu que les composants métier peuvent être structurés de différentes manières (chapitre 4.5.3) et que différents environnements informatiques permettent l'interopérabilité entre composants (chapitre 4.3.1). Une réflexion d'ordre stratégique, débouchant sur des décisions quant aux technologies et à l'architecture informatique sur laquelle s'appuyent les futurs composants métier, est menée. Cette étape ne fait

pas partie du processus itératif : elle constitue une réflexion préliminaire qui n'est menée qu'une seule fois.

- **Phase de construction de l'infrastructure pour composants métier (2).** Se basant sur les décisions prises lors de l'étape précédente, les composants techniques constituant l'infrastructure peuvent être construits (ou acquis et adaptés). Cette étape ne fait elle non plus pas partie du processus itératif.

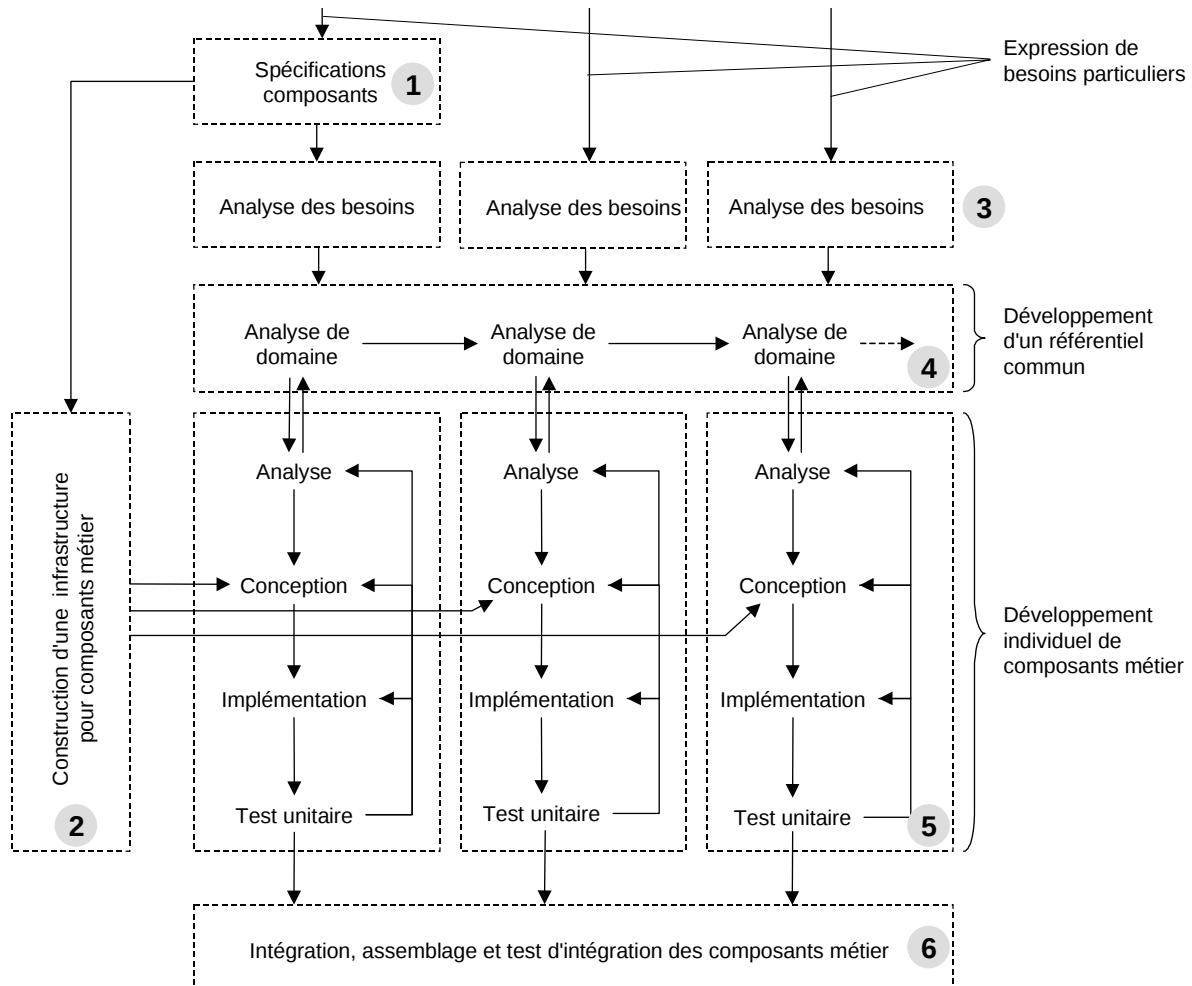


Figure 68. Processus de développement de composants métier.

- **Phase d'analyse des besoins (3).** Pour de raisons de rationalité économique, la décision de concevoir les composants métier ne peut être prise qu'à partir du moment où leur emploi répond à un besoin. Tout cycle de développement de composants métier débute donc par une phase d'analyse des besoins.
- **Phase d'analyse de domaine (4).** Elle a pour objectif l'élaboration d'un modèle commun à tous les composants métier, qui sert de référence et est affiné et complété au fil des itérations successives.
- **Phases de développement de composants métier (5).** Chaque développement de composant métier constitue un "mini-processus" en cascade, et est constitué des quatre étapes classiques - analyse, conception, implémentation, test - adaptées de manière à tenir compte des spécificités relatives à l'approche par composants métier. Les spécificités de l'infrastructure (2) sont prises en compte à partir des étapes de conception de composants métier.

- **Phase d'intégration (6).** Chaque itération conduit à la livraison de un ou plusieurs composants métier qui peuvent soit servir de base à la construction de nouveaux SIAGEM, soit être intégrés à des SIAGEM existants. Nous sommes à cette étape sorti du cycle de développement de composants métier proprement dit. Notons que l'intégration des composants métier doit être suivie d'un test d'intégration, destiné à vérifier que les composants interagissent conformément aux attentes.

Il est à noter que les phases proposées réclament des compétences diverses. Ainsi, les phases d'analyse des besoins et d'analyse de domaine nécessite des connaissances générales de la gestion des eaux et de la modélisation orientée objet. La phase d'analyse demande des connaissances avancées en modélisation orientée objet et en ingénierie des processus, ainsi que des domaines de la gestion des eaux abordés, et nécessitent la collaboration d'informaticiens et de spécialistes du domaine considéré. Les phases de conception, d'implémentation et de test unitaire réclament des connaissances en conception logicielle et en programmation qui ne sont pas nécessairement très pointues, car l'essentiel des difficultés d'ordre technique sont résolues au niveau des étapes de spécification et de développement de l'infrastructure pour composants métier, qui, elles, nécessitent des connaissances en informatique très spécifiques. Enfin, la mise en oeuvre des composants métier développés (intégration dans des applications logicielles) doit être relativement aisée et pouvoir être l'oeuvre de personnes issues du domaine de la gestion des eaux disposant de bases en informatique (des compétences relativement à la mise en oeuvre de composants logiciels sont nécessaires). Ainsi, la production de composants métier nécessite la collaboration d'intervenants aux compétences variées et issus tant des secteurs de l'informatique que de la gestion des eaux.

5.3.2 Méthode de développement

5.3.2.1 Introduction

Nous avons vu que la mise en oeuvre d'une méthode de développement est indispensable pour le développement de composants métier réutilisables. La méthode COWIS que nous proposons possède un certain nombre de particularités :

- l'un de ses objectifs sous-jacents est de contribuer à la formation d'une bibliothèque de composants métier réutilisables dans le domaine de la gestion des eaux. Il est nécessaire, lors de la conception des différents composants, d'élargir la réflexion menée afin de maximiser leur potentiel de réutilisation. De plus, la manière de répartir les concepts entre différents composants doit faire l'objet d'une attention particulière.
- la mise à disposition de tiers de composants réutilisables nécessite la production de documents très soignés. Cela s'applique aussi bien à ceux qui sont générés durant le processus de développement qu'à ceux mis à disposition des intégrateurs¹.
- COWIS a pour ambition de permettre la création de composants métier faciles à utiliser. Ainsi, dès les premières étapes, les concepts importants pour les spécialistes du domaine considéré sont identifiés. Ces concepts peuvent être tracés tout au long du processus de développement (figure 69). En fin de processus, les composants métier proposent leurs services par le biais d'interfaces qui se

1. Nous appelons "intégrateurs" les personnes qui assemblent des composants métier pour former des SIAGEM et adaptent ces assemblages aux besoins spécifiques d'intervenants particuliers.

rapportent à ces concepts, sont intuitives et faciles à mettre en oeuvre.

- COWIS est une méthode qui est à la fois itérative et séquentielle : itérative parce que des composants réutilisables sont construits les uns après les autres, ce qui permet de rapidement disposer de composants métier opérationnels, motive les développeurs qui travaillent plus rapidement et plus efficacement car ils disposent d'objectifs à plus court terme, et diminue les risques liés au développement de logiciels mal adaptés, ce risque étant limité à une itération (Jacobson et al., 2000). La méthode est séquentielle car une fois les interfaces externes de composants métier publiées, il n'est plus question de les modifier. Cela implique une réflexion très poussée concernant ces interfaces. Rappelons que les implémentations de ces interfaces peuvent être modifiées et améliorées, et que les interfaces qui ne répondent plus aux besoins doivent être remplacées par de nouvelles interfaces, et non modifiées (chapitre 4.3.3). De tels mécanismes permettent aux composants d'évoluer tout en garantissant leur interopérabilité.
- enfin, COWIS combine une approche par composants métier avec l'utilisation du territoire et de ses éléments dans un but d'intégration des métiers des différents intervenants. C'est pour cette raison que sont proposés des techniques, une organisation et un ordre de développement particuliers des composants.

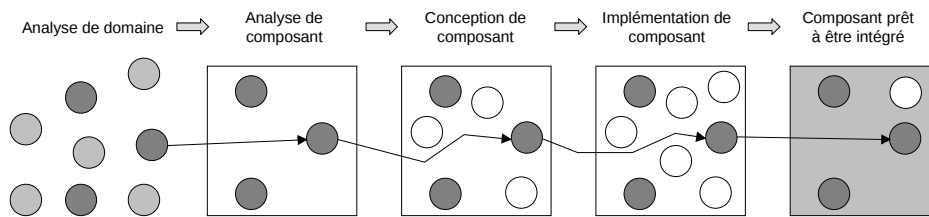


Figure 69. Traçabilité des concepts développés. Un concept du monde réel devient une interface permettant la manipulation d'objets métier et implémentée par un composant métier. Le processus de développement ne dénature pas le concept qui est issu d'une analyse du monde réel.

COWIS se base bien évidemment sur des méthodes existantes, en reprend de nombreux éléments, et en adapte d'autres aux spécificités de la situation. Ont été plus particulièrement considérées les méthodes FUSION (Coleman, 1994) et Objectory (Jacobson et al., 1992), la première en raison de la grande cohérence des modèles proposés, la seconde en ce qui concerne la mise en oeuvre de "cas d'utilisation" (annexe 6). Les propositions de Herzum, Eeles et Sims (Eeles et Sims, 1998; Herzum et Sims, 2000), plus en rapport avec le contexte des objets métier, ont également été prises en compte.

Comme cela a été dit, le but de la démarche est, à terme, de disposer d'une collection de composants métier s'attachant aux différents aspects du domaine de la gestion des eaux, à même de permettre la construction rapide et efficace de SIAGEM robustes, efficaces et évolutifs. Cependant, les composants métier ne peuvent être développés spontanément, le processus doit être initié d'une manière ou d'une autre. Ils doivent être développés dans le cadre de la réalisation d'applications logicielles répondant à une demande précise : il s'agit d'adopter une approche orientée composant, conduisant à la production de composants métier, qui non seulement peuvent être utilisés dans le cadre de l'applications logicielle susmentionnée mais de plus pourront être réutilisés lors de développements ultérieurs.

Les chapitres qui suivent présentent les différentes étapes de COWIS. Pour chacune d'entre elles, l'on montrera ses objectifs, les moyens et techniques à mettre en oeuvre pour les atteindre, et enfin certains

des artefacts générés. Précisons que le chapitre 6 présente un cas d'application de COWIS, et contient donc des exemples de documents générés par la méthode, complémentaires de ceux proposés dans le présent chapitre.

5.3.2.2 Spécifications des caractéristiques des composants métier

Cette phase n'est menée qu'une fois, avant que ne commencent les cycles de développement de composants métier : elle a pour but de déterminer les caractéristiques communes à l'ensemble des composants métier et le contexte dans lequel ils seront mis en oeuvre. Nous avons en effet vu que des composants sont conçus pour interagir au sein d'un environnement pour composants (chapitre 4.3.1) : il s'agit à présent de choisir l'un de ces environnements. Nous avons également vu que des composants métier peuvent être conçus pour être distribués sur différentes machines, le chapitre 4.5.3 en ayant présenté la structure interne et l'extension maximale, mais aussi que certaines couches de composants métier peuvent être regroupées de manière à construire des composants à la structure moins compliquée : les conclusions issues de la présente étape conduisent à choisir la façon de structurer les composants métier.

Le développement de composants métier se base sur un besoin concret, exprimé par un intervenant particulier. Il est nécessaire cependant d'élargir le cadre de la réflexion, et de se demander dans quel type de SIAGEM les composants prévus pourraient être intégrés. Ainsi, il faut préciser les exigences non fonctionnelles¹, de même que les contraintes d'ordre général. Différents points doivent être abordés (adaptés de Robertson et Robertson, 1999) :

- quels sont les intervenants qui sont susceptibles d'utiliser des SIAGEM et dans quel contexte (administration, bureau d'ingénieurs, etc.) ?
- des interfaces utilisateurs doivent-elles faire partie des composants métier, ou faut-il laisser cette responsabilité aux intégrateurs ?
- jusqu'à quel degré faut-il rendre les composants métier intuitifs et simples d'emploi, quels seront exactement les intégrateurs ?
- quelles doivent être les performances et les caractéristiques des composants métier suivant les environnements organisationnels dans lesquels ceux-ci sont mis en oeuvre (nombre d'utilisateurs, organisations centralisées, décentralisées, etc.) ?
- quelles doivent être les mesures de sécurité (contrôle différencié des utilisateurs, enregistrement de méta-données, etc.) ?
- quelles contraintes d'ordre légal (gestion de l'accès à l'information, protection des données) ou commercial (vente d'informations) doivent être prises en considération ?
- quels sont les risques auxquels l'entreprise projetée risque de se heurter : nouvelles technologies, produits concurrents, manque d'intérêt de la part des utilisateurs potentiels ?
- quels sont les moyens financiers disponibles pour le développement des différents composants projetés ?
- en combien de temps chaque jeu de composants-métier devra-t-il être finalisé ?
- sous quelle forme doit se présenter la documentation s'y rapportant ?

Pour pouvoir mener à bien cette réflexion, il est nécessaire de sélectionner quelques acteurs² (annexe 6)

1. Soit les propriétés, ou qualités, que les composants doivent avoir.

utilisateurs potentiels de SIAGEM et d'examiner leur mode de travail ainsi que leur environnement professionnel. Les réponses aux questions ci-dessus doivent être délivrées sous forme d'une documentation textuelle, compréhensible à la fois pour les informaticiens et les acteurs du domaine de la gestion des eaux. Cette documentation appuie le choix d'un environnement informatique, guide la mise en place d'une infrastructure pour composants métier (chapitre 5.3.2.3), et influence grandement le développement de ces derniers.

5.3.2.3 Mise en place d'une infrastructure pour composants métier

L'objectif de la présente phase est de créer l'infrastructure technique qui facilitera la mise sur pied ultérieure des composants métier. Cette infrastructure est constituée de composants logiciels qui pour une partie seront intégrés aux composants métiers eux-mêmes, et pour une autre partie leur offriront des services. L'une des problématiques les plus importantes ici est celle de la gestion de la persistance des objets métier, ce qui peut techniquement être très complexe. Il s'agit de faire en sorte que le développeur de composants métier disposant de cette infrastructure dite de persistance n'ait plus à se soucier de la technologie relative à la manière d'assurer la pérennité de l'information, mais puisse se consacrer à la logique-métier qu'implémentent les composants (figure 70). Les intégrateurs de composants métier doivent pouvoir gérer les aspects liés à la persistance de manière très simple : chaque objet-métier dit "persistant"¹ doit disposer d'une interface intuitive offrant les services nécessaires à la gestion de sa persistance (figure 71). Par exemple, il doit être suffisant, pour sauver un objet métier de type persistant, d'invoquer une opération comme "Mettre à jour", l'objet-métier se chargeant alors, en s'appuyant sur l'infrastructure de persistance, de valider toutes les données, d'appliquer différentes règles de gestion et de contrôle, de transférer toute ces données dans une BD, et, en cas de problème, d'annuler toute l'opération et d'informer l'utilisateur de l'échec de cette dernière. Le but poursuivi est de séparer totalement toute problématique liée à la gestion physique des données de la logique-métier elle-même.

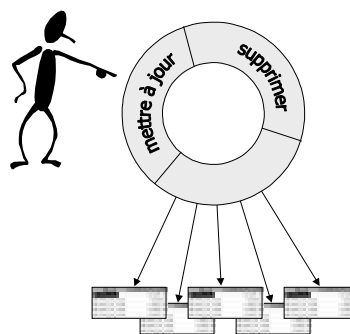


Figure 70. L'utilisateur est déchargé de tout souci relatif à la gestion des données au sein d'une BD: l'objet métier joue le rôle d'intermédiaire entre l'utilisateur et la BD.

De nombreuses informations sont disponibles dans la littérature quant à la manière de concevoir des infrastructures de persistance interagissant avec des BD relationnelles (Ambler, 1998a; Wright, 1998;

2. "A business actor represents a role that someone or something in the environment can play in relation to the business." (Jacobson et al., 1992). Voir annexe 6 pour plus de détail.
1. Un objet métier persistant est un objet métier dont l'état doit être maintenu d'une session à l'autre, et donc dont les valeurs des attributs doivent être enregistrées d'une façon ou d'une autre (BD, système de fichiers).

Yoder et al., 1998). Cette problématique a été étudiée dans le cadre de ce travail, et l'annexe 7 présente différentes possibilités.

Les interactions avec des BD relationnelles ne constituent cependant pas le seul problème à devoir être abordé. Dans le contexte des SIAGEM doivent être manipulées des données dont la pérennité ne peut être assurée selon le modèle relationnel : les données à caractère spatial et les séries temporelles. Il s'agit, lors de la construction de l'infrastructure de persistance, de penser également à ce type de données. Les données thématiques peuvent être stockées par des BD relationnelles, tandis que les données géographiques ou les séries temporelles peuvent devoir l'être par des fichiers. Notons que les SGBD d'entreprise les plus récents, comme Oracle 8 (Abbey et al., 1999), sont à même de gérer différents types de données complexes, dont les données spatiales et les séries temporelles

D'un point de vue technique, la présente étape est certainement la plus complexe et doit être menée comme un projet quasiment indépendant et par des informaticiens expérimentés. Le développement d'une infrastructure basée sur le modèle "Broker Object" (voir annexe 7) ne s'attachant qu'aux interactions avec une BD relationnelle, et non dotée de fonctionnalités sophistiquées, nécessite 2,5 à 4 mois-homme de travail. Le développement d'une infrastructure plus complexe, tenant compte de

```
Module Persistence {
    exception NotEditable {string why};
    exception NotDeletable {string why};
    exception NotUpdatable {string why};

    interface PersistentObject {
        readonly attribute string id;
        // Identifiant de l'objet persistant
        readonly attribute boolean isNew;
        // Vrai si l'objet vient d'être créé et n'existe pas encore dans la BD
        readonly attribute boolean isDirty;
        // Vrai si un attribut au moins de l'objet a été modifié mais pas encore
        // mis à jour dans la BD
        readonly attribute boolean isEdited;
        // Vrai si l'objet est en mode édition : ses attributs peuvent être modifiés
        readonly attribute boolean isDeleted;
        // Vrai si les données correspondant à l'objet dans la BD ont été détruites

        void edit() raises (NotEditable);
        // Met un objet en mode édition
        void delete() raises (NotDeletable);
        // Détruit tout ce qui se rapporte à l'objet dans la BD
        void update() raises (NotUpdatable);
        // Met à jour les informations relatives à l'objet dans la BD. Si l'objet est
        // neuf, enregistre les données relatives à l'objet dans la BD.
    };
};
```

Figure 71. Chaque objet-métier persistant doit implémenter une interface spécifique. Nous présentons ici l'interface de persistance, en OMG IDL, que nous suggérons. La création et la recherche d'objets métier sont confiées à des objets tiers de type "gestionnaires".

multiples cas de figure et prenant en compte de nombreux aspects (gestion de cache, protection, outils d'administration, etc.) nécessite certainement au minimum une année-homme¹ de travail. A noter que, suivant l'environnement pour composants choisi, des infrastructures qui peuvent être adaptées au contexte des SIAGEM (manipulation de données spatiales et temporelles) sont disponibles sur le marché.

Il est important de réaliser que cette infrastructure, dont la mise en place nécessite un gros effort, est

1. Informations issues de forums de discussion sur Internet.

absolument nécessaire au fonctionnement de chaque composant métier. De sa qualité dépendent les performances et le bon fonctionnement de tous les objets métier qui s'appuient sur elle. Une telle infrastructure, une fois disponible, est à l'origine d'une diminution considérable du temps nécessaire au développement de composants-métier de qualité. Elle doit être construite préalablement aux itérations du processus de développement, et peut ensuite être tenue à jour et améliorée parallèlement à ces dernières.

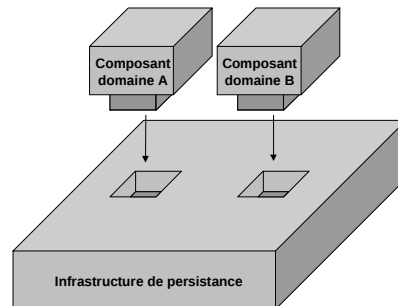


Figure 72. Une fois l'infrastructure de persistance en place, les développeurs peuvent se consacrer à l'implémentation de la logique-métier. Les composants implémentant cette logique métier peuvent s'appuyer sur l'infrastructure.

5.3.2.4 Analyse des besoins

Le développement d'un ensemble de composants métier se fait donc dans le contexte de la construction d'un outil logiciel destiné à soutenir un processus métier particulier. Il s'agit ainsi de mener une analyse des besoins "classique", tout en élargissant la réflexion de manière à concevoir des composants métier qui puissent être réutilisés dans d'autres contextes. La phase d'analyse des besoins a pour objectif de permettre de bien comprendre en quoi consiste le processus-métier qui doit être soutenu, ce qui est attendu de l'application projetée, mais aussi quels sont les autres processus-métier qui pourraient, lors d'une informatisation, s'appuyer sur les composants métier développés.

Cette étape part donc de l'expression d'un besoin. Celui-ci doit être exprimé de manière claire, sous forme d'une description textuelle. Les acteurs concernés doivent être identifiés, et les fonctions attendues ainsi que les exigences non fonctionnelles doivent être précisées. En particulier, le processus-métier que l'on souhaite informatiser doit être décrit par le biais de cas d'utilisation réels étendus (annexe 6). De tels documents décrivent le processus-métier sous forme textuelle, tel qu'il se déroule.

Nous avons également expliqué le rôle que jouent le territoire et ses composantes comme éléments d'articulation et d'intégration entre les visions des différents intervenants (chapitre 2.4.1.3), ainsi que la manière dont ce rôle est dévolu à des composants-entité. Il est impératif, pour développer ces derniers, de considérer non seulement le processus-métier ayant motivé le cycle de développement en cours, mais également ceux, menés par des acteurs différents et dans des contextes différents, qui conduisent à manipuler des concepts identiques.

Avant d'aller plus loin dans notre explication, il est nécessaire de définir le terme de concept, qui est utilisé à plusieurs reprises dans ce rapport. Un concept est "l'idée que l'on a d'un objet de la réalité" (Buogo, 1995). Un concept possède une intension (ou définition) et une extension (les entités qui correspondent au concept défini) et peut être défini par un symbole (illustration 73).

Ces concepts peuvent être (adapté de Brunet, 1993; cité dans Dano, 1997) :

- de type tangible ou entité, qui sont des "abstractions de choses ayant une existence propre dans le monde physique";
- de type rôle ou acteur, qui font référence à des acteurs au sens où cela a été défini à l'annexe 6;
- de type événement ou incident, qui font référence à un événement survenant dans le système à un moment donné;
- de type interaction, ou processus, qui résultent de l'existence d'une relation entre deux concepts ou plus;
- des références à des unités organisationnelles.

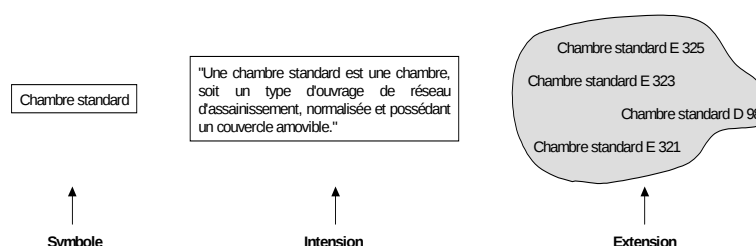


Figure 73. La notion de concept : symbole, intension et extension, adapté de Larman (1997)

Cette explication donnée, notre proposition peut être expliquée par un exemple. Imaginons que le besoin d'un outil informatisé de gestion du cadastre de l'assainissement soit à l'origine du développement en cours. Une analyse des besoins "classique" conduit à faire une description de cette activité, une liste des acteurs dans ce contexte, une description des objectifs fixés à l'application, une liste des fonctions que le système logiciel doit être à même de fournir et une liste des exigences non fonctionnelles s'y rapportant (Larman, 1997). La méthode COWIS propose de mener une identification préliminaire des concepts de type entité qui sont cités dans le but, à terme, de les implémenter dans des composants métier qui reprendront à leur compte le rôle d'intégrateurs de différentes perceptions d'une même réalité. Un exemple nous permet d'illustrer notre propos : soit la phrase, issue de l'analyse des besoins :

"Le système doit assurer la gestion des données relatives aux différents ouvrages du réseau d'assainissement, à savoir les canalisations, chambres, exutoires, bassins de décantation, bassins de rétention, stations de pompage, déversoirs et stations d'infiltration."

Les concepts de type entité "Ouvrage de réseau d'assainissement", "Canalisation", "Chambre", etc. y sont mentionnés. Les renseignements collectés durant une phase "classique" d'analyse des besoins permettent de déterminer comment les acteurs à qui est destinée l'application se représentent de telles entités. La méthode COWIS propose d'identifier en plus les autres acteurs et processus-métier du domaine de la gestion des eaux qui traitent des mêmes types d'entités. Pour faciliter cette tâche, il est suggéré de s'aider de la matrice "processus-système" (figure 74). Celle-ci met en relation les catégories de processus métier avec les différentes composantes du système "Eaux en milieu urbain", et doit être utilisée de la manière suivante : 1°) déterminer à quel sous-système, parmi les "composantes du système" de la matrice "processus-système" présentée à la figure 74, appartiennent les entités identifiées lors de l'analyse "classique"; 2°) sélectionner dans la matrice citée ci-dessus la colonne relative au sous-système identifié (soit dans notre exemple le réseau d'assainissement); 3°) parcourir les différentes catégories de processus-métier reportés en ordonnées, chaque cellule de la colonne correspondant à une catégorie

d'activité dans le contexte d'un sous-domaine de la gestion des eaux; 4°) se demander au niveau de chaque cellule rencontrée quels sont les acteurs et processus-métier concernés. La réflexion doit être menée en gardant à l'esprit les différents niveaux organisationnels de la gestion des eaux (figure 9). Une fois identifiés les processus-métier manipulant les mêmes concepts, ceux-là peuvent être décrits par des cas d'utilisation réels résumés¹ (annexe 6).

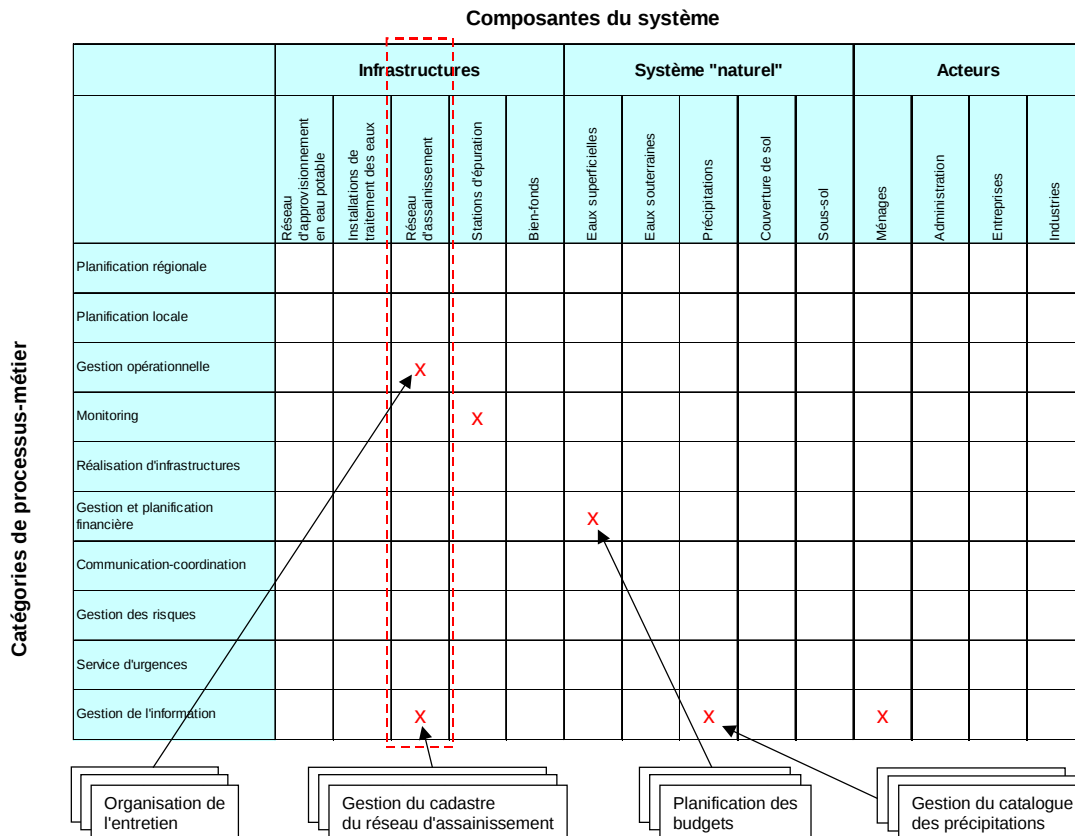


Figure 74. Matrice processus-système. Chaque case de la matrice correspond à un ensemble de processus-métier. La finalité de cette matrice n'est pas d'être exhaustive, mais de fournir un cadre de réflexion à son utilisateur.

La figure 75 présente le modèle des cas d'utilisation identifiés en suivant l'approche proposée, c'est-à-dire en s'interrogeant sur d'autres processus métier que ceux directement à l'origine du développement en cours. Un modèle des cas d'utilisation doit s'accompagner des cas d'utilisation résumés correspondants. La figure 76 présente un exemple d'un tel cas d'utilisation résumé.

Aux documents d'analyse fonctionnelle classique, qui se rapportent à l'application intégrant le ou les composants métier, s'ajoute donc un modèle de cas d'utilisation relatif aux autres processus métier qui manipulent des concepts de type entités similaires à ceux mis en évidence dans la phase d'analyse fonctionnelle classique, et qui permettra aux développeurs de composants métier de cerner les intervenants qui utiliseront ces composants. Les documents issus de la présente étape sont ainsi :

- un ou plusieurs cas d'utilisation réels étendus décrivant le processus métier dont l'informatisation a déclenché le cycle de développement considéré;

1. Un cas d'utilisation réel résumé décrit en quelques phrases le processus-métier, sans rentrer dans des considérations relatives à un système informatique.

- un ensemble de documents textuels qui traduisent les attentes des intervenants ayant demandé la construction d'un outil informatique (selon une analyse "classique"), dont les cas d'utilisation précédemment cités font partie;
- une liste des concepts identifiés;
- un modèle des cas d'utilisation réels résumés, mettant en évidence la relation entre les acteurs et les cas d'utilisation identifiés à l'aide de la matrice processus-système;
- les cas d'utilisation réels résumés des processus-métier qui figurent dans ce modèle.

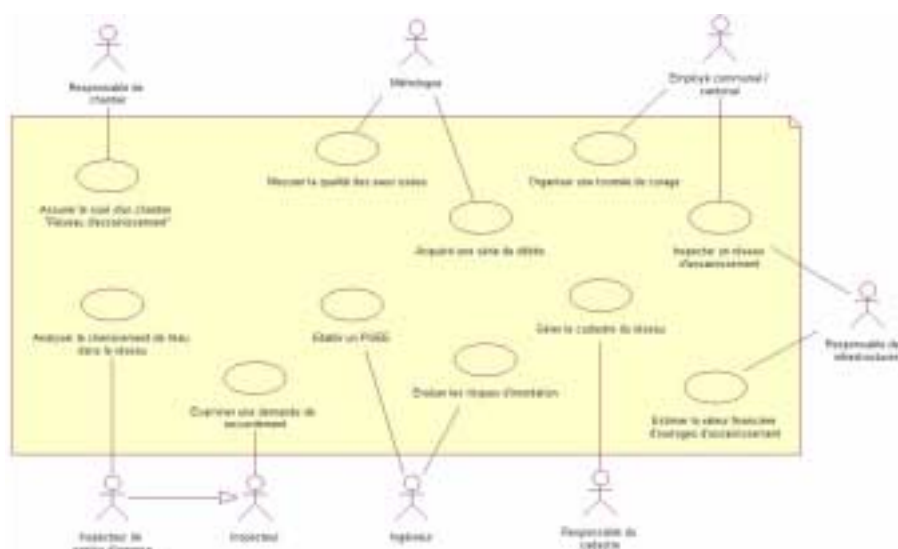


Figure 75. Modèle des cas d'utilisation en relation avec le concept "Réseau d'assainissement".

Nom :	Evaluer les risques d'inondation
Acteur(s) :	Ingénieur
Description :	Un ingénieur reçoit pour mandat d'évaluer les risques de surcharge d'une portion de réseau d'assainissement. Il acquiert l'information dont il a besoin, modélise le réseau d'assainissement, effectue des simulations des écoulements, analyse les résultats, délimite les points du réseau présentant des sous-capacités, en fonction de leur fréquence, et établit une carte des sous-capacités.

Figure 76. Cas d'utilisation résumé de haut niveau "Évaluer les risques d'inondation".

5.3.2.5 Analyse et segmentation de domaine

Cette étape a pour objectif d'affiner l'identification des concepts importants pour notre problématique, et de mener une première délimitation de l'envergure (quant aux concepts manipulés) qui doit être celle des composants métier. La figure 77 présente la manière dont s'enchaînent les différentes activités prenant place au cours de cette phase. La première activité à mener est d'entamer l'élaboration d'un modèle du domaine de la gestion des eaux, si les composants-métier qui vont être créés sont les premiers, ou de compléter le modèle existant, s'il ne s'agit pas là de la première itération au travers de la méthode¹. Ce modèle du domaine sert de référence commune aux différents modèles "partiels" qui seront implémentés par des composants métier, et est enrichi lors de chaque itération. Les concepts jugés suffisamment

1. "Un modèle du domaine décrit, en les reliant les uns aux autres, les concepts essentiels du contexte sous forme d'objets du domaine." (Jacobson et al., 2000)

importants pour figurer dans ce modèle donnent lieu à une classe du domaine. Précisons d'ores et déjà que la représentation de ce modèle ne peut se faire à l'aide d'un seul document : un modèle d'une telle envergure doit au contraire être segmenté en sous-modèles représentés sur des diagrammes différents, et qui ensemble forment un tout cohérent.

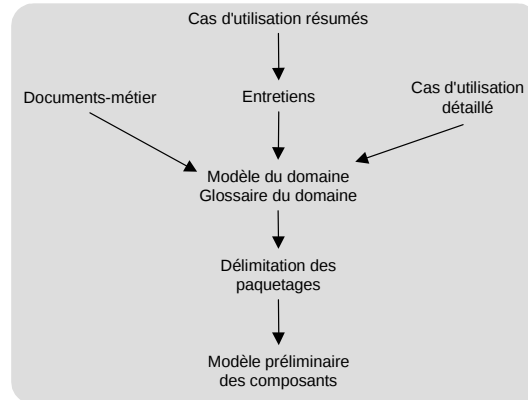


Figure 77. Les différentes étapes de la phase d'identification des composants.

L'identification des classes de domaine se fait à partir de plusieurs sources (figure 77) : des cas d'utilisation détaillés relatifs au processus-métier (ayant engendré le cycle de développement en cours, des entretiens avec les acteurs extérieurs à ce processus métier, mais manipulant dans leur activité professionnelle les mêmes concepts de type entité (ces acteurs ayant été identifiés lors de l'étape précédente), ainsi que des documents de travail utilisés ou générés par les acteurs en question.

- Le cas d'utilisation détaillé est analysé, et les concepts importants en sont extraits.

Les entretiens de la phase d'identification des domaines suivent un canevas extrêmement simple : après une brève introduction présentant les raisons de la démarche, les questions suivantes sont posées à l'interviewé, afin de bien cerner son activité et d'éventuellement identifier de nouveaux processus métier :

1°) "En quoi consiste votre travail ? Quels sont vos responsabilités et vos objectifs principaux"

2°) "Quelles sont les tâches que vous menez à bien pour remplir vos responsabilités et atteindre vos objectifs ?"

3°) "Quels outils informatiques employez-vous dans ce contexte ?"

L'entretien doit ensuite se focaliser sur les processus-métier conduisant à la manipulation des entités identifiées dans le contexte des cas d'utilisation détaillés identifiés à l'étape précédente. Les entretiens peuvent être mis à profit pour évaluer les besoins et attentes des interlocuteurs en matière d'outils informatiques :

4°) Pouvez-vous me décrire << telle tâche >>.

5°) "Qu'attendez-vous d'outils informatiques ?"

Figure 78. Canevas d'entretien suggéré.

- Les entretiens sont une source précieuse d'information, car de nombreuses informations ne se trouvent que dans la mémoire des praticiens (Berztiss, 1999). Les personnes qu'il s'agit d'interviewer sont des représentants des acteurs impliqués dans les cas d'utilisation résumés de l'étape précédente. Un certain nombre d'entretiens ont été menés (figure 78) qui nous ont permis d'en identifier la forme la plus efficace : des entretiens doivent être menés selon un mode non-directif¹, et avoir une durée de deux à trois heures environ.
- Les normes ou recommandations d'associations professionnelles (ASPEE et ASMFA, 1990; GSA, 1994; VSA, 1990; VSA, 1992) constituent des documents très intéressants pour une analyse de

domaine, car ils résultent de réflexions approfondies menées par des spécialistes du domaine. Des documents générés dans le cadre d'une activité particulière, comme ceux présentés à la figure 79¹, contiennent également un grand nombre d'informations.

- Les classes de domaine, qui donc modélisent les concepts les plus importants, sont organisées au sein d'un modèle du domaine (figure 80). Un tel modèle met en évidence les classes et leurs attributs, les relations entre les classes ainsi que les cardinalités des relations. Il doit s'accompagner d'un glossaire qui définit, en quelques phrases précises, les différentes classes et leurs attributs. Précisons qu'à ce stade du processus il ne s'agit pas d'élaborer un modèle complet et exact. Un tel modèle est affiné et complété par étapes, à la faveur des itérations successives. L'analyse plus poussée qui est effectuée dans les étapes suivantes permet de compléter rétroactivement le modèle de domaine.

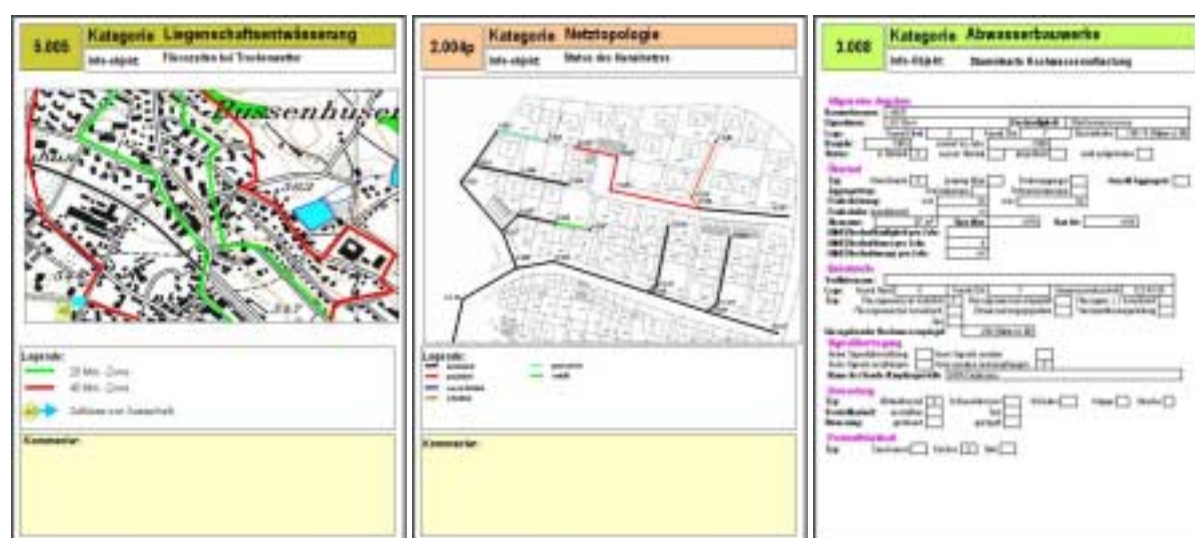


Figure 79. Modèles de documents utilisés dans le contexte de la planification de l'assainissement. A gauche un document montrant les isochrones des temps d'écoulement par temps sec vers la STEP, au centre un document montrant le statut des canalisations (hors service, en service, planifiées), et à droite un formulaire relatif aux ouvrages d'assainissement (Bernasconi et al., 1998).

La figure 80 présente la portion du modèle de domaine de la gestion des eaux en milieu urbain se rapportant au réseau d'assainissement. Il s'agit d'un modèle préliminaire, qui ne sert qu'à identifier les concepts importants, comprendre le domaine et établir son vocabulaire. Ce modèle est en l'état inexact (il ne permet par exemple pas de déterminer la topologie réseau), incomplet (de nombreux concepts manquent), et pas assez détaillé (les cardinalités, contraintes et attributs n'y figurent pas forcément).

Il faut à présent aborder la problématique de la construction des composants métier, et pour cela organiser les classes de domaine en ensembles, les paquetages², qui ensuite feront l'objet d'une analyse

1. Un entretien non-directif (ou libre) est "un entretien au cours duquel l'interviewé développe librement ce qu'il a à dire sur le thème, l'interviewer se bornant à déclencher et à relancer" (Harvatopoulos et al., 1989). L'attitude de l'interviewer doit être la plus neutre possible, ses seules interventions étant destinées soit à soutenir le discours, soit à inviter l'interlocuteur à poursuivre lorsque c'est nécessaire.

1. Ces documents peuvent également être consultés sur le site <http://www.imanagement.org>

2. "Paquetage : mécanisme général pour organiser des éléments dans des groupes. Les paquetages peuvent s'emboîter à l'intérieur d'autres paquetages." (Bouzeghoub et al., 1997)

plus détaillée pour finalement déboucher sur des composants métier. Comment mener ce regroupement? Chaque composant s'organise autour d'un concept-clé (chapitre 4.5.5). Il faut l'identifier à partir du modèle de domaine. Chaque paquetage regroupe logiquement une classe-clé (correspondant au concept-clé) avec un certain nombre de classes de domaine. Comment identifier ces concepts-clé? Les points qui suivent ne constituent pas des règles strictes, mais plutôt des conseils.

- Un concept-clé revient souvent dans le discours des intervenants et dans les documents analysés. Dans notre exemple le concept d'ouvrage d'assainissement, très fréquemment cité, constitue un concept-clé.
- Un concept-clé présente un certain degré d'indépendance : il peut être utile de manipuler un objet métier implémentant ce concept-clé, indépendamment d'autres objets métier. Dans le modèle présenté à la figure 80, des applications logicielles peuvent être imaginées qui ne se rapportent qu'à la station de mesure, à la station d'épuration, aux ouvrages du réseau d'assainissement.
- Dans le modèle de domaine un concept-clé joue fréquemment un rôle central, qui peut être mis en évidence par les nombreuses relations qui lient la classe-clé qui s'y rapporte à d'autres classes de domaine : la classe "Ouvrage de réseau d'assainissement" présente de telles caractéristiques.
- Un objet métier est un objet d'une granularité relativement importante : un concept-clé est ainsi souvent un agrégat d'autres concepts.

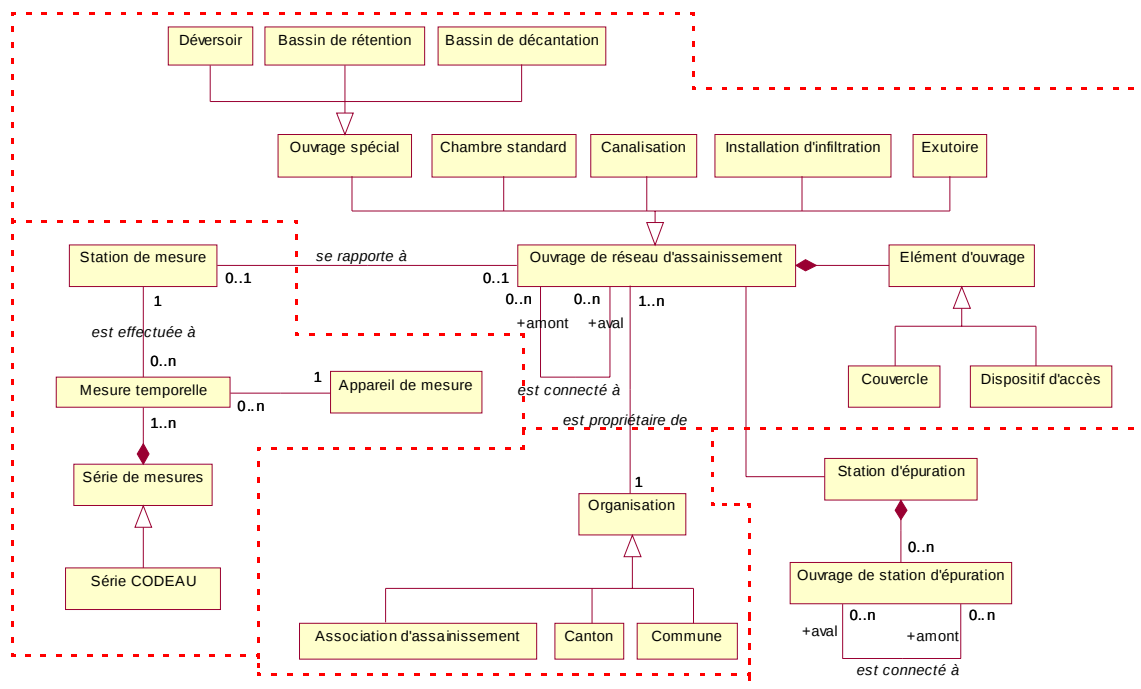


Figure 80. Extrait du modèle de domaine relatif à la gestion des eaux. L'annexe 2 présente le formalisme UML utilisé. À noter toutefois que ce modèle n'est à ce stade qu'une ébauche, et est donc imprécis et inexact. Les concepts sont regroupés logiquement en quatre paquetages.

Une fois les classes-clé identifiées, les autres concepts qui feront partie du même paquetage doivent être regroupés :

- seront ainsi généralement intégrées au même paquetage d'une classe-clé les sous-classes de ces dernières;
- lorsqu'une classe-clé agrège d'autres classes, ces dernières font généralement partie du même

paquetage, à moins qu'il n'y ait un intérêt particulier à les manipuler isolément, ou qu'elles ne soient également agrégées par d'autres classes appartenant à des paquetages différents;

- une classe de domaine qui n'est jamais manipulée de manière indépendante doit être intégrée au paquetage regroupant les classes avec lesquelles elle est manipulée;
- les classes de domaine qui peuvent être considérées comme des attributs de classes sont regroupées dans le même paquetage que ces dernières. Constituent une exception à ce principe les classes "utilitaires" qui, nécessaires à de nombreux composants, font l'objet de paquetages spécifiques.

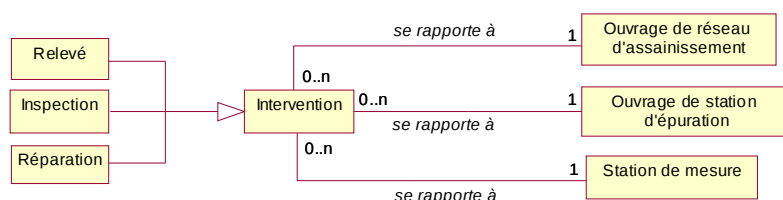


Figure 81. Une autre vue sur le modèle de domaine en donne une perspective différente. Un modèle de domaine ne peut faire l'objet d'un document unique.

Le modèle de domaine, tel que présenté aux figures 80 et 81, rend possible l'identification de cinq paquetages (figure 82). Chacun de ces paquetages peut à présent constituer la base d'une analyse plus détaillée. Les résultats en seront en retour utilisés pour affiner le modèle de domaine.

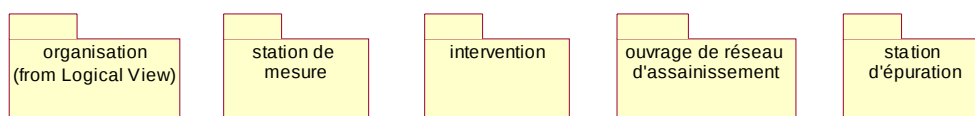


Figure 82. Les paquetages identifiés, à partir desquels seront construits des composants métier.

Constituent les documents finaux de cette étape :

- le modèle de domaine, accompagné d'un glossaire décrivant les classes de domaine;
- la liste des paquetages identifiés, chaque paquetage se rapportant à une portion du modèle de domaine regroupant un ensemble de classes de domaine.

La liste des paquetages constitue un modèle préliminaire des composants métier, puisque à partir de chaque paquetage sera construit au minimum un composant métier. À noter que si la présente phase propose plusieurs paquetages, chacun d'entre eux peut être pour la suite pris en charge par une équipe différente.

5.3.2.6 Analyse orientée composants

Principe général

La présente étape se concentre sur un paquetage et comprend les activités suivantes : élaboration d'un modèle de classes d'analyse, spécification de règles de gestion, tant invariantes que relatives aux opérations dont les différents composants auront la responsabilité, et délimitation précise des composants métier. Le but de cette phase d'analyse est, comme dans de nombreuses méthodes, de traduire et de préciser les exigences issues de l'analyse des besoins de manière complète, cohérente et

dénuée d'ambiguïtés. Le fait de se concentrer à partir de là sur un seul paquetage permet de répondre à l'objectif de développement rapide que nous nous sommes fixé au chapitre 3.2.7.

Aspects structurels

Les premières ébauches d'un modèle de domaine, qui doivent être effectuées rapidement, ne sont ni complètes, ni optimales. Par contre, les modèles sémantiques, matérialisés par des interfaces publiques, que proposent à terme les composants métier, doivent être conçus soigneusement. Un modèle de classes d'analyse, constitué de classes interreliées par un réseau de relations et accompagné d'un glossaire, constitue un moyen de communication entre l'informaticien et le spécialiste du domaine, et permet à l'informaticien de bien comprendre la problématique traitée. Il utilise les mêmes notations et répond aux mêmes règles qu'un modèle de domaine. Les figures 83 et 84 représentent deux modèles de classes d'analyses, qui nous permettent d'en discuter différents aspects : le premier de ces modèles se rapporte au paquetage "Station de mesure", tandis que le second concerne le paquetage "Ouvrage de réseau d'assainissement".

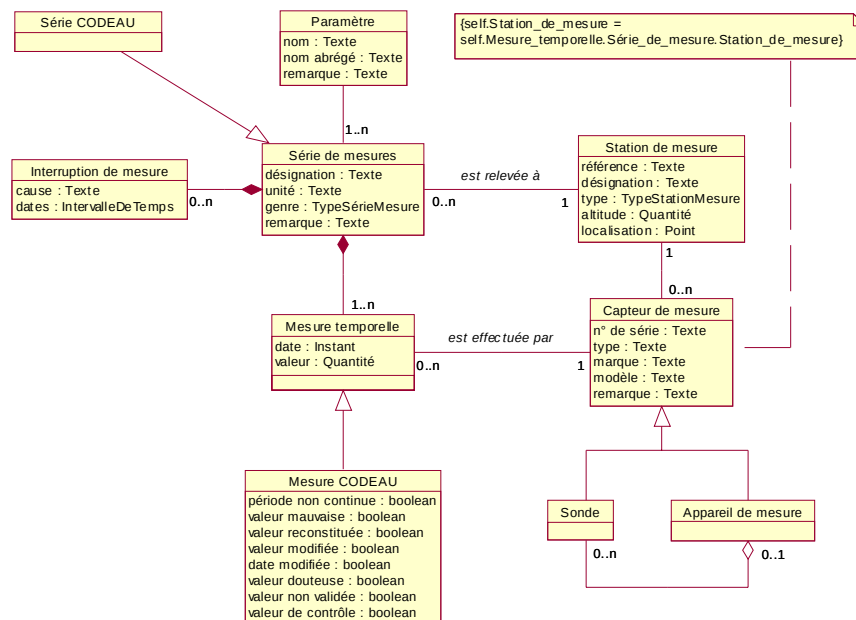


Figure 83. Modèle initial de classes d'analyse de la station de mesure. Le format CODEAU est utilisé pour la gestion des données temporelles en hydrologie (CODEAU, 1996; de Souza et al., 1994). Noter la présence d'attributs qui sont en fait des classes de type utilitaire ou des domaines de valeurs.

- Chaque classe de domaine peut donner naissance à une ou plusieurs classes d'analyse. L'éclatement, souvent nécessaire à une modélisation efficace, d'une même classe de domaine en plusieurs classes d'analyse, peut rendre un modèle difficile à comprendre. Dans la figure 84 par exemple, les classes d'analyse "Noeud" et "Tronçon" de réseau d'assainissement ont été définies afin que la topologie réseau puisse être correctement représentée. Cette idée, proposée par l'ASPEE (VSA, 1999), revient à modéliser par des classes d'analyse distinctes les aspects liés à la topologie-réseau et les caractéristiques "physiques" d'un ouvrage. Une telle modélisation permet de représenter des ouvrages à l'hydraulique compliquée. Citons, à titre d'exemple basique de ce type d'ouvrage une chambre d'assainissement traversée simultanément par une canalisation d'eaux usées et une canalisation d'eaux pluviales (qui bien sûr ne doivent hydrauliquement pas être interconnectées).

- Les concepts qui figurent dans le modèle doivent faire l'objet d'une classe d'analyse, et non d'un attribut d'une classe existante. L'exutoire d'un réseau d'assainissement (pluvial) peut par exemple être un cours d'eau : le concept de cours d'eau ne peut être attribut de la classe d'analyse "Exutoire", mais doit être une classe d'analyse. Les attributs des classes d'analyse doivent en principe être des types de données : chaîne de caractère, nombre, booléen, etc., ou des domaines de valeurs.

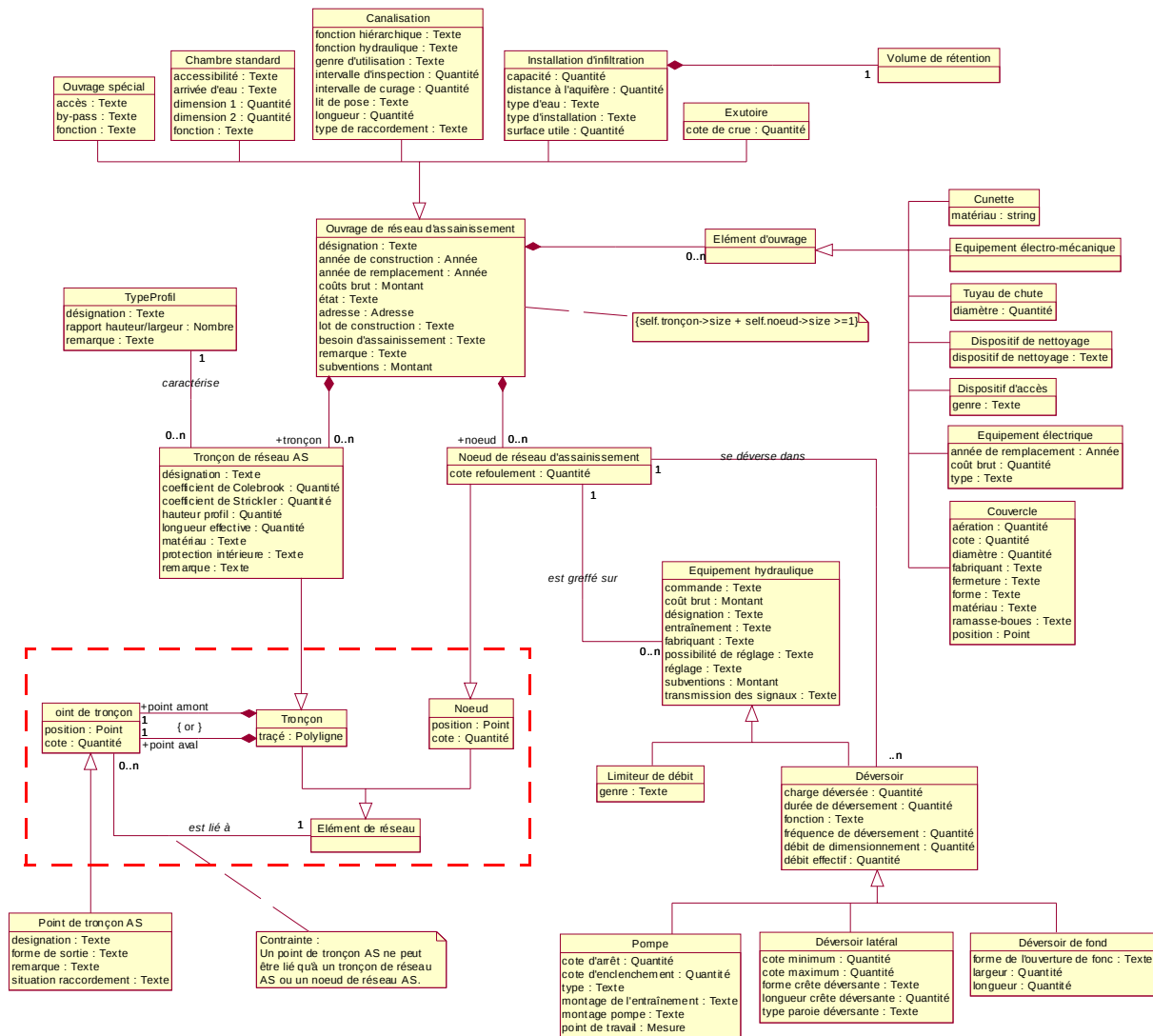


Figure 84. Modèle de classes d'analyse du paquetage "ouvrage de réseau d'assainissement". Les contraintes de type invariant peuvent y être exprimées directement dans le modèle, soit de manière formelle, soit en langage naturel. Remarquer la modélisation qui permet d'isoler les concepts génériques relatifs aux réseaux, ceci de manière à pouvoir ensuite créer un composant qui leur est propre.

- Constituent une exception à ce dernier principe les classes d'analyse dites "utilitaires" : une date, un point, une adresse par exemple. De telles classes, implémentées par la suite par un composant de type utilitaire, figurent dans le modèle en tant qu'attributs des classes d'analyse. La figure 85 présente quelques classes de ce type. On peut distinguer une classe "utilitaire" par le fait que la manipulation indépendante de ses instances, sans connaissance des objets auxquelles elles se rapportent, ne présente aucun intérêt : une quantité, ou une polyligne, ne sert à rien si l'objet auquel elle correspond n'est pas connu. Les entités spatiales sont modélisées par des classes utilitaires. Ceci

permet de ne pas surcharger les diagrammes. Les pictogrammes spatiaux permettent de clairement mettre en évidence les types spatiaux des entités ou des attributs, et pourraient dans les modèles de classes d'analyse (et à l'étape de conception qui suit) être mis en oeuvre avec profit (Bédard, 1999; Parent, 1996). Nous avons cependant ici renoncé à cette option, les outils utilisés n'offrant pas une telle possibilité.

- On trouve sur le marché des composants "techniques" qui permettent la manipulation d'entités spatiales : il est donc superflu de redévelopper des composants implémentant ces mêmes concepts. Les possibilités d'intégration de composants préexistants doivent être examinées lors d'une démarche telle que la nôtre. Mentionnons comme exemples de tels composants les produits MapX de la société MapInfo¹ (voir description en annexe 9) et MapObjects de la société Esri².

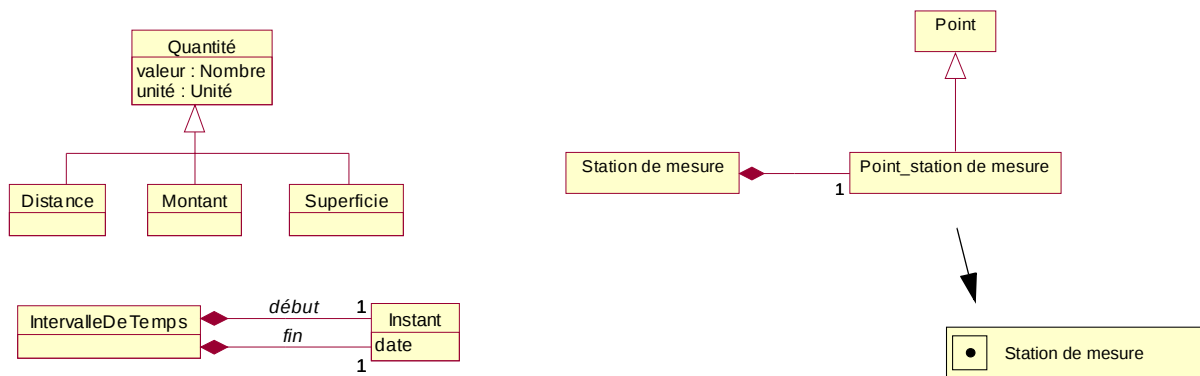


Figure 85. Modèles se rapportant à des classes utilitaires : principe de la quantité adapté de Fowler (1997), aspects spatiaux et temporels selon Gayte (1996). La spatialité et la temporalité peuvent être représentées par des pictogrammes comme ceux proposés par Bédard (2000).

- Un même paquetage peut être à l'origine de plusieurs composants, et le modèle de classes d'analyse permet de délimiter la sémantique de ces composants. En effet, il faut mener l'analyse en gardant à l'esprit la notion de réutilisation : *"Reusable components not only satisfy the immediate needs of the current development but those of the future as well"* (Coleman et al., 1994). Ainsi, en menant

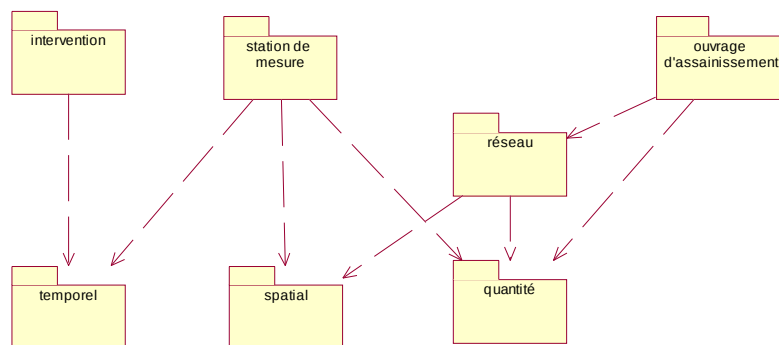


Figure 86. Identification de paquetages et des relations qui les lient : des paquetages de domaine s'appuyent sur des paquetages utilitaires.

l'analyse relative au paquetage "Ouvrage de réseau d'assainissement", il nous a semblé utile de

1. <http://www.mapinfo.com>
2. <http://www.esri.com>

remanier le modèle de manière à pouvoir créer un composant implémentant les différents aspects relatifs au réseau qui peut être réutilisé dans d'autres contextes que celui de l'assainissement (figure 84). En effet un tel composant intègre des algorithmes de navigation qui peuvent être complexes, devoir faire l'objet d'optimisations, et être utiles dans d'autres contextes (voies de communication, réseaux d'approvisionnement en eau potable, etc.). Cela justifie le développement d'un composant indépendant.

Les règles de gestion invariantes, soit, comme leur nom l'indique, toujours vérifiées quel que soit l'état du système, doivent être déterminées. Ces règles peuvent être reportées directement dans le modèle de classes d'analyse (voir les exemples à la figure 84) ou dans un document séparé. De telles règles peuvent être rédigées en langage naturel, et sont ainsi faciles à comprendre mais présentent le risque d'être imprécises et ambiguës, ou dans un langage de description de contraintes comme OCL¹ (Object Constraint Language). Des langages de ce genre rendent possible une description parfaitement précise et dépourvue d'ambiguïtés des différentes contraintes qui complètent les modèles. La figure 87 présente à titre d'exemple un extrait des règles invariantes relatives à la classe "Ouvrage d'assainissement".

```
Context SewersystemStructure inv:

    self.comment.ocIsTypeOf(String)

    self.constructionState.ocIsTypeOf(TypeSewersystemStructureConstructionState)

    self.constructionYear.ocIsTypeOf(Integer) and
    self.constructionYear > 1600 and
    self.constructionYear < 2100

    self.designation.ocIsTypeOf(String) and
    self.sewersystem.allInstances -> forAll (p1, p2 | p1 <> p2 implies
    p1.designation <> p2.designation)

    self.grants.ocIsTypeOf(Money)

    self.needForSanitation.ocIsTypeOf(TypeSewersystemStructureNeedForSanitation)

    self.replacementYear.ocIsTypeOf(Integer) and
    self.replacementYear > 1600 and
    self.replacementYear < 2100

    self.status.ocIsTypeOf(TypeSewersystemStructureStatus)
```

Figure 87. Description en OCL des invariants relatifs aux objets de type "Ouvrage de réseau d'assainissement". Les domaines de valeurs qui sont cités ont été définis préalablement.

Aspects fonctionnels

Nous avons, lors de la phase précédente, décrit la structure d'un sous-système particulier. Les composants métier qui constituent l'aboutissement de l'itération doivent être à même de proposer des services et de remplir des fonctions. Il s'agit ici de déterminer quels sont les services qui doivent être assurés par ce sous-système.

L'analyse des aspects fonctionnels est différente selon que le composant métier en voie d'élaboration appartient au type entité ou processus. En effet, dans le cas d'un composant-entité, l'analyse des aspects

1. OCL est un langage formel dont la syntaxe a de plus été conçue de manière à être facile à appréhender (OMG, 1999; Warmer et Kleppe, 1998)

structurels représente la majeure partie du travail, les fonctions attendues se résumant principalement à l'acquisition, la gestion et, éventuellement, la visualisation de l'information relative à ces entités. Dans le cas des composants-processus, les aspects structurels sont très limités, leur diagramme de classe se limitant souvent à quelques classes d'analyse chargées de piloter le processus ou proposant à l'utilisateur les différentes fonctions à même de le soutenir dans son activité. En revanche, la spécification des aspects fonctionnels constitue la partie la plus importante de l'analyse. L'objectif est ici : 1°) d'identifier les services, ou opérations, que le composant sera amené à fournir; et 2°) de spécifier ces opérations.

Lorsque l'on se trouve dans le contexte de l'analyse d'un paquetage de type entité, les principales opérations sont des opérations qui autorisent la gestion de la persistance des objets métier (dites CRUD pour Create, Update, Read, Delete). Il faut également s'interroger quant aux services immédiats que de tels objets métier devraient pouvoir rendre. Par exemple, les ouvrages d'assainissement font partie d'un réseau : il est évident que doit être offerte la possibilité de connaître les ouvrages situés en amont ou en aval de chacun d'eux, et qu'une opération peut être créée à ce sujet. En ce qui concerne le domaine de la station de mesure, il est clair que l'on doit pouvoir connaître la période ayant fait l'objet de mesures, ou le nombre de mesures que comporte une série de mesures. L'identification de telles opérations se fait principalement par : 1°) l'analyse des documents de travail collectés en début de processus; 2°) des discussions avec les spécialistes du domaine; et 3°) de la réflexion personnelle du modélisateur.

Lorsque l'analyse porte sur un paquetage se référant à un processus, il est judicieux de l'appuyer sur un ou plusieurs cas d'utilisation-système (annexe 6). Un cas d'utilisation-système décrit la manière dont on prévoit que l'intervenant interagira avec le système informatique projeté. Un cas d'utilisation-système résulte donc d'une réflexion, basée sur les cas d'utilisation réels, quant à la manière de soutenir les processus-métier par un outil informatique. Le tableau 8 montre un extrait du cas d'utilisation-système relatif à une intervention d'urgence suite à un accident de la circulation impliquant un camion-citerne transportant un liquide polluant : l'inspecteur de piquet, par téléphone ou sur le terrain, utilise le SIAGEM pour déterminer les risques et moyens d'intervention.

Cas d'utilisation	
Nom	Identifier les possibilités d'intervention
Numéro	CU_43
Utilisé par	Intervention d'urgence
Date dernière modification	21.11.2000
Version	1.0
Statut	En révision
Type	Système, développé
Description succincte	Suite à un accident risquant de conduire à un écoulement de liquide potentiellement dangereux, le service d'intervention communal est alerté, intervient sur le terrain en renfort des services de police et de pompiers, et fournit des renseignements permettant de maîtriser la pollution.
Acteur principal	Inspecteur de service d'urgence (Inspecteur)
Acteurs secondaires	Officier de police, officier de pompiers.

Tableau 8. Extrait du cas d'utilisation (système, développé) relatif au contexte de l'intervention d'urgence. À gauche (partie scénario), on trouve les actions de l'acteur principal, et à droite celles du système.

Déclencheur	L'inspecteur de service est alerté (en principe par téléphone) d'un accident.
Précondition	Le SIAGEM est opérationnel.
Résultat en cas d'échec	Le SIAGEM ne fournit pas les indications demandées.
Résultat en cas de succès	Le SIAGEM soutient le processus d'intervention en fournissant les informations demandées.
Scénario principal	
.....	
4. L'inspecteur identifie le point d'entrée dans le réseau d'assainissement dans lequel s'écoule la substance polluante.	
5. L'inspecteur demande au SIAGEM la liste des exutoires dans le milieu naturel situés en aval du point d'entrée identifié.	6. Le SIAGEM renvoie la liste des exutoires en aval du point de déversement, dans l'ordre amont-aval, avec les descriptions des milieux naturels concernés (lacs, étangs, rivières, etc.), leur distance du lieu de déversement et une estimation du temps d'écoulement jusqu'à ces exutoires.
7. L'inspecteur demande au SIAGEM la liste des points du réseau où il serait possible d'intervenir pour bloquer la pollution.	8. Le SIAGEM renvoie la liste des ouvrages du réseau où une intervention serait possible, dans l'ordre amont-aval, avec leur distance du lieu de déversement et une estimation du temps d'écoulement jusqu'à ces ouvrages.
Scénario alternatif	
7.a. L'inspecteur demande au SIAGEM la station d'épuration qui recevra la pollution, la distance jusqu'à cette dernière ainsi qu'une estimation du temps d'écoulement jusqu'à elle (cas du réseau unitaire).	8.a Le SIAGEM renvoie la station d'épuration, la distance et une estimation du temps d'écoulement jusqu'à ladite station.
...	

Tableau 8. Extrait du cas d'utilisation (système, développé) relatif au contexte de l'intervention d'urgence. À gauche (partie scénario), on trouve les actions de l'acteur principal, et à droite celles du système.

À partir des documents de travail, des cas d'utilisation-système, des discussions avec les intervenants et de la réflexion du modélisateur, on définit les événements qui conduisent à invoquer des opérations sur le système informatisé : sont utilisés pour représenter cela des diagrammes de contexte du système (figure 88). Chaque opération représentée dans un tel diagramme est ensuite définie par un schéma

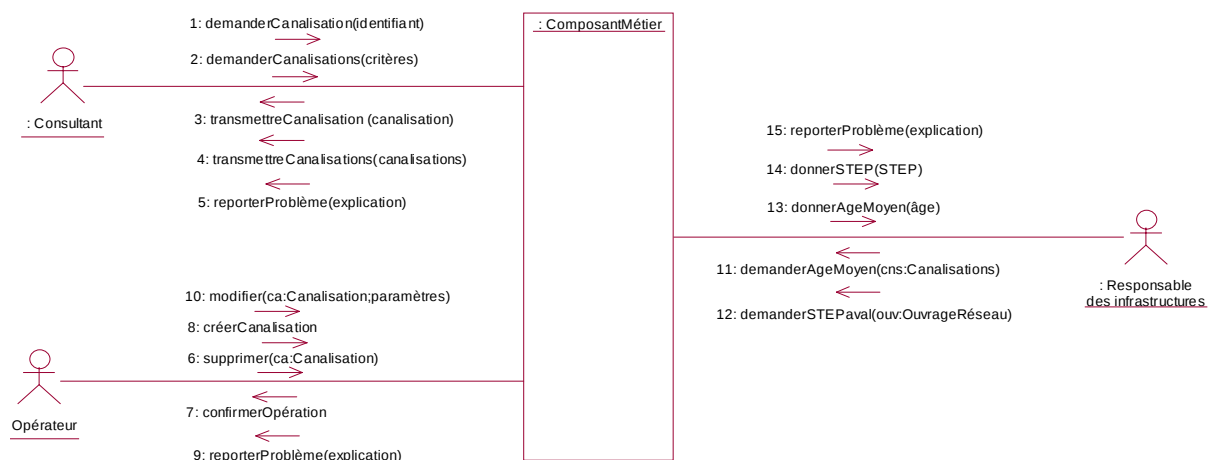


Figure 88. Diagramme de contexte du système.

d'opérations (tableau 9), l'ensemble de ces schémas formant le modèle des opérations du futur système informatisé : *"The operation model defines the semantics of each system operation in the system interface"* (Coleman et al., 1994). Le diagramme de contexte du système fournit une vue d'ensemble des interactions entre le système et les utilisateurs que le composant métier en cours de développement sera chargé de gérer, tandis que les schémas d'opération décrivent les opérations qui peuvent être invoquées par les utilisateurs. Une opération du système peut conduire à créer de nouvelles instances de classes, changer la valeur d'attributs d'objets existants, ajouter ou supprimer des occurrences d'associations, ou envoyer des événements à des acteurs (Strohmeier, 1999). Dans la mesure où cette partie de l'analyse - création des diagrammes de contexte du système et du modèle d'opérations - ne présente pas de particularité (mis à part le fait que nous nous intéressons non au système dans son ensemble, mais au sous-système délimité par un paquetage), nous renvoyons le lecteur à la littérature pour plus de détail (Coleman et al., 1994; Larman, 1997; Strohmeier, 1999; Sendall et Strohmeier, 2000).

Opération	créerCanalisation (noeud_amont : Noeud de réseau d'assainissement; noeud_aval : Noeud de réseau d'assainissement; données_canalisation)
Description	L'opérateur introduit une nouvelle canalisation dans le système. Les noeuds amont et aval sont passés en paramètre, ainsi que les données relatives à la canalisation. Le système enregistre les données, et signale la réussite ou l'échec de l'opération. Des méta-informations quant à la réalisation de la présente opération sont enregistrées (identité de l'opérateur, date et type d'opération).
Acteur	Opérateur
Événements générés	Un message est envoyé à l'opérateur pour signaler la réussite ou expliquer l'échec de l'opération. Un message de mise en garde est envoyé si la cote du noeud aval est plus élevée que la cote du noeud amont.
Préconditions	L'utilisateur du système est un opérateur ou un responsable du cadastre.
Postconditions	• une canalisation est créée; • un tronçon et ses points de tronçons amont et aval sont créés; • le tronçon a été lié à la canalisation; • une liaison est établie entre le point de tronçon amont et le noeud de tronçon amont passé en paramètre; • une liaison est établie entre le point de tronçon aval et le noeud de tronçon aval passé en paramètre; • les attributs sont modifiés suivant les données passées en paramètre; • l'identité de l'opérateur, le moment et la description de l'opération sont enregistrés.

Tableau 9. Opération de création de canalisation d'assainissement. Ici également il est possible d'employer un langage naturel ou formel, les pré- et post-conditions étant en fait des règles de gestion. Nous préférons utiliser ici le langage naturel, plus simple à comprendre pour les experts du domaine.

La phase d'analyse permet ainsi de préciser les caractéristiques des futurs composants métier, les concepts qui seront implémentés par ces derniers et mis à disposition des utilisateurs sous forme d'interfaces, et enfin, pour chaque composant, les opérations dont devra être assumée la responsabilité. Les documents délivrés à l'issue de cette phase sont :

- des cas d'utilisation-système développés, ainsi que le modèle de ces cas d'utilisation;
- un modèle définitif des composants métier, toujours représentés par des paquetages;
- un modèles de classes d'analyse par paquetage;
- pour chaque composant métier (représenté par un paquetage), une description détaillée des opérations dont il sera chargé (diagramme de contexte et schémas d'opérations).

5.3.2.7 Conception orientée composants

Mentionnons en préambule que les phases de conception, d'implémentation et de test-validation décrites ci-après ne le sont que succinctement : les principes qui guident ces phases dans des méthodes de développement orientées objet disponibles à l'heure actuelle sont également de mise en ce qui concerne la conception de composants métier. Nous nous bornons de ce fait à souligner les points importants ou qui sont propres à l'approche proposée.

La portion de "monde réel" qui sera implémentée par chaque composants métier a été décrite à l'aide de modèles de classes d'analyse, et les comportements et services attendus de ces composants ont été définis par des diagrammes de contexte et des schémas d'opérations. Il s'agit à présent de concevoir un modèle du système informatisé qui soit à même de remplir ces services et de déterminer la manière dont doivent collaborer les objets informatisés, au sein des composants, pour accomplir les opérations demandées.

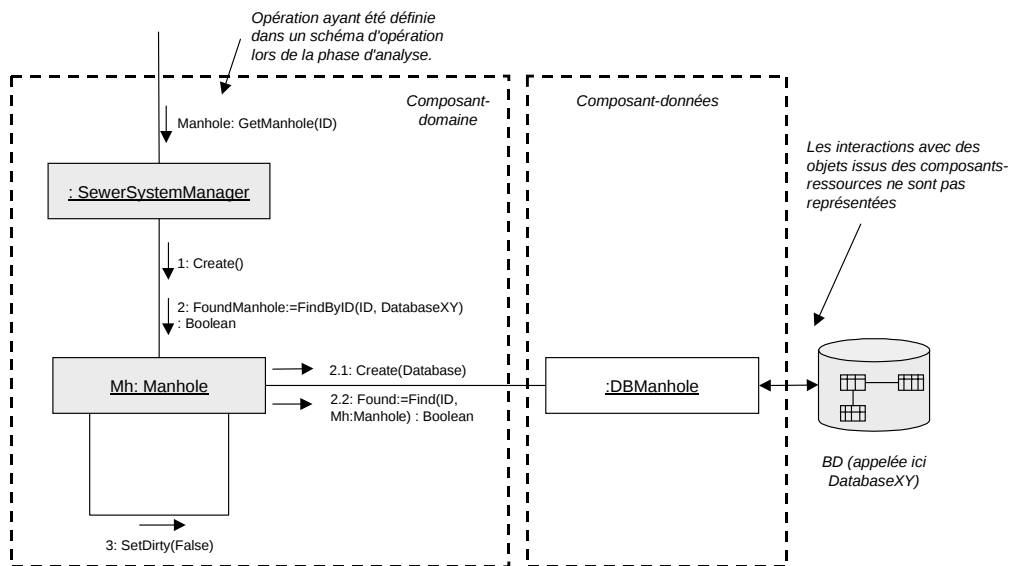


Figure 89. Exemple de diagramme de collaboration relatif à l'opération de recherche d'une chambre normée à partir de son identifiant.

L'étape de conception débute par l'élaboration de diagrammes qui montrent les interactions qui doivent avoir lieu entre les objets pour que les opérations demandées puissent être menées à bien. Chaque diagramme a pour base l'un des schémas d'opération décrits à la phase précédente. Les diagrammes d'interaction peuvent être des diagrammes de collaboration (figure 89) ou des diagrammes de séquence (figure 57). À partir du diagramme présenté à la figure 89 sont discutés les points suivants :

- la responsabilité de chaque opération doit être attribuée à une classe particulière. Chaque opération sera trouvée sur l'une des interfaces publiques de ces classes. Cette phase d'attribution de responsabilités est extrêmement importante dans le contexte de composants métier réutilisables. En effet, d'une part l'utilisateur de tels composants doit pouvoir facilement trouver les opérations dont il a besoin, et l'attribution de ces responsabilités doit répondre à la logique du domaine. D'autre part, une fois qu'une opération fait partie d'une interface et que cette dernière est publiée, il n'est plus possible de modifier cette interface, et pas judicieux d'attribuer la responsabilité de l'opération également à une autre interface.
- la phase de conception conduit à la proposition de nouvelles classes (classe "Gestionnaire de réseau

d'assainissement" par exemple) qui ne sont pas issues de modélisations de concepts réels du domaine de la gestion des eaux, mais permettent au système de mener à bien les opérations demandées. Cela conduit à construire un modèle de classes de conception. Alors que le modèle de classes d'analyse représentait essentiellement des concepts "réels" du domaine de la gestion des eaux, le modèle de classes de conception décrit des abstractions informatiques.

- lors de la construction des diagrammes d'interaction, le concepteur doit garder à l'esprit les objectifs d'une approche par composants et éviter les interactions directes entre instances de classes appartenant à des paquetages différents. Pour cela, il est possible de mettre en oeuvre avec profit des modèles de conception (Gamma et al., 1995; Larman, 1997), et les concepts de façade et de médiateur, présentés aux chapitres 4.5.4, doivent être gardés à l'esprit.
- la même opération conduit à de nombreuses interactions entre différents objets. Un composant doit rester simple d'utilisation : il convient donc de décider de la visibilité ou non des différentes classes de conception et de leurs opérations depuis l'extérieur du composant métier. Rappelons que le composant métier doit proposer des interfaces qui ensemble forment un modèle facile à comprendre pour le spécialiste du domaine de la gestion des eaux.

Figurent ainsi dans un modèle de classes de conception non seulement les attributs mais aussi les opérations dont les différentes classes ont la responsabilité, leur visibilité, et la navigabilité des relations. Il peut de plus être judicieux de séparer, au sein même d'un composant, les interfaces de leurs classes d'implémentation, et de rajouter des interfaces (par exemple une interface propre aux classes se rapportant à des entités du monde réel qui peuvent faire l'objet d'interventions humaines, voir figure 83), d'une part dans un but de bonne conception, mais aussi pour regrouper en ensembles cohérents les services proposés par les composants métier.

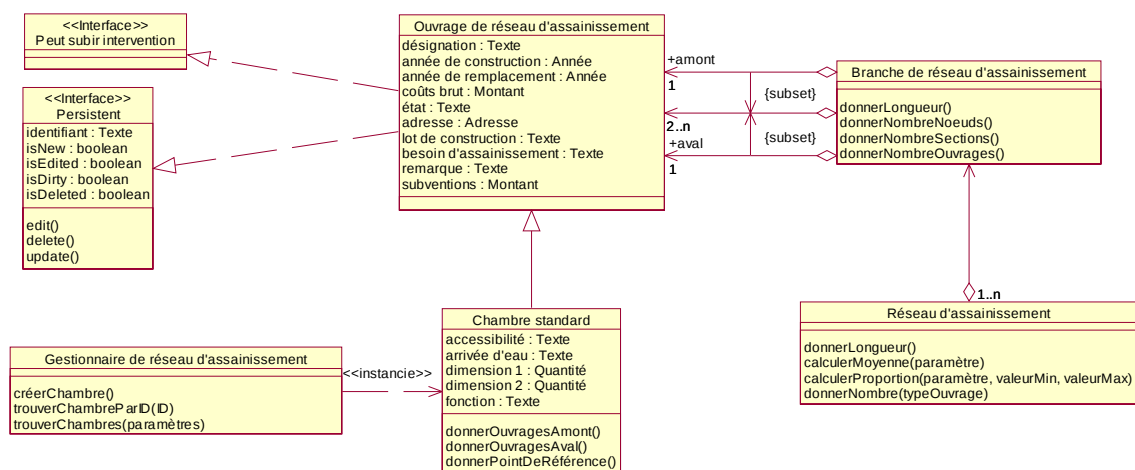


Figure 90. Extrait du modèle de classes de conception.

Une fois les modèles de classes de conception relatifs aux composants en cours de développement élaborés, les classes définies peuvent être décrites par le biais de leurs interfaces, spécifiées dans le langage qui sera utilisé lors de l'implémentation : Java, Corba IDL, etc. Les interfaces publiques du composant métier, qui sont d'un intérêt tout particulier dans le contexte de ce travail puisque ce sont elles qui seront visibles des utilisateurs de composants métier, servent à la manipulation des objets métier, et permettent l'interopérabilité des composants métier, doivent faire l'objet d'une documentation séparée.

Nous verrons au chapitre 7.2 de quelle manière nous proposons de gérer ces interfaces publiques.

C'est également au niveau de la conception que doivent être définis les modèles de persistance des différents composants métier : parmi les objets métier, il s'en trouve dont la persistance doit être assurée par une BD ou un système de fichiers. Le composant données se base sur un modèle logique de données. Généralement, lors de la conception de bases de données, trois niveaux de modélisation des données sont considérés : les niveaux conceptuels, logiques et physiques (Spaccapietra, 1996). À l'heure actuelle, le modèle logique de données est généralement de type relationnel. Nous proposons de passer directement du modèle de classes de conception au modèle logique de données : les modèles objet et les modèles conceptuels de données, en particulier les évolutions récentes du modèle entité-relation (Spaccapietra et Parent, 1992), tout deux de niveau conceptuel, sont très proches, et l'on gagne donc du temps à passer directement d'un modèle objet à un modèle logique de données. Quantité d'auteurs se sont penchés sur la problématique de la correspondance entre modèles objet et relationnel, et proposent des règles de traduction du premier vers le second (Ambler, 1998b; Gupta et Scumniotales, 1998; Rational, 2000; Yoder et al., 1998). Notons cependant que la gestion de la persistance des données spatiales et temporelles peut devoir être abordée différemment, lorsque les BD relationnelles qu'il est prévu de mettre en oeuvre ne permettent pas la gestion de tels types de données.

La phase de conception présentée ci-dessus ne diffère pas fondamentalement de celles proposées par d'autres méthodes. Il faut cependant être attentif aux points suivants : 1°) le modèle de classes d'analyse doit autant que possible être respecté lors de sa transposition en modèle de classes de conception afin que les concepts dont le composant métier permettra la manipulation soient autant que possible conformes à la réalité du spécialiste du domaine de la gestion des eaux; 2°) le nombre d'interfaces se rapportant à des concepts absents du modèle de classes d'analyse "visibles" depuis l'extérieur du composant doit être réduit au strict minimum, et 3°) la volonté d'assurer une indépendance maximale aux composants doit guider les développeurs durant la phase de conception. Ces trois précautions facilitent la conception de composants qui soient réellement réutilisables et faciles à mettre en oeuvre.

Font partie des documents livrés à l'issue de cette phase, pour chacun des paquetages :

- un modèle objet des classes de conception, accompagné des interfaces de chacune de ces classes et des règles de gestion que les implémentations de ces interfaces devront respecter;
- un document décrivant les interfaces qui seront proposées par le composant (interfaces publiques);
- un modèle logique de données.

5.3.2.8 Implémentation de composants métier

Lors de cette phase, les différents modèles et documents générés à l'étape de conception sont traduits dans le langage de programmation choisi. C'est l'étape de transformation du paquetage : tous les modèles sont codés, puis compilés, donnant ainsi naissance aux composants constituant le composant métier lui-même (figure 91). La méthode proposée ne présentant pas de spécificité à ce niveau-là, nous renvoyons le lecteur à la littérature spécifique.

5.3.2.9 Test et documentation de composants métier

Les composants métier sont conçus pour être réutilisés dans de nombreuses applications. Ils doivent de

ce fait être l'objet d'une phase de test approfondie. De plus, après leur diffusion, leur implémentation peut devoir être soumise à des modifications et des améliorations. Nous proposons par conséquent que soit développé un applicatif qui serve à tester l'ensemble des fonctionnalités offertes par les interfaces publiques de chaque composant métier délivré. L'expérience nous a montré que les gains de temps qui résultent de l'utilisation d'un tel outil de test, ainsi que l'amélioration de la qualité des composants métier qui s'ensuit, justifient amplement son développement.

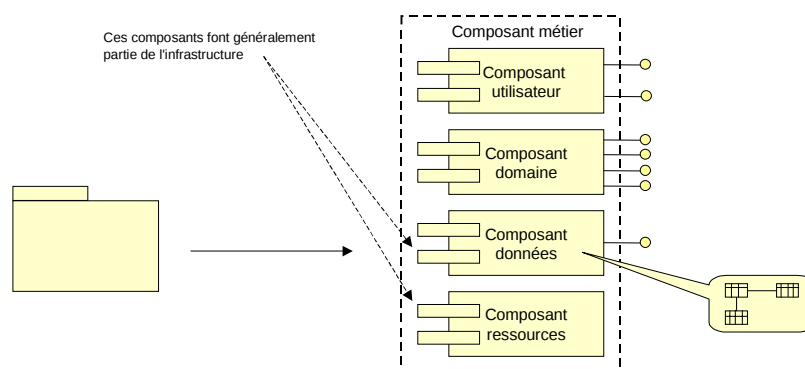


Figure 91. À l'issue de la phase d'implémentation, le composant est développé, prêt à être testé puis intégré dans un SIAGEM et à interagir avec une BD.

Afin que les utilisateurs soient à même de mettre en oeuvre les composants métier délivrés, il faut également joindre à ceux-ci une documentation soignée des services offerts par leurs différentes interfaces. Cette documentation doit comprendre : une vue d'ensemble du modèle objet implémenté par le composant métier, une description du modèle logique de données sur lequel il s'appuie, une description des différents services proposés par les interfaces, ainsi que des exemples d'utilisation. Enfin, toute la documentation générée durant les différentes phases du processus de développement doit être archivée, car elle sera indispensable à la maintenance du composant métier.

5.4 Synthèse

D'une part la construction de composants logiciels de qualité et réellement réutilisables nécessite l'adoption d'une démarche rigoureuse, et d'autre part le domaine de la gestion des eaux présente un certain nombre de particularités, présentées au chapitre 3, qui influencent la manière de développer de tels composants. C'est pourquoi nous proposons ici une méthode de développement, baptisée COWIS, axée sur le développement de composants réutilisables et interopérables, prenant en compte la notion de territoire et de ses éléments, et autorisant le développement indépendant et en des cycles relativement courts desdits composants. Bien sûr COWIS n'a pas été créée *ex nihilo*, mais s'appuie sur des méthodes préexistantes. En particulier les phases habituelles du cycle de développement ont été suivies, et des formalismes existants adoptés (figure 92). Les particularités de COWIS sont :

- un partitionnement du problème effectué en début de processus, de manière à pouvoir développer des composants métier indépendamment les uns des autres et en des temps relativement courts;
- l'adoption d'un référentiel commun, le modèle de domaine, qui assure la cohérence entre les différents composants métier;
- une réflexion élargie, en début de processus, en regard des objectifs immédiats auxquels devra

répondre le composant métier, de manière à anticiper de futures utilisations et à maximiser son potentiel de réutilisation;

- un suivi, tout au long du processus de développement, des concepts réels issus du domaine de la gestion des eaux, dont des modèles se retrouvent à chaque étape. Cela débouche sur des composants d'utilisation plus intuitive;
- une attention particulière prêté aux services, matérialisés en fin de processus par des interfaces publiques, que proposent les composants métier. Ces interfaces permettent l'interopérabilité entre composants métier, et ne doivent, une fois publiées, plus faire l'objet de modifications.

Analyse des besoins	Analyse de domaine	Analyse composant	Conception composant
• cas d'utilisation réels étendus • documents d'analyse des besoins • liste préliminaire des concepts • modèle des cas d'utilisation • cas d'utilisation réels résumés	• modèle de domaine et glossaire • liste des paquetages	• cas d'utilisation-systèmes développés • modèle des "paquetages" métier • modèles de classes d'analyse • diagrammes de contexte • schémas d'opérations	• modèle de classes de conception • description des interfaces publiques • modèle logique de données

Figure 92. Documents issus des quatre phases de la méthode COWIS présentant les particularités les plus marquées.

Notre proposition implique certes un travail important lors des phases initiales du processus de développement. Cependant, il faut bien voir que si la construction de composants métier de qualité nécessite un investissement plus élevé qu'une application "classique" (toutes choses étant égales par ailleurs), de tels composants peuvent être réutilisés dans de nombreuses applications, et les frais de maintenance des applications construites par assemblages de composants grandement diminués : des composants soigneusement conçus présentent peu de déficiences, et un composant ayant fait l'objet d'une amélioration pourra être modifié en conséquence et facilement remplacé dans les applications le mettant en oeuvre. Comme les frais de maintenance représentant à l'heure actuelle la majeure partie des dépenses dans le cycle de vie d'une application logicielle (figure 93), une approche par composants métier conduit à une réduction des coûts totaux résultant de l'informatisation de la gestion des eaux.

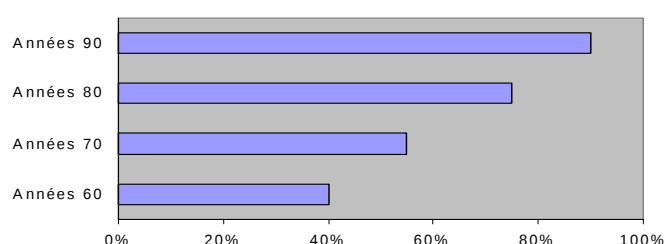


Figure 93. Proportion des coûts de maintenance dans le cycle de vie du logiciel, d'après (Datamation, 1990, cité dans Strohmeier, 1999) : ces coûts sont proportionnellement de plus en plus importants.

L'approche proposée distingue clairement les interfaces publiques, proposées par les composants, des composants eux-mêmes. Son but est de favoriser l'interopérabilité de composants, qui ne sont pas forcément conçus par des équipes similaires. Comment organiser la production et la gestion de ces composants métier ? Quelles institutions doivent être impliquées ? Où intervient la notion de standardisation dans un tel processus ? Ces questions seront abordées au chapitre 7.2.

Mise en oeuvre du concept de composant métier

Les réflexions présentées aux deux chapitres précédents se sont appuyées sur le développement de prototypes. Le présent chapitre donne un bref aperçu de ces développements. Un prototype de composant métier relatif au réseau d'assainissement a été construit, puis intégré dans un outil logiciel destiné à la gestion des données se rapportant aux réseaux d'assainissement. Une synthèse des commentaires et impressions suscités par ces produits chez différents intervenants clôture ce chapitre.

6.1 Contexte

Un composant métier de type entité, destiné à faciliter la gestion de l'information relative au réseau d'assainissement, a été construit. Cette démarche a servi à la mise au point des concepts et de la méthode proposés et autorisé une première validation de cette dernière. L'intégration de ce composant dans un prototype d'outil logiciel nous a ensuite permis d'une part de vérifier la véracité des avantages attribués par la littérature à une approche par composants métier (développement rapide, facilité de mise en oeuvre, qualité des applications développées), et d'autre part de disposer d'un support concret dans les discussions avec des spécialistes du domaine de la gestion des eaux.

En ce qui concerne la méthode de développement COWIS, il a été vérifié : 1°) que les phases proposées sont adéquates; 2°) qu'elles s'enchaînent logiquement; 3°) que les étapes recommandées à chacune des phases mènent au résultat désiré; 4°) que le volume de travail impliqué par ces différentes phases n'est pas disproportionné en regard du résultat obtenu; 5°) que l'ensemble du processus conduit effectivement à la conception de composants logiciels réutilisables; 6°) et ce, dans des délais raisonnables.

Le présent chapitre se compose de trois parties. Dans la première, nous présentons le développement du composant métier selon la méthode COWIS. Dans la deuxième, nous présentons l'outil logiciel que nous avons construit et qui intègre ce composant métier. Enfin, nous clôturons ce chapitre par une discussion des enseignements que nous avons pu tirer, tant de la construction et l'utilisation de ces prototypes que de discussions tenues avec différents intervenants après leur avoir présenté ou après avoir mis à leur disposition ces développements.

Nous soulignons ici le fait que nos prototypes ne reflètent pas nécessairement les choix adéquats pour une application opérationnelle. D'une part, en effet, ils ont servi à tester des concepts et mettre en place une méthode, et non à concevoir un produit directement utilisable; d'autre part, des choix techniques ont été faits dans le but d'accélérer le processus de développement et non d'aboutir à un composant commercialisable.

6.2 Réalisation d'un prototype de composant métier

6.2.1 Spécification des caractéristiques et mise en place de l'infrastructure

Les aspects techniques relatifs à la conception de composants métier distribués sont déjà abordés par différents auteurs (Eeles et Sims, 1998; Herzum et Sims, 2000; Szyperski, 1997). En conséquence, et pour simplifier notre démarche, il a été décidé de développer un composant métier destiné à la mise en place d'applications mono-poste.

Les outils logiciels (SIG, modèles de comportement, etc) utilisés dans le contexte de la gestion des eaux fonctionnent généralement sous les systèmes d'exploitation Windows (95, 98, NT, 2000). Il n'a donc pas été nécessaire, dans le contexte de notre travail, de tenir compte des problèmes d'interopérabilité entre plate-formes informatiques. De plus, les intervenants susceptibles de donner un avis sur les concepts proposés étaient plus familiers des environnements COM (annexe 3) que des environnements Java ou Corba. Nous avons par conséquent choisi de nous appuyer sur COM, un environnement pour composants spécifiquement destiné aux différents systèmes Windows.

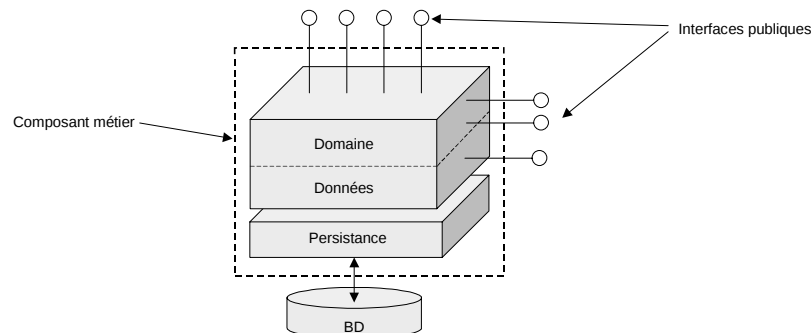


Figure 94. Structure du prototype de composant métier développé. Les composants domaine et données sont agrégés.

Le composant métier proposé aura pour mission l'acquisition, la validation et l'édition d'informations relatives aux réseaux d'assainissement (chapitre 6.2.2). De telles fonctions impliquent d'assurer la persistance de données, et une infrastructure de persistance a dû être définie : elle a été basée sur le modèle "DB Object" (annexe 7). Selon un tel modèle, à chaque classe du domaine (implémentée par le composant domaine) correspond une classe du composant données chargée d'assurer la correspondance entre les objets du domaine et les données de la BD. Cette solution constitue un bon compromis entre des composants métier basés sur le modèle "Persistent Object", peu modulaires, difficiles à maintenir, mais très rapidement construits, et des composants métier basés sur le modèle "Broker Object", très

souple, très modulaire, mais dont la mise en place aurait nécessité trop de temps en regard de l'objectif fixé.

Pour simplifier la mise en oeuvre du composant métier, les composants constituant domaine et données ont été agrégés (figure 94). Enfin, il a été décidé de ne pas intégrer au composant métier de couche d'interface-utilisateur.

Les phases de spécification des caractéristiques et de mise en place de l'infrastructure ont été présentées ci-dessus de manière très succincte. Il va de soi qu'en réalité une réflexion beaucoup plus poussée serait nécessaire, la construction ou l'adaptation d'une infrastructure pour composants constituant à elle seule un véritable projet informatique.

6.2.2 Analyse des besoins

Les besoins pris en compte dans le cadre du présent cycle de développement sont ceux de l'un des intervenants mentionnés au tableau 1 (chercheur dans une institution fédérale). Ce dernier procède, dans le cadre de son activité professionnelle, à de nombreuses simulations des écoulements dans les réseaux d'assainissement, entre autres dans le but de déterminer les tronçons sous-dimensionnés. Ces simulations nécessitent l'acquisition de nombreuses données, relatives tant aux infrastructures d'assainissement qu'aux surfaces contributives, à la pluviométrie et aux débits dans les canalisations.

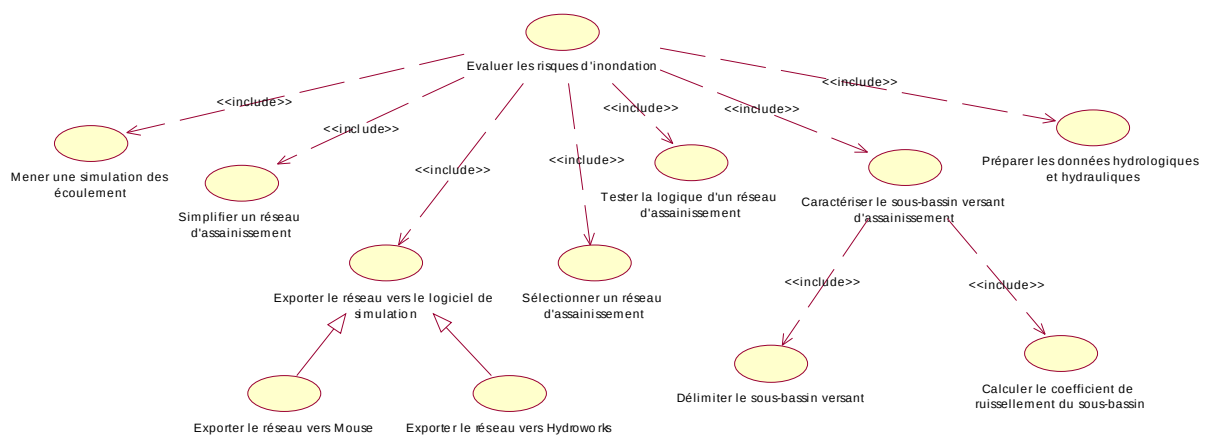


Figure 95. Modèle des cas d'utilisation se rapportant au processus métier "Évaluer les risques d'inondation". Notons que les niveaux d'abstraction des différents cas d'utilisation sont variables.

Afin de mieux comprendre les processus métier relatifs à cet intervenant, des cas d'utilisation réels étendus ont été identifiés et rédigés (quatre d'entre eux figurent à l'annexe 10) : il ne s'agit pas de développer à cette étape l'ensemble des cas d'utilisation déterminés, mais uniquement ceux qui sont nécessaires à une bonne compréhension du processus métier. Celui qui nous intéresse ici ("Évaluer les risques d'inondation") peut être décomposé en un grand nombre de cas d'utilisation interreliés par des relations d'inclusion ou de dépendance (figure 95). Tout ces cas d'utilisation ne se situent pas au même niveau d'abstraction : celui qui décrit le processus métier "de base" recouvre un vaste ensemble d'activités de haut niveau. De sa décomposition résultent des cas d'utilisation de niveau d'abstraction moindre, qui peuvent recouvrir une succession d'activités clairement définies (cas d'utilisation

"Simplifier un réseau d'assainissement" par exemple), ou encore dont l'informatisation conduirait à une fonction unique (cas d'utilisation "Sélectionner un réseau d'assainissement" par exemple). À partir des cas d'utilisation peuvent être identifiés les concepts fréquemment manipulés. Ceux qui se rapportent aux éléments de réseau d'assainissement (canalisation, ouvrage d'assainissement, etc.) ainsi qu'aux surfaces contributives (sous-bassins versants d'assainissement, etc.) comptent parmi les plus fréquents. Précisément, les concepts de réseau d'assainissement, branche de réseau d'assainissement, ouvrage de réseau d'assainissement, chambre, ouvrage spécial, exutoire, canalisation, bassin versant, sous-bassin versant d'assainissement, zone de construction, bâtiment, station de mesure, sol, toit, voie de circulation, débit, habitant et événement pluvial sont identifiés. Nous décidons dans la suite de cet exemple de nous focaliser sur le réseau d'assainissement.

La matrice processus-système (figure 74) est ensuite mise en oeuvre : les processus métier qui conduisent à manipuler ces mêmes concepts, ainsi que les acteurs concernés, sont identifiés, représentés sous forme de modèle de cas d'utilisation (voir figure 74) et décrits par des cas d'utilisation réels résumés (voir figure 75). Ainsi sont identifiées les catégories d'intervenants qui devront être prises en compte pour que l'informatisation relative au sous-système "réseau d'assainissement" conduise à des composants logiciels qui leur soient utiles à eux aussi.

A la fin de la présente étape, une bonne compréhension des besoins à considérer et du processus métier à soutenir est acquise. Les processus métier qui présentent un cadre de réutilisation potentiel pour le composant métier en voie de développement ont également été identifiés.

6.2.3 Analyse et segmentation de domaine

A partir des cas d'utilisation réels résumés et étendus définis à l'étape précédente, d'entretiens supplémentaires avec quatre représentants des acteurs concernés (trois intervenants issus de bureaux d'ingénieurs chargés d'effectuer des modélisations et d'élaborer des PGEE, et un intervenant concerné par la gestion du cadastre de l'assainissement), de documents de travail fournis par ces mêmes intervenants, et de recommandations éditées par des associations professionnelles (VSA, 1992), un modèle de domaine a été élaboré. Des vues sur ce modèle de domaine ont été présentées aux figures 80 et 81. La segmentation de ce domaine est issue du regroupement des classes de domaines identifiées autour de concepts clé (figure 80). Précisons que le but n'est pas d'être exhaustif, mais d'obtenir un modèle préliminaire des aspects structurels du domaine. Les différentes classes de domaine identifiées, regroupées au sein de paquetages, doivent être décrites dans un glossaire qui sera complété tout au long du processus de développement.

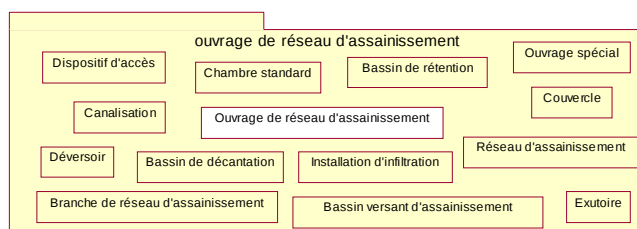


Figure 96. Le paquetage "ouvrage de réseau d'assainissement" et les classes de domaine identifiées jusqu'ici. "Ouvrage de réseau d'assainissement" est la classe-clé de ce paquetage.

Les étapes suivantes focaliseront sur le concept-clé "Ouvrage de réseau d'assainissement" et les concepts qui gravitent autour de celui-ci (figure 96). L'informatisation de cas d'utilisation comme ceux modélisés à la figure 95 peut donner naissance à des composants métier de type processus. Ces derniers doivent pouvoir s'appuyer sur des composants de type entité pour fonctionner. Les premiers cycles de développement conduisent ainsi à des composants métier de type entité, tandis que les cas d'utilisation sont réutilisés dans les cycles suivants pour servir de base à la conception de composants de type processus.

6.2.4 Analyse orientée composant

Les composants-entité permettent principalement de gérer les données relatives aux concepts implémentés, mais peuvent également offrir des services à condition que ceux-ci ne nécessitent pas le recours à des informations fournies par des composants métiers tiers. Des cas d'utilisation-système sont ici élaborés, qui montrent comment l'utilisateur sera en mesure d'interagir avec le composant en cours de développement (par l'intermédiaire d'une interface graphique). Les cas d'utilisation principaux sont dans le cas de composants-entité ceux qui permettent la gestion de l'information (créer, supprimer et modifier un ouvrage de réseau d'assainissement). Deux exemples de tels cas figurent à l'annexe 10.

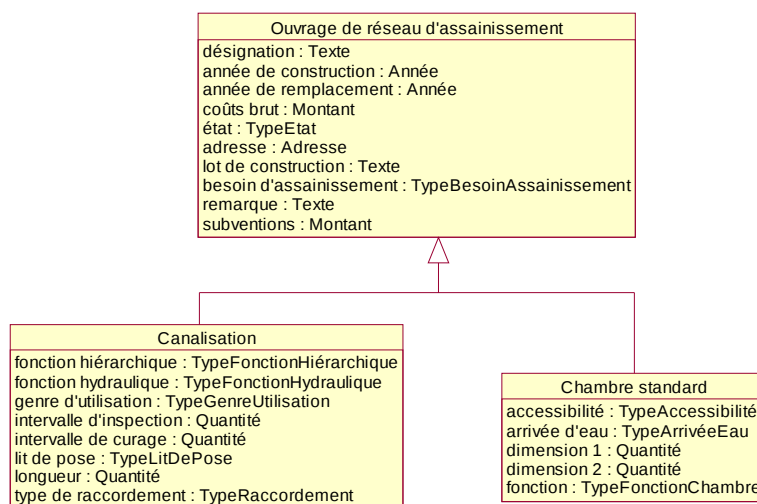


Figure 97. Extrait du modèle de classes d'analyse relatif au paquetage "Ouvrage de réseau d'assainissement". Les types d'attributs dont le nom commence par "Type" sont des domaines de valeurs.

Le modèle de classes d'analyse proposé en ce qui concerne le paquetage en cours d'étude a été présenté à la figure 80. Le glossaire doit être complété par les classes d'analyse ainsi que les règles de gestion de type invariant déterminées. La figure 97 et le tableau 10 montrent respectivement un extrait du modèle de classes d'analyse et du glossaire. Au niveau du prototype, il n'a pas été tenu compte de la possibilité de segmentation (figure 80) entre éléments du réseau proprement dit et ouvrages d'assainissement.

Classe : Canalisat	
Définition	Canalisation ouverte ou fermée dont la fonction est de conduire les eaux à évacuer d'un ouvrage du réseau d'assainissement à l'autre.

Tableau 10. Extrait du glossaire relatif aux classes d'analyse. Il est recherché une conformité maximale aux recommandations de l'ASPEE (VSA, 1999).

Règles générales	• Une canalisation possède au moins un tronçon. • Une canalisation est liée à au maximum un ouvrage d'assainissement à l'amont. • Une canalisation est liée à au maximum un ouvrage d'assainissement à l'aval.	
Attribut	fonction hiérarchique	
Définition	fonction de la canalisation dans le système d'assainissement	
Type	TypeFonctionHiérarchique : {conduite de raccordement, collecteur principal, collecteur, conduite d'assainissement, autres}	
Attribut	fonction hydraulique	
Définition	fonction hydraulique de la canalisation	
Type	TypeFonctionHydraulique : {conduite de drainage, conduite d'étranglement, siphon inversé, conduite à écoulement gravitaire, conduite de refoulement, conduite de rétention, conduite de rinçage}	
Attribut	genre d'utilisation	
Définition	type d'eau à évacuer	
Type	TypeUtilisationCanalisation : {eaux industrielles, eaux mixtes, eaux pluviales, eaux propres, eaux usées}	
Attribut	intervalle d'inspection	
Définition	intervalle de temps qui devrait s'écouler entre deux inspections, en général exprimée en années	
Type	Quantité	
Règle	est compris entre 1 mois et 100 ans	
.....		

Tableau 10. Extrait du glossaire relatif aux classes d'analyse. Il est recherché une conformité maximale aux recommandations de l'ASPEE (VSA, 1999).

Enfin, les opérations dont le composant métier aura la responsabilité doivent être précisées : déterminées à partir des cas d'utilisation, elles doivent aussi figurer dans le glossaire et peuvent être résumées dans un diagramme de contexte du système (le système étant ici le paquetage considéré) semblable à celui présenté à la figure 88.

Opération	Description
créer_ouvrage	Créer un ouvrage de réseau d'assainissement du type spécifié.
obtenir_ouvrage	Rechercher un ouvrage identifié de manière univoque.
obtenir_ouvrages	Rechercher des ouvrages à partir des critères spécifiés.
mettre_à_jour	Mettre à jour les attributs de l'ouvrage.
supprimer	Supprimer l'ouvrage et toutes ses composantes.
ajouter_noeud	Ajouter un noeud de réseau à l'ouvrage spécifié.
ajouter_section	Ajouter un tronçon de réseau à l'ouvrage spécifié, connecté à l'amont au noeud amont et à l'aval au noeud aval.
obtenir_ouvrages_amont	Rechercher les ouvrages situés immédiatement en amont de l'ouvrage spécifié.
obtenir_ouvrages_aval	Rechercher les ouvrages situés immédiatement en aval de l'ouvrage spécifié.
obtenir_noeud_amont	Rechercher le noeud situé immédiatement en amont du tronçon spécifié.

Tableau 11. Extrait du glossaire reportant les opérations identifiées à partir des cas d'utilisation.

Opération	Description
obtenir_noeud_aval	Rechercher le noeud situé immédiatement en aval du tronçon spécifié
obtenir_tronçons_amont	Rechercher les tronçons situés immédiatement en amont du noeud spécifié.
obtenir_tronçons_aval	Rechercher les noeuds situés immédiatement en aval du noeud spécifié.
obtenir_longueur	Calculer la longueur totale de la canalisation spécifiée.
obtenir_longueur	Calculer la longueur totale du tronçon spécifié.
obtenir_longueur	Calculer la longueur totale de canalisations du réseau d'assainissement spécifié.
obtenir_moyenne	Obtenir la moyenne relative au paramètre spécifié pour un réseau d'assainissement donné.
obtenir_proportion	Obtenir la proportion de canalisations (par rapport à la longueur) obéissant au critère spécifié et appartenant au réseau donné.

Tableau 11. Extrait du glossaire reportant les opérations identifiées à partir des cas d'utilisation.

Chacune des opérations est détaillée dans un schéma d'opération (tableau 12).

Schéma d'opération	
Nom	CalculerAgeMoyen(type d'ouvrage : TypeOuvrage, réseau : RéseauAssainissement)
Description	L'opérateur demande au système le calcul de l'âge moyen d'un certain type d'ouvrage du réseau d'assainissement
Acteur	Opérateur
Événements	Opérateur :: {ÂgeMoyen; Erreur}
Pré-conditions	- Un réseau est identifié - L'attribut "année de construction" des ouvrages est renseigné
Post-conditions	Si le calcul est possible - donne l'âge moyen des ouvrages correspondant au type demandé, en fonction du nombre d'ouvrages de ce type sinon - explique la cause de l'impossibilité du calcul

Tableau 12. Schéma de l'opération de calcul de l'âge moyen d'un réseau d'assainissement.

Ce qui est attendu de chaque composant métier en gestation est à présent décrit de manière précise. Les étapes ultérieures de la méthode s'attachent à la manière de mener à bien les services demandés.

6.2.5 Conception orientée composant

La conception est l'étape du processus à laquelle l'informatisation débute. Pour des raisons de diffusion, il a été décidé de développer le composant métier en utilisant une terminologie anglophone en ce qui concerne les noms de classes, d'attributs, et d'opérations.

Le premier pas est l'attribution de la responsabilité de chaque opération reportée dans le glossaire (tableau 11) à une classe particulière. Des modèles d'attribution de responsabilités doivent ici être mis en oeuvre. Par exemple :

- le modèle "Expert" (Larman, 1997) recommande d'attribuer la responsabilité d'une opération à la classe qui dispose de l'information nécessaire. L'opération "obtenir_moyenne" (tableau 11), par exemple, peut raisonnablement être attribuée à la classe "Réseau d'assainissement", puisque les

instances de cette classe ont accès à l'information nécessaire.

- le modèle "Créateur" recommande qu'une instance de classe B soit responsable de la création d'instances de la classe A si : 1°) B contient des instances de A; 2°) B agrège des instances de A; 3°) B possède les données nécessaires à l'initialisation d'instances de A; 4°) B utilise intensivement des instances de A. Une opération d'ajout de noeud de réseau ou d'élément d'ouvrage (couvercle, etc.) peut selon ce modèle raisonnablement être attribuée à la classe "Ouvrage de réseau d'assainissement" elle-même.

L'étape de conception conduit fréquemment à la création de classes de conception n'ayant pas d'équivalent au niveau des classes d'analyse : une classe "SewersystemManager" a par exemple été créée et chargée des opérations de recherche et de création d'ouvrages de réseau, cette responsabilité ne pouvant être attribuée à aucune autre classe.

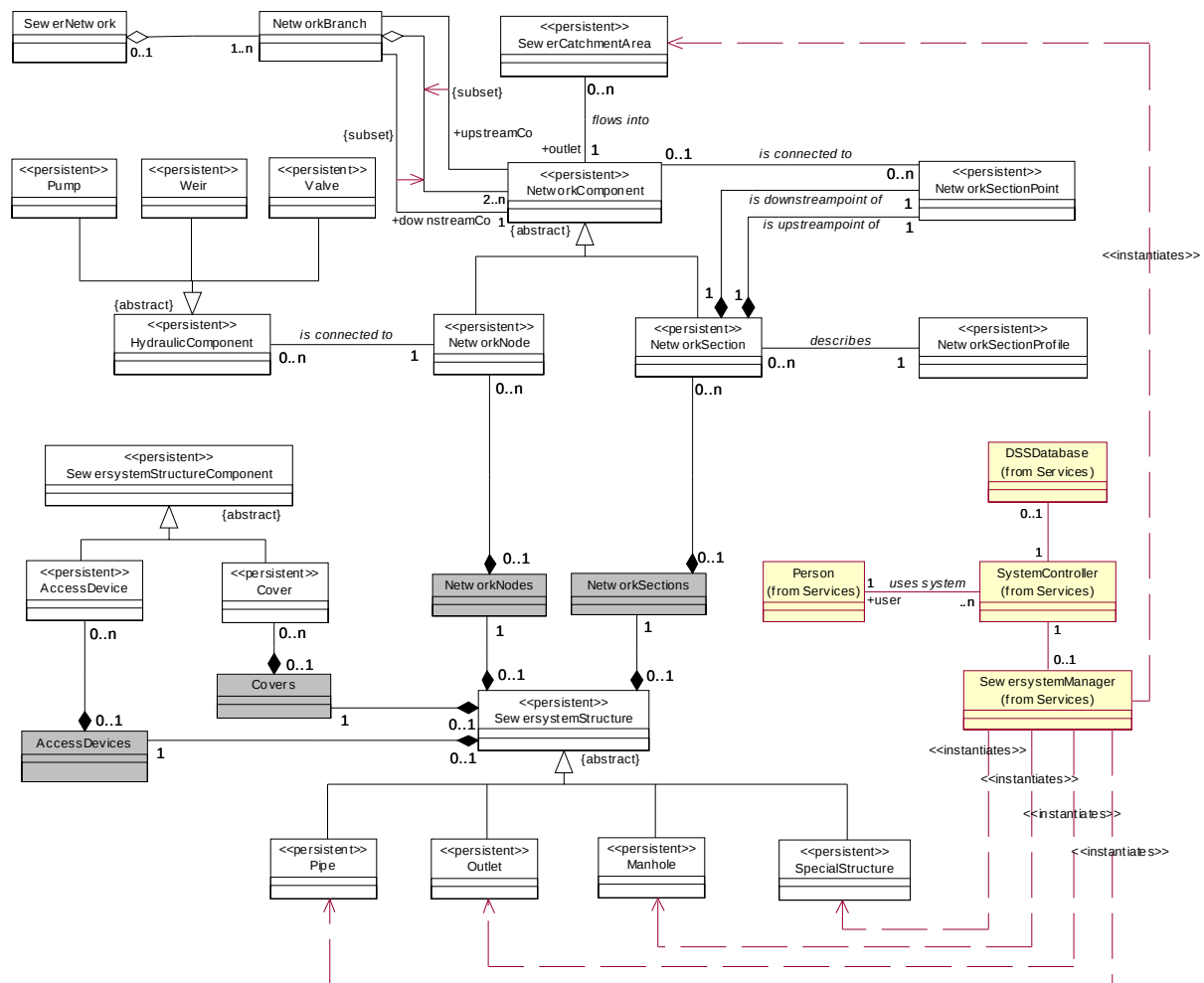


Figure 98. Partie "publique" du modèle de classes de conception. Il peut être constaté que les classes issues de l'analyse de domaine (en blanc) sont majoritaires, l'étape de conception ayant conduit à l'ajout de classes de type collection (en gris) et de classes permettant le contrôle du système informatique (en jaune). Les stéréotypes "persistant" montrent les classes dont la persistance doit être assurée.

Une fois les responsabilités d'opérations attribuées, la manière dont ces opérations sont effectuées doit être déterminée. C'est à cet effet principalement la technique des diagrammes de collaboration qui a été

mise en oeuvre (voir exemple figure 89). Différentes classes techniques, dont celles issues des composants techniques faisant partie de l'infrastructure pour composants métier (chapitre 6.2.1) sont mises à contribution. De nombreuses classes servant à mener à bien les opérations demandées doivent être créées, mais restent invisibles de l'extérieur du futur composant métier, leurs interfaces n'étant pas exportées par ce dernier.

Sont créées des classes dont la manipulation par l'intégrateur présente un intérêt, même si elles ne sont identifiées qu'à cette étape de conception : les classes "Chambres standard" ou "Canalisations", par exemple, servent au regroupement d'ouvrages de type similaire. L'existence de telles classes se justifie, car il leur est attribué des responsabilités propres : l'utilisateur peut désirer obtenir une information globale sur de tels ensembles d'ouvrages, par exemple la longueur totale que représente un groupe de canalisations.

La figure 98 présente le modèle de classes de conception, dont les interfaces seront proposées par le composant métier développé : on y retrouve bien les concepts du monde réel issus des étapes d'analyse. On note également la présence de classes dont la connotation est liée au système informatique : une classe "SystemController", qui sert à initialiser le composant et à en contrôler les utilisateurs, et une classe "DSSDatabase", qui est une abstraction de la BD et sert à en régler les paramètres.

La figure 99 présente un extrait, plus détaillé, du modèle présenté en figure 98.

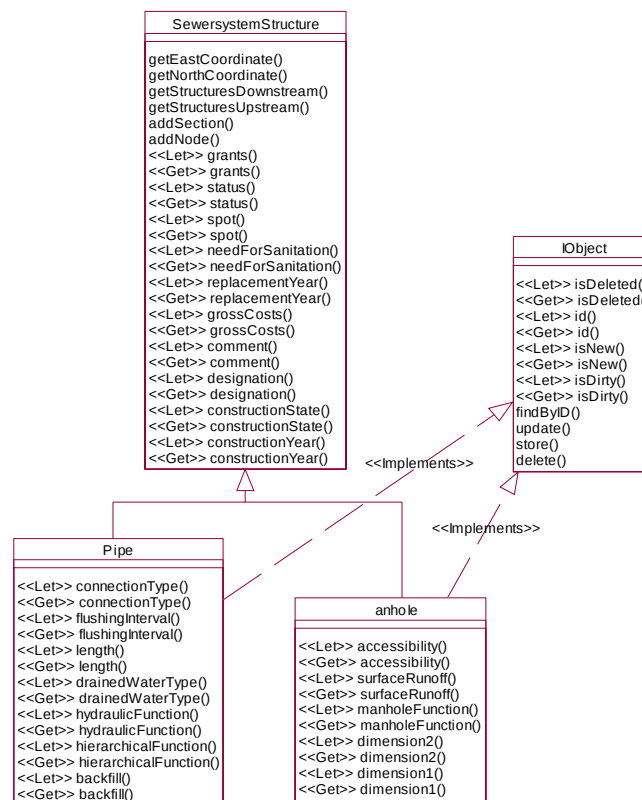


Figure 99. Extrait du modèle de classes "publiques" de conception (figure 98) correspondant au modèle de classes d'analyse de la figure 97. À noter l'absence d'attributs : ceux-ci sont accessibles uniquement par le biais d'opérations (stéréotypes <<Get>> et <<Set>>). Le principe d'encapsulation est ainsi respecté.

Les interfaces publiques qui assurent la compatibilité entre composants métier peuvent être définies au

cours de l'étape de conception : elles doivent être décrites de manière extrêmement précise, et sont susceptibles de faire l'objet d'un processus de standardisation (dans le but d'assurer l'interopérabilité entre composants) dont nous proposons une ébauche au chapitre 7.2. Ayant choisi l'environnement pour composants COM (chapitre 6.2.1), les interfaces publiques du composant développé peuvent être spécifiées en MIDL¹ (Microsoft Interface Definition Language). Le tableau 13 présente un exemple d'interface et de composant logiciel définis dans ce langage. Une telle description doit s'accompagner d'un glossaire, des schémas d'opérations et des règles de gestion qui permettent de préciser la sémantique de ces interfaces.

```
[object, uuid (133BCFF0-9B5B-11D3-80ED-00C04F43ABBB)]

interface ISewerSystemStructure : IUnknown
{
    HRESULT GetStructuresDownstream ([out, retval] ISewersystemStructures** );
    HRESULT GetStructuresUpstream ([out, retval] ISewersystemStructures** );
    AddSection ([in, out, optional] INetworkNode** ,[in, out, optional]
    INetworkNode** ,[out, retval] INetworkSection** );
    AddNode ([out, retval] INetworkNode** node);
    [propget] HRESULT Grants ([out, retval] double* );
    [propput] HRESULT Grants ([in] double );
    [propget] HRESULT Designation ([out, retval] BSTR* );
    [propput] HRESULT Designation ([in] BSTR );
    [propget] HRESULT ConstructionState ([out, retval] BSTR* );
    [propput] HRESULT ConstructionState ([in] BSTR );
    ...
};
...

// Exemple d'implémentation spécifique d'un composant SEWERSYSTEM

[uuid(C8BA9164-40F4-11D3-AF26-00104BF8778D),
 version (1.0)
]
library Sewersystem
{
    coclass Manhole
    {
        [mandatory] interface ISewersystemStructure;
        [mandatory] interface IPersistent;
        [default, mandatory] interface IManhole;
    };
    coclass Pipe
    {
        [mandatory] interface ISewersystemStructure;
        [mandatory] interface IPersistent;
        [default, mandatory] interface IPipe;
    };
};
}
```

Tableau 13. Exemple de spécification, en MIDL, de l'interface "SewersystemStructure" et d'un composant implémentant entre autres cette interface.

Nous avons également proposé que soit spécifié un modèle logique de données (MLD) pour chaque composant dont la persistance des objets doit être assurée. Ce MLD doit pouvoir être dérivé du modèle de classes de conception (notons à la figure 98 l'indication, par un stéréotype², des classes dont la persistance doit être assurée), il peut être influencé par des normes ou recommandations existantes. Des

1. http://msdn.microsoft.com/library/psdk/midl/mi-laref_1r1h.htm

outils AGL peuvent être mis à contribution pour générer, à partir du modèle de classes, le MLD.

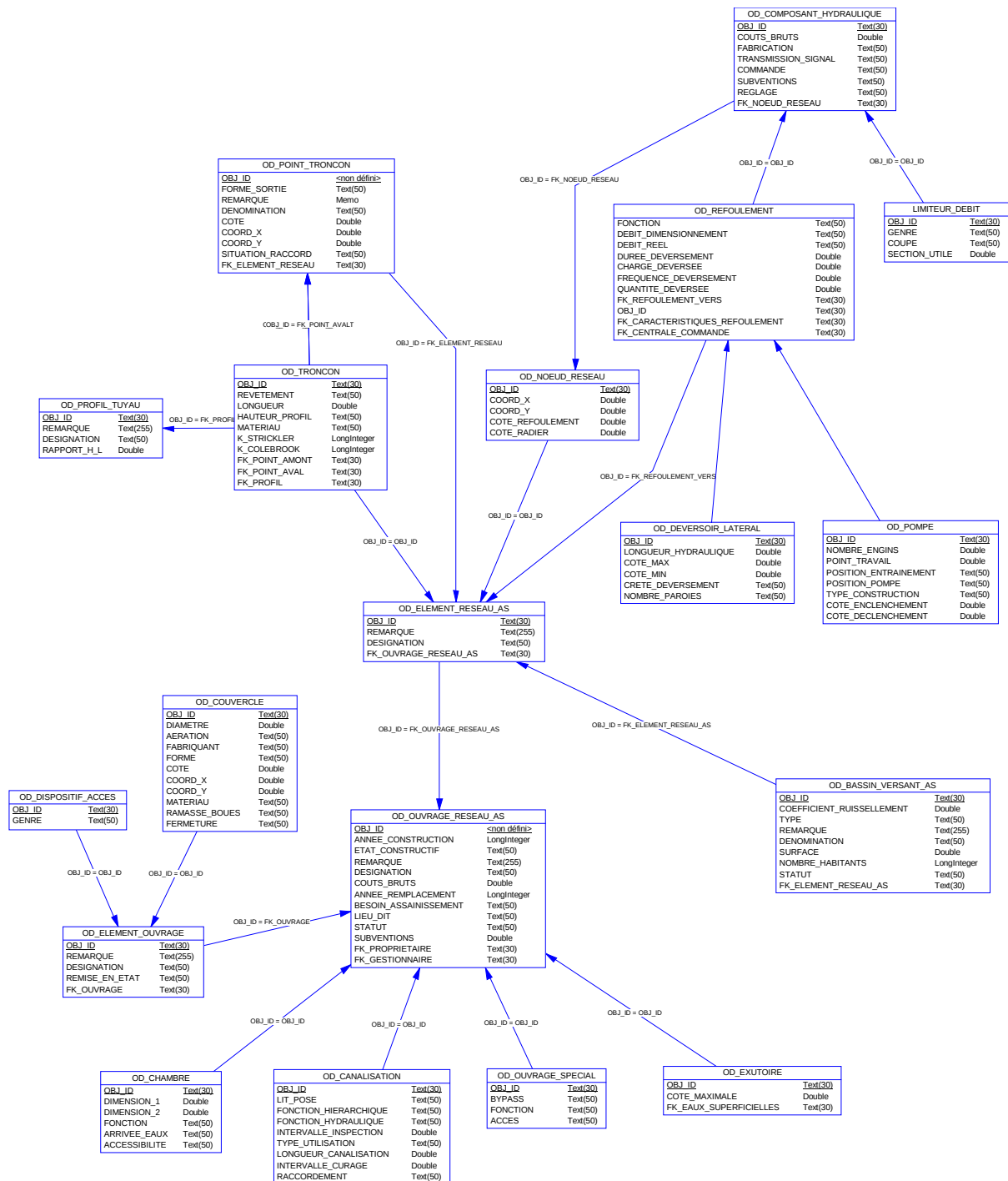


Figure 100. Modèle logique de données (relationnel) proposé. La persistance des objets métier peut être assurée par ce modèle.

Nous nous sommes ici fixé comme contrainte le respect de la structure de données proposée par l'ASPEE (VSA, 1999). Le modèle de données proposé ici (figure 100) constitue de ce fait un sous-ensemble de celui proposé par cette association professionnelle. Il constitue "l'îlot de données" dont il a

2. Les stéréotypes permettent de distinguer des constructions qui sont dérivées de constructions existantes, mais sont propres à la problématique traitée (Booch et al., 1999).

été question au chapitre 4.5.3.3.

6.2.6 Implémentation, test et documentation

L'implémentation du composant métier a été effectuée au moyen du langage de programmation Visual Basic 6 de Microsoft : un composant logiciel COM (annexe 3), de type ActiveX, agrégeant à la fois les composants données et domaine (chapitre 6.2.1), a été construit. Ce composant logiciel implémente 57 classes et met à disposition des intégrateurs 39 interfaces publiques (en plus des interfaces spécifiques à tout composant ActiveX), suivant le modèle présenté à la figure 98. Parmi ces interfaces publiques, 20 permettent la manipulation d'objets représentant des concepts réels issus de l'analyse de domaine. Le modèle de données a, quant à lui, été implémenté dans une BD Access. Enfin, nous nous sommes appuyés, pour permettre l'interaction entre le composant et la BD précitée, sur le composant ressource ActiveX Data Objects (ADO) de Microsoft.

Le composant métier développé a fait l'objet de tests approfondis, a été documenté (figure 101) et mis en oeuvre dans le cas de diverses applications (chapitre 6.3).

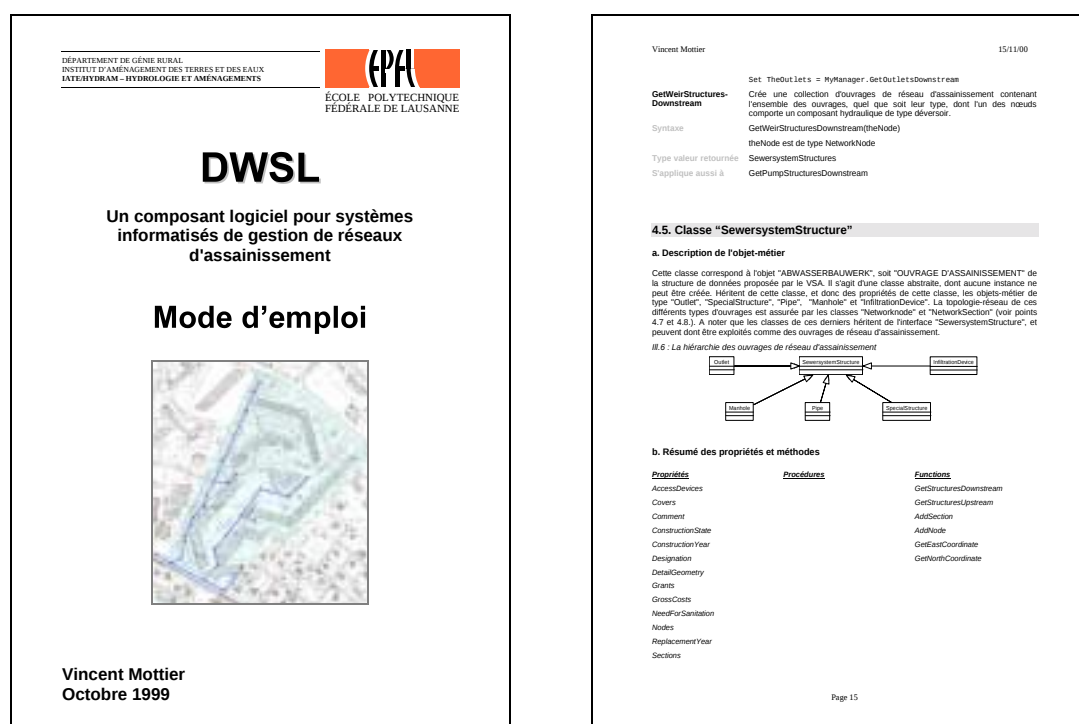


Figure 101. Extraits du mode d'emploi accompagnant le composant métier développé (Mottier, 1999).

L'illustration 102 montre le composant métier tel que vu par le biais de "l'explorateur d'objets" de l'environnement de développement Visual Basic : la technologie COM rend possible une interrogation de la sémantique des interfaces proposées par un composant, propriété qui est mise en oeuvre par les différents outils d'aide à la programmation. L'explorateur d'objets peut ainsi obtenir des renseignements sur les fonctionnalités offertes par tout composant COM, et les présenter à l'intégrateur.

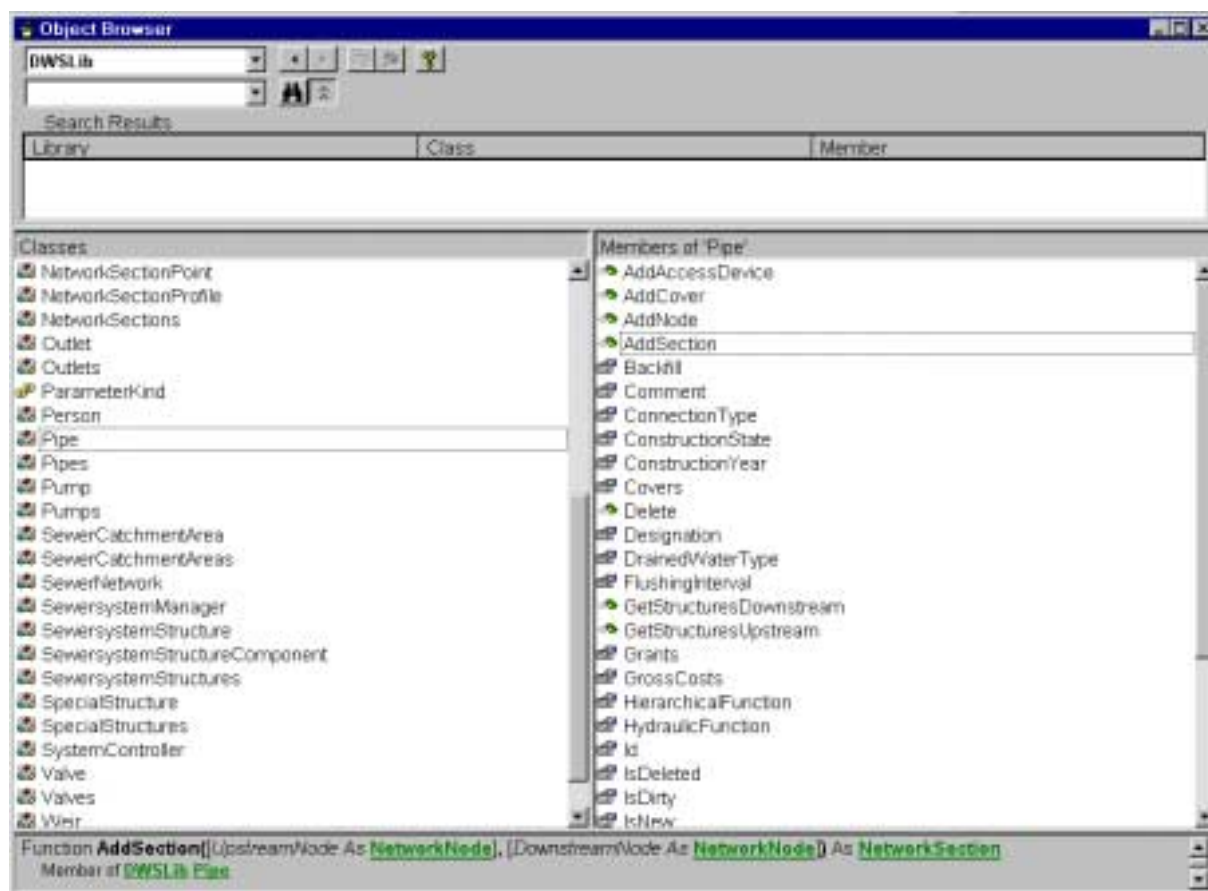


Figure 102. Le composant métier dans l'explorateur d'objets MS VB 6. À gauche, liste des interfaces publiques implémentées par le composant, à droite, liste de leurs propriétés (ici l'interface "Canalisation"); en bas, détail de l'opération d'ajout de section (de l'interface "Ouvrage de réseau d'assainissement" dont hérite l'interface "Canalisation").

6.3 Intégration du composant métier dans une application

6.3.1 Objectifs et architecture générale

Le composant métier (chapitre 6.2) présenté ici, que nous avons baptisé DWSL (chapitre 6.2), a été intégré dans un prototype d'application logicielle destiné à l'intervenant cité au chapitre 6.2.2. Cette intégration a d'une part rendu possible la vérification de l'intérêt de l'utilisation d'un composant métier, et d'autre part offert la possibilité de disposer d'un outil logiciel qui puisse être présenté et servir de support à la discussion avec différents acteurs du domaine de la gestion des eaux.

L'intervenant en question effectue régulièrement, dans le cadre de son activité professionnelle, des simulations des écoulements dans les réseaux d'assainissement, et utilise principalement, pour ce faire, le logiciel de simulation Hydroworks. Un tel processus métier implique l'acquisition des diverses données nécessaires à la simulation, relatives aux canalisations, aux ouvrages du réseau d'assainissement et aux

bassins versants d'assainissement, puis leur introduction dans le logiciel Hydroworks. Cette démarche, telle que menée usuellement, présente un certain nombre d'inconvénients :

- la manipulation des données directement dans le logiciel Hydroworks n'est pas aisée. Les possibilités de visualisation des données, ainsi que les possibilités d'interactions avec ces dernières, sont limitées;
- il est difficile de réutiliser les données car elles sont spécifiquement adaptées au logiciel de simulation. Or l'intervenant souhaite pouvoir les valoriser au moyen de logiciels tiers;
- la gestion des données sous forme de fichiers n'offre pas les avantages des BD et SGBD tels que sécurité, requêtes et sélections, etc.

L'intervenant en question a donc émis le vœu d'un outil qui lui serve de plate-forme d'acquisition et de gestion des données, soit indépendant des autres logiciels et lui permette de choisir librement ceux à utiliser pour les valoriser (figure 103).

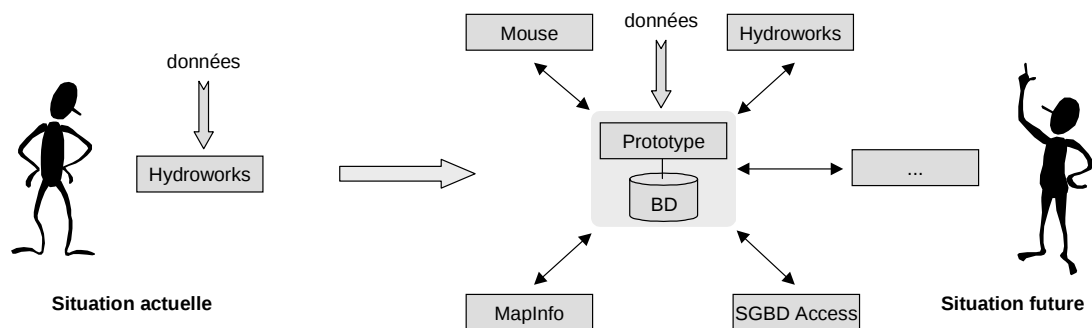


Figure 103. Le prototype a pour objectif de faciliter la gestion et la valorisation des données en les rendant indépendantes du logiciel Hydroworks. Il s'appuie principalement sur une BD Access, contrôle l'intégrité et la cohérence des données, et en facilite l'utilisation par les logiciels tiers.

Ses exigences étaient les suivantes :

- disposer d'un outil fonctionnant sous Windows en mode mono-utilisateur;
- pouvoir assurer la pérennité de l'information relative au réseau d'assainissement (canalisations, chambres, réservoirs, exutoires, déversoirs et sous-bassins versants d'assainissement), y compris la topologie réseau, et ce de la manière la plus simple possible;
- pouvoir visualiser l'information relative au réseau d'assainissement sous forme cartographique et choisir parmi différentes cartes d'arrière-plan;
- simplifier au maximum l'exportation des données vers le logiciel de simulation utilisé;
- gérer les données dans une BD relationnelle et pouvoir le cas échéant les manipuler à l'aide d'un SGBD;
- disposer d'outils permettant de vérifier la topologie du réseau et d'obtenir de l'information sur ce dernier.

En considération de ces besoins, il a donc été construit un prototype d'application (figure 104), basé principalement sur trois composants : 1°) le composant DWSL, dont il a été question tout au long du chapitre 6.2, pour la manipulation des objets métier relatifs au réseau d'assainissement; 2°) le composant technique MapX (annexe 9) pour la gestion de la composante spatiale de ces objets métier; et 3°) un autre composant métier, appelé par la suite HydroW, de type processus, également réalisé dans le cadre de cette recherche (Rakotondrafara, 1999), et conçu pour permettre la génération des fichiers nécessaires

au logiciel de simulation Hydroworks à partir des objets métier issus du composant DWSL.

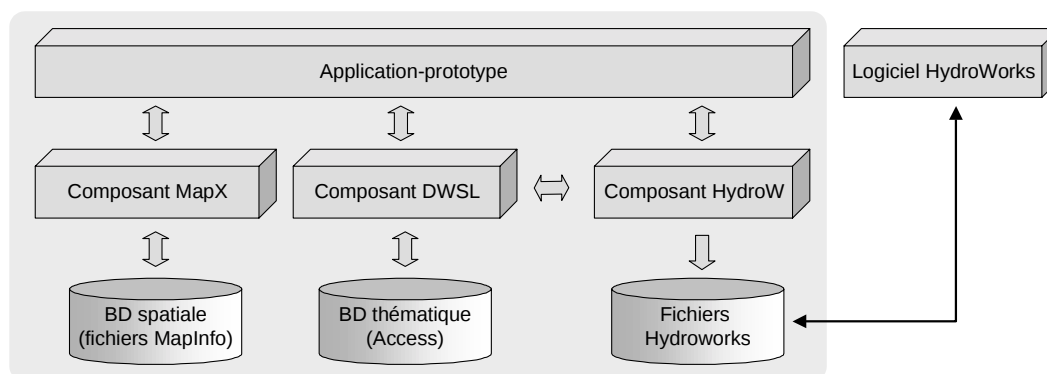


Figure 104. Architecture générale de l'application-prototype, qui est donc basée sur le composant métier présenté au chapitre précédent (DWSL), le composant métier permettant les interactions avec Hydroworks (HydroW) et le composant technique MapX. Le couplage avec Hydroworks est "faible" (Pouliot, 1999), c'est-à-dire basé sur l'échange de fichiers, raison pour laquelle ce dernier logiciel est extérieur au système considéré.

L'application-prototype a été implémentée avec le langage Visual Basic 6.0. Le composant métier DWSL, de type entité, assure la pérennité des données par le biais d'une BD Access. Le composant technique MapX utilise, quant à lui, des fichiers spécifiques au logiciel MapInfo. L'application logicielle assure elle-même la correspondance entre l'attribut spatial et l'objet métier correspondant (solution qui s'est à vrai dire révélée peu satisfaisante). Le composant HydroW, de son côté, enregistre les données sous forme de fichiers spécifiques au logiciel de simulation Hydroworks.

6.3.2 Présentation de l'application

Les paragraphes qui suivent présentent les différentes fonctionnalités de l'application développée et montrent, à l'aide de diagrammes de collaboration et de séquence (Booch et al., 1999), comment le composant métier DWSL est mis en oeuvre. L'objectif de ce chapitre est double : d'une part montrer que l'application offre les fonctionnalités demandées, et d'autre part illustrer la simplicité avec laquelle les composants métier permettent l'intégration de ces fonctionnalités dans l'application. Cinq points sont abordés : présentation générale de l'interface-utilisateur, gestion des données, fonctionnalités topologiques, calculs d'indicateurs, et soutien de la modélisation des comportements. Finalement, la mise en oeuvre d'un logiciel commercial tiers, en l'occurrence le tableur Excel, montre une autre forme d'utilisation du composant DWSL.

Interface-utilisateur

Une fenêtre cartographique occupe la majeure partie de l'interface-utilisateur (figure 105) et permet la visualisation des différents objets qui composent un réseau d'assainissement. L'utilisateur dispose, sur le coté gauche, de quatre palettes interchangeables, qui autorisent : 1°) la manipulation des fonds de plan (cartes, orthophotos, etc.); 2°) la sélection des éléments du réseau que l'on souhaite afficher ou masquer; 3°) la manipulation de ces éléments, soit leur création, leur édition ou leur destruction; 4) la navigation dans le réseau et la vérification de la topologie. En haut à gauche de l'écran sont regroupés les boutons offrant les fonctions de navigation et les outils de sélection fréquemment utilisés. Les fonctionnalités

moins usuelles ou conduisant à l'ouverture de formulaires tiers sont accessibles par le biais des menus.

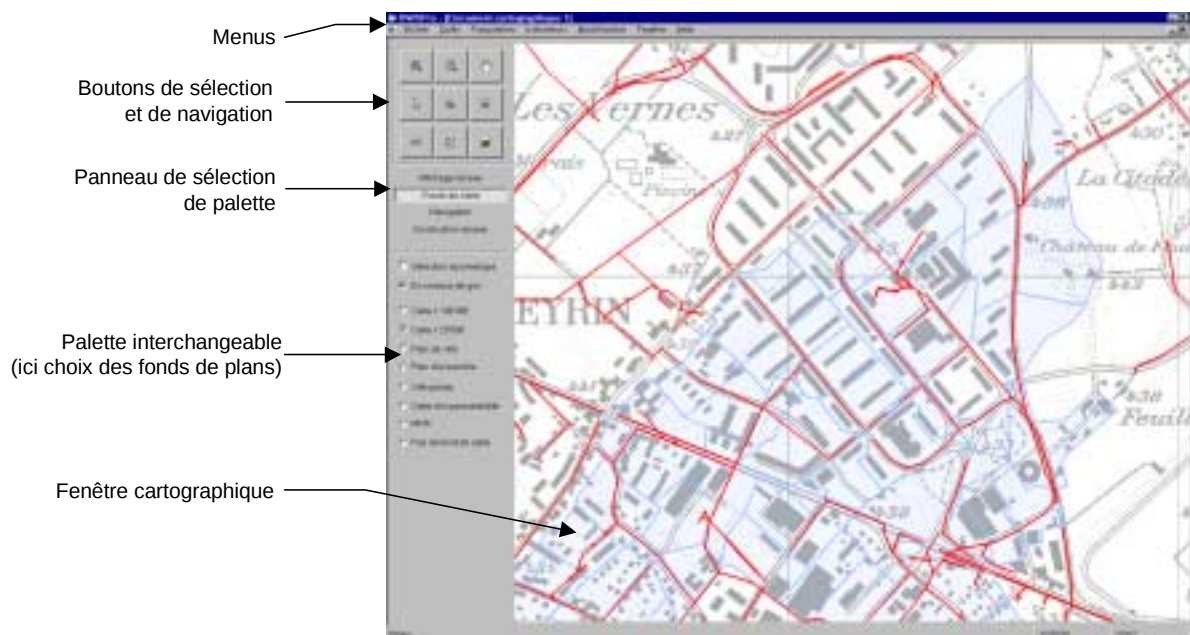


Figure 105. Une vue générale de l'application-prototype développée. Sont superposés à une carte au 1:25'000 (Office fédéral de la topographie) le réseau d'assainissement, en rouge, et les bassins versants d'assainissement, en bleu (Commune de Meyrin, GE).

Gestion des données

Aux différents fonds de carte peuvent être superposés les objets dont le composant métier permet la manipulation : la liste de ces objets est accessible par le biais de la palette visible à la figure 106. Un utilisateur, pour visualiser ou éditer de l'information thématique relative à un objet, peut utiliser le formulaire *ad hoc*.

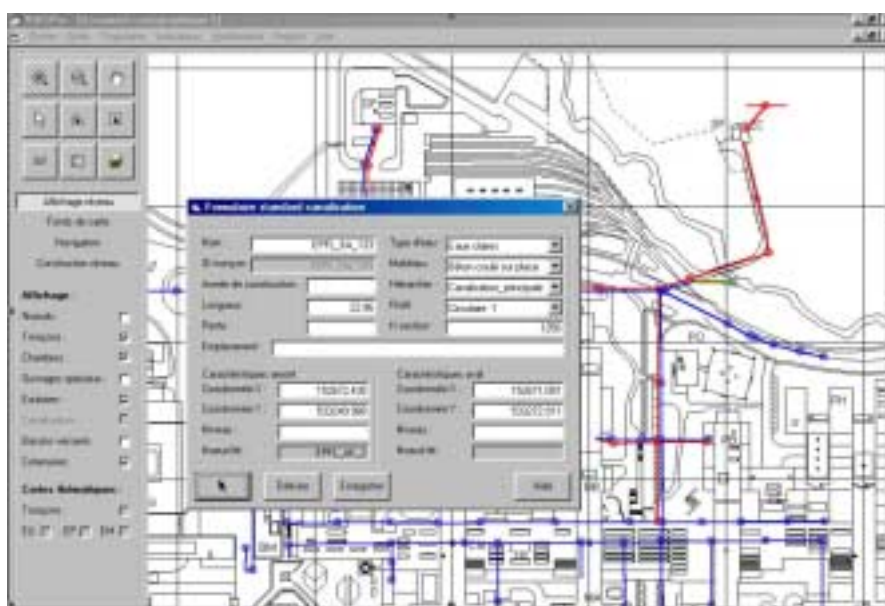


Figure 106. Affichage du réseau et utilisation du formulaire relatif aux canalisations (Davoli, 2000).

Pour l'intégrateur, la mise en place de ces fonctions est extrêmement simple, bien qu'un objet métier de type canalisation soit en réalité un agrégat dont les données sont disséminées dans de nombreuses tables relationnelles. Un tel objet métier gère lui-même les objets qui le composent, chacun de ces derniers étant responsable de sa persistance. L'intégrateur peut donc, par le biais d'un unique message, obtenir l'ensemble des données relatives à un concept donné, sous la forme d'un objet métier : il n'a absolument pas besoin d'interagir avec la BD ou de se préoccuper du respect d'une quelconque règle de gestion; il ne doit s'occuper que de la manière de présenter à l'utilisateur les données mises à disposition par l'objet métier (figure 107).

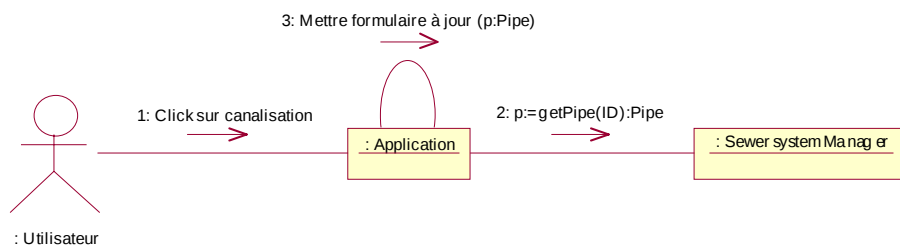


Figure 107. Diagramme de collaboration de la fonction "Afficher les caractéristiques thématiques d'une canalisation dans un formulaire". L'intégrateur, pour implémenter cette fonction, n'a pas besoin de se préoccuper d'interactions avec la BD.

Cette simplicité se retrouve au niveau des fonctions de création, d'édition et de destruction d'objets métier. L'utilisateur dispose de boutons (figure 108) qui lui permettent de créer ou de détruire interactivement les différentes composantes d'un réseau d'assainissement (chambres, canalisations, exutoires, déversoirs et sous-bassins d'assainissement). Il peut par exemple créer des ouvrages, soit par le biais du formulaire adéquat, soit en effectuant une digitalisation directement à l'écran. Les attributs thématiques sont spécifiés ou modifiés par le biais des champs correspondants des formulaires.

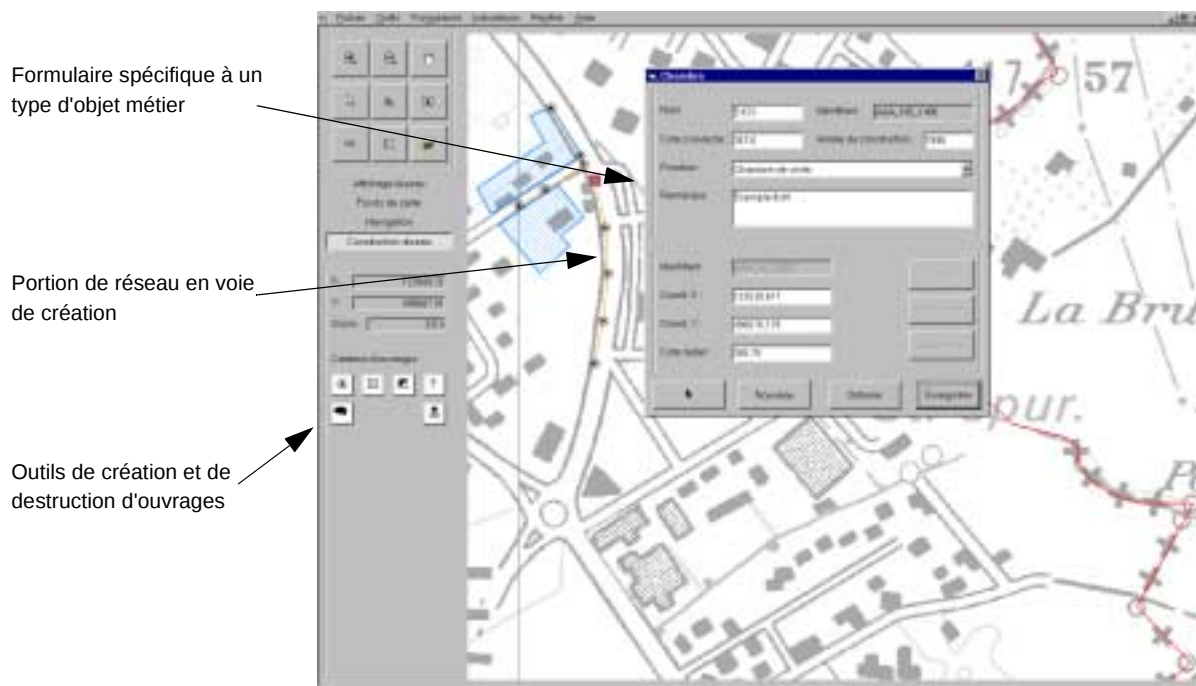


Figure 108. Plusieurs possibilités pour l'introduction de données dans le système. En fait, les outils de création, de destruction ou d'édition permettent d'agir sur les objets métier correspondants.

À nouveau, du point de vue de l'intégrateur, le travail de développement se trouve fortement réduit grâce à la mise en oeuvre d'un composant métier : seul le transfert des messages entre l'interface graphique et les composants métier sous-jacents doit être programmé. Le diagramme de séquence (figure 109) montre ainsi la succession d'opérations que l'intégrateur doit considérer pour créer et enregistrer un nouvel ouvrage (ici une chambre standard).

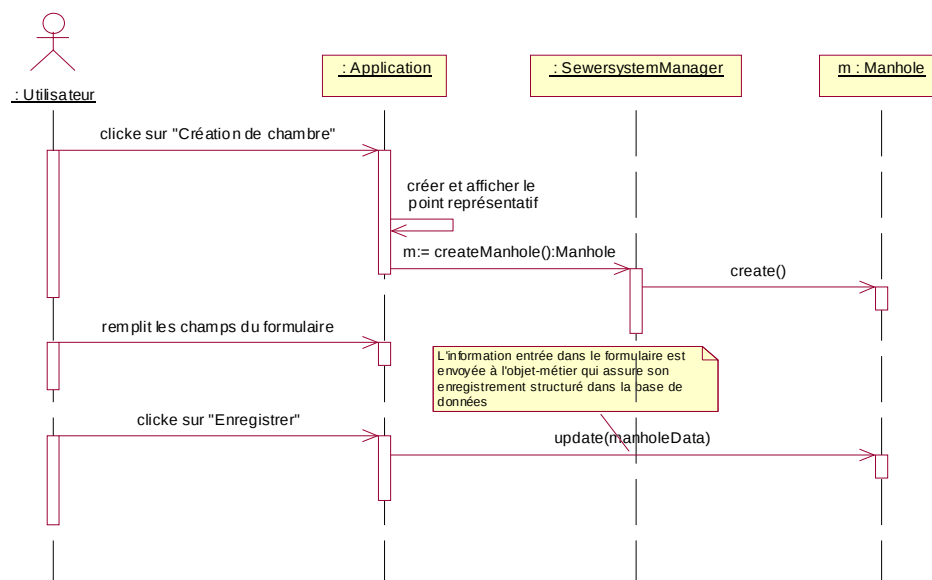


Figure 109. Diagramme de séquences pour la création et l'enregistrement de chambres standard, du point de vue de l'intégrateur. Les opérations internes au composant métier ne sont pas montrées.

Les outils proposés permettent la création interactive des relations topologiques. La création d'une canalisation par exemple se fait directement dans la fenêtre cartographique (en respectant le sens de l'écoulement) : il en résulte une relation topologique explicite entre les objets métier concernés.

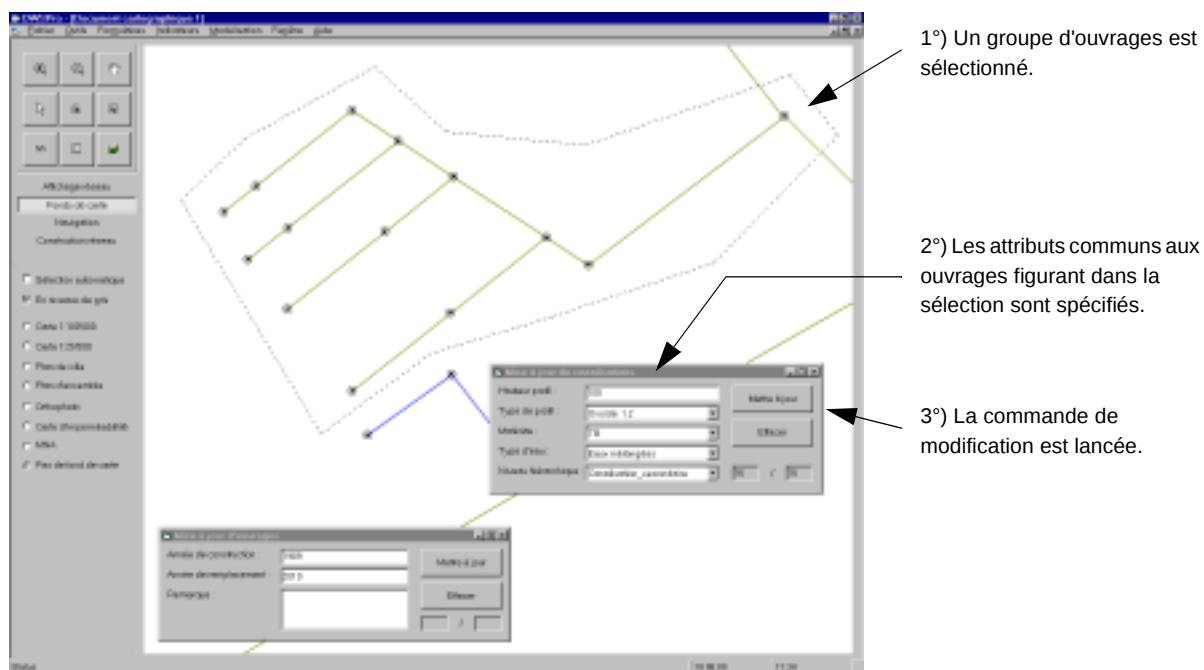


Figure 110. Modifications groupées des propriétés de canalisations et d'ouvrages de réseau.

Des outils destinés à simplifier l'acquisition de l'information ont été adjoints à l'application. En effet, les ouvrages d'assainissement possèdent généralement des caractéristiques similaires (diamètre, matériau, année de construction, etc.) dans un périmètre donné. Certains objets métier offrent la possibilité de modifier simultanément les propriétés relatives à des groupes d'objets métier : il devient ainsi très facile de mettre en place une fonction qui permet de sélectionner interactivement un ensemble d'ouvrages et d'indiquer au logiciel quelles propriétés doivent être modifiées. L'application regroupe en fait les ouvrages sélectionnés dans un objet métier de type "Collection d'ouvrages" et ordonne à ce dernier de procéder aux modifications demandées (figure 110).

Topologie

Les fonctions faisant intervenir les relations topologiques entre les ouvrages du réseau d'assainissement occupent une place très importante dans le prototype. Les objets métier gèrent cette topologie et proposent un ensemble de services faisant appel à cette topologie. Le composant métier offre de plus la possibilité de manipuler les objets "Réseau d'assainissement" et "Branche de réseau d'assainissement" qui gèrent des ensembles d'ouvrages d'assainissement corrélés topologiquement.

Chaque objet de type "Ouvrage de réseau d'assainissement" dispose par exemple d'une opération permettant d'obtenir l'ensemble des sous-bassins versants d'assainissement situés à son amont. Grâce à cette opération, un intégrateur peut très simplement prévoir une fonction permettant de visualiser cet ensemble des bassins versants d'assainissement situés en amont d'un ouvrage sélectionné (figure 111).



Figure 111. Détermination automatique de bassins versants d'assainissement.

Dans le même ordre d'idées, un objet de type "Branche de réseau d'assainissement" peut être utilisé pour visualiser un cheminement dans le réseau à partir d'un certain point. Dans notre application, un ouvrage du réseau est sélectionné par l'utilisateur, qui demande ensuite simplement au système de descendre le réseau à partir de ce point. L'application demande au composant métier, de manière transparente pour

l'utilisateur, la création d'un objet métier de type "Branche de réseau d'assainissement". Une fois que ce dernier a automatiquement identifié les ouvrages qui le composent, l'application se charge de le représenter graphiquement (figure 112). Des règles de gestion sophistiquées peuvent être intégrées à tout composant métier - par exemple relativement au comportement des flux dans des ouvrages hydrauliques complexes - ce qui ouvre la voie à de nombreuses possibilités. Une fonction pourrait ainsi être créée dans le but de soutenir les interventions d'urgence, en déterminant par exemple le cheminement de polluants à la suite d'un déversement accidentel dans le réseau et en identifiant les zones d'intervention possibles.

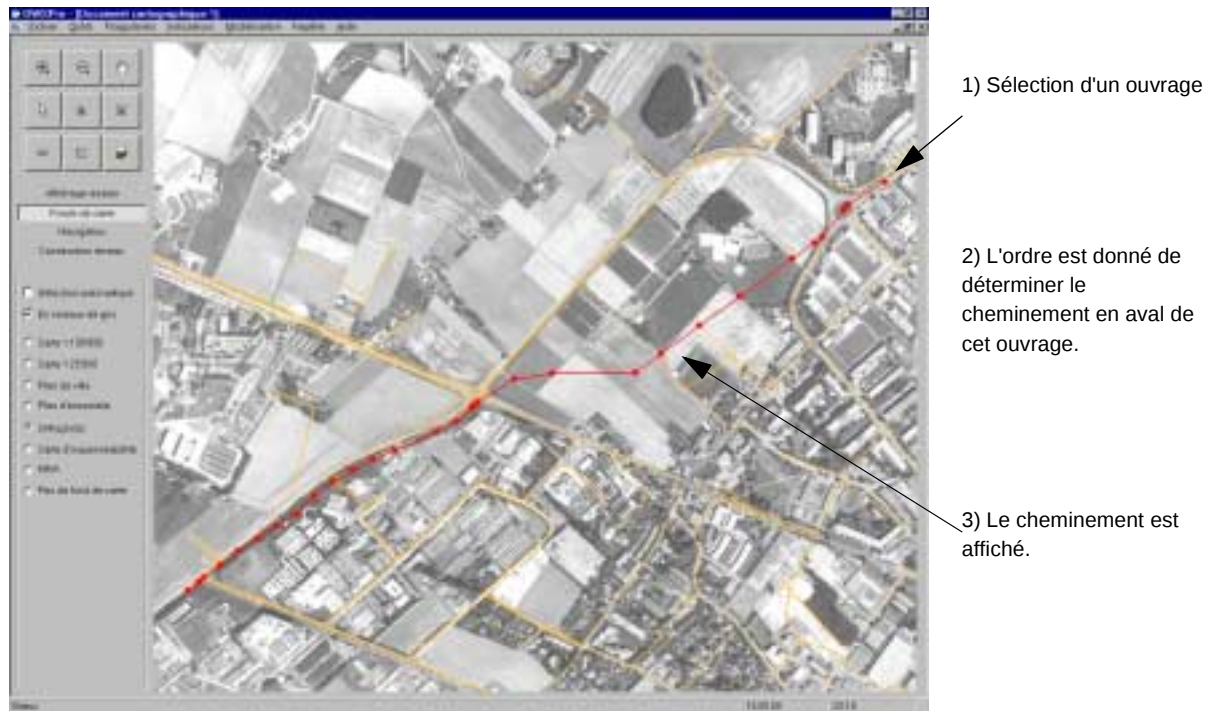


Figure 112. Identification automatique du cheminement de l'eau dans un réseau d'assainissement. L'utilisateur du système sélectionne le point à partir duquel il souhaite connaître le cheminement, en l'occurrence vers l'aval, et demande à visualiser ce cheminement (en rouge).

Calcul d'indicateurs

Des objets métier peuvent être chargés du calcul d'indicateurs. La figure 114 présente, à titre d'exemple, l'utilisation d'une opération dont sont pourvus les objets métier de type "Réseau d'assainissement": le calcul de la proportion de canalisations dont le diamètre est compris entre les valeurs passées en paramètre. La figure 113 présente un diagramme de collaboration mettant en oeuvre une telle possibilité et montre le peu d'opérations qui doivent être prévues en vue d'implémenter cette fonction.

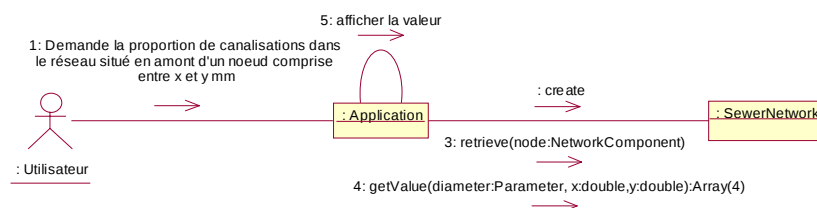


Figure 113. Diagramme de collaboration pour l'exploitation d'un indicateur relatif au réseau d'assainissement.

1) L'affichage du formulaire ad hoc est demandé.

2) Un ouvrage est sélectionné.

3) L'ordre est donné de rechercher le réseau amont à l'ouvrage sélectionné.

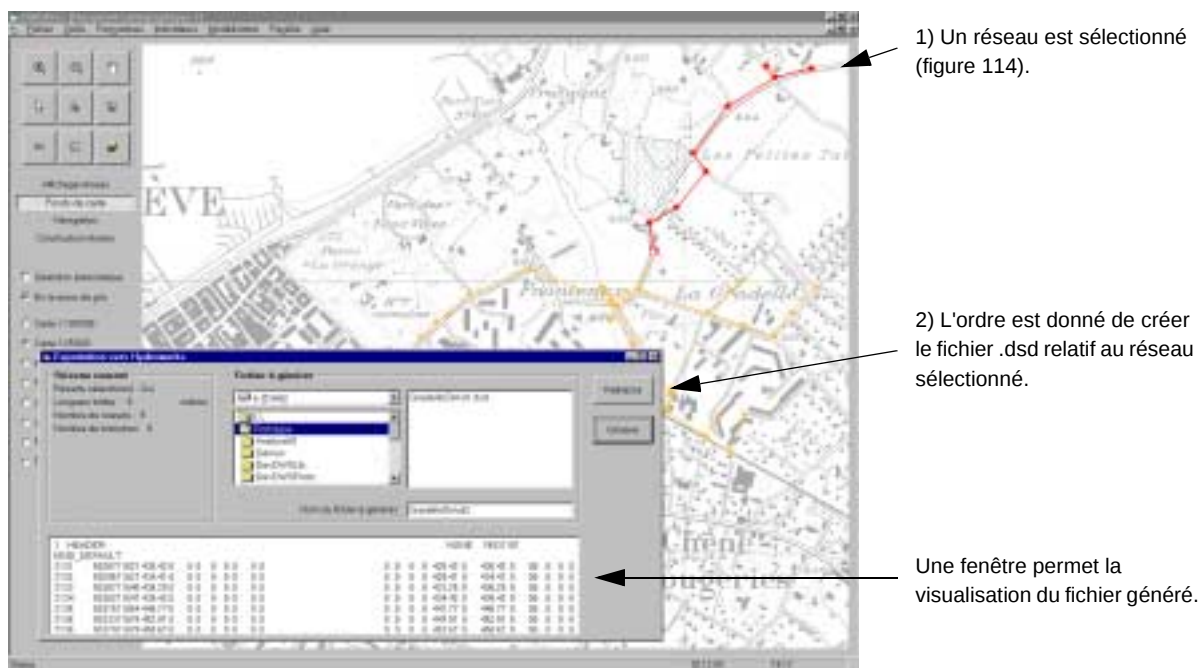
4) Le formulaire affiche la répartition des diamètres ainsi que quelques indications relatives au réseau identifié.



Figure 114. Calcul de la répartition des diamètres dans le réseau de canalisations sélectionné (en rouge). Les possibilités de navigation dans le réseau sont par ailleurs visibles sur la palette à gauche.

Aide à la simulation des comportements

La figure 115 présente l'utilisation de la fonctionnalité de génération automatique de fichiers de paramètres qui peuvent ensuite être exploités par le logiciel Hydroworks.



1) Un réseau est sélectionné (figure 114).

2) L'ordre est donné de créer le fichier .dsd relatif au réseau sélectionné.

Une fenêtre permet la visualisation du fichier généré.

Figure 115. Sélection interactive d'un réseau d'assainissement (en rouge) et génération du fichier de paramètres pour le logiciel de simulation de comportement Hydroworks (Rakotondrara, 1999).

L'ingénieur, tout d'abord, sélectionne le réseau d'assainissement pour lequel il souhaite effectuer une

simulation : ayant choisi le noeud aval, il demande la détermination automatique du réseau localisé en amont de ce dernier. Un objet métier de type "Réseau d'assainissement" est créé, puis passé en paramètre lors de l'invocation d'une opération particulière de l'objet métier "HydroworksController". Celui-ci se charge de créer et d'enregistrer un fichier propre au logiciel Hydroworks contenant les données relatives au réseau et prêt à être utilisé pour une simulation. Les paramètres qui ne peuvent être générés à partir des informations dont dispose l'objet métier "Réseau d'assainissement" devront être introduits par le modélisateur directement dans le logiciel Hydroworks.

Mise en oeuvre de logiciels tiers

L'illustration 116 présente enfin un exemple d'utilisation du composant métier DWSL dans un contexte totalement différent : s'intercalant entre l'application MS Excel et la BD contenant les données relatives au réseau, DWSL permet d'utiliser le tableur pour effectuer des traitements sur les données contenues dans la BD. La feuille de calcul présentée permet à l'utilisateur de sélectionner, dans des menus déroulants, deux ouvrages du réseau d'assainissement. Il peut ensuite être donné l'ordre au composant métier de générer un objet métier de type "Bras de réseau d'assainissement" recensant tous les ouvrages compris entre ceux venant d'être cités. Les données relatives à ces canalisations, qui constituent ce bras de réseau, sont ensuite affichées suivant l'ordre topologique. Cette petite application a été développée suite à la demande d'un ingénieur qui souhaitait pouvoir simplifier la préparation des données en vue de pouvoir faire différents calculs dans le logiciel Excel. Elle nous permet de montrer que les composants métier peuvent non seulement être assemblés pour former des applications nouvelles, mais également être intégrés à certaines applications existantes.

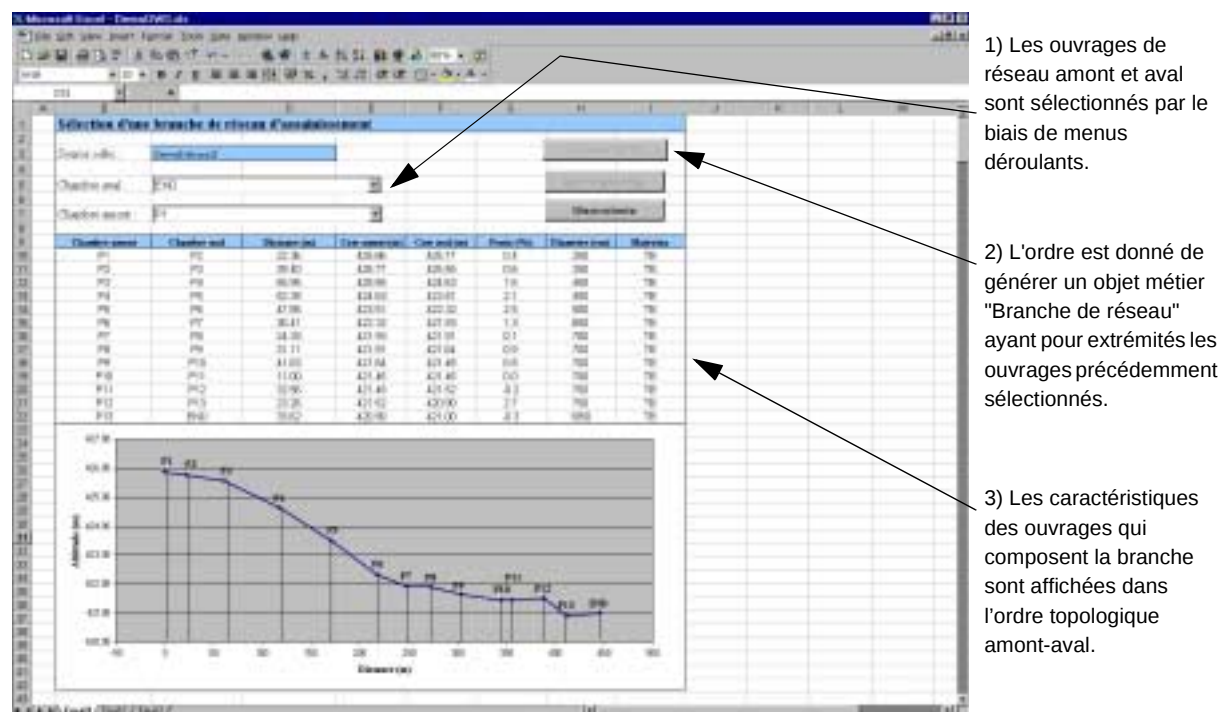


Figure 116. Utilisation du composant métier DWSL conjointement à l'application MS Excel. Le profil en long est factice.

6.4 Synthèse du chapitre et conclusion

La première partie de ce chapitre a présenté le processus de développement ayant conduit à la construction du composant métier de type entité DWSL. Ce composant a pour mission de permettre la manipulation des différents objets métier qui composent un système d'assainissement, soit les ouvrages du réseau (canalisations, chambres, ouvrages spéciaux, exutoires), les bassins versants d'assainissement, ainsi que des agrégats de plus haut niveau, à savoir les réseaux d'assainissement proprement dits et les bras de réseau. Ce composant métier offre, en plus des fonctions de gestion et de manipulation des données, des opérations de navigation au sein du réseau qui exploitent les relations topologiques existant entre les différents objets métier, des fonctions de calcul d'indicateurs destinées à faciliter la gestion des infrastructures.

Trois composants, le composant métier cité ci-dessus, un second composant métier développé dans le cadre de ce travail et permettant un couplage avec le logiciel de simulation des écoulements Hydroworks, ainsi que MapX, composant technique du domaine commercial offrant des fonctionnalités de type SIG, ont été intégrés dans un prototype d'application. La mission de cette dernière est d'apporter une assistance aux modélisateurs de comportements en réseau (modélisation hydraulique) dans leur tâche d'acquisition et de gestion de données.

Le développement du composant métier a permis les constatations suivantes :

- les différentes étapes de la méthode proposée s'enchaînent de manière logique, et sont bien équilibrées quant au temps nécessaire pour être menées à bien.
- la production des différents documents demande une rigueur certaine, et dans ce contexte les outils AGL (CASE tools) sont bienvenus.
- cette documentation des différentes étapes du processus de développement, qui peut paraître contraignante lors du premier cycle de développement, s'est révélée extrêmement précieuse dès que des itérations au sein même du cycle de développement du composant métier ont débuté. D'une part la rigueur imposée par la méthode proposée a favorisé le développement d'un composant métier ne contenant dès le départ que très peu d'erreurs, et d'autre part la documentation a permis de localiser très rapidement celles ayant été mises en évidence lors de la phase de test.
- la nette séparation entre les phases tournées vers une analyse du métier - analyse des besoins, analyse et segmentation de domaine, puis analyse-composant - et les phases comportant une forte composante informatique - spécification et mise en place de l'infrastructure, conception, implémentation, test et validation - a facilité l'implication des spécialistes du domaine de la gestion des eaux dans la réflexion.
- l'étape de spécification et de mise en place d'une infrastructure pour composants métier représente la phase la plus complexe du point de vue informatique, et nécessite un investissement important. Cependant, une fois cette infrastructure en place, le processus de développement de composants métier devient, du point de vue technique, beaucoup plus simple. Nous estimons que la durée nécessaire au développement d'un composant métier, une fois l'infrastructure pour composants disponible, est en moyenne de deux semaines à quatre mois-homme (en fonction de la complexité du composant).

La mise en place du prototype d'application basée sur l'assemblage des composants métier cités précédemment a permis de vérifier :

- qu'une approche par composants permet la création extrêmement rapide d'applications opérationnelles. À titre d'exemple, le prototype d'application qui a été développé - utilisé intensivement dans le cadre d'un travail de diplôme (Davoli, 2000) - a nécessité seulement trois semaines de développement (s'agissant d'un prototype, il serait toutefois nécessaire, pour le rendre commercialisable, d'y investir plus de temps). La notion de composant métier permet ainsi d'envisager la construction d'applications à courte durée de vie, répondant spécifiquement à un besoin, mais s'appuyant sur des éléments plus stables, les composants métier.
- que le rôle d'intermédiaire entre l'interface utilisateur et la BD que jouent les composants métier simplifie considérablement le travail de l'intégrateur, puisque la complexité des structures de données, inévitable dans un contexte de gestion intégrée des eaux, peut être maîtrisée.
- que le fait de baser une application sur des composants métier permet de construire des applications très fiables. Les composants métier intègrent en effet la majeure partie de la logique métier, et prennent en charge pour l'intégrateur les interactions avec les BD. Les problèmes qui se sont révélés lors de l'intégration du composant avaient essentiellement pour origine la gestion de l'interface utilisateur.

Le prototype de composant métier, accompagné de sa documentation (figure 101) a été mis à disposition de cinq ingénieurs travaillant dans le domaine de la gestion des eaux et conduits par leur activité professionnelle à développer dans ce contexte des outils informatiques. L'objectif de cette démarche était de recueillir leurs impressions quant aux concepts proposés. L'approche par composants métier a été plébiscitée par les différents intervenants. La possibilité de disposer de composants qui peuvent être réutilisés est considérée comme très intéressante. Par contre, le prototype de composant métier a été peu testé. Plusieurs raisons peuvent l'expliquer : 1°) la mise en oeuvre de composants logiciels aurait nécessité, de la part des intervenants, des connaissances spécifiques que, faute de temps il n'ont pu acquérir; 2°) même si les interfaces proposées par le composant métier sont basées sur des concepts réels, l'exploration de ces interfaces nécessite un investissement assez important; 3°) le prototype développé se base sur une infrastructure peu souple en regard des évolutions des structures de données sous-jacentes auxquelles il s'agit de s'adapter : les intervenants ne pouvaient donc l'utiliser pour exploiter leurs propres BD; et 4°) le développement d'interfaces graphiques est considéré - à juste titre - comme une tâche nécessitant du temps, et le prototype développé n'offre, à cet égard, aucun support. Nous pouvons donc déduire que :

- les composants métier s'adressent à une catégorie particulière d'intervenants, les intégrateurs, qui sont spécialisés dans le développement d'outils informatiques. Les composants métier ne doivent pas avoir pour objectif de permettre à chaque intervenant de concevoir l'outil logiciel répondant à ses besoins propres.
- la documentation accompagnant tout composant métier revêt une importance considérable : elle doit décrire les interfaces publiques proposées de manière exhaustive, et s'accompagner d'exemples d'utilisation. La documentation proposée dans le cadre de ce travail a été jugée austère.
- les composants métier doivent intégrer des composantes d'interface graphique, qui peuvent (mais ne doivent pas) faciliter la conception de l'interface utilisateur des outils logiciels intégrant lesdits composants.
- enfin, l'infrastructure pour composants doit être basée sur un modèle qui permette à l'intégrateur d'adapter les composants métier à des structures de données particulières, pour autant, bien sûr, que ces structures respectent certaines règles. Il s'avère en effet qu'il est très difficile d'imposer pour la

structuration de bases de données un modèle particulier, ces structures évoluant constamment dans la pratique.

Le prototype d'application nous a donné la possibilité de présenter les potentialités qu'offre une approche par composants métier aux intervenants intéressés. Il a été constaté à cette occasion qu'un tel prototype peut jouer un rôle très important dans le cadre d'une définition des besoins : l'interlocuteur dispose d'un support concret pour appuyer et stimuler sa réflexion.

Ainsi, il s'avère qu'une approche par composants métier possède réellement les avantages qui lui sont attribués (chapitre 4.6), mais que de tels composants s'adressent à un public particulier : les intégrateurs, chargés de concevoir des solutions informatiques spécifiques. Il n'est pas dans l'optique d'une approche par composants de permettre à chacun de construire son propre outil logiciel, étant donné que de tels composants sont malgré tout relativement complexes et guère accessibles par pure intuition.

Les fonctions proposées par le prototype de composant métier développé peuvent paraître simplistes, et le lecteur aura certainement noté que des logiciels commerciaux sont à même d'offrir des services similaires (aide à la gestion des données, navigation, etc.). Cependant, les composants métier possèdent la faculté d'interagir et de pouvoir être assemblés. Ainsi, les premiers composants métier qui sont mis sur le marché doivent offrir ces services de base. À partir de là pourront être développés des composants métier s'appuyant sur les premiers cités et offrant des services plus sophistiqués : une approche par composants métier offre la possibilité de pouvoir réutiliser aisément des développements antérieurs, y compris des développements issus d'organisations différentes, pour produire des services toujours plus avancés.

Enfin, la démarche adoptée dans ce travail, laquelle a consisté à appuyer notre réflexion sur un processus itératif intégrant mise en place de prototypes, analyse de ces derniers et interactions avec différents intervenants, a certes nécessité un volume de travail important mais s'est révélée très fructueuse. En effet, des solutions techniques ont pu être proposées, et le degré d'acceptation que ces intervenants leur ont réservé a été estimé. Et si une validation complète de la méthode proposée nécessiterait que celle-ci soit mise en oeuvre lors de la réalisation de quelques projets réels, notre manière de procéder a également permis de vérifier son adéquation avec les objectifs fixés.

Perspectives et conclusion

En premier lieu est présentée une synthèse de la recherche. Puis nous proposons un processus destiné à encourager, accompagner et coordonner la production de composants métier dans le domaine de la gestion des eaux. En effet, nous estimons que le but à viser est la création d'une palette de composants métier dans laquelle les intégrateurs pourraient puiser pour construire de manière efficiente des outils de gestion des eaux. Finalement, une conclusion clôture ce travail.

7.1 Synthèse

L'optique dans laquelle la gestion des eaux est actuellement menée est en train de changer : les différentes tâches sont abordées non plus de manière sectorielle, mais selon une approche plus globale, tenant compte des multiples interactions qui existent entre les différentes composantes du système hydrologique (chapitre 2.1). Cette approche intégrée de la gestion des eaux est encouragée par les récents textes législatifs ainsi que par diverses recommandations et normes d'associations professionnelles.

Les praticiens se trouvent cependant confrontés à des difficultés lorsqu'ils souhaitent adopter une telle approche : 1°) le système hydrologique est un système complexe, ce qui rend l'évaluation des implications des interventions difficile et oblige à tenir compte d'un nombre élevé de paramètres (chapitre 2.2); 2°) les intervenants concernés par une gestion intégrée sont très nombreux et appartiennent généralement à des organisations différentes, ce qui rend les collaborations difficiles (chapitre 2.2.3); et 3°) les pratiques actuelles en matière de gestion des eaux sont remises en question, car elles ne sont pas conformes aux exigences induites par une perspective de développement durable (chapitre 2.1.3.2).

Les outils informatiques, qu'il s'agisse de SGBD, de SIG, de logiciels de CAO, de logiciels de simulation des comportements ou autres (chapitre 2.4.2), apportent bien évidemment dans un tel contexte un soutien bienvenu, de par leur capacité à gérer et à manipuler de gros volumes de données. Ils ne donnent cependant pas entière satisfaction : les échanges d'informations entre intervenants restent peu aisés, et les fonctionnalités à même de faciliter les processus de gestion des eaux manquent cruellement, la complexité même du domaine rendant la mise sur pied de tels outils difficile et coûteuse.

L'objectif de ce travail est de **proposer des concepts et une démarche permettant, malgré ces difficultés, de développer des outils logiciels qui puissent efficacement soutenir une gestion intégrée des eaux**, en particulier dans les zones urbanisées. Les exigences en regard de ces outils sont qu'ils : 1°) soient à même d'évoluer parallèlement à la façon d'aborder la gestion des eaux; 2°) puissent être adaptés aux besoins spécifiques de chaque utilisateur; 3°) soient compatibles avec différentes configurations informatiques; 4°) soient ouverts sur les applications logicielles existantes; 5°) soient simples à mettre en oeuvre 6°) puissent être conçus et modifiés rapidement; et 7°) restent d'un coût acceptable (chapitre 3.2).

Notre réflexion nous a conduits à proposer des entités logicielles indépendantes, les composants métier. **Chaque composant métier permet la manipulation d'un ensemble précis et cohérent de concepts du domaine de la gestion des eaux**. Les composants métier peuvent être assemblés, en fonction des besoins et ces assemblages configurés de manière à former des outils logiciels destinés à soutenir une gestion intégrée des eaux, tout en restant spécifiquement adaptés aux besoins d'utilisateurs particuliers.

D'un point de vue théorique, notre proposition repose sur trois idées principales :

- **maîtriser la complexité inhérente au système considéré par un partitionnement.** Une gestion intégrée des eaux implique la prise en compte simultanée de très nombreux facteurs. Les concepts qu'il est nécessaire de manipuler dans un tel contexte sont regroupés en ensembles cohérents qui font l'objet de processus d'informatisation indépendants et conduisent aux composants métier. Ces derniers sont à même d'interagir avec d'autres composants métier. Un système logiciel supportant une gestion intégrée des eaux peut, à partir de là, être construit par assemblage de ces différents composants.
- **concevoir les composants métier comme des entités hautement réutilisables.** À partir du moment où les outils logiciels sont construits à partir de composants métier, il est judicieux de concevoir ces derniers de manière à pouvoir les réutiliser le plus souvent possible. En effet, ceci permet d'en améliorer la qualité tout en diminuant les coûts spécifiques, chaque composant métier étant destiné à être intégré dans de nombreuses applications.
- **utiliser le territoire et ses composantes comme facteur d'intégration entre les différentes catégories d'intervenants.** Dans le domaine de la gestion des eaux, les nombreuses catégories d'intervenants manipulent, dans le contexte de processus-métier différents, des concepts qui se rapportent aux mêmes éléments du territoire. Certains composants métier permettent spécifiquement la manipulation de ces concepts, les présentant sous des formes adaptées aux besoins des utilisateurs. Ainsi un composant métier se rapportant à un ensemble de concepts particuliers sera mis en oeuvre par les différents intervenants qui manipulent ces concepts, et pourra assurer un rôle intégrateur entre ces différents intervenants.

D'un point de vue technique, notre proposition se base sur le paradigme orienté objet, les composants logiciels, la notion d'interface et les composants métier :

- **le paradigme orienté objet** conduit à une correspondance très étroite entre le système "réel" et le système "logiciel" : le système logiciel permet la manipulation d'abstractions informatiques qui ont leur équivalent dans le monde réel (chapitre 4.2). Cela facilite la compréhension des systèmes logiciels, ainsi que leur modularité et leur maintenance.
- les **composants logiciels** sont des entités logicielles aux caractéristiques et fonctionnalités bien définies qui peuvent être réutilisées et assemblées pour former des applications logicielles

opérationnelles. Raisonner en termes de composants logiciels permet de concevoir des applications "sur mesure" à partir de pièces "standardisées".

- les composants logiciels proposent leurs services par le biais d'**interfaces**, qui peuvent être vues comme des contrats définissant les services que le composant s'engage à assurer (figure 117). Les interfaces sont clairement séparées de leur implémentation, constituée du code informatique qui permet d'accomplir les services proposés (chapitre 4.3). L'interopérabilité entre composants est basée sur cette notion d'interface. Nous nous intéressons ici aux composants objet, c'est-à-dire à ceux dont les interfaces permettent, comme leur nom l'indique, la manipulation d'objets informatisés.
- les **composants métier**, constitués de différents composants logiciels (chapitre 4.5.3), rendent possible la manipulation d'objets informatiques à forte granularité, les objets métier, qui sont des abstractions informatiques de concepts utilisés par des intervenants dans le contexte de leur activité professionnelle.

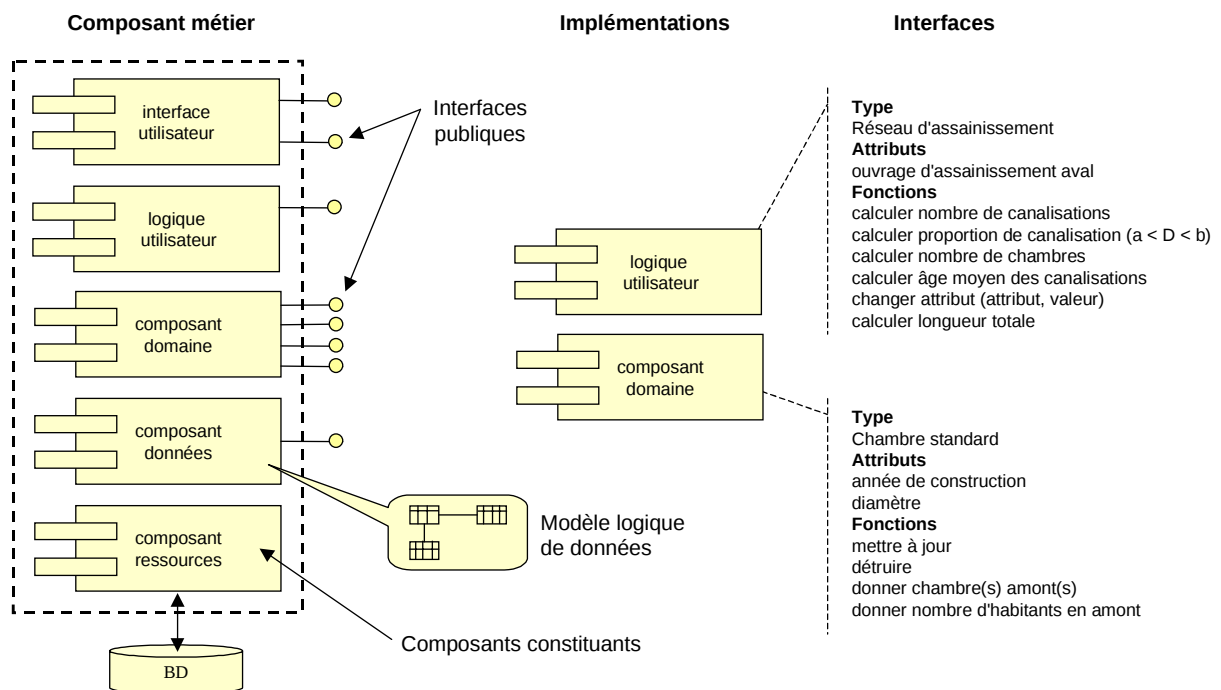


Figure 117. À gauche, exemple de composant métier avec extension maximale, c'est-à-dire comprenant un représentant de chaque catégorie de composant constituant. En pratique, suivant le type de composant métier, certains de ces composants constitutants seront absents. À droite, exemples d'interfaces publiques (incomplètes et simplifiées) proposées par le composant métier.

Les composants métier sont ainsi des unités de réutilisation spécifiquement conçues pour soutenir certains métiers et pour permettre la manipulation de concepts issus de l'activité professionnelle des intervenants concernés. Ceux que nous proposons sont plus spécifiquement destinés au domaine de la gestion des eaux en milieu urbain. Ils sont destinés à être intégrés dans un système logiciel qui est ensuite personnalisé en fonction des besoins et vœux de l'utilisateur final.

Une approche par composants métier comporte de nombreux avantages :

- la compréhension et la prise en main de composants métier est aisée, car ceux-ci sont basés sur une modélisation des concepts et activités du domaine professionnel de l'intervenant;

- la construction de SIAGEM devient, si l'on utilise une telle approche, relativement simple, car tant la complexité informatique que celle due au système réel considéré sont maîtrisées lors de la conception des composants métier eux-mêmes;
- la qualité des SIAGEM constitués de composants métier est élevée, car ces derniers sont destinés à être intégrés dans de nombreuses applications et font de ce fait l'objet d'une conception et de tests soigneux;
- les coûts spécifiques et les temps de développements de SIAGEM se trouvent réduits, de par l'utilisation de composants préexistants dont les coûts de développement peuvent être répartis sur différentes applications;
- ces SIAGEM sont modulaires et flexibles, car les composants métier sont interchangeables.

Des composants métier, pour pouvoir être effectivement réutilisés, doivent répondre à certains critères. En particulier, ils doivent être les plus indépendants possibles les uns des autres et pouvoir être adaptés à différentes configurations informatiques. Le chapitre 4.5.3 présente ainsi une architecture interne de composant métier qui leur procure la souplesse nécessaire. Chacun de ces composants met à disposition de l'intégrateur de composants métier des interfaces publiques qui autorisent soit la manipulation d'objets métier, soit leur configuration. À noter que certains composants métier doivent pouvoir s'appuyer sur une BD et intègrent donc, de façon plus ou moins configurable, un modèle de données assurant la correspondance entre les objets métier et les données contenues dans la BD.

Les composants métier doivent être conçus de manière très soignée car ils sont destinés à être réutilisés de nombreuses fois. Il est proposé une méthode de développement, baptisée COWIS, spécifiquement adaptée au développement de composants métier dans le domaine de la gestion des eaux (chapitre 5). Bien sûr, cette méthode n'a pas été créée *ex nihilo*, mais s'appuie sur celles qui sont disponibles. Elle a pour principales caractéristiques de permettre des cycles de développement rapide, chaque cycle conduisant à la livraison de composants métier opérationnels, et d'être axée sur le développement de composants intuitifs et réutilisables. Outre le fait de sensibiliser les intervenants ne venant pas du domaine de l'informatique à la nécessité d'appuyer le développement de composants métier sur une méthode, les principaux apports de ce travail, à ce niveau, sont : 1°) d'autoriser la mise en place, au fil des itérations successives, d'un référentiel commun, le modèle de domaine, qui est progressivement affiné; 2°) de fournir une technique à même de maximiser le potentiel de réutilisation des composants développés; et 3°) de proposer une hiérarchisation des composants métier qui permet de pleinement tirer profit du potentiel intégrateur du territoire et d'augmenter l'indépendance des composants.

Pour appuyer notre raisonnement, un prototype de composant métier, de type entité, se rapportant au domaine des réseaux d'assainissement, a été développé (chapitre 6). Son intégration dans un prototype d'application destinée à l'acquisition, à l'édition et à la validation de données relatives au réseau d'assainissement, et comprenant également des fonctions d'exportation de données et de navigation dans le réseau, a montré d'une part que l'utilisation de composants métier permet une construction rapide d'applications performantes et fiables, et d'autre part que leur mise en oeuvre, bien qu'a priori relativement simple, puisque la majeure partie de la complexité informatique est intégrée aux composants métier, requiert tout de même un certain bagage de la part des intégrateurs dans le domaine des composants logiciels. Il s'est également avéré que la qualité de la documentation de tels composants revêt une grande importance.

7.2 Vers un "marché" de composants métier ?

La gestion des eaux recouvre des activités très diverses qui impliquent la collaboration d'intervenants aux compétences variées. L'élaboration d'une collection de composants métier qui permettraient de concevoir des SIAGEM spécifiquement adaptés aux besoins de ces intervenants, et autorisant une gestion réellement intégrée des eaux, nécessite une collaboration entre ces différents intervenants. En raison de la dispersion des compétences entre diverses organisations, un tel processus est difficile à initier. De plus, un projet de ce type implique des connaissances informatiques dont peu d'organisations actives dans le domaine de la gestion des eaux disposent. Quelles stratégies peuvent être mises en place pour amorcer et accompagner un processus de production intensive de composants métier ?

Autre obstacle, le développement de composants métier constitue un investissement relativement important. De ce fait, de tels composants doivent nécessairement être intégrés dans des applications différentes aussi nombreuses que possible. Comment assurer une diffusion suffisante et économiquement rentable des composants métier ?

À partir du moment où il faut que des intervenants n'appartenant pas aux mêmes organisations puissent collaborer, que des composants métier d'origines différentes puissent être assemblés, il devient nécessaire d'envisager un certain degré de standardisation des composants métier, et plus particulièrement des interfaces publiques proposées par ces composants métier. Comment aborder une telle tâche ?

Nous proposons ici un processus, inspiré de celui mis en oeuvre par l'OGC (Open GIS Consortium, 1998b, voir également l'annexe 8), et avons identifié quatre groupes d'intervenants (figure 118):

- le processus s'enclenche lorsque l'organisation, en charge du développement d'une application spécifique, estime que les fonctions demandées présentent un potentiel de réutilisation et qu'il vaut la peine de développer des composants métier. Cette organisation intégrera à terme les composants métier pour former l'application opérationnelle. Les acteurs au sein de cette organisation sont les **intégrateurs** de composants métier, et celle-ci est généralement de type privé, bureau d'ingénieurs ou de géomatique par exemple.
- le développement de composants métier réclame des compétences pointues en informatique, et doit de ce fait être mené par des intervenants spécialisés dans ce domaine. Ces **groupes de développement** de composants métier peuvent être issus de sociétés d'informatique ou de géomatique, voire de gros bureaux d'ingénieurs qui disposent des compétences nécessaires.
- à partir du moment où il est question de standardisation, il faut qu'existe une autorité qui encourage l'approche par composants métier, édicte des règles et des recommandations en matière de documentation et de conception, contrôle la conformité des composants métier qui sont développés et dispose d'une vue d'ensemble de ceux qui existent. Ce rôle est dévolu à l'**autorité de contrôle et de validation**, qui peut être la commission *ad hoc* d'une association professionnelle.
- le domaine de la gestion des eaux est extrêmement vaste, et l'autorité de contrôle et de validation ne peut être compétente pour toutes les questions qui peuvent se poser. Elle constitue donc, lorsqu'un processus de développement de composants métier s'enclenche, un **groupe d'intérêt**, composé d'experts du domaine concerné et de spécialistes en modélisation objet, qui assistent l'équipe en charge du développement des composants métier et donnent un préavis quant aux spécifications dont la standardisation est demandée. De tels groupes peuvent être dissous en fin de processus de

développement.

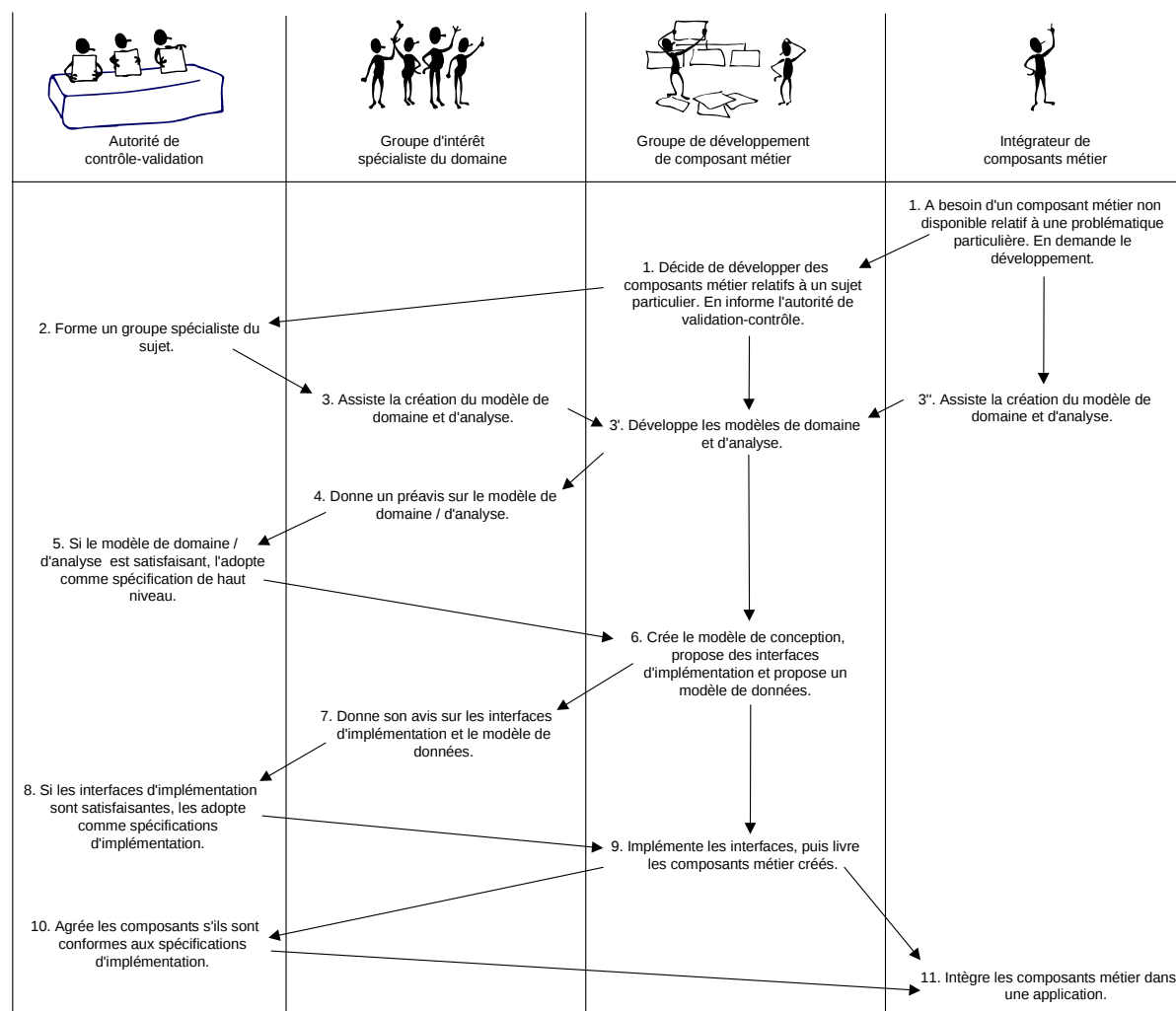


Figure 118. Aspects organisationnels d'une approche par composants métier dans le domaine de la gestion des eaux. Déroulement des activités dès le moment où la décision est prise de construire un ou plusieurs composants métier non disponibles et dont les interfaces ne sont pas standardisées.

Les groupes d'intérêt étant clairement identifiés, le processus proposé à la figure 118 peut être détaillé :

- un organisme est chargé de développer un outil logiciel se rapportant au domaine de la gestion des eaux. Il décide d'adopter une approche orientée composants métier (une telle approche peut également être demandée par le mandant). Il charge alors un groupe de développement de composants métier de développer ceux dont il a besoin.
- le groupe de développement de composants métier informe alors l'autorité de contrôle-validation de son objectif, à savoir créer des composants métier pour tel ou tel domaine puis les faire admettre comme standard. L'autorité de contrôle-validation peut former alors, si besoin est, le groupe d'intérêt, qui assistera à la création des composants métier en apportant son expertise dans le domaine considéré. Mentionnons que l'intégrateur de composants participe généralement, en tant que futur utilisateur, au développement des composants métier.
- le groupe de développement, après avoir pris connaissance des spécifications déjà normalisées qui pourraient exister auprès de l'autorité de contrôle-validation, entame le processus de développement décrit au chapitre 5.3.2. Il soumet, après préavis du groupe d'intérêt et pour chaque composant

métier projeté, le modèle de classes d'analyse (accompagné du glossaire et des règles de gestion) ainsi que la liste des opérations dont le composant sera chargé. Si les documents déposés sont cohérents, complets, présentent un intérêt pour la communauté du domaine de la gestion des eaux, et ne sont pas redondants par rapport à des documents déjà déposés, ils sont validés sous le terme de "spécifications de haut niveau" (figure 119).

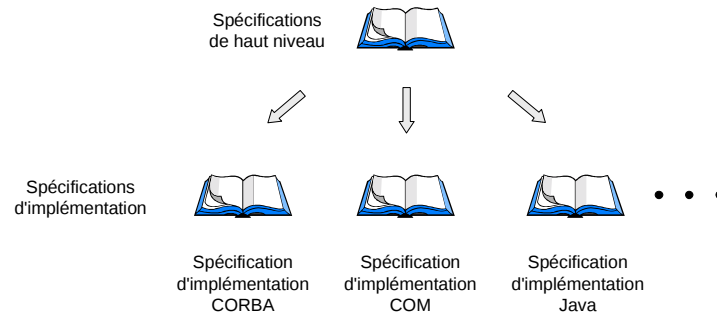


Figure 119. Pour chaque composant, une spécification de haut niveau est déposée et peut être à l'origine d'une spécification d'implémentation relative à un environnement pour composants (adapté de Open GIS Consortium, 1998b).

- le groupe de développement poursuit le processus de développement de composants métier. Dans le contexte de l'étape de conception, qui tient compte de l'environnement pour composants choisi (chapitre 4.3), il définit précisément les interfaces publiques qui seront implémentées par chaque composant métier : la responsabilité de chaque opération est attribuée à une interface. La description des interfaces, les règles de gestion à respecter lors de l'implémentation (invariants, préconditions et postconditions), ainsi qu'un modèle logique de données si nécessaire, constituent les documents qui sont déposés, après préavis du groupe d'intérêt, auprès de l'autorité de contrôle-validation, dans le but de les faire adopter comme "spécifications d'implémentation". Celles-ci impliquent que des composants métier construits ultérieurement, dans un but similaire, devront être conformes à ces spécifications pour être compatibles avec les composants existants.

Pourquoi intégrer aux spécifications un modèle de données? Les interfaces rendent possible l'interopérabilité au sein d'un même système logiciel, et permettent d'échanger non seulement des données, mais également des services. Cependant, les données doivent pouvoir être transférées entre systèmes logiciels non compatibles : un modèle de données standardisé facilite les échanges de données avec des systèmes tiers et autorise une sauvegarde aisée des données dans un système externe.

L'approche décrite ci-dessus permet à tous les groupes d'intervenants de préserver leurs intérêts, tout en facilitant leur collaboration :

- l'intégrateur dispose de composants métier qui lui permettent de construire des applications de qualité beaucoup plus rapidement et à meilleur compte que si le développement avait été mené de manière conventionnelle, et profite de tous les avantages que procure une approche par composants. De plus, si un composant ne donne pas satisfaction, par exemple quant à ses performances, l'intégrateur est assuré de rapidement trouver un composant de remplacement, implémentant les mêmes interfaces.
- le propriétaire d'un composant métier (qui peut être soit le groupe de développement du composant métier, soit l'organisation en ayant mandaté le développement) peut rentabiliser son investissement

en réutilisant lui-même le composant métier dans le cadre de différents projets ou en le distribuant. La standardisation assure aux composants métier conformes une publicité certaine, l'autorité de contrôle-validation tenant à jour la liste des implémentations opérationnelles qui respectent les spécifications d'implémentation. Enfin, le fait de rendre publiques les interfaces des composants métier ne constitue pas un problème vis-à-vis de la concurrence : d'une part, la société dont les travaux ont conduit à la publication de spécifications dispose d'une avance sur ses concurrents, et son savoir-faire est principalement encapsulé et protégé au sein du composant. D'autre part, le fait de proposer des composants conformes à des "spécifications officielles" met les intégrateurs en confiance : ils se trouvent ainsi plus enclins à mettre en oeuvre ces composants.

- l'autorité de contrôle-validation a pour objectif d'optimiser les processus dans le domaine de la gestion des eaux et de coordonner les efforts allant dans ce sens. Elle agit de manière à ce que soit disponible, à terme, une collection, la plus complète possible, de composants métier dans laquelle les intégrateurs pourront puiser en fonction de leurs besoins. Le problème du financement de cette autorité, ainsi que des groupes d'intérêt qui assistent le développement de composants métier, reste toutefois ouvert : faut-il prélever un dédommagement pour l'enregistrement de spécifications, pour le test et l'agrégation de conformité de composants métier, une taxe sur les ventes de composants métier ayant été certifiés conformes aux spécifications ?

Un processus comme celui qui est proposé - lequel doit bien sûr être affiné - permet d'envisager une réelle collaboration entre les différentes catégories d'intervenants et la création à terme d'outils informatisés de gestion des eaux toujours plus complets et efficaces, et préserver en même temps les intérêts de chacun. Un tel processus peut certainement stimuler la mise en place d'un véritable marché de composants métier. La question se pose cependant de savoir à quelle échelle il doit être mis en place: le territoire helvétique n'est-il pas trop restreint pour que puisse se développer un tel marché ?

7.3 Conclusion

La présente recherche s'est fixée pour but de faciliter la mise en place d'outils informatisés d'aide à la gestion des eaux. Pour cela, les processus métier dans ce contexte et les exigences en regard de tels outils ont fait l'objet d'une analyse détaillée. Nous proposons de construire ces outils à partir de composants logiciels réutilisables, les composants métier. Véritables briques logicielles, de tels composants peuvent être assemblés pour construire des systèmes informatiques performants, fiables, ouverts, et ce dans des délais raisonnables et à des coûts moindres que par le biais d'approches classiques.

La manière de concevoir de tels composants métier a fait l'objet d'une recherche approfondie. En particulier une grande attention a été portée à la manière d'assurer à ces composants une indépendance aussi grande que possible, gage d'un grand potentiel de réutilisation ainsi que de modularité des outils logiciels les intégrant. L'interopérabilité entre composants métier est rendue possible par une indépendance entre interfaces et implémentations combinée à une standardisation des interfaces publiques proposées par ces composants. Cela autorise la construction de systèmes logiciels à partir de composants d'horizons divers.

Un apport essentiel de ce travail, outre la proposition d'utiliser des composants métier dans le domaine

de la gestion des eaux, est de suggérer la mise en oeuvre du potentiel intégrateur du territoire non plus seulement au niveau des modèles de données, mais au niveau du système logiciel lui-même, et plus particulièrement à celui des composants métier. Ces composants sont en effet beaucoup plus aptes à articuler entre elles les différentes visions des multiples intervenants, offrant à chacun les interfaces dont il a l'utilité et assurant la cohérence et l'intégration de ces différentes visions en arrière-plan. De plus, la possibilité de regrouper des concepts logiquement corrélés dans des composants à même d'interagir avec des composants tiers offre un moyen pour aborder la complexité inhérente au système naturel considéré. Ainsi, cette complexité, qui a des répercussions sur le système logiciel, peut bien mieux être maîtrisée et abordée.

L'intérêt d'une approche par composants métier démontré, une méthode de développement, baptisée COWIS, a été mise en place. Des prototypes ont permis d'effectuer un début de validation de la méthode proposée. Cependant, il est maintenant nécessaire, pour compléter cette validation et affiner la méthode, de mettre cette dernière à contribution dans un certain nombre d'applications. L'approche proposée a été mise sur pied dans le domaine de la gestion des eaux. Elle peut cependant être mise en oeuvre dans d'autres domaines, à partir du moment où des composantes du territoire sont à même de jouer le rôle d'éléments de fédération entre différents métiers.

La mise en place de structures destinées à promouvoir, favoriser et encadrer le développement de composants métier, à encourager leur utilisation est, dans le domaine de la gestion des eaux, indispensable. En effet, si l'utilité de développer des composants métier a été démontrée, d'une part ce processus nécessite la collaboration de nombreux intervenants, et d'autre part la fabrication de ces composants nécessite des investissements qui ne peuvent être rentabilisés que si leur diffusion est assurée. Un encouragement, un encadrement et une optimisation de l'approche par composants métier dans le domaine de la gestion des eaux est nécessaire.

Partant de cette constatation, nous avons ébauché un processus conduisant à la mise en place par étapes d'une véritable collection de composants métier (chapitre 7.2). Dans un premier temps seront conçus des composants qui assureront de manière efficace des fonctions indispensables, mais que les systèmes d'information actuels fournissent déjà (gestion de l'information relative aux infrastructures, fonctionnalités simples). Puis l'on verra apparaître des composants métier qui, s'appuyant sur ceux qui viennent d'être cités, offriront de nouveaux services, par exemple le calcul d'indicateurs, l'intégration de logiciels de simulation des comportements, des interactions avec des systèmes informatiques tiers, ou un appui intégral à des processus-métier. Une approche par composants métier est ainsi à même de conduire à une véritable valorisation des données et de faciliter les flux d'information entre intervenants au sein de ou entre différents niveaux organisationnels.

Une approche par composants métier constitue ainsi une réelle opportunité d'améliorer les processus de gestion des eaux, et de disposer de véritables moyens de suivi et de pilotage du système "Eaux en milieu urbain". Elle permet de construire, de manière à la fois décentralisée et coordonnée, des composants métier compatibles entre eux. Ceux-ci peuvent être assemblés pour former des outils informatisés spécifiquement adaptés aux besoins des intervenants tout en favorisant leur collaboration, et donc par ce biais une approche intégrée de la gestion des eaux.

Bibliographie

- Abbey, M., Corey, M.J., Abramson, I.**, 1999. *Oracle 8i, a beginners guide*. Mc Graw-Hill, Berkeley, 765 pp.
- AGW**, 1997. *Vollzugskonzept Siedlungsentwässerung VOKOS*. Bau-, Verkehrs- und Energiedirektion des Kantons Bern, 167 pp.
- Almeida, M.C., Butler, D., Friedler, E.**, 1999. *At-source domestic wastewater quality*. Urban Water, 1: 49-55.
- Ambler, S.W.**, 1998a. *The design of a robust persistence layer for relational databases*. White paper, AmbySoft Inc.
- Ambler, S.W.**, 1998b. *Mapping objects to relational databases*. White paper, AmbySoft Inc.
- Andro, T., Chauvet, J.-M.**, 1998. *Objets métier*. Eyrolles, Paris, 232 pp.
- Apothéloz, B., Stettler, A.**, 1995. *Maîtriser l'information comptable*. Collection Diriger l'Entreprise, Presses polytechniques et universitaires romandes, Lausanne, 380 pp.
- Arlow, J., Emmerich, W., Quinn, J.**, 1998. *Literate modelling - capturing business knowledge with the UML*. Proceedings <<UML>>'98.
- ASPEE, ASMFA**, 1990. *Norme suisse 592 000. Evacuation des eaux des biens-fonds. Conception et réalisation d'installations*. Association suisse des professionnels de l'épuration des eaux, Zurich.
- Azar, C., Holmberg, J., Lindgren, K.**, 1996. *Socio-ecological indicators for sustainability*. Ecological Economics, 18: 89-112.
- Baumann, R., Keller, S.F.**, 1998. *Anforderungen an ein GIS. Aus der Sicht des Schweizerischen Vereins des Gas- und Wasserfaches (SVGW)*. GWA(11/98): 859-866.
- Bédard, Y.**, 1982. *Recommandations relatives à l'implantation d'un système d'information sur le territoire pour des fins de gestion municipale*. Thèse, Université Laval, Québec.
- Bédard, Y.**, 1987. *Sur les différents types de systèmes d'information à référence spatiale*. Actes du congrès conjoint de Carto-Québec et de l'Association canadienne de cartographie, Québec: 73-87.
- Bédard, Y.**, 1999. *Principles of spatial database analysis and design*. In: P.A. Longley, M.F. Goodchild, D.J. Maguire, D.W. Rhind (Editors), *Geographical information systems*. John Wiley and Sons, New York, pp. 413-424.
- Bédard, Y.**, 2000. *Guide à l'usager du logiciel Perceptory 2000*. Université Laval, Centre de recherche en géomatique, Laval.
- Benagli, C.**, 1997. *Un modèle de données pour l'approvisionnement en eau potable*. Travail de séminaire. Département de génie rural, Ecole Polytechnique Fédérale, Lausanne.
- Bernasconi, D.**, 1999. *Rahmenkonzept zur Gestaltung eines Datenmanagementsystems Siedlungsentwässerung*. Dissertation. Eidgenössische technische Hochschule Zürich, 175 pp.
- Bernasconi, D., Fritschi, P., Schmid, A.**, 1998. *Datenmanagement Siedlungsentwässerung. Nutzungsanalyse. Katalog der Informationsobjekte*, Büro für Management-Technologie-Siedlungsplanung, Allschwill.
- Bernasconi, D., Gujer, W.**, 1998. *Integriertes Datenmanagement für den GEP*. GWA(3/98): 188-192.
- Bertziss, A.T.**, 1999. *Domain analysis for business software systems*. Information Systems, 24(7): 555-

568.

- Blanc, V., Clabaut, A.,** 1997. *Des exemples d'utilisation du SANDRE dans le bassin Rhin-Meuse*. Informatique pour l'environnement '97. Conférence européenne sur les technologies de l'information pour l'environnement, Strasbourg: 337-345.
- Boehm, B.W.,** 1988. *A spiral model of software development and enhancement*. IEEE Computer, 21(5): 61-72.
- Boller, M.,** 1997. *Tracking heavy metals reveals sustainability deficits of urban drainage systems*. Water Science and Technology, 35(9): 77-87.
- Booch, G., Rumbaugh, J., Jacobson, I.,** 1999. *The Unified Modeling Language user's guide*. Addison Wesley, 482 pp.
- Bourrier, R.,** 1985. *Les réseaux d'assainissement. Calculs, applications, perspectives*. Tec & Doc. Lavoisier, Paris, 482 pp.
- Bouzeghoub, M., Gardarin, G., Valduriez, P.,** 1997. *Les objets*. Eyrolles, 450 pp.
- Brashear, R.W., Promise, J., Roesner, L.A.,** 1999. *Utilizing the world wide web to organize and convey urban storm water management principles to diverse audiences*. Height International Conference on Urban Storm Drainage, Sydney, Australia, 30 August - 3 September 1999: 210-217.
- Brooks, F.P.,** 1996. *Le mythe du mois-homme. Essai sur le génie logiciel*. International Thomson Publishing France, 276 pp.
- Brunet, J.,** 1993. *Analyse conceptuelle orientée objet*. Thèse de 3ème cycle, Université de Nantes, 157 pp.
- Buogo, A.,** 1995. *Essai sur l'amélioration de la représentation des connaissances spatiales dans les SIRS par l'intégration de l'information*. Mémoire de maîtrise. Université de Laval, 150 pp.
- Butler, D., Parkinson, J.,** 1997. *Towards sustainable urban drainage*. Water Science and Technology, 35(9): 53-63.
- Cardoso, M.A., Coelho, S.T., Matos, J.S.S., Matos, R.S.,** 1999. *A new approach to the diagnosis and rehabilitation of sewerage systems through the development of performance indicators*. Height International Conference on Urban Storm Drainage, Sydney, Australia, 30 August - 3 September 1999: 610-617.
- Caron, C.,** 1991. *Nouveau formalisme de modélisation conceptuelle adapté aux SIRS*. Mémoire de maîtrise, Université Laval.
- Chen, J., Beck, M.B.,** 1997. *Towards designing sustainable urban wastewater infrastructures : a screening analysis*. Water Science and Technology, 35(9): 99-112.
- Chen, P.P.-S.,** 1976. *The Entity-Relationship Model - Toward a unified view of data*. ACM Transactions on Database Systems.
- Chow, V.T., Maidment, D.R., Mays, L.W.,** 1998. *Applied hydrology*. McGraw Hill Series in Water Resources and Environmental Engineering. McGraw Hill, 572 pp.
- CIPEL,** 1997. *Rapports sur les études entreprises dans le bassin lémanique, campagne 1996*. Commission internationale pour la protection des eaux du Léman, Lausanne.
- Cockburn, A.,** à paraître. *Writing effective use cases*. Addison Wesley.
- CODEAU,** 1996. *CODEAU. Logiciel de traitement de données hydrologiques. Guide de l'utilisateur*. Institut d'Aménagement des Terres et des Eaux (IATE/HYDRAM), Ecole Polytechnique

- Fédérale, Lausanne.
- Cohen, J., Steward, I.**, 1994. *The collapse of chaos. Discovering simplicity in a complex world.* Viking, New-York.
- Coleman, D., Arnold, P., Bodoff, S., Dollin, C., Gilchrist, M., Hayes, F., Jeremaes, P.**, 1994. *Object-oriented development. The FUSION method.* Prentice Hall Object Oriented Series. London, 314 pp.
- Coleman, D.J.**, 1999. *Geographical information systems in networked environments.* In: P.A. Longley, M.F. Goodchild, D.J. Maguire, D.W. Rhind (Editors), *Geographical information systems.* John Wiley and Sons, New York, pp. 317-329.
- Collongues, A., Hugues, J., Laroche, B.**, 1989. *Merise. 1. Méthode de conception.* Bordas, Paris, 240 pp.
- Crausaz, P.-A.**, 2000. *Du rôle intégrateur des systèmes d'information à référence spatiale dans la gestion institutionnelle des eaux : analyse, méthode, limites et perspectives. Application à l'Administration cantonale vaudoise.* Thèse de doctorat. Ecole Polytechnique Fédérale de Lausanne, 279 pp.
- Crausaz, P.-A., Musy, A.**, 1996. *GESREAU: un outil d'aide à la gestion des eaux appliquée au canton de Vaud.* Vermessung, Photogrammetrie, Kulturtechnik.
- Crausaz, P.-A., Musy, A.**, 1997. *GESREAU, un outil de gestion des eaux par une modélisation du territoire.* Revue internationale de géomatique, 7(2): 127-139.
- Da Costa, J.R., Jesus, H.B., Lacerda, M.**, 1996. *Integrating GIS and time series analysis for water resources management in Portugal.* HydroGIS 96: Application of Geographic Information Systems in Hydrology and Water Resources Management, Vienna, April 1996, 235: 289-297.
- Dano, B.**, 1997. *Une démarche d'ingénierie des besoins orientée objet guidée par les cas d'utilisation.* Thèse de doctorat, Université de Nantes, Nantes, 157 pp.
- Dano, B., Briand, H., Barbier, F.**, 1997. *A use case driven requirements engineering process.* Requirements Eng, 2: 79-91.
- Date, J.C.**, 1999. *An introduction to database systems.* 7th ed., Addison-Wesley, 975 pp.
- Davoli, F.**, 2000. *Plan général d'évacuation des eaux (PGEE) de l'EPFL-UNIL.* Mémoire de diplôme. Département de génie rural, Ecole Polytechnique Fédérale de Lausanne.
- de Heer, J., Wyss, A.**, 1995. *Archivage des données de pluies.* GWA, 3/95: 213-218.
- de Rosnay, J.**, 1977. *Le microscope.* Collection Essais. Le Seuil, 346 pp.
- de Souza, P., Consuegra, D., Musy, A.**, 1994. *CODEAU: a database package for the assessment and analysis of hydrometeorological data.* In: A. Verwey, A.W. Minns, V. Babovic (Editors), *Hydroinformatics '94. Proceedings of the first international conference on hydroinformatics,* Delft, Netherlands, 19-23 september 1994: 103-110.
- Degrémont**, 1990. *Mémento technique de l'eau.* Degrémont, 1596 pp.
- Descartes**, 1637. *Discours de la méthode.* Livre de Poche.
- Durant-Dastes, F.**, 1992. *Les eaux douces. Abondances, sécheresses et conflits.* Planète verte.
- Edwards, J., Clements, P., Gascoigne, J., Coutts, I.**, 1998. *Component-based systems. The basis of future manufacturing systems.* Component-based software development. Managing object technology. Cambridge University Press, Cambridge, pp. 105-110.

- Eeles, P., Sims, O.**, 1998. *Building business objects*. Wiley Computer Publishing. John Wiley and Sons, Inc, New-York, 342 pp.
- Firesmith, D.G.**, 1999. *Use case modeling guidelines*. TOOLS, Technology of Object-Oriented Languages and Systems: 184-193.
- Fischer, M.M., Nijkamp, P.**, 1993. *Design and use of geographic information systems and spatial models. Geographic information systems, spatial modelling and policy evaluation*. In: M.M. Fischer, P. Nijkamp (Editors). Springer Verlag, pp. 1-13.
- Fowler, M.**, 1997. *Analysis patterns: reusable object models*. The Addison-Wesley series in object-oriented software engineering. Addison-Wesley, 357 pp.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J.**, 1995. *Design patterns. Elements of reusable object-oriented software*. Professional Computing Series. Addison-Wesley, 395 pp.
- Gane, C., Sarson, T.**, 1979. *Structured systems analysis: tools and techniques*. Prentice-Hall Software Series. Prentice-Hall, 241 pp.
- Gardels, K.**, 1996. *The Open GIS approach to distributed geodata and geoprocessing*. Third International Conference on Integrating GIS and Environmental Modeling, Santa Fe, New Mexico, USA, 21-25 January 1996.
- Gayte, O., Libourel, T., Cheylan, J.P., Lardon, S.**, 1996. *POLLEN. Méthode de conception des systèmes d'information sur l'environnement*. IARE, Montpellier, 109 pp.
- Geldof, G.D.**, 1995. *Adaptive water management: integrated water management on the edge of chaos*. Water Science and Technology, 32(1): 7-13.
- Geldof, G.D.**, 1997. *Coping with uncertainties in integrated urban water management*. Water Science and Technology, 36(8-9): 265-269.
- GEOEurope**, 2000. *Thames Water gets a clearer Vision*. GEOEurope(June): 12.
- GERMINAL**, 1995. *Approche et méthodes pour la réalisation de Systèmes d'Information à Référence Spatiale*. Rapport. Ecole Polytechnique Fédérale de Lausanne.
- Golay, F.**, 1992. *Modélisation des systèmes d'information à référence spatiale et de leurs domaines d'utilisation spécialisés. Aspects méthodologiques, organisationnels et technologiques*. Thèse de doctorat. Ecole Polytechnique Fédérale de Lausanne, 95 pp.
- Grauer, M., Schreiber, D., Michels, I.**, 1996. *The analysis of time-series in watermanagement - Data modelling and implementation issues*. In: Müller (Editor) Hydroinformatics'96: 187-193.
- Grottke, M., Otterpohl, R.**, 1996. *Integrated urban water concept*. In: F.S.a.H.-R. Verworn (Editor), Seventh International Conference on Urban Storm Drainage, Hannover, Germany, September 9-13, 1996: 1801-1806.
- GSA**, 1994. *Infiltration et rétention des eaux pluviales*. Direction des travaux publics, des transports et de l'énergie du canton de Berne, 32 pp.
- Gupta, S., Scummiotales, J.**, 1998. *Integrating objects and relational technologies*. White paper, Vigor Technology.
- Gustafsson, L.-G., Winberg, S., Refsgaard, A.**, 1996. *Towards a distributed physically based model description of the urban aquatic environment*. In: F.S.a.H.-R. Verworn (Editor), Seventh International Conference on Urban Storm Drainage, Hannover, Germany, September 9-13, 1996: 1467-1472.
- Harremoës, P.**, 1997. *Integrated water and waste management*. Water Science and Technology, 35(9):

11-20.

- Harvatopoulos, Y., Livian, Y.-F., Sarnin, P.,** 1989. *L'art de l'enquête. Guide pratique*. Eyrolles.
- Hermann, E.,** 1998. *Datenerhebung und Kanalnetz-Simulationstechnik*. GWA(3/98): 178-183.
- Herzum, P., Sims, O.,** 2000. *Business component factory. A comprehensive overview of component-based development for the enterprise*. Wiley Computer Publishing, 579 pp.
- Hislop, G.W.,** 1998. *Analysing existing software for software reuse*. The Journal of Systems and Software, 41: 33-40.
- Jacobson, I., Booch, G., Rumbaugh, J.,** 2000. *Le Processus unifié de développement logiciel*. Technologies objet. Eyrolles, Paris, 488 pp.
- Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G.,** 1992. *Object-oriented software engineering. A use case driven approach*. Addison-Wesley, 524 pp.
- Jacobson, I., Griss, M., Jonsson, P.,** 1997. *Software reuse. Architecture, process and organization for business success*. Addison Wessley Longman, New York, 497 pp.
- Jalade, A.,** 1998. *Elaboration du cahier des charges d'un SIRS destiné à la gestion et au contrôle des sources petentielles de pollution des eaux*. Travail de séminaire. Département de génie rural, Ecole Polytechnique Fédérale, Lausanne.
- Kaiser, M.,** 1997. *Requirements and possibilities of best management practises for storm water run-off from the view of ecological townplanning*. Water Science and Technology, 36(8-9): 319-323.
- Keller, S.F., Thalmann, H.,** 1999. *Modeling and sharing graphic presentations of geospatial data*. In: A. Vckovski, K.E. Brassel, H.-J. Schek (Editors), *Interoperating geographic information systems. Proceedings of the Second International Conference, INTEROP'99. Lecture notes in computer science*. Springer, Zurich, Switzerland, pp. 151-162.
- Keller, W.,** 1997. *Mapping objects to tables : a pattern language*. Second European Conference on Pattern Languages of Programming and Computing (Eurolop '97), Bad Irsee, Germany.
- Keller, W.,** 1998. *Object/relational access layers. A road map, missing links and more patterns*. Third European Conference on Pattern Languages of Programming and Computing (EuroPlop '98), Bad Irsee, Germany, 9 - 11 July 1998.
- Keller, W., Coldewey, J.,** 1996. *Relational database access layers. A pattern language*. The Joint Pattern Languages of Programs Conferences (PloP '96), Allerton Park, Illinois, Sept. 4-6, 1996.
- Kettani, N., Mignet, D., Paré, P., Rosenthal-Sabroux, C.,** 1998. *De Merise à UML*. Eyrolles, 433 pp.
- Korwing, J.L., Wiggers, J.B.M.,** 1999. *Information needs for strategic management of water systems in the urban environment*. Height International Conference on Urban Storm Drainage, Sydney, Australia, 30 August-3 September 1999: 379-386.
- Kotler, P., Dubois, B.,** 1997. *Marketing management*. Publi-Union, 741 pp.
- Krebs, P., Larsen, T.,** 1997. *Guiding the development of urban drainage systems by sustainability criteria*. Water Science and Technology, 35(9): 89-98.
- Krejci, V., Schilling, W.,** 1992. *Einführung in die moderne Siedlungsentwässerung. Grundkurs : Modelle Siedlungshydrologischer Prozesse*. EAWAG, Dübendorf.
- Kurata, D.,** 1998. *Doing objects in Visual Basic 6, the autoritative solution*. SAMS Publishing, 642 pp.
- Lam, D.C.L., Swayne, D.A., Mayfield, C.I., Cowan, D.D.,** 1996. *Integration of GIS with other software systems: integration versus interconnection*. Third International Conference on

- Integrating GIS and Environmental Modeling, Santa Fe, New Mexico, USA, 21-25 January 1996.
- Langeveld, J.G., Wiggers, J.B.M.,** 1999. *Appropriate urban water systems: an approach for generating and screening promising concepts*. Height International Conference on Urban Storm Drainage, Sydney, Australia, 30 August - 3 September: 137-142.
- Larman, C.,** 1997. *Applying UML and patterns : an introduction to object oriented analysis and design*. Prentice Hall, 507 pp.
- Larsen, L.,** 1999. *GIS in environmental monitoring and assessment*. In: P.A. Longley, M.F. Goodchild, D.J. Maguire, D.W. Rhind (Editors), *Geographical information systems*. John Wiley and Sons, New York, pp. 999-1007.
- Larsen, T.A., Gujer, W.,** 1996. *Separate management of anthropogenic nutrient solutions*. Water Science and Technology, 34(3-4): 87-94.
- Larsen, T.A., Gujer, W.,** 1997. *The concept of sustainable urban water management*. Water Science and Technology, 35(9): 3-10.
- Larsen, T.A., Gujer, W.,** 2000. *Waste design and source control lead to flexibility in wastewater management*. Accepted for the First World Congress of the International Water Association, Paris, 3-6 July 2000.
- Laurini, R., Thompson, D.,** 1992. *Fundamentals of spatial information systems*. The APIC Series. Academic Press, 670 pp.
- Lawrence, A.I., Ellis, J.B., Marsalek, J., Urbonas, B., Phillips, B.C.,** 1999. *Total urban water cycle based management*. Height International Conference on Urban Storm Drainage, Sydney, Australia, 30 August - 3 September 1999: 1142-1149.
- Le Moigne, J.-L.,** 1984. *La théorie du système général. Théorie de la modélisation*. Systèmes-Décisions. Presses Universitaires de France, 320 pp.
- Le Moigne, J.-L.,** 1990. *La modélisation des systèmes complexes*. Bordas, Paris, 178 pp.
- Leffingwell, D., Widrig, D.,** 2000. *Managing software requirements : a unified approach*. Object Technologies Series. Addison Wesley, 528 pp.
- Lewis, T.,** 1999. *VB COM*. Wrox Press, 345 pp.
- Lhotka, R.,** 1998. *Visual Basic 6 business objects*. Wrox Press, 640 pp.
- Lim, W.C.,** 1998. *Managing software reuse. A comprehensive guide to strategically reengineering the organization for reusable components*. Prentice Hall PTR, 552 pp.
- Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W.,** 1999. *Epilogue: seeking out the future*. In: P.A. Longley, M.F. Goodchild, D.J. Maguire, D.W. Rhind (Editors), *Geographical Information Systems: principles, techniques, applications and management*. John Wiley & Sons, New York, pp. 1009-1021.
- Luiten, J.P.A.,** 1995. *The water system exploration - a new dutch project (the aquatic outlook) for combining monitoring, research and policy analysis for integrated water management*. Water Science and Technology, 31(8): 329-344.
- Lundin, M., Molander, S., Morrison, G.M.,** 1997. *Indicators for the development of sustainable water and wastewater systems*. Sustainable development research conference, Manchester, April 1997.
- Lundin, M., Molander, S., Morrison, G.M.,** 1999. *A set of indicators for the assessment of temporal*

- variations in the sustainability of sanitary systems*. Water Science and Technology, 39(5): 235-242.
- Maguire, D.J.**, 1999. *GIS customisation*. In: P.A. Longley, M.F. Goodchild, D.J. Maguire, D.W. Rhind (Editors), *Geographical information systems*. John Wiley and Sons, New York, pp. 359-369.
- Makrygiannis, N.**, 1998. *Toward mass-customized information systems*. Component-based software development. Managing object technology. Cambridge University Press, Cambridge, pp. 91-104.
- Mallaun, T., Burgermeister, W.**, 1994. *Das Kanalinformationssystem KIS der Stadtentwässerung Zürich*. GWA(11/94): 943-952.
- Managlia, D.**, 1998. *Base de données "Forages". Proposition d'une base de données à références spatiales pour la gestion et l'exploitation des données "Forages" du Canton de Genève. Application à un secteur du bassin versant du Nant D'Avril*. Travail de séminaire. Département de génie rural, Ecole Polytechnique Fédérale, Lausanne.
- MapInfo Corporation**, 1999a. *MapX developer's guide*. Troy, New-York, 347 pp.
- MapInfo Corporation**, 1999b. *MapX reference guide*. Troy, New-York, 467 pp.
- Matheron, J.-P.**, 1991. *Approfondir MERISE. Tome 1. Modèles conceptuels de données ou la représentation statique du réel*. Eyrolles, Paris, 284 pp.
- Maystre, L.-Y.**, 1984. *L'épidémie de typhoïde à Zermatt : une leçon de l'histoire à ne pas oublier*. Document de cours, Ecole Polytechnique Fédérale de Lausanne.
- McFadden, F.R., Hoffer, J.**, 1993. *Modern database management*. Benjamin Cummings, Redwood City, 596 pp.
- Meadows, D.H., Meadows, D.L., Randers, J.**, 1972. *The limits to growth : A report for the Club of Rome's project on the predicament of mankind*. Universe Books, New-York.
- Merz, P., Gujer, W.**, 1997. *Fliessgewässer und die Einwirkungen der Siedlungsentwässerung*. GWA(3): 181-192.
- META Group**, 1995. *Making the case for use case*. Advanced management, File 324 (13 february 1995).
- Michelbach, S.**, 1996. *Interaction between combined sewer system, waste water treatment plant and receiving water in a rural catchment*. In: F.S.a.H.-R. Verworn (Editor), *Seventh International Conference on Urban Storm Drainage*, Hannover, Germany, September 9-13, 1996: 1187-1192.
- Mikkelsen, P.S. et al.**, 1996. *Pollution of soil and groundwater from infiltration of highly contaminated stormwater - a case study*. In: F.S.a.H.-R. Verworn (Editor), *Seventh International Conference on Urban Storm Drainage*, Hannover, Germany, September 9-13, 1996: 707-712.
- Mottier, V.**, 1999. *DWSL, un composant logiciel pour systèmes informatisés de gestion de réseaux d'assainissement. Mode d'emploi*. Rapport. Département de génie rural, Ecole Polytechnique Fédérale de Lausanne, Lausanne.
- Mottier, V., Boller, M.**, 1992. *Les eaux de ruissellement de toits : qualité et dynamique de la charge polluante*. Rapport. EAWAG, Dübendorf.
- Mull, R.**, 1996. *Water exchange between leaky sewers and aquifers*. In: F.S.a.H.-R. Verworn (Editor), *Seventh International Conference on Urban Storm Drainage*, Hannover, Germany: 695-700.
- Musy, A., Laglaine, V.**, 1992. *Hydrologie générale*. Institut d'Aménagement des Terres et des Eaux, Ecole Polytechnique Fédérale, Lausanne.
- Mylopoulos, J.**, 1995. *Conceptual modelling for information systems engineering*. University of

- Namur, Namur.
- Niemczynowicz, J.**, 1999. *Urban hydrology and water management - present and futur challenges*. Urban Water, 1: 1-14.
- Noy, A.**, 1990. *Théorie des systèmes d'organisation*. Cours CNAM.
- Oestereich, B.**, 1998. *Objektorientierte Geschäftsprozessmodellierung mit der UML*. OBJECTSpektrum, 2/1998.
- OFS/OFEFP**, 1997. *L'environnement en Suisse 1997 - chiffres, faits, perspectives*. Office fédéral de la statistique, Office fédéral de l'environnement, des forêts et du paysage, 370 pp.
- OMG**, 1999. *OMG Unified Modeling Language Specification. Version 1.3*. OMG Document bom/97-12-04, Framingham, U.S.A.
- OMG**, 1995. *OMG business object application architecture*. White paper, OMG.
- OMG**, 1997. *Business Object DTF. Common Business Objects*. OMG Document bom/97-12-04, Framingham, U.S.A.
- Open GIS Consortium**, 1998a. *Open GIS Simple feature specification for CORBA. Revision 1.0*.
- Open GIS Consortium**, 1999a. *Open GIS Simple feature specification for OLE/COM. Revision 1.1*.
- Open GIS Consortium**, 1999b. *The OpenGIS abstract specification*. Kurt Buehler (Editor).
- Open GIS Consortium**, 1998b. *The OpenGIS guide. Introduction to interoperable geoprocessing and the OpenGIS specification*. Kurt Buehler (Editor).
- Otterpohl, R., Grottker, M., Lange, J.**, 1997. *Sustainable water and waste management in urban areas*. Water Science and Technology, 35(9): 121-133.
- Pantazis, D., Donnay, J.-P.**, 1996. *La conception de SIG, méthode et formalisme*. Collection Géomatique, Paris, 342 pp.
- Parent, C.**, 1996. *MADS, un modèle de données pour modéliser les applications à données spatiales*. Rapport. Département d'informatique, Ecole Polytechnique Fédérale de Lausanne.
- Parent, C., Spaccapietra, S., Zimányi, E.**, 1999a. *Spatio-temporal conceptual models: data structures + space + time*. 7th ACM Symposium on Advances in GIS, Kansas City, November 5-6, 1999.
- Parent, C. et al.**, 1999b. *MADS ou l'information spatio-temporelle à portée de ses utilisateurs*. Quatrièmes rencontres de ThéoQuant, Besançon, France, Février 1999.
- Payne, J., Gardiner, J.**, 1996. *Sustainable urban runoff management*. In: F.S.a.H.-R. Verworn (Editor), Seventh International Conference on Urban Storm Drainage, Hannover, Germany, September 9-13, 1996: 1635-1640.
- Petit Robert**, 1996. *Le nouveau Petit Robert. Dictionnaire alphabétique et analogique de la langue française*. Dictionnaires Le Robert, Paris.
- Pfister, C., Szyperski, C.**, 1998. *Why objects are not enough. Component-based software development*. Managing object technology. Cambridge University Press, Cambridge, pp. 57-66.
- Pornon, H.**, 1998. *Systèmes d'information géographique, pouvoir et organisations. Géomatique et stratégies d'acteurs*. L'Harmattan, 255 pp.
- Pouliot, J.**, 1999. *Définition d'un cadre géosémantique pour le couplage des modèles prévisionnels de comportement et des SIG - Application pour les écosystèmes forestiers*. Thèse de doctorat. Ecole Polytechnique Fédérale, Lausanne, 171 pp.

- Prélaz-Droux, R.**, 1995a. *Conception d'un système d'information à référence spatiale pour l'aménagement et la gestion du territoire. Approche systémique et procédure de réalisation*. Thèse. Ecole Polytechnique Fédérale de Lausanne, Lausanne, 172 pp.
- Prélaz-Droux, R.**, 1995b. *Système d'information et gestion du territoire*. Collection Méta. Presses Polytechniques et Universitaires Romandes, Lausanne, 156 pp.
- Pressman, R.S.**, 1997. *Software engineering*. McGraw-Hill, 4th ed., 852 pp.
- Prins, J.G., Reil, J.D., Madison, M.J., Swimley, C.E.**, 1998. *Developing a sewer master plan using ArcView GIS*. ArcUser (July-September 1998): 26-29.
- Rakotondrafara, J.-L.**, 1999. *Contribution à la conception d'un outil informatique d'aide à la simulation des écoulements en zones urbaines. Application à l'étude du réseau d'assainissement de Frontenex*. Mémoire de diplôme. Département de génie rural, Ecole Polytechnique Fédérale, Lausanne.
- Rational Software Inc.**, 2000. *Mapping object to data models with the UML*. White paper.
- Resweber, J.-P.**, 1995. *La recherche-action*. Que sais-je. Presses Universitaires de France.
- Revaz, J.M., Storelli, S., Claivaz, S., Dion, Y.**, 2000. *L'approche systémique, les télétransmissions, la micro-informatique au service du génie urbain*. DM in UCE 2000, 2ème conférence internationale sur l'aide à la décision., Lyon, 20-22 novembre 2000, 1: 41-50.
- Riedo, M., Mottier, V.**, à paraître. *Langages de développement*. Traité IGAT : Information géographique et aménagement du territoire. Interfaces homme/machine et langage. Hermès, Paris.
- Rijsberman, M.A., Ven, F.M.H.v.d.**, 1999. *Concepts and approaches to sustainable development in urban water management*. Height International Conference on Urban Storm Drainage, Sydney, Australia, 30 August - 3 September 1999: 42-49.
- Rine, D.C., Sonnemann, R.M.**, 1998. *Investment in reusable software. A study of software reuse investment success factors*. The Journal of Systems and Software, 41: 17-32.
- Robertson, S., Robertson, J.**, 1999. *Mastering the requirements process*. Addison-Wesley, New-York, 352 pp.
- Rolland, C.**, 1986. *Introduction à la conception de systèmes d'information et panorama des méthodes disponibles*. Génie logiciel(4).
- Rossi, L.**, 1998. *Qualité des eaux de ruissellement urbaines*. Thèse de doctorat. Ecole Polytechnique Fédérale de Lausanne, 313 pp.
- Rossi, L.**, 1999. *Accord de participation sur l'assainissement du plateau de Frontenex. Etude du rendement d'un séparateur particulaire*. Rapport technique. DGR-IGE / EGS, Ecole Polytechnique Fédérale, Lausanne.
- Rumbaugh, J.**, 1994. *OMT. Modélisation et conception orientées objets*. Prentice Hall, 516 pp.
- Rumbaugh, J., Jacobson, I., Booch, G.**, 1999. *The Unified Modeling Language Reference Manual*. Object Technology Series. Addison-Wesley, Reading, 550 pp.
- Sawatzki, J.**, 1996. *Assessing the value of sewers and drains*. World Water and Environmental Engineering: 22-23.
- Schach, S.R.**, 1996. *Classical and object-oriented software engineering*. McGraw-Hill, 3rd edition.
- Schmid, H.A.**, 1999. *Business entity and process components*. In: Springer (Editor) OOPSLA'99, Business Object Design and Implementation, Denver, USA, 2 novembre 1999: 131-145.

- Scholl, M., Voisard, A., Peloux, J.P., Raynald, L., Rigaux, P., 1996.** *SGBD géographiques. Spécificités*. International Thomson Publishing. 185 pp.
- Schütze, M., Butler, D., Beck, M.B., 1996.** *Development of a framework for the optimisation of runoff, treatment and receiving waters*. In: F.S.a.H.-R. Verworn (Editor), Seventh International Conference on Urban Storm Drainage, Hannover, Germany: 1419-1424.
- Schuurmans, W., Nelen, A.J.M., 1996.** *Hydroinformatic framework for urban and rural water systems*. In: Müller (Editor) Hydroinformatics'96, Balkema, Rotterdam: 251-256.
- Sendall, S., Strohmeier, A., 2000.** *From use cases to system operation specifications*. In: S. Kent and A. Evans (Editors), UML'2000, Third International Conference, York, UK, 1999: 1-15.
- Serain, D., 1997.** *Le middleware. Concepts et technologies*. Collection Systèmes Distribués. Masson, Paris, 214 pp.
- Shekhar, S., Vatsavai, R.R., Chawla, S., Burk, T.E., 1999.** *Spatial pictogram, enhanced conceptual data models and their translation to logical data models*. In: P. Agouris, A. Stefanidis (Editors), Integrated spatial databases. International Workshop ISD'99. Lecture Notes in Computer Science. Springer, pp. 77-102.
- SIA, 1999.** *GEO405. Catalogue des données et des représentations pour les conduites de réseaux souterrains*. Zurich.
- Sieker, F., 1998.** *On-site stormwater management as an alternative to conventional sewer systems: a new concept spreading in Germany*. Water Science and Technology, 38(10): 65-71.
- Somlyódy, L., 1995.** *Water quality management: can we improve integration to face future problems ?* Water Science and Technology, 31(8): 249-259.
- Sommerville, I., 1995.** *Software engineering*. International Computer Science Series. Addison Wesley, 5th edition, 742 pp.
- Spaccapietra, S., Parent, C., 1992.** *ERC+: an object based entity relationship approach*. In: P. Loucopoulos and R. Zicari (Editors), Conceptuel Modelling, Databases and CASE: An integrated View of information Systems Development. John Wiley.
- Spaccapietra, S., 1996.** *Bases de données. Volume 1*. Document de cours. Ecole Polytechnique Fédérale, Lausanne.
- Spring, U., Kaufmann, U., 1998.** *GIS - ein Gefäss für den GEP. Visualisierung von GEP-Resultaten*. GWA(3/98): 184-187.
- SSIGE, 1992.** *Statistique 1990 : Résultats statistiques des services des eaux en Suisse*. Société Suisse de l'Industrie du Gaz et des Eaux, Zurich, 389 pp.
- Stephens, W., Hess, T., 1999.** *Systems approaches to water management research*. Agricultural Water Management, 40: 3-13.
- Stoiber, S., 1999.** *Datenbankanbindungsmuster - Untersuchung alternativer Ansätze und Realisierung im Rahmen eines verteilten Kalender-Managers*. Diplomarbeit. Johannes Kepler Universität, Institut für angewandte Informatik, Abteilung für Informationssysteme, Linz.
- Strohmeier, A., 1996.** *Cycle de vie du logiciel*. In: A. Strohmeier, D. Buchs (Editors), Génie logiciel : principes, méthodes et techniques. Presses Polytechniques et Universitaires Romandes, Lausanne, pp. 1-28.
- Strohmeier, A., 1999.** *Notes du cours de génie logiciel*, Ecole Polytechnique Fédérale, Département d'informatique, Laboratoire de génie logiciel, Lausanne.

- Sturm, J.**, 1999. *VB6 UML, design and development using frameworks and patterns*. Programmer to programmer. Wrox Press. 581 pp.
- Sutherland, J.**, 1999. *The emergence of a business object component architecture*. In: Springer (Editor) OOPSLA'99, Business Object Design and Implementation, Denver, USA, 2 november 1999: 39-59.
- Szyperski, C.**, 1997. *Component software. Beyond object-oriented programming*. ACM Press Books, New-York, 411 pp.
- Tabourier, Y.**, 1986. *De l'autre coté de Merise. Systèmes d'information et modèles d'entreprise*. Les Editions d'Organisation.
- Tanrikorur, T.**, 1999. *Concepts for simpler design*. Software Development (June 99).
- Taylor, D.A.**, 1995. *Business engineering with object technology*, New-York, 188 pp.
- Tercier, P.**, 1989. *Droit 1. Introduction générale au droit*. Polycopié de cours. Ecole Polytechnique Fédérale de Lausanne, Lausanne, 47 pp.
- Thériault, M.**, 1996. *Systèmes d'information géographique*. LATIG, Département de Géographie, Université de Laval, 165 pp.
- Trauth, R., Xanthopoulos, C.**, 1996. *Non-point pollution of groundwater in urban areas*. In: F.S.a.H.-R. Verworn (Editor), Seventh International Conference on Urban Storm Drainage, Hannover, Germany, September 9-13, 1996: 701-706.
- United Nations**, 1992. *Agenda 21 : program of action for sustainable development*. New-York.
- Valiron, F., Affholder, M.**, 1994. *Guide de conception et de gestion des réseaux d'assainissement unitaires*. Tec & Doc. Lavoisier, Paris, 374 pp.
- van der Graaf, J.H.J.M., Meester-Broertjes, H.A., Bruggeman, W.A., Vles, E.J.**, 1997. *Sustainable technological development for urban-water cycles*. Water Science and Technology, 35(10): 213-230.
- van Lier, J.B., Lettinga, G.**, 1999. *Appropriate technologies for effective management of industrial and domestic waste water: the decentralised approach*. Water Science and Technology, 40(7): 171-183.
- Vangenot, C.**, 1998. *Concepts pour la description de bases de données multi-représentation*. Revue internationale de géomatique, 8(1-2): 121-147.
- Varis, O., Somlyódy, L.**, 1997. *Global urbanization and urban water : can sustainability be afforded ?* Water Science and Technology, 35(9): 21-32.
- VSA**, 1990. *Plan général d'évacuation des eaux. Directives concernant l'élaboration et les honoraires*. Verband Schweizer Abwasser- und Gewässerschutzfachleute, Zurich.
- VSA**, 1992. *Plan général d'évacuation des eaux (PGEE). Manuel d'explication*. Verband Schweizer Abwasser- und Gewässerschutzfachleute, Zurich.
- VSA**, 1994. *Directive concernant le financement de l'assainissement au niveau des communes et de leurs groupements*. Verband Schweizer Abwasser- und Gewässerschutzfachleute, Zurich.
- VSA**, 1999. *Datenstruktur Siedlungsentwässerung (VSA-DSS)*. Verband Schweizer Abwasser- und Gewässerschutzfachleute, Zurich.
- VSA, OFEFP**, 1999. *Le plan régional d'évacuation des eaux (PREE), élément de la gestion des eaux*. Verband Schweizer Abwasser- und Gewässerschutzfachleute, Zurich.

- Vu, T., Dar-Biau, L.,** 1997. *A procurement-centric model for engineering component-based software systems*. Proceedings of the 5th International Symposium on Assessment of Software Tools (SAST '97), June 3-5, 1997.
- Warmer, J., Kleppe, A.,** 1998. *The Object Constraint Language. Precise modeling with UML*. Object Technology Series. Addison Wesley, New York, 112 pp.
- Williams, S., Kindel, C.,** 1994. *The Component Object Model : a technical overview*. White paper, Microsoft Corporation.
- Wisserhof, J.,** 1995. *Enhancing research utilization for integrated water management*. Water Science and Technology, 31(8): 311-319.
- Wolf-Schumann, U., Vaillant, S.,** 1996. *TimeView: a time series management system for GIS and hydrological systems*. HydroGIS 96: Application of Geographic Information Systems in Hydrology and Water Resources Management, Vienna, April 1996, 235: 79-87.
- World Commission on Environment and Development,** 1987. *Our common future*, Oxford University Press, Oxford.
- Wright, P.,** 1998. *Beginning Objects with Visual Basic 5*. Programmer to programmer. Wrox Press, 352 pp.
- Yoder, J.W., Johnson, R.E., Wilson, Q.D.,** 1998. *Connecting business objects to relational databases*. The 1998 Pattern Languages of Programs Conference (PLoP '98), Monticello, Illinois, USA, August 11-14, 1998.

Annexes

Annexe 1	Description des processus métier.....	198
Annexe 2	Les formalismes UML	205
Annexe 3	COM et ses évolutions.....	211
Annexe 4	Systèmes d'information et architectures en couches.....	214
Annexe 5	Principaux types de processus de développement	217
Annexe 6	Notions d'acteur et de cas d'utilisation.....	222
Annexe 7	Infrastructures de persistance	225
Annexe 8	La vision de l'Open GIS Consortium (OGC).....	227
Annexe 9	Le composant logiciel SIG MapX	231
Annexe 10	Exemples de cas d'utilisation.....	235

Annexe 1 Description des processus métier

Dans cette annexe sont décrits les processus-métier qui sont menés par les différents intervenants rencontrés dans le cadre de cette recherche. Ils y sont classés par domaines d'activité. A noter que le calcul d'indicateurs, qui n'est décrit que sous une seule rubrique, est un processus-métier que l'on retrouve dans différentes catégories d'activités : on rencontre en effet des indicateurs financiers, environnementaux, ainsi que des indicateurs relatifs aux infrastructures ou au comportement des ménages.

• *La planification générale*

La réalisation de PGEE et de PREE : ces deux instruments de planification pour la gestion des eaux (voir chapitre 2.2.2), le premier à l'échelle communale, le second à l'échelle régionale, représentent à l'heure actuelle une part très importante des activités menées par les bureaux d'ingénieurs travaillant dans le domaine de l'eau.

Le processus d'établissement d'un PGEE implique l'élaboration d'un document qui constituera *"pour la collectivité publique concernée la ligne directrice à suivre lors de la planification, la construction et l'exploitation de l'évacuation des eaux des agglomérations"* (VSA, 1990). Les documents de base d'un tel plan sont constitués du cadastre des eaux usées, ce terme recouvrant des informations sur les stations d'épuration, les canalisations, les industries et entreprises d'artisanat, les bâtiments d'habitation, et des rapports d'état relatifs aux cours d'eau, aux canalisations, aux eaux claires parasites, à l'infiltration, aux bassins versants, aux zones de danger, ainsi qu'aux débits d'eaux usées par temps sec et par temps de pluie dans le réseau d'assainissement. Sur la base de ces renseignements est proposée la planification de l'assainissement au niveau communal, soit un descriptif du concept futur en matière d'assainissement communal ainsi que divers avants-projets (VSA, 1992).

Un PREE est *"un plan d'action visant à harmoniser, coordonner et optimiser les coûts des mesures à prendre dans un bassin versant hydrologique. Il porte sur des bassins versants hydrologiques complets"* (VSA et OFEFP, 1999). Un PREE a donc pour objectif de coordonner différents PGEE qui sont localisés dans la même unité hydrologique, et donc se base principalement sur les informations présentes dans les PGEE. Un PREE contient de l'information sur l'état des cours d'eau, les sources de pollution, sur le territoire et les mesures de protection contre les crues, et enfin sur l'exploitation des eaux (souterraines, loisirs, transports).

L'établissement d'un PGEE est une tâche du ressort de la commune, confiée généralement à un bureau d'ingénieurs, tandis qu'un PREE est du ressort du service cantonal en charge de la protection des eaux, ses partenaires pouvant être des associations intercommunales, voire des communes.

Planification du développement des réseaux d'approvisionnement en eau potable : il existe une planification générale de l'approvisionnement en eau potable. Cet aspect n'est cependant pas ressorti des entretiens qui ont été menés. Il n'empêche que la planification du développement des réseaux d'approvisionnement en eau potable est une activité qui est mentionnée dans les documents étudiés.

Le calcul et l'acquisition d'indicateurs divers : le suivi d'indicateurs est nécessaire pour assurer de

manière correcte la gestion du système "Eaux en milieu urbain". Des processus-métier ayant pour objectif le calcul d'indicateurs se retrouvent en conséquence au nombre des différentes activités. Il peut s'agir d'agréger de l'information relative au réseau d'assainissement, de manière à cibler les interventions, d'extraire des renseignements des séries temporelles ou de modélisations des écoulements, comme la proportion d'eaux claires parasites, d'estimer des paramètres comme la durée de vie probable ou la valeur de remplacement d'ouvrages, d'estimer la consommation en eau potable par rue ou par ménage pour identifier des anomalies par rapport aux valeurs mesurées, etc. Mentionnons ici, à titre d'anecdote, le prélèvement et l'analyse d'échantillons de biofilm des canalisations, dont il est possible de tirer des conclusions, et qui sont menés par certaines municipalités en Suisse. Dans le contexte de l'approvisionnement en eau potable, divers indicateurs ont été mentionnés, comme l'évaluation des temps de séjour de l'eau dans le réseau ou le nombre de ruptures de canalisations pour des secteurs donnés. Les indicateurs peuvent également servir au contrôle et au suivi des systèmes naturels. En fait, des exemples d'indicateurs se rapportant aux différents compartiments du système "Eaux en milieu urbain" peuvent être trouvés, et peuvent servir à appuyer, entre autres, la planification générale.

- ***La réalisation d'infrastructures***

L'attribution de projets : la décision de réaliser une infrastructure entraîne une succession de processus, initiés par la collectivité publique, et dont le résultat est l'attribution du projet (conception et/ou réalisation) à une entreprise particulière.

Le dimensionnement d'infrastructures et les tâches d'expertise : les tâches de dimensionnement et d'expertise requièrent de nombreuses données, quelle que soit la méthode de dimensionnement utilisée. Elles sont souvent le fait d'organismes privés, de bureaux d'étude en particulier, qui les exécutent sous mandat des autorités.

La gestion, le suivi et la réception des chantiers : la réalisation d'infrastructures relatives à l'assainissement occasionne des chantiers importants, qui sont suivis par des représentants des institutions concernées. Le suivi de ces chantiers, pour être efficace, nécessite une organisation sans faille.

- ***La gestion des infrastructures***

La gestion d'un cadastre des infrastructures souterraines : il est absolument nécessaire pour toute communauté urbaine de connaître les infrastructures contenues dans son sous-sol. Les réseaux d'assainissement et d'approvisionnement en eau font partie de ces dernières, et si les cadastres relatifs à ces derniers ont longtemps été (et sont encore souvent à l'heure actuelle) tenus manuellement, de nombreuses applications informatiques consacrées à ces problématiques ont vu le jour.

L'acquisition de séries temporelles relativement aux écoulements dans le réseau d'assainissement: des séries temporelles de débits dans les canalisations sont établies, pour permettre d'identifier d'éventuels problèmes et caractériser le comportement hydraulique des réseaux suite à des événements pluviaux, ainsi que dans certains cas pour déterminer la part d'eaux claires parasites dans l'écoulement. Ces séries temporelles sont validées puis archivées afin de pouvoir être utilisées à d'autres fins, par exemple pour le calibrage de logiciels de simulation hydraulique des écoulements, ou simplement pour être mises à disposition de personnes en ayant besoin. Cette activité peut impliquer la gestion d'un réseau

de stations de mesures.

Le contrôle du réseau d'assainissement : les réseaux d'assainissement sont des infrastructures très coûteuses, dont la durée de vie moyenne est longue, et dont il faut assurer l'entretien et contrôler l'état et le bon fonctionnement régulièrement. A cet effet, les services cantonaux ou communaux disposent d'équipes chargées d'inspecter les ouvrages du réseau, et en particulier les déversoirs d'orage. Ces derniers jouent en effet un rôle très important dans la mesure où ils peuvent être responsables d'une part importante de la pollution des eaux de surface. Relèvent également de ce thème les contrôles des canalisations effectués à l'aide de caméras vidéo, qui génèrent de gros volumes de données mais sont extrêmement précieux pour repérer les déficiences des infrastructures. Ces contrôles vidéo sont parfois directement effectués après la pose d'un nouveau réseau, avec pour objectif de contrôler le travail effectué.

La réfection des ouvrages de réseaux d'assainissement : des défauts ou le vieillissement des ouvrages peuvent rendre nécessaire la réfection d'éléments du réseau d'assainissement. Ces réfections sont soit effectuées directement par les services publics, soit confiées à des partenaires privés. On peut alors retrouver les processus cités sous le point "Réalisation d'infrastructures". Les chantiers de réfection doivent être gérés.

Le contrôle des performances des stations d'épuration : les services cantonaux contrôlent généralement à intervalles réguliers les performances des stations d'épuration des eaux usées situées sur le territoire cantonal. Cette activité, qui nécessite dans le cas de cantons de grande taille une logistique importante, conduit à la gestion de gros volumes de données.

Le contrôle de la qualité des boues d'épuration : les boues d'épuration, qui résultent des processus d'épuration des eaux dans les stations d'épuration, sont en Suisse soit incinérées, soit valorisées comme engrais agricole. Ce dernier usage nécessite que les concentrations de ces boues en substances polluantes ne dépassent pas certaines valeurs limites fixées par la législation. De plus, les polluants qui peuvent être décelés dans les boues d'épuration constituent un excellent indicateur de déversements non conformes effectués dans le réseau d'assainissement, ou d'autres problèmes de pollution. Les services cantonaux effectuent donc régulièrement des contrôles de la qualité des boues d'épuration des différentes stations d'épuration situées sur le territoire cantonal.

L'examen des demandes de construction et de transformation : lors de toute nouvelle construction ou transformation d'immeuble, une demande d'autorisation est adressée aux services administratifs compétents. Ces demandes d'autorisation circulent dans les différents services techniques concernés, et en particulier dans les services compétents en matière d'assainissement. Ces différents services examinent l'objet et les détails de la demande d'autorisation, et émettent un préavis quant à cette dernière. Notons que les services hydrogéologiques doivent également émettre un préavis lors de constructions nouvelles si l'ouvrage projeté présente un risque pour les eaux souterraines, et particulièrement s'il est situé dans une zone de protection des eaux et/ou que ses fondations atteignent une certaine profondeur.

L'organisation des tournées de nettoyage des canalisations d'assainissement : afin d'assurer le bon fonctionnement du réseau de collecteurs, les communautés urbaines gèrent un service de nettoyage de ce dernier. L'organisation optimale des tournées de nettoyage est une tâche complexe, dans la mesure où

les appareils de nettoyage sont spécialement conçus ou réglés pour certains diamètres de canalisations et où l'intervalle de temps entre deux nettoyages peut varier considérablement selon les caractéristiques de la canalisation et du réseau. A l'heure actuelle, ces tournées de nettoyage sont menées à intervalles fixes selon une rotation constante.

L'organisation des tournées de prélèvement d'échantillons destinés à une analyse : ce processus mentionné dans le cadre du contrôle des performances des stations d'épuration, nécessite le fréquent prélèvement d'échantillons en entrée et en sortie de station, qui doivent rapidement être analysés par le laboratoire cantonal. Les services responsables de l'approvisionnement en eau potable organisent également de telles tournées, afin de surveiller la qualité des ressources en eau ainsi que de l'eau dans les différentes composantes des réseaux d'approvisionnement.

Le contrôle des réseaux d'approvisionnement en eau potable : les services industriels procèdent régulièrement à des campagnes d'analyse des pertes d'eau dans les réseaux d'approvisionnement en eau potable (à titre indicatif, ces pertes sont de l'ordre de 12 à 15% en ce qui concerne le réseau de la Ville de Lausanne).

D'autres activités faisant partie de la gestion ou de l'entretien des infrastructures, jouant un rôle dans le contexte de la gestion des eaux, peuvent être citées, comme par exemple la gestion du parc d'équipements sanitaires publics, le nettoyage des voies de circulation ou le choix et la planification de l'emploi des boues d'épuration.

- ***La planification financière***

L'estimation et le suivi de la valeur du patrimoine : la valeur du patrimoine constitué par les différentes infrastructures est utilisée dans différents contextes. En ce qui concerne les réseaux d'assainissement, cette valeur sert d'indicateur, les responsables s'efforçant de la maintenir ou de l'élever au cours du temps. Cette valeur peut également contribuer au calcul de la péréquation financière entre les différents partenaires d'un syndicat d'assainissement ou servir à planifier les investissements financiers dans le temps. Enfin, à l'heure actuelle, la gestion commune, voire la fusion de réseaux d'assainissement, est fréquemment évoquée, et l'estimation de la valeur du patrimoine permet d'avoir une idée précise des apports des participants à un tel processus.

Planifier les investissements concernant les infrastructures : les réseaux et leurs composants ont une durée de vie relativement longue. Les services cantonaux et communaux responsables de ces infrastructures tentent de planifier leurs investissements sur une durée de dix à vingt ans, et utilisent pour cela divers indicateurs : état moyen du réseau, coûts spécifiques par mètre linéaire, espérance de vie moyenne et valeur de remplacement des ouvrages, etc.

Administrer les subventions : les subventions constituent le principal moyen d'incitation de la Confédération et des cantons pour promouvoir l'application de la loi. Le calcul, l'administration et le contrôle de la bonne utilisation des fonds attribués constitue la finalité de différents processus.

- ***Le monitoring environnemental***

La surveillance des débits et hauteurs d'eau dans les eaux de surface, l'acquisition de séries

temporelles relativement à ces paramètres : les services administratifs responsables de la gestion des eaux de surface surveillent les niveaux (lacs, rivières) et débits (rivières) dans les eaux superficielles, et veillent à ce que ces derniers ni ne dépassent un seuil maximum, de manière éviter les inondations, ni ne descendent en dessous d'un seuil minimum, de manière à éviter les étiages. Ces mesures sont fréquemment effectuées par le biais de stations automatisées, et les séries temporelles acquises sont validées et archivées, et permettent l'établissement de statistiques ou peuvent être utilisées pour calibrer des modèles de simulation des écoulements.

L'acquisition et la gestion de séries temporelles relatives aux précipitations : ces données sont absolument nécessaires pour pouvoir comprendre les différents processus que sont les débits dans les cours d'eau et les eaux superficielles, les niveaux des lacs et nappes souterraines. De ce fait, différents services administratifs gèrent des réseaux de pluviomètres, et archivent les séries temporelles recueillies. Différentes personnes insistent sur la nécessité de valider les données avant leur archivage. L'acquisition de séries pluviométriques nécessite la gestion d'un réseau d'appareils de mesure, dont l'exploitation est parfois confiée à des partenaires privés.

La surveillance des niveaux des nappes d'eau souterraines, l'acquisition de séries temporelles relatives à ce paramètre : les nappes d'eau souterraines servent fréquemment de ressource en eau potable, et tant leur qualité que leur quantité font l'objet d'un suivi régulier de la part des administrations. Les séries temporelles résultant de ce suivi sont archivées et servent à l'élaboration de statistiques.

L'inventaire des interaction entre les eaux superficielles et les infrastructures de prélèvement d'eau et d'assainissement : les services administratifs responsables des eaux de surface souhaitent avoir une parfaite connaissance des points de rejets et des exutoires du réseau d'assainissement dans les eaux superficielles, ainsi que des prélèvements d'eau dans ces dernières. Les informations relatives aux rejets sont également l'objet d'attention de la part d'associations de protection de la nature. Aussi est-il tenu des inventaires de ces rejets et prélèvements.

La réalisation de bilans pollutifs : les services compétents réalisent des bilans de pollution en différents endroits des infrastructures d'assainissement (aval de bassins versants, déversoirs d'orages, exutoires), afin d'identifier d'éventuels dysfonctionnements et d'établir un ordre de priorité pour les interventions.

La surveillance de la conformité des lieux de baignade : la population doit pouvoir se baigner dans une eau de qualité adéquate dans les lieux de baignade.

- ***Les services d'urgence***

Les interventions en cas d'urgences : les cantons sont tenus par la LEaux de mettre sur pied un service d'intervention en cas d'accident. Les procédures suivies lors d'une intervention de ce service dans le Canton de Genève ont été examinées en détail. Dans un tel cas, lorsque par exemple un camion-citerne se renverse et que des hydrocarbures se déversent dans le réseau d'assainissement, ou lorsque, suite à un incendie, des eaux utilisées pour éteindre ce dernier ruissellent dans le réseau, les services de police ou de pompiers avertissent l'inspecteur de piquet au service des eaux, qui se déplace sur le lieu de l'accident et, en fonction du type de substance en jeu, conseille les pompiers sur les mesures à prendre : bloquer la pollution dans le réseau d'assainissement ou au contraire la laisser s'écouler jusqu'au cours d'eau (dans le cas d'hydrocarbures présentant des risques d'explosion), en fonction de la configuration du réseau

d'assainissement, des exutoires dans le milieu naturel et de la sensibilité de ce dernier. Cette intervention sur le terrain est suivie d'une procédure administrative destinée à déterminer les responsabilités.

A noter que, dans le domaine de l'approvisionnement en eau potable, les outils informatiques sont à l'heure actuelle abondamment mis en oeuvre, et que, en cas de crise, des scénarios de répartition de l'eau sont prêts à être suivis.

- ***La gestion des risques***

La tenue d'un cadastre des industries : les services administratifs mettent sur pied et gèrent des registres des industries présentant un risque de rejet de substances polluantes dans l'environnement, en particulier dans les réseaux d'assainissement et les eaux superficielles en particulier. Ces industries disposent généralement d'installations de prétraitement, qui font également l'objet d'un archivage et de contrôles réguliers. Les types d'entreprises concernées sont par exemple les industries, les entreprises agricoles, les garages, les cabinets de dentistes, les gravières, les entreprises de nettoyage textile.

La tenue d'un cadastre des installations d'assainissement autonomes : dans certains cas, des immeubles ne peuvent être raccordés au réseau d'assainissement et doivent être équipés d'installations d'assainissement autonomes. Ces installations font l'objet d'un registre particulier et de contrôles réguliers.

La tenue d'un cadastre des citernes : les citernes contenant des liquides polluants, en particulier des hydrocarbures, font l'objet d'un contrôle administratif et sont recensées dans des registres.

La tenue d'un cadastre des sondages géologiques : dans le canton de Genève par exemple, lorsqu'un géologue effectue un sondage préalablement à la construction d'un bâtiment ou d'une infrastructure quelconque, il est tenu d'envoyer les données relatives à ce sondage au Service cantonal de géologie, qui enregistre cette information. Cela permet à ce service, qui n'effectue lui-même que très peu de forages, de disposer d'une information riche relativement au sous-sol du canton (Managlia, 1998). Ce cadastre est mentionné ici en raison de l'importance que revêt à l'heure actuelle le concept d'infiltration artificielle des eaux de ruissellement et dont l'implantation nécessite de bonnes connaissances hydrogéologiques.

Le contrôle des installations de prétraitement : les eaux usées qui résultent de certains processus industriels ou activités ne peuvent être rejetées telles quelles dans le réseau d'assainissement public et doivent donc faire l'objet d'un pré-traitement. Tel est par exemple le cas des eaux provenant des entreprises de nettoyage de textiles ou des cabinets de dentistes (Jalade, 1998). Ces installations de pré-traitement sont donc gérées par les entreprises privées, mais font l'objet de contrôles réguliers de la part des autorités cantonales, qui en tiennent un registre.

La gestion des risques : l'évaluation des risques d'inondation en zones urbaines ou rurales, ainsi que des risques de pollution des eaux par des entreprises, sont des activités menées soit par les autorités administratives, soit, sur mandat, par des entreprises privées.

- ***La communication et la coordination***

La promotion pour la réalisation d'infrastructures concernant la protection des eaux : cet aspect

est mentionné essentiellement par les représentants des services administratifs au niveau cantonal, qui encouragent les communes à améliorer leurs infrastructures d'assainissement. Ces incitations revêtent principalement la forme de séances d'informations destinées à sensibiliser la collectivité en question et de subventions. L'attribution de subventions entraîne toute une procédure administrative d'examen et de traitement des demandes présentées.

La transmission d'information : les autorités parfois cantonales, mais plus souvent communales, se doivent de répondre aux questions du public (particuliers, bureaux d'ingénieurs, architectes) qui par exemple souhaiteraient raccorder un bâtiment au réseau d'assainissement et aimeraient connaître les possibilités de raccordement, ou alors recevoir des données dont ils auraient besoin dans le cadre de leurs activités professionnelles. Cette transmission d'information doit également être assurée.

L'information et la sensibilisation du public en matière de protection des ressources en eau : cet objectif est cité par différents services administratifs cantonaux et communaux.

Annexe 2 Les formalismes UML

• Introduction

UML (Unified Modeling Language) est *"un langage standard de modélisation des logiciels, c'est-à-dire un langage de visualisation, de spécification, de construction et de documentation des artefacts de systèmes à forte composante logicielle. Fondamentalement, le langage UML permet aux développeurs de visualiser les produits de leur travail sous forme de plans d'élaboration et de construction ou de diagrammes standardisés"* (Jacobson et al., 2000). Accepté en novembre 1997 par l'OMG comme standard industriel pour la modélisation dans le domaine logiciel, UML est principalement basé sur les concepts développés par Rumbaugh (1994), Jacobson (1992) et Booch, mais introduit également de nouveaux concepts. UML résulte d'un large consensus et est le fruit du travail d'experts reconnus. UML propose une notation graphique, soit un ensemble de symboles qui constituent une syntaxe et s'accompagnent d'une sémantique.

Quatre catégories de diagrammes peuvent être créés avec UML :

- les diagrammes de cas d'utilisation, qui décrivent les aspects fonctionnels d'un système;
- les diagrammes de classes, permettant d'en modéliser les aspects structurels;
- les diagrammes qui en décrivent les comportements, et qui recouvrent les diagrammes d'interaction (diagrammes de séquence et diagrammes de collaboration), les diagrammes d'état et les diagrammes d'activité;
- les diagrammes d'implémentation que sont les diagrammes de composants et les diagrammes de déploiement.

Précisons qu'un diagramme n'est pas un modèle complet, mais une vue sur un modèle : plusieurs diagrammes sont nécessaires pour décrire un modèle. Nous décrivons ci-dessous brièvement les différentes notations graphiques de UML nécessaires à une bonne compréhension des figures illustrant notre recherche. Nous ne décrivons pas ici les notions d'acteur et de cas d'utilisation, car ces concepts font l'objet de l'annexe 6.

Pour une description plus détaillée, le lecteur intéressé pourra se tourner soit vers la spécification officielle de UML¹ (OMG, 1999), soit vers l'un des nombreux ouvrages consacrés à ce langage (à titre d'exemple Booch et al., 1999; Larman, 1997; Rumbaugh et al., 1999).

• Classes, objets et interfaces

La figure 120 montre les différentes façons de représenter les classes et les objets. Notons que les classes définies durant les phases initiales d'un processus de développement ne se présentent pas de la même manière que les classes de conception : les classes d'analyse ne possèdent pas d'opérations, tandis que les attributs de classes de conception, privés, ne sont modifiables le cas échéant que par le biais d'opérations. Une classe de conception n'a donc normalement que des opérations.

1. [ftp://ftp.omg.org/pub/docs/formal/00-03-01.pdf](http://ftp.omg.org/pub/docs/formal/00-03-01.pdf)

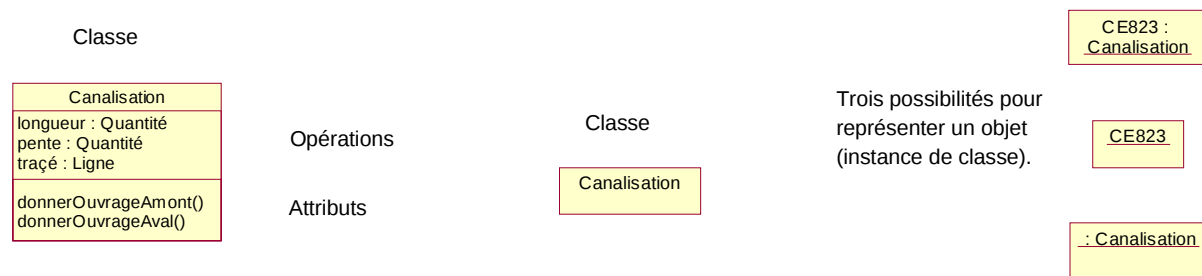


Figure 120. Différentes façons de représenter classes et objets.

• Associations

Une association modélise une relation entre objets. Une association est définie par un nom et est représentée par un lien entre classes. Une flèche sous ce nom peut préciser une direction de lecture. La direction de navigation peut également être exprimée par une flèche à l'extrémité de l'association. Chacune des extrémités de cette dernière est caractérisée par une cardinalité, qui décrit combien d'objets de la classe-cible peuvent être associés à un objet de la classe-source. Ainsi, sur l'exemple à droite de la figure 121, un nombre de bassins versants d'assainissement compris entre zéro et n peuvent avoir comme exutoire un noeud de réseau d'assainissement particulier.

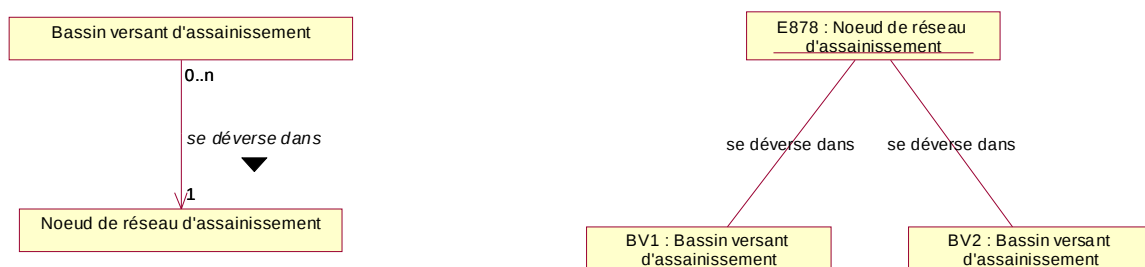


Figure 121. Association entre classes (à gauche) et liens entre objets (à droite).

Un lien entre objets représente une instance d'association. Mentionnons que des rôles peuvent être attribués aux associations, et que des contraintes peuvent les préciser.

• Agrégation

Une relation d'agrégation est une catégorie particulière d'association spécifiant une appartenance entre un tout (l'agrégat) et ses parties. Une telle relation est transitive et antisymétrique. Deux types d'agrégation sont distinguées : l'agrégation "faible", dans laquelle les parties sont indépendantes de l'agrégat, et la composition, dans laquelle les parts dépendent de l'objet composite.

La sémantique de ces deux catégories de relation d'agrégation diffère : dans le cas de la composition, l'existence de la partie est liée à celle de l'agrégat. De plus, cette partie ne peut appartenir, à un instant donné, qu'à un seul objet composite. Dans le cas de l'agrégation faible, la partie peut appartenir à plusieurs objets composites et son existence n'est pas liée à ce dernier. Dans l'exemple de la figure 122,

la relation de composition qui existe entre un ouvrage de réseau d'assainissement et ses couvercles implique que la suppression de cet ouvrage entraîne la suppression de ces derniers. Par contre, la relation d'agrégation faible entre un réseau d'assainissement et un tronçon de réseau d'assainissement révèle un autre choix : un tronçon d'assainissement peut appartenir à plusieurs réseaux d'assainissement, et la destruction d'un tel réseau n'entraîne pas celle des tronçons (différents réseaux peuvent se chevaucher, et donc comprendre les mêmes tronçons).



Figure 122. A gauche, relation de composition, et à droite, relation d'agrégation "faible".

• Généralisation

Une relation de généralisation est une relation taxonomique entre une classe spécialisée et une classe plus générale. Les attributs, opérations et associations des classes les plus générales sont héritées par les classes spécialisées. Des contraintes peuvent préciser la relation de généralisation (figure 123).

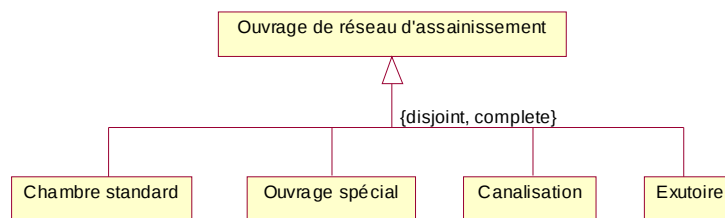


Figure 123. Relation de généralisation : les canalisations, les exutoires, les ouvrages spéciaux, les chambres standard sont des ouvrages de réseau d'assainissement.

• Relations de dépendance

On utilise une telle relation lorsque l'on veut montrer qu'un élément dépend d'un autre élément. Ce cas se produit par exemple lorsqu'une classe utilise une autre classe comme argument dans le contexte de la signature de l'une de ses opérations.

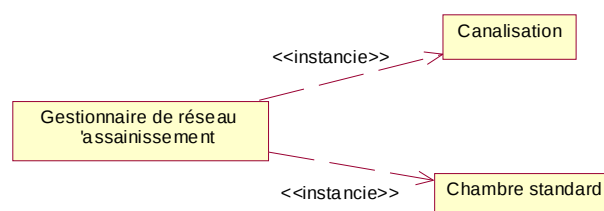


Figure 124. Relation de dépendance. La classe "Gestionnaire de réseau d'assainissement" dépend des classes "Canalisation" et "Chambre standard", car elle les utilise dans le contexte de l'une de ses opérations.

• Interfaces

Les interfaces spécifient des opérations et permettent ainsi de préciser le comportement attendu de classes, de composants ou de paquetages. Ces éléments peuvent "réaliser" les interfaces en question.



Figure 125. L'interface "Persistent", qui définit les opérations nécessaires à la gestion de la persistance des objets, est réalisée par la classe Canalisation.

• Les mécanismes d'extension dans UML

UML est un langage de modélisation ouvert, c'est-à-dire qu'il peut être étendu pour répondre à des besoins particuliers. Trois mécanismes d'extension existent : les stéréotypes, les valeurs marquées et les contraintes. "Les stéréotypes permettent de définir de nouveaux éléments en élargissant ou en affinant la sémantique d'éléments existants, tels que des éléments ou des relations. Les valeurs marquées permettent, quant à elles, de définir de nouvelles propriétés d'éléments existants. Enfin, les contraintes offrent un moyen d'imposer des règles à des éléments et à leurs propriétés" (Jacobson et al., 2000).

• Les paquetages

Les paquetages permettent d'organiser des éléments en groupes : il s'agit donc d'unités de regroupement. Deux types de relations peuvent exister entre paquetages : les dépendances d'accès et d'importation, utilisées pour importer dans un paquetage des éléments exportés d'un paquetage tiers, et les généralisations, utilisées pour spécifier des familles de paquetages.

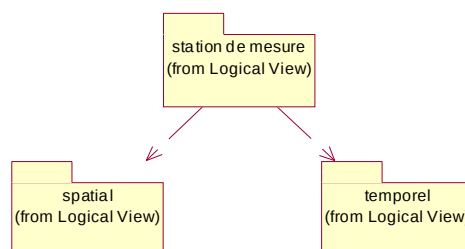


Figure 126. Relation de dépendance entre paquetages. Le paquetage "station de mesure" s'appuie sur les classes proposées par les paquetages "temporel" et "spatial".

• Les interactions

Les diagrammes comportementaux décrivent le comportement et les interactions des objets. On peut en distinguer quatre types : les diagrammes de séquence, de collaboration, d'état et d'activité. Les diagrammes d'état permettent également de préciser les comportements au sein même de l'objet. Nous avons dans le cadre de ce travail mis en oeuvre les diagrammes d'interaction (diagrammes de séquence

et de collaboration), raison pour laquelle nous nous concentrons sur eux.

Les objets, dans un système, interagissent généralement par le biais d'échanges de messages. Une interaction est un comportement qui englobe un ensemble de messages et possède un but précis. Un message (figure 127) est une spécification de communication entre objets qui véhicule une certaine information, et qui a pour objectif l'initiation d'une certaine activité (Booch et al., 1999).

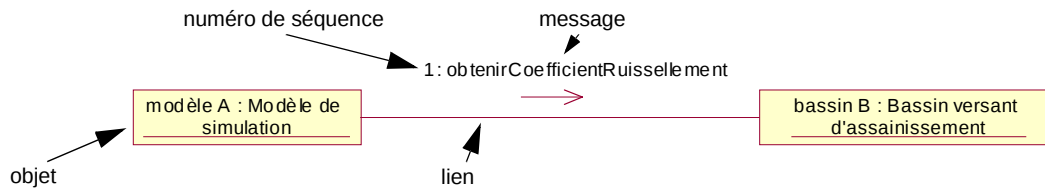


Figure 127. Message entre deux objets.

Les diagrammes de séquence (figure 128) mettent l'accent sur la succession temporelle des messages. Les objets qui participent à l'interaction sont placés sur la partie supérieure du diagramme. Les messages échangés entre objets sont placés de haut en bas en fonction de leur chronologie.

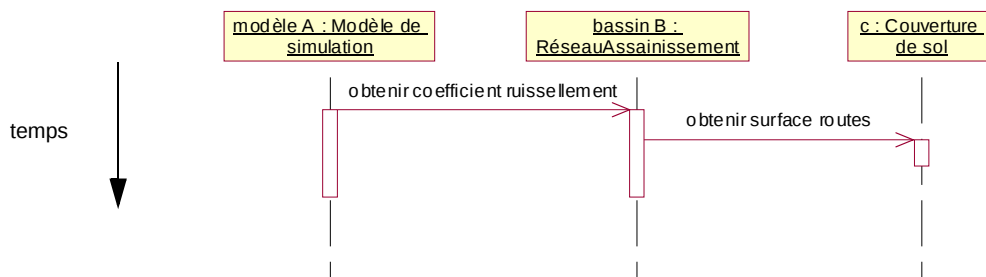


Figure 128. Diagramme de séquence.

Les diagrammes de collaboration (figure 129) se focalisent quant à eux sur l'organisation des objets qui participent à une interaction. Dans un tel diagramme les objets constituent les noeuds de graphes, et les liens qui connectent ces objets constituent les arcs de ces graphes. Les messages que s'échangent les objets sont positionnés sur ces arcs. De tels diagrammes donnent une image claire de la manière dont les objets collaborent lors d'une interaction.

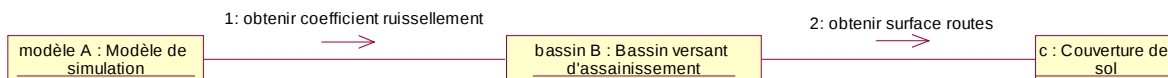


Figure 129. Diagramme de collaboration.

• Les composants

Les diagrammes de composants, auxquels nous nous intéressons en raison de la thématique de notre recherche, sont l'une des deux sortes de diagrammes de UML qui s'attachent aux aspects physiques des systèmes orientés objet. Un diagramme de composants montre l'organisation et les dépendances entre composants. Dans UML, tous les artefacts logiciels "physiques" sont modélisés par le biais de

composants. Les composants ressemblent en fait, de par leurs propriétés, aux classes : ils se situent cependant à un niveau d'abstraction différent, beaucoup plus proche du système logiciel physique.

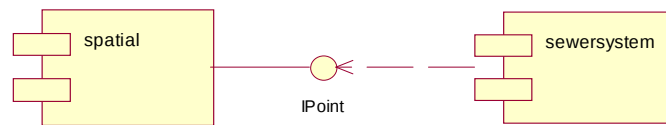


Figure 130. Composants ("spatial" et "sewersystem") et interface ("IPoint"). Reprise de la figure 37.

Annexe 3 COM et ses évolutions

COM (Component Object Model) est la proposition de la société Microsoft pour permettre la conception d'applications logicielles par combinaison de composants. Il s'agit d'un environnement proposant des concepts standard et assurant des mécanismes qui permettent l'interopérabilité des composants : COM d'une part fournit un mécanisme de communication entre composants, et d'autre part recouvre un ensemble de services ("COM runtime") qui doivent être présents sur l'ordinateur pour permettre à la communication de s'établir (Lewis, 1999). Un composant COM est une portion de code compilé qui propose un certain nombre de services au reste du système et peut donc être intégré dans plusieurs applications à la fois. Un objet COM est une instance de composant COM. L'architecture COM rend ainsi possible la création de composants capables d'interagir durant la phase d'exécution.

Une caractéristique de COM est qu'un objet COM n'accède jamais directement aux fonctionnalités d'un autre objet COM : il doit pour ce faire s'adresser à une interface. Il s'agit là d'une propriété fondamentale, qui permet de respecter strictement le principe d'encapsulation : les données ainsi que la manière de produire les services demandés ne sont accessibles que par le biais des propriétés publiques proposées par les interfaces. Une interface COM est définie comme étant un contrat fortement typé entre composants logiciels offrant un ensemble d'opérations utiles et sémantiquement corrélées (Williams et Kindel, 1994). De ce fait, une interface constitue la définition d'un comportement et de fonctionnalités attendus. Lewis (1999) insiste sur le fait qu'une interface ne fait que décrire ce qu'un composant possède comme fonctionnalités, sans décrire la manière dont ces fonctionnalités sont implémentées.

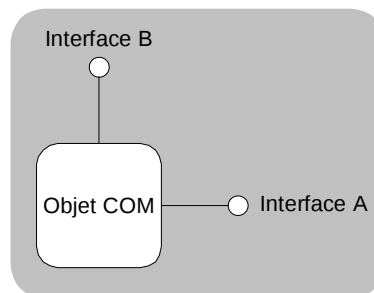


Figure 131. Représentation usuelle d'un objet COM.

Un objet COM peut aussi bien être un client qu'un serveur (voir fig. 132) : en effet un serveur est défini comme un objet qui expose ses interfaces et propose un ensemble de fonctionnalités auxquelles un client peut accéder. Un client est donc un autre objet, ou une application dont la conception lui permet d'accéder à cette interface (voir fig. 132). Le client peut lui-même posséder ses interfaces.

En raison de l'importance de la notion d'interface dans le contexte de COM, il est nécessaire d'en développer les caractéristiques :

- Une interface n'est pas une classe car elle ne possède pas d'implémentation. Une interface ne peut donc pas être instanciée seule, et un composant implémentant cette interface doit être instancié pour que cette interface puisse être utilisée. A noter que différents composants peuvent implémenter la même interface, ce qui permet de mettre en oeuvre le concept de polymorphisme (Williams et Kindel, 1994).

COM est un standard défini au niveau binaire, permettant un accès aux interfaces par le biais de pointeurs. Ainsi, tout langage à même d'appeler des fonctions par le biais de pointeurs (C, C++, Visual Basic, Delphi,...) peut être utilisé pour développer des composants qui peuvent interagir avec d'autres composants (éventuellement développés à l'aide d'autres langages) obéissant aux mêmes spécifications binaires. C'est grâce à cette spécification des interfaces au niveau binaire que l'indépendance de COM vis-à-vis du langage de programmation utilisé peut être atteinte.

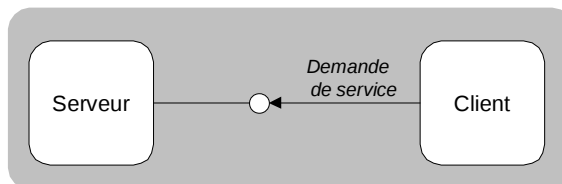


Figure 132. La notion de client et de serveur COM.

- Un composant COM peut implémenter plusieurs interfaces et ainsi proposer plusieurs ensembles de services logiquement regroupés. Par exemple un composant "Canalisation" pourrait implémenter une interface relative à sa persistance (par exemple IPersistObject), définissant toutes les fonctionnalités permettant d'assurer l'interaction d'un objet avec le système chargé d'assurer sa persistance (base de données, système de fichiers,...), et une interface IPipe, intégrant toutes les fonctionnalités spécifiques à une canalisation.
- L'implémentation d'un composant peut être le fait d'une classe unique, mais ça n'est pas obligatoire. En général un composant contient plusieurs classes, qui collaborent pour implémenter les interfaces souhaitées (Szyperski, 1997).
- Les interfaces dans COM sont fortement typées, ce qui veut dire que chaque interface est identifiée de manière univoque par un identifiant globalement unique (GUID). La possibilité de conflits de noms est de ce fait écartée.
- Les interfaces ne peuvent être modifiées. Dès que l'on modifie la sémantique d'une interface, un nouvel identifiant pour cette interface est généré, et l'on se retrouve en fait face à une nouvelle interface. Ainsi des conflits de version peuvent être évités : en effet si l'on veut modifier un composant (par exemple pour lui adjoindre de nouvelles fonctionnalités), tout en restant compatible avec les applications utilisant ce composant, il suffit de faire cohabiter sur ce composant les deux versions de l'interface, qui seront considérées comme deux interfaces différentes.

Le concept de délégation d'interface joue un rôle très important dans le cadre de ce travail, dans la mesure où il facilite grandement la réutilisation de composants. *"Deux cas peuvent être distingués, qui sont l'agrégation d'interface [...] réalisant simplement le transfert de l'appel du délégataire vers le délégant, et le containment d'interface où du code peut être ajouté au délégataire avant appel de la fonction du délégant."* (Bouzeghoub et al., 1997). La figure 133 représente ces deux alternatives. Dans le cas du containment, un objet possède une référence unique vers un autre objet, et, conceptuellement tout au moins, contient le second. Un appel adressé à l'objet externe, mais devant être accompli par l'objet interne, est simplement transmis à l'objet interne par l'objet externe, qui peut au passage intervenir sur cet appel (en valider ou modifier les paramètres par exemple). Dans le cas de l'agrégation par contre, l'objet interne met son interface directement à disposition d'objets clients de l'objet externe. De ce fait les

appels vont directement à l'objet interne, et l'objet externe ne peut pas intercepter ces derniers.

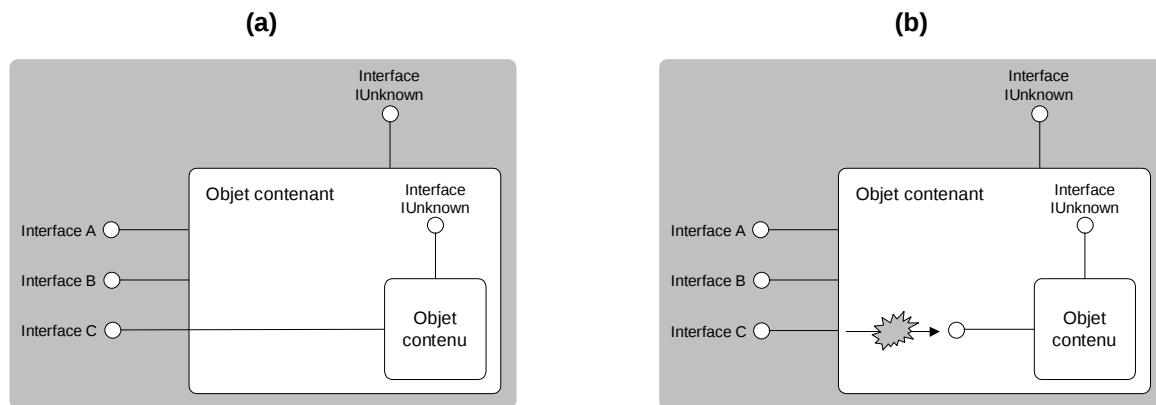


Figure 133. Agrégation (a) et containment (b), adapté de Bouzeghoub (1997) et Szyperski (1997). A remarquer, les interfaces IUnknown. Chaque objet est muni d'une interface standard IUnknown qui lui permet de découvrir les autres interfaces.

A noter que COM ne supporte pas l'héritage d'implémentation. Une relation de généralisation peut toutefois être simulée par combinaison d'héritage d'interface et de délégation.

COM constitue la technologie de base sur laquelle reposent les développements de Microsoft relativement aux composants logiciels : OLE, ActiveX, DCOM, COM+. Parmi ces dernières, DCOM (pour Distributed Component Object Model) et COM+ permettent à des applications ou composants de faire appel à des composants qui ne sont pas localisés sur la même machine.

Annexe 4 Systèmes d'information et architectures en couches

La majorité des applications actuelles dans le domaine de la gestion des eaux interagissant avec des BD sont conçues selon une architecture en deux couches : la première de celles-ci comprend les interfaces utilisateurs, tandis que la seconde intègre la BD (Fowler, 1997). Les portions de code qui sont responsables de la présentation de l'information et des interactions avec la BD, et qui intègrent la logique-métier sont, dans ce type d'application, intimement liées (figure 134). La maintenance de telles applications est extrêmement difficile, la moindre modification au niveau de la BD entraînant des modifications en de nombreux points du logiciel. Cet inconvénient a fait que très rapidement a dû être proposée une architecture à trois couches comportant une couche intermédiaire qui, intégrant la sémantique de l'application, s'intercale entre les interfaces-utilisateurs et la BD.

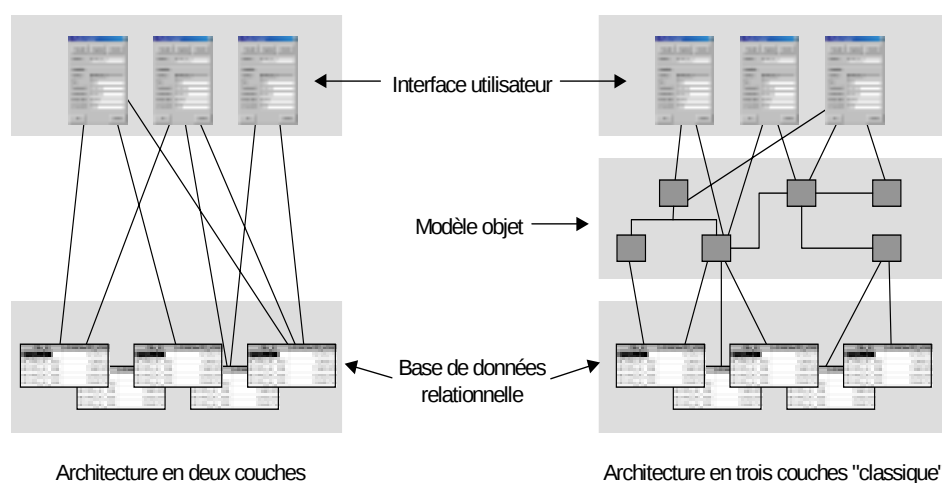


Figure 134. Architectures logicielles en deux et trois couches (adapté de Gupta et Scumniotales, 1998).

L'approche orientée objet s'insère très bien dans ce schéma : la couche intermédiaire peut être conçue comme un système objet basé sur une modélisation orientée objet du domaine de l'application; les interfaces-utilisateurs interagissent étroitement avec le modèle objet, qui lui-même gère les interactions avec la BD. Cette approche ôte aux concepteurs des interfaces utilisateurs le souci de se préoccuper d'interactions avec la BD, puisqu'ils travaillent avec des objets du domaine. Le modèle de données structurant la BD peut de son côté évoluer sans que cela affecte les applications (à condition bien sûr que le modèle objet subisse les adaptations nécessaires). Notons que d'une part une telle approche donne la possibilité d'envisager plus facilement la répartition des données entre différentes bases, et que d'autre part différentes alternatives sont possibles quant à la localisation physique de ces différentes couches.

Cette approche ne résout pas tous les problèmes : il a été constaté que les concepteurs de la couche intermédiaire consacrent 30% de leur temps en moyenne à créer des fonctionnalités permettant d'assurer la correspondance entre les objets de la couche intermédiaire et les données leur correspondant dans la BD, et cela alors que de telles fonctionnalités peuvent être généralisées. Des infrastructures peuvent être développées qui ont pour principale fonction d'assurer la persistance des objets de la couche intermédiaire (Gupta et Scumniotales, 1998). La majeure partie des BD qui sont mises en place à l'heure actuelle sont de type relationnel. Un modèle de données relationnel a pour but une organisation cohérente

et sans redondance des données, tandis qu'un modèle orienté objet a pour but de représenter le monde réel de la manière la plus fidèle possible (en fonction bien sûr d'un certain objectif). Cette divergence entre les approches relationnelles et objet conduit à certaines difficultés lors de la conception de systèmes d'information s'appuyant, pour la gestion des données, sur des BD relationnelles. Cette problématique est abondamment discutée dans la littérature (Ambler, 1998a; Ambler, 1998b; Gupta et Scumniotales, 1998; Keller, 1997; Keller, 1998; Keller et Coldewey, 1996; Stoiber, 1999; Yoder et al., 1998). De telles infrastructures, qui constituent la couche dite "de persistance", peuvent être réutilisées en étant transférées d'une application à l'autre, et se chargent des aspects suivants (Gupta et Scumniotales, 1998) :

- persistance : un objet à même de maintenir son état entre deux sessions d'utilisation d'un système d'information peut être qualifié de persistant. La persistance, assurée par des BD ou des fichiers, permet donc à un utilisateur de sauvegarder l'état d'un objet entre deux sessions. Les aspects qui doivent être considérés dans ce contexte sont la connection aux bases (ou autres sources) de données, la génération d'objets à partir de ces sources, l'enregistrement et l'effacement d'objets, cette dernière fonction devant entraîner la destruction des données qui leurs correspondent dans les BD.
- requêtes : il s'agit de développer un mécanisme qui permette aux utilisateurs de retrouver les objets dont ils ont besoin dans les BD.
- transactions : une transaction est, dans le domaine des BD, un ensemble d'opérations qui doivent être soit accomplies intégralement, soit ne pas être accomplies du tout. Une infrastructure de persistance doit être à même d'assurer la gestion des transactions.
- concurrence : dans le cas de systèmes multi-utilisateurs, plusieurs utilisateurs peuvent vouloir manipuler le même objet en même temps. Il est possible d'adopter une approche optimiste, partant du principe que cela est rarement le cas, ou au contraire une approche pessimiste. Ces deux approches conduisent à des façons différentes de gérer les problèmes de concurrence.
- relations entre objets : dans un modèle orienté objet, les objets sont liés par différentes catégories de relations. Une infrastructure de persistance gèrera ces relations pour agir d'une manière adaptée en ce qui concerne les données : l'effacement d'un objet, par exemple, peut entraîner l'effacement de nombreux objets qui lui correspondent, et donc des données s'y rapportant.
- gestion des identifiants : chaque occurrence dans une base de donnée relationnelle est identifiée de manière univoque par une clé, et chaque objet a sa propre identité. L'infrastructure de persistance peut avoir pour fonction de générer et d'attribuer des identifiants, tâche particulièrement complexe lorsque de multiples BD sont impliquées (Ambler, 1998a).

Enfin, dans un contexte multi-utilisateur, il se peut que les applications spécifiques liées aux différents utilisateurs doivent intégrer un certain nombre de fonctionnalités, auquel cas il faudra que celles-ci fassent l'objet d'une couche distincte. En effet, pour des raisons de maintenance, de telles fonctionnalités doivent être distinguées des interfaces utilisateurs proprement dites, et tout de même être localisées sur les ordinateurs des utilisateurs. Ainsi, il est possible de distinguer une couche supplémentaire, qui intègre les fonctionnalités spécifiques liées à des applications particulières. Une application complexe devrait donc être structurée selon cinq couches principales (figure 135), et répondre aux caractéristiques suivantes:

- une forte indépendance, soit un couplage le plus faible possible, entre les couches;
- des interfaces clairement définies et documentées entre les différentes couches, ceci afin d'assurer une certaine souplesse au système, dont chaque couche doit pouvoir faire l'objet de modifications sans que les autres couches en soient affectées;

- ces couches doivent pouvoir être installées sur des machines différentes, et donc communiquer par le biais de réseaux informatiques;
- certaines couches doivent être conçues de manière à pouvoir être réutilisées dans de multiples applications (en particulier la couche de persistance).

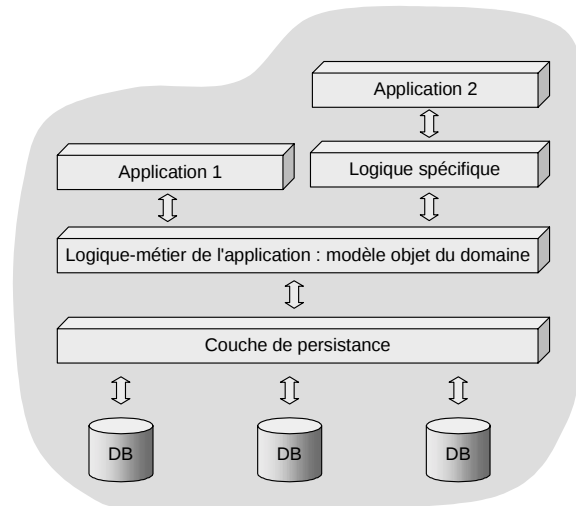


Figure 135. Architecture en couches des systèmes d'information.

Cette volonté de regrouper les fonctionnalités en unités logiques et à en définir précisément les interfaces n'est pas sans rappeler l'approche par composants présentée dans ce travail. Chacune des couches peut ainsi être "encapsulée" par un ou plusieurs composants, et il peut être accédé à leurs fonctionnalités par le biais de leurs interfaces.

Dans un SI conçu selon une architecture en deux couches, les règles de gestion et la logique de l'application sont disséminées dans la couche utilisateur et/ou le mécanisme de persistance, ce qui ne favorise pas son évolution ou sa maintenance. Dans un système conçu selon une architecture en trois couches, la déconnection totale du mécanisme de persistance des applications intégrant la logique-métier des utilisateurs permet 1°) de modifier, voire de remplacer le mécanisme de persistance (changement de la structure des tables dans le cas d'une BD relationnelle, changement de BD) sans aucune influence sur les applications des utilisateurs; 2°) de modifier les applications spécifiques liées aux différents utilisateurs de manière simple et sans influence sur le mécanisme de persistance; 3°) de mieux maîtriser la logique de l'application dans la mesure où les fonctionnalités et règles de gestion sont regroupées logiquement et facilement accessibles; et 4°) lorsque l'application implique de nombreux utilisateurs, de gérer ceux-ci de manière beaucoup plus simple.

Annexe 5 Principaux types de processus de développement

Le cycle, ou processus de développement logiciel, constitue le cadre au sein duquel s'organise une méthode de développement. Sont en général abordées dans un processus de développement les étapes d'analyse, de conception, d'implémentation, de test et éventuellement d'installation et de maintenance. Cet annexe présente succinctement les principaux modèles de développement cités dans la littérature (Bouzeghoub et al., 1997; Pressman, 1997; Schach, 1996; Sommerville, 1995).

• *L'approche "construction-correction"*

L'approche "construction-correction" ("build and fix model") correspond en fait à l'absence de processus de développement. Dans un tel cas, le développeur de l'application ne procède à aucune analyse des besoins, et construit rapidement une première version de l'application. Celle-ci est ensuite soumise aux utilisateurs, puis modifiée en fonction des désirs et observations de ces derniers.

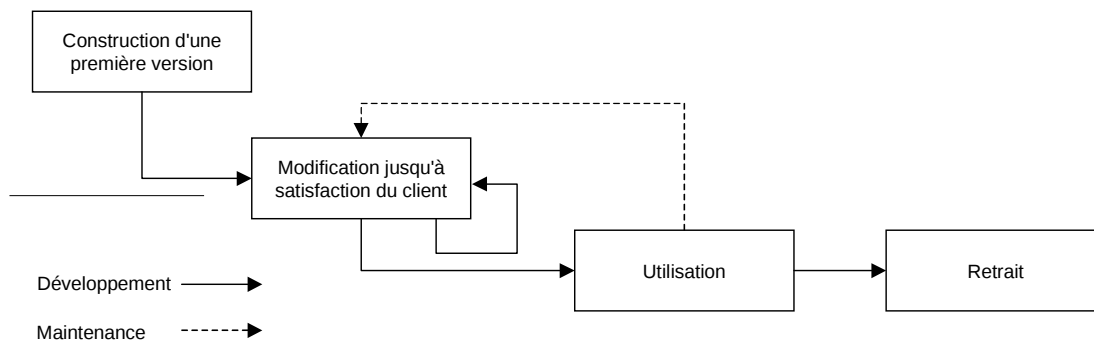


Figure 136. L'approche "construction-correction" (adapté de Schach, 1996).

Cette approche est acceptable pour les projets de petite taille, comprenant moins de 100 à 200 lignes de code (Schach, 1996), mais induit au-delà des coûts de développement qui deviennent rapidement inacceptables. En effet, les coûts des modifications sont de plus en plus élevés à mesure que ces dernières sont effectuées tard. De plus, en l'absence de documentation précise, la maintenance de telles applications est très difficile.

• *Le modèle en cascade*

Le modèle en cascade ("waterfall model"), également appelé modèle de cycle de vie classique, a commencé à être utilisé à la fin des années 60 et l'est toujours à l'heure actuelle (Pressman, 1997). Ce modèle est caractérisé par une succession d'étapes bien définies et formalisées : à une étape d'analyse des besoins ("requirements") succède une phase de spécifications, qui décrit de manière précise les caractéristiques futures du SI développé. Une fois le mandataire du SI d'accord avec les spécifications, un plan de développement est établi et la phase de conception du SI peut commencer. Durant celle-ci, les aspects relatifs à l'architecture logicielle, aux structures de données, aux interactions avec les utilisateurs et aux détails algorithmiques sont étudiés. Durant la phase d'implémentation, les produits de la phase de conception sont transformés de manière à pouvoir être interprétés par l'ordinateur : c'est le codage. Le SI est ensuite testé puis déployé chez les utilisateurs, moment à partir duquel il est

opérationnel et fait l'objet d'une maintenance, jusqu'à son obsolescence et donc son retrait.

L'une des particularités du modèle en cascade, hormis son mode de développement linéaire, tient au fait que chaque étape est vérifiée et documentée. L'étape suivante ne débute pas tant que celle qui est en cours n'a pas été approuvée et que les documents s'y rapportant n'ont pas été produits. Ce modèle de développement permet, malgré son côté séquentiel, de revenir à l'étape précédente lorsque l'on se rend compte que c'est nécessaire, et les documents descriptifs doivent être mis à jour. Lorsque des erreurs, des problèmes ou des changements des besoins sont détectés durant la phase d'exploitation de l'application, une modification du logiciel est possible, pour autant qu'elle fasse l'objet d'une documentation. De telles interventions, ayant lieu alors que l'application est déjà opérationnelle, appartiennent à la maintenance.

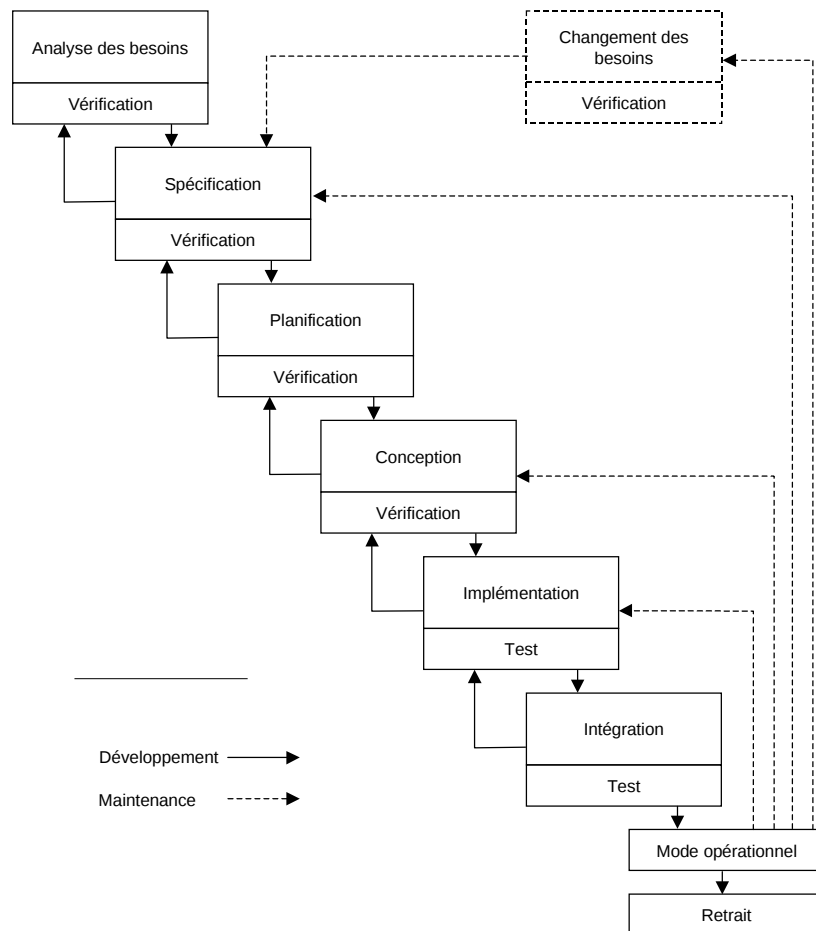


Figure 137. Modèle de développement en cascade, d'après Schach (1996).

Ce processus, de par la discipline imposée aux équipes de développeurs et aux contrôles qui ponctuent ses différentes étapes, permet de mettre en place des SI de qualité et dont la maintenance est, de par l'abondante documentation disponible, facilitée. Le modèle en cascade n'est cependant pas exempt de faiblesses :

- des erreurs qui ne sont pas détectées lors des différents contrôles peuvent avoir des impacts considérables si elles impliquent de devoir reprendre le processus à son début.
- le mode de développement séquentiel implique que le mandataire du SI ne voit fonctionner l'application qu'au dernier moment. Lors de gros projets, le temps nécessaire au processus peut être très long.

- lorsque le SI délivré ne répond pas aux attentes, le processus doit être repris à la base.
- les spécifications, qui résultent de l'analyse des besoins du mandataire du SI, devraient pouvoir être validées par ce dernier. Or les documents de spécifications sont difficiles à comprendre, en particulier lorsque des langages formels sont utilisés. Le mandataire accepte donc souvent des documents de spécifications dont il ne comprend pas les implications (Schach, 1996).
- le processus en cascade implique que les besoins du mandataire puissent être identifiés dès le début du cycle. Or dans la pratique un certain nombre d'incertitudes accompagnent les débuts de projets de développement de SI (Pressman, 1997), et les besoins réels peuvent être difficiles à cerner.
- les exigences en terme de documentation sont tellement élevées que dans la pratique elles ne sont que rarement satisfaites et que les différentes phases ne sont pas suivies avec autant de rigueur que nécessaire.

Selon Bouzeghoub [1997 #169], *"le principal inconvénient de ce modèle est son incapacité à prendre en charge des systèmes complexes comportant un grand nombre d'applications interagissant les unes avec les autres, de façon non nécessairement hiérarchique"*. Ceci provient du fait d'une part que la notion de système n'est en fait pas prise en compte dans ce modèle, et d'autre part que l'on passe directement d'une phase d'analyse des besoins, trop informelle, à une phase de spécifications techniques trop détaillée.

• ***Le modèle à prototypage rapide***

Un prototype est une application-test comprenant un nombre réduit de fonctionnalités par rapport à l'application qui doit être produite. Harel, cité dans Brooks (1996) complète la notion de prototype par la définition suivante :

"[C'est une version d'un programme qui] reflète seulement les décisions de conception prises durant la préparation du modèle conceptuel, et non celles qu'engendrent les problèmes d'implémentation."

Le modèle à prototypage rapide propose de développer rapidement un prototype du SI, en insistant sur les aspects relatifs aux interactions avec l'utilisateur et en négligeant, pour gagner du temps, les aspects "invisibles" (validation des données, optimisation, etc.), puis de mettre ce prototype à disposition de l'utilisateur afin de lui permettre de se rendre compte des caractéristiques de l'application qui sera développée. Une fois le futur utilisateur satisfait du prototype, les spécifications du SI sont rédigées et ce dernier est développé selon un déroulement similaire à celui du modèle en cascade. Cette approche présente plusieurs avantages :

- souvent les utilisateurs d'un SI expriment en termes généraux leurs attentes mais n'arrivent pas à définir précisément leurs besoins. La manipulation d'un prototype leur permet de mieux exprimer leurs attentes et donc de produire des documents de spécifications répondant mieux à leurs besoins.
- le prototype permet aux concepteurs de tester de nouvelles technologies, que la prudence leur interdirait de mettre en oeuvre directement dans une application opérationnelle.

Cependant, il faut bien se rendre compte que le prototype réalisé est destiné à être abandonné, ne peut servir de base à la construction du SI définitif. Or, dans la mesure où le prototype est souvent visuellement très attractif, il est difficile pour le mandataire d'accepter les frais inhérents au nouveau

développement que constitue la mise en place de l'application opérationnelle. Il réclame de ce fait souvent une "amélioration" du prototype, alors que les bases techniques sur lesquelles ce dernier est construit sont souvent insuffisantes, voire inadaptées. De plus, les concepteurs font souvent des choix lors du prototypage qui sont plutôt guidés par des impératifs de rapidité de développement et ne sont du point de vue technique pas les plus adaptés. Une fois le prototype disponible, ce fait est souvent oublié et les développeurs renoncent à baser le produit final sur des techniques différentes (Pressman, 1997).

• **Le modèle incrémental**

Le modèle de développement incrémental, qui fait partie des modèles dits "évolutionnaires", se distingue des modèles présentés précédemment par le fait qu'il permet la conception de SI selon un mode itératif: des cycles se succèdent qui autorisent le développement de parties réduites, mais pleinement opérationnelles, du SI. Ces parties sont successivement ajoutées au SI, ce dernier étant à chaque fois testé, puis mis à disposition des utilisateurs. Ce modèle a été proposé sur la base de la constatation qu'un SI est souvent construit pièce par pièce, à la manière d'un bâtiment.

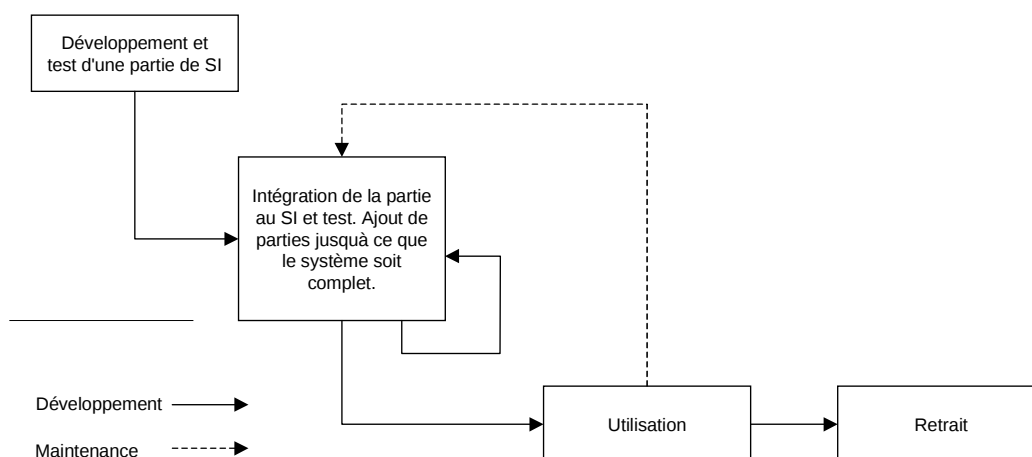


Figure 138. Modèle incrémental (adapté de Schach, 1996).

Contrairement aux méthodes de développement précédentes, qui conduisent à la livraison de SI complets, terminés, cette méthode permet la livraison à la fin de chaque cycle de produits partiels mais opérationnels. Schach (1996) estime que le nombre de cycles menés habituellement par les équipes qui utilisent ce modèle varie entre dix et cinquante.

Ce mode de développement permet aux utilisateurs de s'approprier très rapidement leur SI, les premiers éléments en étant livrés dans des délais très brefs. De plus, la mise en contact des utilisateurs avec le SI est moins brutale que dans le cas d'un développement basé sur un modèle en cascade : elle se fait au rythme de livraison des modules et l'adaptation est progressive. Enfin, comme l'intégration des nouveaux éléments fait l'objet d'une validation à chaque cycle, le développement du SI peut à tout moment être stoppé, lorsque le mandataire estime que les objectifs en sont atteints.

Une telle approche nécessite une conception extrêmement soignée, afin que les parties développées puissent sans problème être ajoutées au SI opérationnel.

• Le modèle en spirale

L'idée principale du modèle en spirale est de vouloir minimiser le risque tout au long du processus de développement. Ce modèle résulte en fait de la combinaison d'un processus en cascade avec une approche évolutionnaire, dont les différentes phases sont systématiquement précédées d'une analyse de risque.

De manière plus détaillée, ce processus est représenté par une spirale parcourue dans le sens des aiguilles d'une montre par l'équipe de développement. La spirale est découpée en un certain nombre de régions, qui correspondent à des types d'activités particuliers : planification de l'étape à venir, détermination des objectifs, alternatives et contraintes, analyse de risque, puis développement et vérification. Chacune des itérations de la spirale, qui correspond à l'une des phases classiques d'un processus de développement (spécifications, analyse, conception, etc.), passe au travers de ces zones d'activité. A noter que contrairement aux modèles de développement présentés dans les chapitres précédents, le modèle en spirale s'intéresse également aux phases qui suivent la mise en service d'un SI, à savoir la maintenance et l'amélioration du SI.

Le grand avantage de ce processus de développement est l'intégration de la notion de risque. Par contre il s'agit d'un modèle complexe, qui réclame la participation de spécialistes (du risque en particulier), et qui ne peut être mis en oeuvre que dans le cadre de projets de grande envergure.

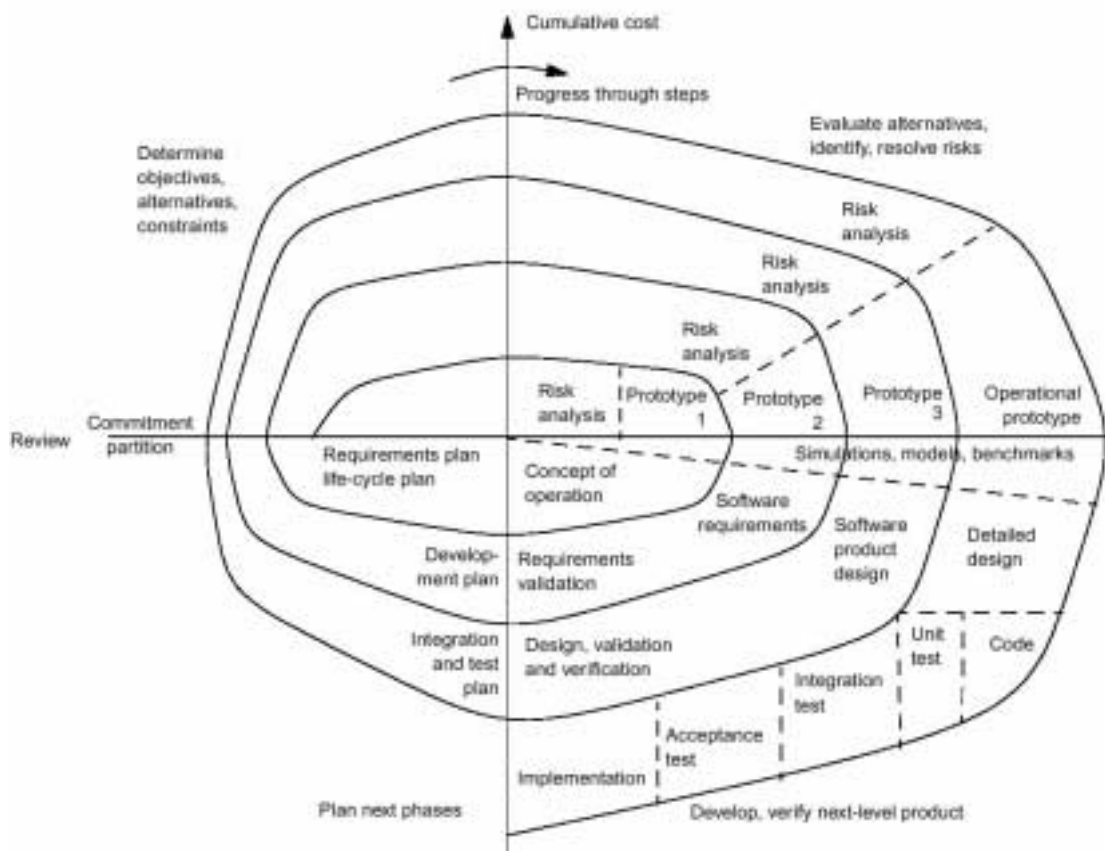


Figure 139. Le modèle en spirale (Boehm, 1988; cité dans Strohmeier, 1999).

Annexe 6 Notions d'acteur et de cas d'utilisation

Un cas d'utilisation présente les actions successives qui ont pour but de permettre à un acteur d'atteindre un objectif donné (Dano et al., 1997). Toutes les possibilités d'atteindre cet objectif sont décrites. Un scénario par contre représente une manière d'atteindre l'objectif, et constitue donc une occurrence particulière d'un cas d'utilisation. Les cas d'utilisation ("use cases") sont des outils très bien adaptés à l'identification de spécifications fonctionnelles. Ils ont été utilisés dans le contexte du génie logiciel pour la première fois par Jacobson (1992) et sont abondamment discutés dans la littérature (Cockburn, 2000; Firesmith, 1999; Jacobson et al., 1997; Kettani et al., 1998; Larman, 1997; Leffingwell et Widrig, 2000; Sendall et Strohmeier, 2000). La notion d'acteur est quant à elle définie par Jacobson de la manière suivante :

"A business actor represents a role that someone or something in the environment can play in relation to the business." (Jacobson et al., 1992)

Cockburn complète la définition donnée par Jacobson, précisant qu'un acteur n'est pas forcément un individu :

"An actor is anything having behaviour. An actor might be a person, a company or organisation, a computer program or a computer system, hardware or software or both." (Cockburn, 2000)

Enfin, Kettani (1998) précise qu'un acteur *"représente une abstraction qui réside juste en dehors du système à modéliser"*.

Une personne peut ainsi jouer plusieurs rôles, et donc être représentée par des acteurs différents (figure 140). Des relations de généralisation peuvent être définies entre acteurs, selon le niveau de détail nécessaire.

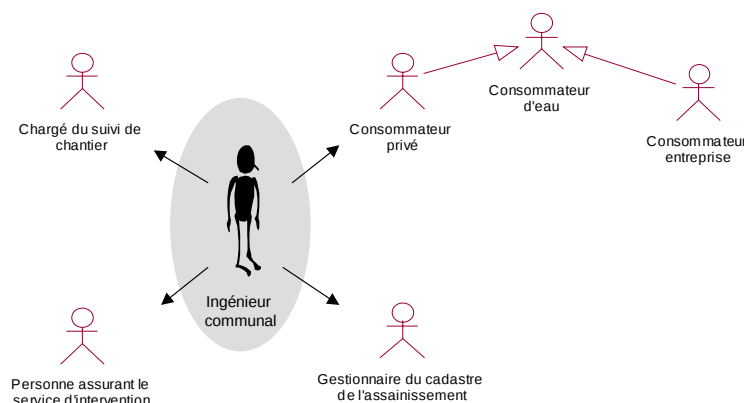


Figure 140. Représentation des acteurs selon le formalisme UML. Un intervenant peut incarner différents acteurs.

Les cas d'utilisation peuvent être représentés de trois différentes manières : sous forme développée, sous forme résumée, et enfin selon le formalisme UML (figure 141). Ces derniers ne sont cependant que des

"table des matières" renvoyant le lecteur vers les descriptions textuelles (développées ou résumées) que sont en fait les cas d'utilisation (Cockburn, 2000).

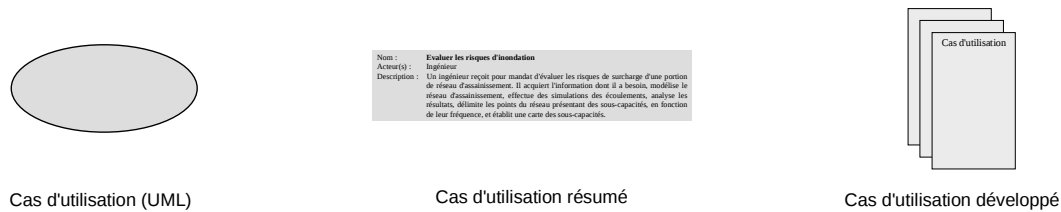


Figure 141. Les cas d'utilisation (systèmes ou réels) peuvent se présenter sous trois formes : symbole UML, résumés ou développés.

Un modèle des cas d'utilisation (figure 142) définit les cas d'utilisation, les acteurs concernés, ainsi que les relations entre ces derniers : généralisation-spécialisation, inclusion et extension de cas d'utilisation, généralisation-spécialisation d'acteurs (Booch et al., 1999). La relation d'inclusion permet la hiérarchisation des cas d'utilisation et est fréquemment employée les cas d'utilisation de base faisant référence aux cas inclus. La relation d'extension permet de définir des comportements optionnels. Dans ce cas, c'est le cas d'utilisation d'extension qui fait référence au cas de base. Enfin, la généralisation, utilisée lorsqu'il est judicieux que des cas d'utilisation génériques soient définis, est dans la pratique peu mise en oeuvre.

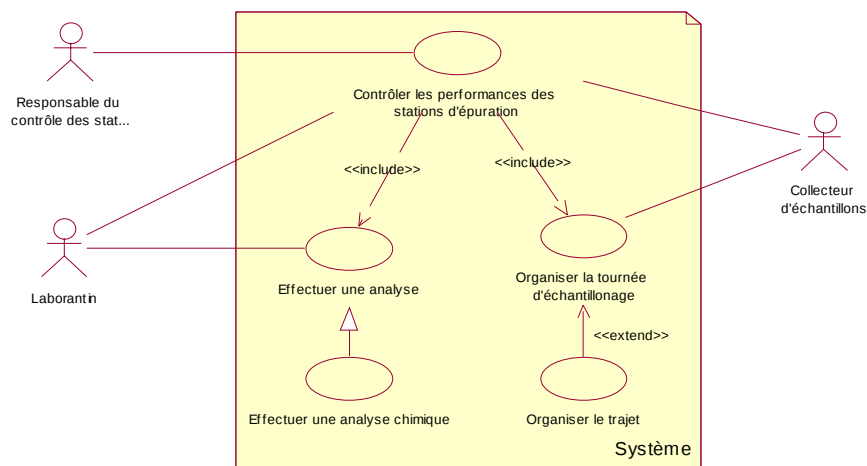


Figure 142. Exemple de représentation de cas d'utilisation, d'acteurs et de relations entre ces derniers.

Deux catégories de cas d'utilisation peuvent être distinguées (Cockburn, 2000) : les cas d'utilisation-système ("system use cases"), et les cas d'utilisation réels ("business use cases"). Les premiers traduisent les interactions successives d'un acteur avec le système informatique projeté, en vue d'atteindre un objectif particulier, tandis que les seconds décrivent les séquences d'actions telles qu'elles se succèdent dans le processus-métier actuel, sans référence au futur outil informatique. L'élaboration de cas d'utilisation permet de comprendre le fonctionnement et les structures d'un système (réel ou informatique), aide à l'identification des concepts du domaine, sert à mettre en place une base de compréhension commune entre les informaticiens et les futurs utilisateurs du SI projeté, et facilite enfin l'analyse des besoins.

La compréhension de diagrammes de classes ou de modèles de données nécessite de gros efforts de compréhension de la part d'interlocuteurs non familiarisés avec de tels formalismes, ce qui ne facilite pas les échanges entre informaticiens et praticiens. Or, l'une des missions des modèles cités précédemment est précisément de faciliter la communication entre ces deux catégories de personnes. Les cas d'utilisation sont par contre des descriptions textuelles, ce qui constitue à cet égard un avantage certain. Ils peuvent en effet être rédigés par les concepteurs du SI, sur la base d'entretiens avec des acteurs, puis corrigés par ces derniers, quelles que soient leurs connaissances en informatique. La figure 143 montre que les cas d'utilisation sont nettement mieux compris par les non-informaticiens que les différents formalismes de modélisation proposés par UML, et en particulier que les diagrammes de classes. Ils constituent un moyen efficace pour capturer les connaissances des praticiens, et permettent une communication aisée avec ces derniers.

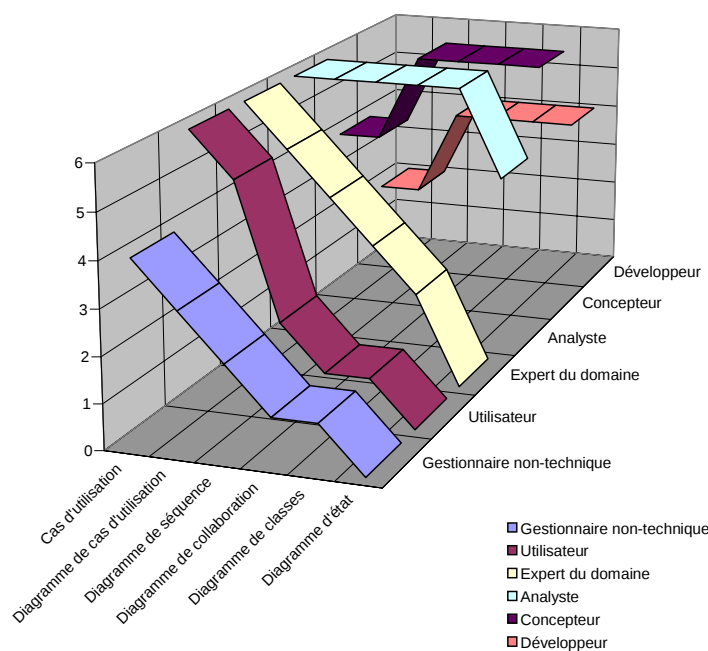


Figure 143. Degré de compréhension de différents types de modèles par différents intervenants, selon leur rôle dans le processus de développement (d'après Arlow et al., 1998).

Annexe 7 Infrastructures de persistance

De nombreuses informations sont disponibles dans la littérature quant à la manière de concevoir des infrastructures de persistance interagissant avec des BD relationnelles (Ambler, 1998a; Wright, 1998; Yoder et al., 1998). Stoiber (1999) compare les alternatives : tout d'abord cet auteur rappelle qu'une telle infrastructure a non seulement pour mission de gérer la persistance des objets informatisés, mais également d'autoriser des requêtes conduisant à l'identification d'objets (isolés ou en grand nombre) et de gérer les transactions. Ces infrastructures doivent aussi rendre possible l'introduction de nouveaux types d'objets métier persistants dans les systèmes sans qu'il soit nécessaire de les modifier. Stoiber distingue trois "philosophies" en matière de création d'infrastructures de persistance, baptisées "Persistent Object", "DB Object" et "Persistence Broker" (figure 144) :

- **Persistent Object** : chaque objet métier gère lui-même sa persistance, les requêtes permettant d'accéder aux bases de données étant nichées au sein du code informatique. Les composants domaine s'appuient selon ce schéma directement sur les composants-ressources qui permettent la liaison physique avec les BD. Cette approche, très peu souple, est à déconseiller car tout changement dans la structure de données implique de devoir modifier le composant-domaine : le code lié à la logique métier est mélangé au code permettant l'interaction avec la BD.
- **DB Object** : selon ce modèle, à chaque objet métier implémenté par le composant-domaine correspond un objet données implémenté par le composant données. Les objets données assurent la correspondance entre les attributs des objets métier et les attributs du modèle de données, et s'appuient sur les composants-ressources pour interagir avec la BD. Un changement de la structure de données nécessite uniquement un changement du composant-données. Cette approche, proposée par Wright (1998), a été adoptée lors du développement du prototype présenté au chapitre 6.2.

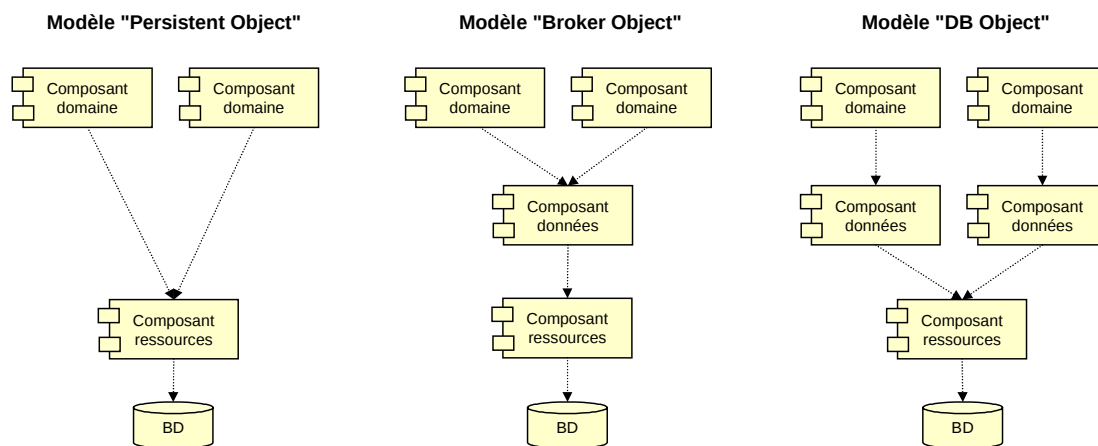


Figure 144. Les trois modèles de persistance présentés induisent des configurations différentes en ce qui concerne les composants constituant un composant métier.

- **Persistence Broker** : c'est le modèle le plus complexe, mais aussi le plus souple. Le composant données se charge des interactions entre les objets métier et la BD, mais est conçu de manière à ne pas devoir être modifié lorsque de nouveaux types d'objets métier sont introduits ou lorsque le modèle de données structurant la BD subit de légères modifications. Selon ce modèle, chaque objet métier persistant implémente une interface spécifique. Un objet "courtier d'objets persistants",

collaborant avec d'autres objets du composant données, est alors à même de se charger du transfert de cet objet-métier vers la BD. Un descriptif de la correspondance entre attributs des objets métier persistants et modèle de données est disponible soit au niveau de la BD elle-même, soit sous forme d'un fichier indépendant. Ce descriptif peut ainsi être modifié sans qu'il soit nécessaire de modifier le code de l'application. Un tel modèle est décrit par Ambler (1998a).

Annexe 8 La vision de l'Open GIS Consortium (OGC)

• Description générale

Le domaine des SIG évolue à l'heure actuelle dans la même direction que celle qui a été prise par l'industrie du logiciel en général, c'est-à-dire des applications monolithiques, propriétaires, aux applications ouvertes, construites par intégration de composants et pouvant être réparties sur différentes machines. Une association à but non lucratif, l'Open GIS Consortium (OGC) a été créée en 1994 avec pour objectif de faciliter l'interopérabilité des applications manipulant de l'information à caractère spatial. Cette association comptait 225 membres en mai 2000, parmi lesquels des sociétés privées, fabricants et intégrateurs de logiciels, des organismes publics et des institutions de recherches issus du monde entier. L'OGC se propose de définir un cadre conceptuel et un ensemble de spécifications sur lesquels les fabricants de SIG pourront s'appuyer pour développer leurs produits. L'adoption et le suivi des spécifications de l'OGC, baptisées spécifications OpenGIS, devraient permettre à ces fabricants de proposer des produits compatibles entre eux, c'est-à-dire à même d'échanger de l'information spatiale et des services de manière transparente, et éventuellement d'être assemblés pour former des systèmes complets répondant à de multiples besoins.

L'approche proposée par l'OGC est basée sur le paradigme orienté objet, et plus particulièrement sur les concepts d'encapsulation et d'interfaces. Ces notions veulent que les méthodes et données propres à un objet soient dissimulées derrière des interfaces et ne soient atteignables que par le biais de ces dernières. Ceci permet d'une part au développeur d'application qui utilise cet objet de ne se préoccuper que des services offerts par l'objet, sans considération de la manière dont ces services sont implémentés, et d'autre part au développeur de l'objet de protéger son implémentation. L'OGC se propose donc de fournir des spécifications consistant principalement en la description d'un ensemble d'interfaces standardisées se rapportant au domaine de l'information spatiale. Les différentes applications ou composants logiciels qui souhaitent se conformer à ces spécifications doivent interagir par le biais de ces interfaces standardisées. Ceci assure leur interopérabilité tout en garantissant aux développeurs une grande liberté dans la manière d'implémenter ces interfaces et le choix du langage de développement.

L'OGC propose un modèle opérationnel, et non un modèle de données. Les données géographiques peuvent être stockées dans les différentes structures de données spécifiques aux différents SGBD et SIG disponibles sur le marché. L'OGC propose la mise en place d'un ensemble d'outils logiciels qui permettent de transférer de manière dynamique des données géographiques issues de différents systèmes de stockage dans un modèle objet commun (Gardels, 1996).

Afin de pouvoir être mises en oeuvre dans le contexte des environnements pour composants (distribués) qui existent à l'heure actuelle (dont les principaux sont CORBA, COM et Java, voir chapitre 4.3), les spécifications OpenGIS sont définies selon trois niveaux conceptuels différents, représentés dans la figure 145 (Open GIS Consortium, 1998b) :

- un modèle essentiel (essential model), qui représente les entités du "monde réel", leurs attributs et leurs relations tels qu'ils sont perçus par les personnes qui écrivent les spécifications. Ce modèle permet de faire le lien entre le "monde réel" et le "monde logiciel", et décrit la succession d'abstractions qui permettent de passer des entités "du monde réel" aux entités du "monde logiciel";
- un modèle abstrait (abstract model), qui est un modèle générique du système logiciel, indépendant

de toute considération d'implémentation, avec ses objets, la description des états dans lesquels ces derniers peuvent se trouver et des stimuli auxquels ils peuvent répondre.

- des modèles d'implémentation, qui sont des traductions du modèle abstrait spécifiques aux différents environnements distribués (figure 145). Ces modèles sont des modèles objet, et un modèle d'implémentation doit être disponible pour chaque infrastructure logicielle.

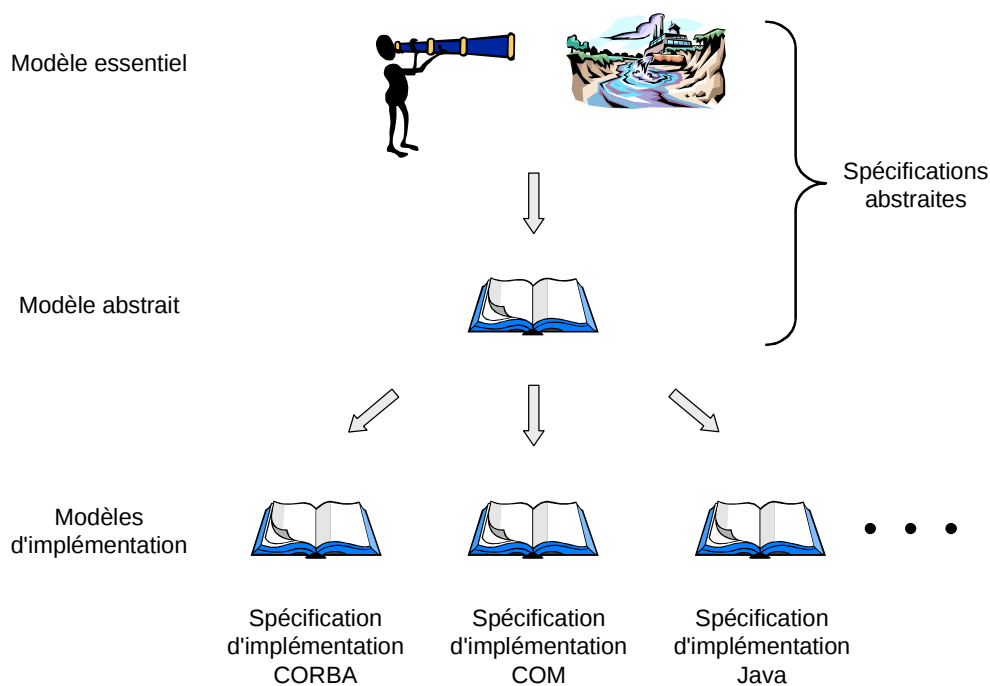


Figure 145. La structuration en trois niveaux des spécifications OpenGIS.

L'OGC ne propose donc pas un environnement et une infrastructure de plus pour la mise en place d'applications distribuées, mais des spécifications permettant d'utiliser les infrastructures existantes dans le contexte de l'interopérabilité des applications manipulant de l'information à caractère spatial.

Le modèle abstrait et le modèle essentiel sont regroupés sous le terme de "spécifications abstraites". Ces dernières sont donc neutres vis-à-vis des technologies informatiques disponibles. Leur finalité est de proposer et de documenter un modèle conceptuel relatif à l'information spatiale qui soit suffisant pour permettre la rédaction de spécifications d'implémentation. Les spécifications abstraites sont organisées en différents volumes traitant de sujets spécifiques (Open GIS Consortium, 1999b) et utilisent le formalisme UML (Booch et al., 1999) ainsi que le langage naturel pour définir les modèles. A noter que ces différents volumes n'en sont pas au même stade d'avancement, certains d'entre eux étant en phase d'élaboration, d'autres étant terminés et aptes à servir de base à l'élaboration de spécifications d'implémentation. Les sujets traités sont par exemple la géométrie des entités spatiales (feature geometry), les systèmes de références (spatial reference systems), les métadonnées (metadata), la photogrammétrie (earth imagery).

• *Les entités spatiales*

L'unité fondamentale pour l'OGC, dans le domaine de l'information spatiale, est l'entité (feature) (Open GIS Consortium, 1999b) :

"Features are digitally coded abstractions of real-world objects and phenomena that have a geometric representation and space/time and other attribution associated with them. Features are created, managed, accessed, interchanged and manipulated by the OpenGIS Services [...]" (Open GIS Consortium, 1998b).

Selon cette définition, un réseau d'assainissement, une photographie aérienne, un déversoir d'orage sont des entités. L'OGC distingue deux catégories particulières d'entités, les "entités avec géométrie" (feature with geometry), dont l'extension spatiale peut être décrite par le biais de primitives géométriques bien définies (points, lignes, polygones), et les "couvertures" (coverage) :

"A coverage is an association of points within a spatial/temporal domain to a value [...]. That is, in a coverage each point has particular simple or complex value. [...] a coverage in the OpenGIS Specification is simply a function which can return its value at a geometric point." (Open GIS Consortium, 1998b)

Une image satellitaire, un modèle numérique d'altitude, une carte de répartition des populations sont des "couvertures".

• Les spécifications

Nous nous intéressons à partir de là plus spécifiquement aux entités avec géométrie.

L'OGC propose, dans ses **spécifications abstraites**, un ensemble d'interfaces permettant de décrire et de manipuler les caractéristiques géométriques d'entités spatiales (Open GIS Consortium, 1999b). GM_Object est la classe de base des objets géométriques (figure 146), et contient de ce fait des données et supporte des interfaces qui sont communes à tous les objets géométriques possédant une référence spatiale.

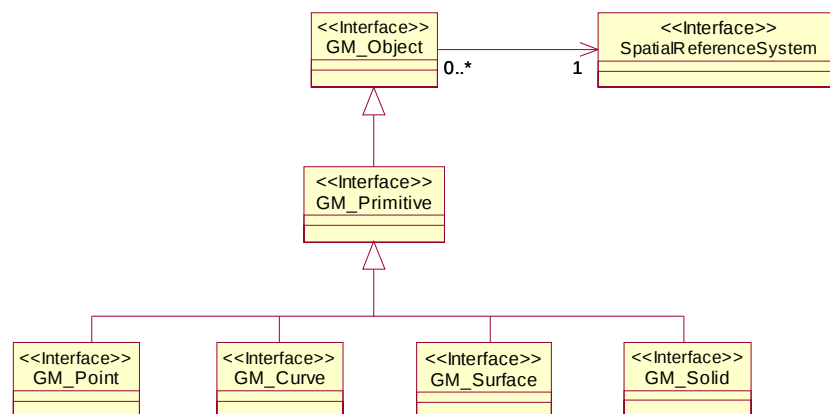


Figure 146. Extrait du modèle objet relatif aux entités spatiales simples proposé par l'OGC (adapté de Open GIS Consortium, 1999b).

On notera en particulier qu'un objet GM_Object possède un lien (unidirectionnel) vers un objet de type SpatialReferenceSystem. La base de la géométrie est le point; les points peuvent être assemblés en séquences pour former des lignes, qui peuvent être groupées pour former des polygones, qui à leur tour peuvent être groupés pour former des polyèdres : ainsi les primitives GM_Curve, GM_Surface et

GM_Solid sont-il aussi définis, directement ou indirectement, par des objets de type GM_Point.

Lorsque des spécifications abstraites sont définies, l'OGC, suivant en cela une procédure bien établie (Open GIS Consortium, 1998b), émet une invitation à destination de l'industrie à lui soumettre les **spécifications d'implémentation** correspondantes. Par exemple, des spécifications d'implémentation pour les "entités simples" (simple features) ont été proposées pour Corba (Open GIS Consortium, 1998a) et pour COM (Open GIS Consortium, 1999a). Des exemples d'interfaces sont donnés aux tableaux 14 et 15. En se basant sur les spécifications d'implémentation, les interfaces définies peuvent être implémentées.

```
exception OutOfDomain {};
```

```
interface Curve : Geometry {
    exception WKSNotImplemented {};
```

```
    readonly attribute Double length;
    readonly attribute Point start_point;
    readonly attribute Point end_point;
    readonly attribute WKSPoint start_point_as_WKS;
    readonly attribute WKSPoint end_point_as_WKS;
```

```
    boolean is_planar
```

```
    Point value (in double r) raises (OutOfDomain);
    WKSPoint value_as_WKS (in double r) raises (OutOfDomain);
```

Tableau 14. Spécification Corba de l'interface Curve (Open GIS Consortium, 1998a).

```
// Description de la co-classe LineString, qui implémente différentes interfaces, // dont
l'interface ICurve
coType LineString

{
    [mandatory] interface IGeometry;
    [mandatory] interface ICurve;
    [mandatory] interface ILineString;
    [mandatory] interface IWks;

    [optional] interface ISpatialRelation;
    [optional] interface ISpatialRelation2;
    [optional] interface ISpatialOperator;
};

// Description de l'interface ICurve

[ object, uuid(6A124036-FE38-11d0-BECE-00805F7C4268) ]
interface ICurve : IGeometry
    [propget] HRESULT Length([out, retval] double* value);
    HRESULT StartPoint([out, retval] IPoint** sp);
    HRESULT EndPoint([out, retval] IPoint** ep);
    [propget] HRESULT IsClosed([out, retval] VARIANT_BOOL * isClosed);
    HRESULT Value([in] double t, [out, retval] IPoint** p);
};
```

Tableau 15. Spécification OLE/COM de l'interface ICurve implémentée par la "coclasse" LineString (Open GIS Consortium, 1999a).

Annexe 9 Le composant logiciel SIG MapX

MapX est un contrôle ActiveX développé par la société MapInfo (MapInfo Corporation, 1999a; MapInfo Corporation, 1999b). Ce composant logiciel permet aux développeurs d'intégrer dans leurs applications logicielles des fonctionnalités de visualisation et de manipulation d'objets géoréférencés. Bien que d'autres composants SIG soient disponibles sur le marché, le présent chapitre est essentiellement consacré à MapX en raison de l'expérience de l'auteur avec ce dernier dans le cadre de divers projets. L'objectif des lignes qui suivent est d'illustrer quelques-unes des possibilités qu'offre un tel composant.

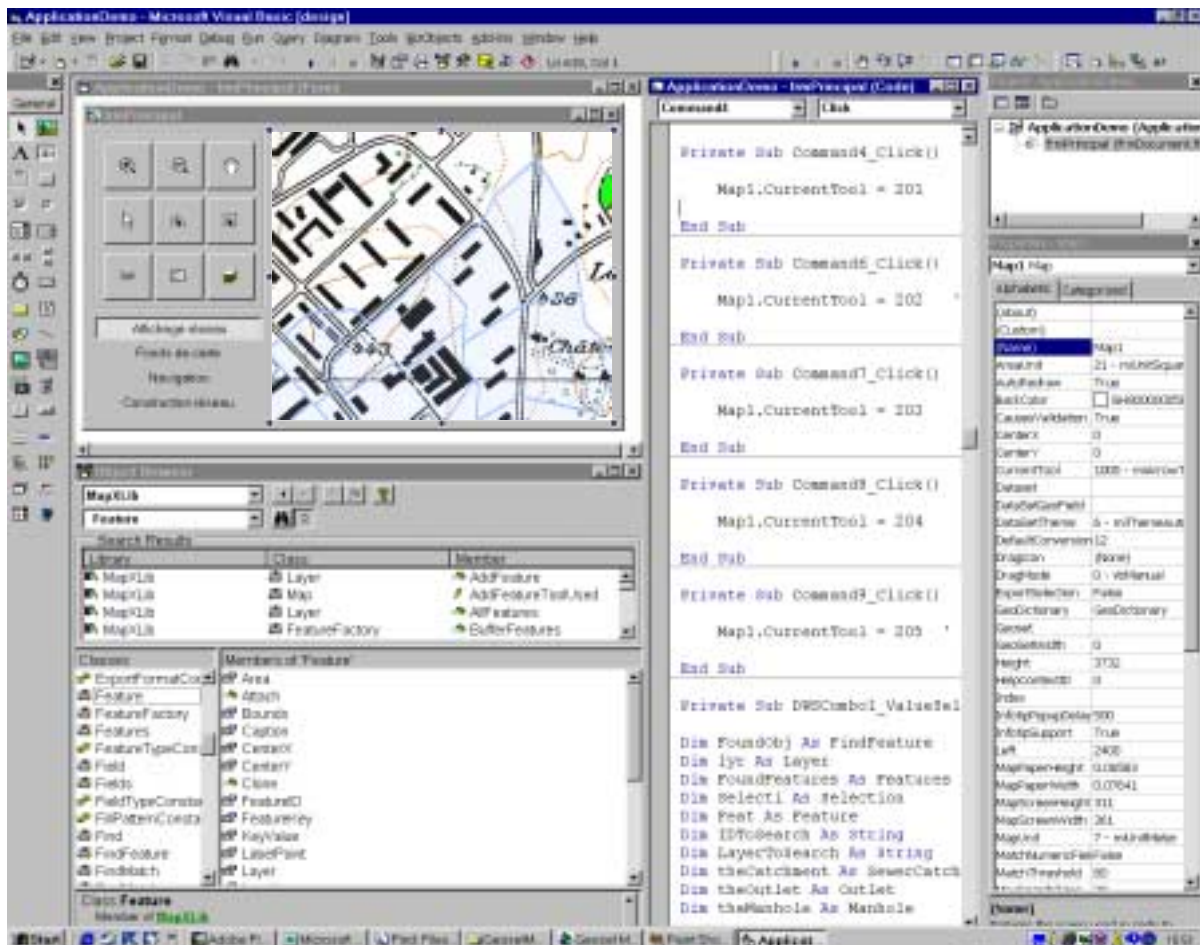


Figure 147. MapX dans l'environnement de développement Visual Basic 6 de Microsoft.

Sa qualité de contrôle ActiveX, basée sur la technologie COM (voir annexe 3) fait que MapX peut être mis en oeuvre par le biais de nombreux langages de programmation, lors du développement d'applications destinées à fonctionner sous environnement Windows (Microsoft). La société MapInfo mentionne la compatibilité de MapX avec les langages Visual Basic, Visual C++, PowerBuilder, Delphi et Lotus Notes. MapX n'est pas prévu pour être déployé sur Internet (la société MapInfo proposant d'autres produits à cet effet), mais est conçu pour être intégré à des applications côté client. Son intégration côté serveur est cependant également envisageable. Au titre de ses fonctionnalités, on citera :

- les capacités de visualisation et d'édition cartographique de données;
- la possibilité de créer des cartes thématiques, en associant aux entités géographiques des données

thématiques, extraites par exemple d'une base de données;

- la possibilité d'annoter et de documenter les cartes par le biais de légendes, labels et autres zones de texte;
- la possibilité d'effectuer différentes sortes de recherches et de requêtes spatiales;
- le développement très aisé d'outils "sur mesure" permettant d'interagir avec la carte;
- la gestion de différents systèmes de projection.

La figure 147 présente l'utilisation de MapX dans le contexte d'un développement en Visual Basic 6.0 de Microsoft. Le composant est proposé dans les outils mis à disposition par la boîte à outils de l'environnement de programmation (sur la droite de l'illustration), et une instance de ce composant, se présentant sous la forme d'une carte, est intégrée au formulaire en cours de développement.

L'examen du modèle objet de MapX (MapInfo Corporation, 1999b), dont la figure 148 présente les caractéristiques principales, révèle que l'objet "Map" en est le constituant principal. Cet objet, dont l'interface graphique est la carte intégrée dans une application, est constitué d'un certain nombre de couches ("Layer"), qui peuvent être vues comme des transparents et sont superposées les unes aux autres pour constituer la carte telle que vue par l'utilisateur.

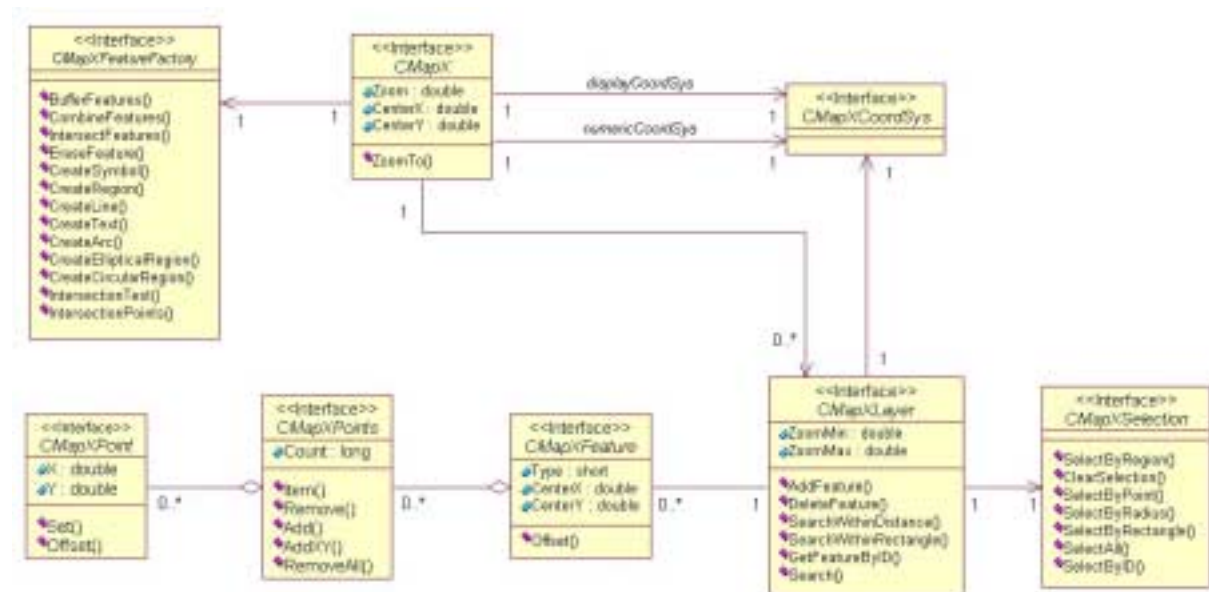


Figure 148. Les interfaces principales (simplifiées) implémentées par le composant ActiveX MapX. Les attributs (losange bleu) et opérations (losanges rouges) de ces interfaces sont représentés. Ces interfaces permettent la manipulation d'entités géographiques de différents types.

Les entités géoréférencées ("Features") représentent les entités réelles que l'on souhaite représenter sur la carte (communes, routes, stations de mesure,...). Elle peuvent être de type point, ligne, région, mixte, inconnu ou étiquette de texte. Ces entités ne représentent toutefois que la composante géographique des entités réelles. Afin de pouvoir visualiser la répartition géographique des différents attributs thématiques, MapX intègre un mécanisme de liaison de données ("data binding"). Le résultat d'un processus de liaison de données conduit à la création d'objets de type jeu de données "Dataset" contenant les attributs thématiques relatifs aux entités géographiques, qui peuvent être mis en relation avec ces dernières, ce qui permet de caractériser chaque entité géographique par des attributs thématiques issus

d'une base de données ou d'un fichier. Le mécanisme de liaison de données ouvre la voie à la cartographie thématique, c'est-à-dire à la représentation cartographique d'informations thématiques.

Les entités géoréférencées, quelles qu'elles soient, sont basées sur un ou plusieurs objets primitifs de type point, dont les coordonnées sont caractérisées par deux valeurs (MapX gère deux dimensions). Ces coordonnées ne donnent un sens à la localisation d'une entité géoréférencées qu'à partir du moment où ce géoréférencement dépend d'un système de coordonnées bien défini. Le modèle objet de MapX inclut donc un type d'objet Système de coordonnées ("CoordSys"). Des instances de ce dernier doivent être associées directement au composant "Map" pour permettre la manipulation et l'affichage des entités géoréférencées.

Le type d'objet Fabrique d'objets ("FeatureFactory") permet, comme son nom l'indique, la création d'entités géoréférencées et constitue à ce titre un élément très important du modèle objet de MapX. Les opérations de ce type d'objet acceptent comme paramètre soit un point ou une collection de points, soit d'autres entités géographiques, et renvoient en résultat l'entité nouvellement créée. Ces opérations permettent par exemple de créer des zones-tampons autour d'une entité géoréférencée, d'agréger des entités géoréférencées, de créer des entités à géométrie particulière, comme par exemple des ellipses, etc. La figure 149 et le tableau 14 présentent un tel exemple de manipulation d'entités spatiales.

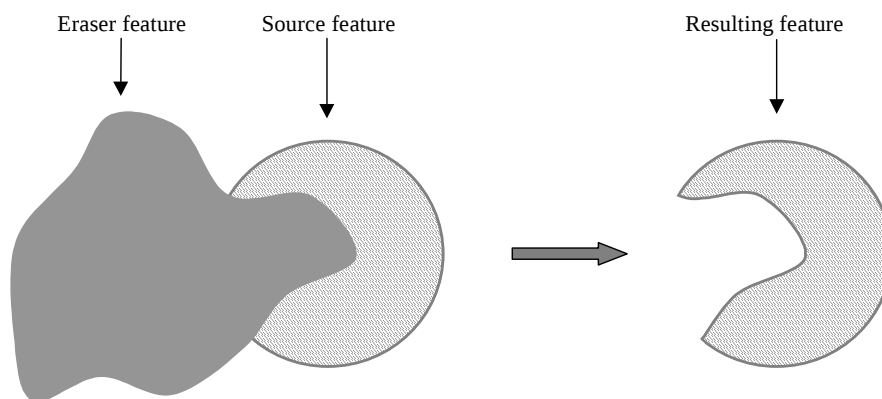


Figure 149. Exemple d'utilisation d'un objet de type *FeatureFactory*. Sa méthode *EraseFeature*, de paramètres *SourceFeature* et *EraserFeature*, retourne comme résultat un objet résultant du découpage de l'objet source par l'objet découpeur (adapté de MapInfo Corporation, 1999).

Enfin, pour terminer cet aperçu des fonctionnalités que peut offrir un composant tel que MapX, mentionnons la possibilité d'effectuer des recherches et sélections, par le biais de l'objet "Selection", qui permet de rechercher des entités géoréférencées sur la base de critères spatiaux (sélection des entités à l'intérieur d'une région, d'un cercle, d'un rectangle, en un point précis, sur la base de l'identifiant de l'entité). Un objet Sélection est en fait une collection d'entités géoréférencées offrant des opérations particulières lui permettant de mener à bien ces recherches. Notons enfin qu'il est également possible d'effectuer des requêtes thématiques, mettant en oeuvre des objets de type "Dataset" et "Find".

```
Private Sub Effacer_Click()  
  
Dim EraserFeature As Feature  
Dim SourceFeature As Feature  
Dim ResultingFeature As Feature  
  
    ' Définition des objets cible et source  
    Set EraserFeature = layer1.Selection.Item(1)  
    Set SourceFeature = layer1.Selection.Item(2)  
  
    ' Opération de découpage  
    Set ResultingFeature = Map1.FeatureFactory.EraseFeature(SourceFeature, _  
        EraserFeature)  
  
    ' Affichage de la surface de l'objet découpé  
    MsgBox ("Surface : " & ResultingFeature.Area)  
  
End Sub
```

Tableau 16. Procédure, en Visual Basic, effectuant l'opération décrite à la figure 149. A noter l'utilisation de l'objet de type *FeatureFactory* appartenant à la carte nommée *Map1*.

Annexe 10 Exemples de cas d'utilisation

La figure 95 du présent rapport a présenté le modèle des cas d'utilisation considéré lors du développement du prototype élaboré dans le cadre de ce travail. Nous nous concentrons ci-dessous sur quatre de ces cas d'utilisation, qui sont de type réel développé, et ont été élaborés à partir d'entretiens menés avec certains des intervenants cités au tableau 1. Sont également présentés deux cas d'utilisation relatifs à la gestion des données de l'assainissement, de type système développé. La figure 150 montre le modèle de ces cas d'utilisation. Les renvois à des cas d'utilisation inclus sont signalés par soulignement dans les cas de base (bien qu'ils ne figurent pas dans le présent rapport).

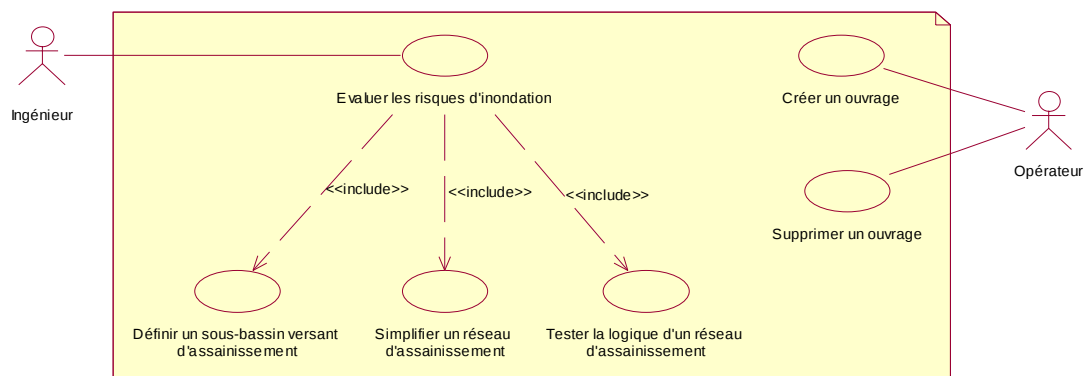


Figure 150. Modèle des cas d'utilisation présentés dans l'annexe 10.

Cas d'utilisation :	Evaluer les risques d'inondation
Numéro :	CU_4
Date dernière modification:	16.02.2000
Version :	1.0
Statut :	En révision
Type :	Réel, développé
Objectif :	Caractériser le fonctionnement hydraulique d'un réseau d'assainissement. Le mandant souhaite en particulier savoir où le réseau est sous-dimensionné et présente des risques de dysfonctionnement (débordements) par temps de pluie.
Acteur principal :	Ingénieur
Acteurs secondaires :	Ingénieur, mandant, logiciel de simulation hydraulique
Pré-condition :	
Résultat en cas de succès :	L'ingénieur fournit au mandant une information relative à la capacité hydraulique du réseau qu'il a été chargé d'étudier, discrétisée par portions de réseau et pour différents temps de retour.

- Résultat en cas d'échec : L'ingénieur informe le mandant qu'il ne lui est pas possible de fournir d'appréciation quant à la capacité hydraulique des tronçons du réseau d'assainissement.
- Déclencheur : Le mandant (représentant des autorités communales ou cantonales) demande à l'ingénieur de lui fournir une analyse de la capacité hydraulique d'un réseau d'assainissement qu'il identifie (voir CE6).

Scénario principal

1. *L'ingénieur identifie ou reporte grossièrement, sur un plan, les limites globales du bassin versant d'assainissement qu'il est chargé d'étudier.*
2. *L'ingénieur recherche les documents nécessaires à son étude qui correspondent au bassin versant d'assainissement qu'il a délimité : cadastre des canalisations, cartes topographiques, éventuellement photographies aériennes, éventuellement information topographique type modèle numérique d'altitude.*
3. *L'ingénieur simplifie le réseau d'assainissement (voir CE7) à l'intérieur de la zone d'étude : il "fusionne" les canalisations qui se succèdent sans que leurs caractéristiques (pente, diamètre, matériau) ne varient beaucoup, il sélectionne les ouvrages de réseau qui sont d'un niveau de détail adéquat (élimine les canalisations correspondant à un niveau de détail trop élevé). L'ingénieur calcule les caractéristiques du réseau simplifié : pour chaque canalisation simplifiée, il relève le radier de l'ouvrage amont, le radier de l'ouvrage aval; il calcule une pente moyenne; il choisit un diamètre moyen ainsi qu'un matériau représentatif (ceci en tenant compte des diamètres et matériaux des canalisations "réelles" qui sont fusionnées dans la canalisation simplifiée); il reporte ces caractéristiques sur un plan.*
4. *L'ingénieur délimite les sous-bassins versants (voir CE8). Il se sert pour cela des ouvrages qui n'ont pas disparu lors de la simplification précédente, et qui peuvent faire office d'exutoires (pour la modélisation hydraulique) pour les sous-bassins versants qui sont délimités. Il tient compte du relief, des couvertures de sol, du réseau de canalisation existant, des bâtiments assainis, de son expérience personnelle. L'ingénieur calcule les caractéristiques des sous-bassins versants d'assainissement qu'il a identifiées. Pour chaque sous-bassin versant, il calcule la superficie, il estime le coefficient de ruissellement, il estime éventuellement la proportion de toits et de voies de circulation, il estime éventuellement le nombre d'équivalents-habitants, ce qui lui permet d'évaluer le débit par temps sec. Il reporte ces caractéristiques sur un plan ou des documents annexes.*
5. *L'ingénieur teste la logique du réseau qu'il a sélectionné (voir CE9), c'est-à-dire qu'il vérifie si sa topologie est correcte.*
6. *L'ingénieur importe le réseau de simulation simplifié dans le logiciel de simulation (voir CE 10) : il introduit les paramètres qu'il a déterminés en ce qui concerne les sous-bassins versants d'assainissement, le réseau d'assainissement, il fixe les paramètres réseau et bassin spécifiques au logiciel.*
7. *L'ingénieur sélectionne les données hydrologiques (voir CE 11) qu'il va utiliser : il se procure les pluies isolées ou séries de pluies qu'il veut utiliser; il les importe dans le logiciel de simulation utilisé (voir CE 12).*
8. *L'ingénieur sélectionne les données hydrauliques (voir CE 13) qui vont servir au calage du modèle; il se procure les événements de débits ou séries de débits qu'il veut utiliser; il les importe dans le logiciel de simulation utilisé (voir CE 14).*
9. *L'ingénieur cale le modèle (voir CE 15) à l'aide d'événements pluviaux isolés : il effectue quelques simulations, examine les résultats et s'il n'en est pas satisfait, modifie les paramètres de calage et refait des simulations; s'il est satisfait, il enregistre le réseau de modélisation avec les paramètres définitifs.*

10. L'ingénieur effectue les simulations "prévisionnelles" (voir CE 16), soit pour des événements pluviaux isolés (pluies historiques), soit pour des séries de pluies de longue durée: il examine les résultats et les reporte. Il indique en particulier les canalisations qui sont sous-dimensionnées.
11. L'ingénieur fournit les résultats au mandant.

Scénarios alternatifs

- 2.a. Il n'existe pas de cadastre des canalisations pour la zone étudiée:
- 2.a.1.a. L'ingénieur va chercher l'information dans d'autres documents (plans d'exécution,...).
- 2.a.1.b. L'ingénieur organise et effectue une campagne de relevé du réseau (CE 17) et de ses infrastructures.
- 3.a. L'ingénieur n'effectue pas de simplification du réseau, mais se contente de définir le niveau hiérarchique de canalisations à partir duquel il effectue la modélisation (typiquement il ne considère que les réseaux primaires et secondaires). Si le réseau n'a pas été simplifié (voir point 3), on vérifie simplement que les attributs nécessaires sont disponibles, et si nécessaire on complète l'information.
- 7.a. L'ingénieur ne dispose pas de pluies historiques ou de séries de pluies historiques :
- 7.a.1.a. L'ingénieur génère des séries de pluies (CE 18).
- 7.a.1.b. L'ingénieur crée des averses synthétiques (CE 19).
- 8.a. L'ingénieur ne dispose pas de séries de débits :
- 8.a.1.a. L'ingénieur effectue ses simulations sans procéder au calage.
- 8.a.1.b. L'ingénieur organise et effectue une campagne de mesures (CE 20).
- 9.a. L'ingénieur ne procède pas au calage de son modèle.

Cas d'utilisation :	Simplifier un réseau d'assainissement
Numéro :	CE7
Date :	16.02.2000
Version :	1.0
Statut :	En révision
Type :	Réel, développé
Objectif :	Simplifier un réseau afin de pouvoir faire une simulation des écoulements. La simplification effectuée n'est pas spécifique à un logiciel de simulation.
Acteur principal :	Ingénieur
Acteurs secondaires :	
Pré-condition :	Une portion du réseau d'assainissement a été identifiée, et tous les documents et renseignements nécessaires sont disponibles.
Résultat en cas de succès :	La portion de réseau d'assainissement isolée a été simplifiée, c'est-à-dire qu'un réseau virtuel simplifié, avec des paramètres calculés, a été identifié et sélectionné.
Résultat en cas d'échec :	La portion de réseau d'assainissement n'a pas été simplifiée.
Déclencheur :	L'ingénieur décide de simplifier la portion de réseau d'assainissement qui a été identifiée.

Scénario principal

1. *L'ingénieur décide des niveaux hiérarchiques qu'il considère dans le cadre de sa simplification de réseau. Par exemple, il peut renoncer à inclure les conduites de raccordement.*
2. *L'ingénieur identifie et "marque" les ouvrages "ponctuels" (chambres, ouvrages spéciaux, exutoires,...) qui appartiennent au réseau qu'il a pour objectif de simplifier et qui doivent figurer dans le réseau simplifié : ouvrages importants en taille, situés en des endroits stratégiques : intersections, extrémités de réseau, exutoires de zones de construction d'un type particulier,...*
3. *L'ingénieur examine les branches de réseau d'assainissement situées entre les ouvrages qu'il a identifiés au point précédent : il identifie et "marque" les ouvrages qui sont situés de part et d'autre de successions de canalisations aux caractéristiques - profil, diamètre, matériau, pente - relativement constantes.*
4. *L'ingénieur crée des tronçons virtuels entre les ouvrages qu'il a "marqués", c'est-à-dire qu'il agrège l'information caractérisant les canalisations entre les ouvrages "marqués" : pour chaque branche de réseau située entre deux ouvrages marqués, il calcule une pente moyenne, un diamètre moyen, une longueur totale et choisit un matériau représentatif.*
5. *L'ingénieur reporte sur un plan les ouvrages "marqués", les canalisations fictives ainsi que les caractéristiques calculées de ces dernières.*

Cas d'utilisation :	Définir un sous-bassin versant d'assainissement
Numéro :	CE8
Date :	06.02.2000
Version :	1.0
Statut :	En révision
Niveau :	Réel, développé
Objectif :	Identifier et caractériser le sous-bassin versant d'un ouvrage de réseau d'assainissement. Le sous-bassin est identifié par son contour et caractérisé par des paramètres.
Acteur principal :	Ingénieur
Acteurs secondaires :	
Pré-condition :	Un ouvrage considéré comme l'exutoire du sous-bassin versant d'assainissement qui doit être défini a été identifié et les documents nécessaires (cadastre, plans d'ensemble,...) sont disponibles.
Résultat en cas de succès :	Le sous-bassin versant d'assainissement de l'ouvrage spécifié a été identifié, et ses caractéristiques calculées.
Résultat en cas d'échec :	Le sous-bassin versant d'assainissement n'a pas été identifié.
Déclencheur :	L'ingénieur décide de déterminer le sous-bassin versant d'assainissement d'un ouvrage de réseau d'assainissement particulier.

Scénario principal

1. *L'ingénieur délimite le périmètre du sous-bassin versant d'assainissement : il examine le réseau d'assainissement et détermine à quelles canalisations sont connectés les bâtiments et surfaces imperméables. Il examine les courbes de niveau, lorsqu'elles sont disponibles. Il délimite le sous-bassin versant recherché par une surface qui comprend tous les bâtiments et les parties de surfaces imperméables qui sont connectées au réseau d'assainissement entre l'ouvrage de réseau considéré et l'ouvrage suivant en se déplaçant vers l'amont; de plus, cette surface comprend les surfaces perméables et imperméables dont l'eau pluviale ruisselle jusqu'au tronçon de canalisation cité précédemment. Enfin, il est attentif à ce que le sous-bassin versant qu'il identifie ne recouvre pas, même partiellement, un autre sous-bassin versant. Si besoin est, il corrige les limites du sous-bassin versant en fonction de son intuition et de son expérience personnelle.*
2. *L'ingénieur nomme le sous-bassin versant, calcule sa surface, calcule son coefficient de ruissellement (voir CE 22), calcule la proportion de toits, de surfaces routières et de surfaces perméables dans le sous-bassin versant.*
3. *L'ingénieur détermine le nombre d'équivalents-habitants hydrauliques du sous-bassin versant: il détermine le nombre d'habitants et de places de travail qu'il y a dans le bassin versant. A partir de là, il calcule le nombre d'équivalents-habitants hydrauliques.*
4. *L'ingénieur estime un débit théorique d'eaux usées par temps sec.*

Cas d'utilisation :	Vérifier la logique du réseau d'assainissement
Numéro :	CE9
Date dernière modification:	06.02.2000
Version :	1.0
Statut :	En révision
Type :	Réel, développé
Objectif :	Etre sûr que le réseau identifié est cohérent du point de vue topologique, défini correctement, et que les attributs nécessaires sont renseignés.
Acteur principal :	Ingénieur
Acteurs concernés :	
Pré-condition :	Un réseau d'assainissement ou une zone englobant un certain nombre d'ouvrages d'assainissement a été identifié.
Résultat en cas de succès :	L'ingénieur sait si oui ou non son réseau d'assainissement est cohérent du point de vue topologique et que les attributs minimums ont été renseignés. Si non, il dispose d'une information sur les incohérences identifiées.
Résultat en cas d'échec :	L'ingénieur ne dispose pas de l'information quant à la cohérence de la portion de réseau qu'il examine.
Déclencheur :	L'ingénieur décide de vérifier la cohérence d'un réseau d'assainissement ainsi que la complétude des attributs des éléments de ce réseau.

Scénario principal

1. *L'ingénieur vérifie que tous les ouvrages qui sont sélectionnés sont connectés entre eux.*
2. *L'ingénieur vérifie qu'il n'y ait pas d'incohérence dans les sens d'écoulement.*

3. *L'ingénieur vérifie que tous les ouvrages du réseau d'assainissement sélectionné aient les attributs suivants renseignés : pour les ouvrages ponctuels : au moins un jeu de coordonnées, au moins une cote de radier, au moins une cote de débordement (niveau couvercle, niveau du sol); pour les canalisations : une cote amont, une cote aval, une longueur ou des coordonnées amont et des coordonnées aval, un matériau.*
4. *L'ingénieur identifie les endroits où la cote du radier de l'ouvrage aval est supérieure à celle de l'ouvrage amont. Lorsqu'il identifie une telle situation, il contrôle qu'il ne s'agisse pas d'une erreur.*
5. *L'ingénieur vérifie que les sous-bassins versant d'assainissement ne se chevauchent pas, et contrôle que les endroits non couverts par un sous-bassin versant d'assainissement ne révèlent pas des erreurs.*

Déroulements alternatifs

- 1.a. *L'ingénieur décèle des ouvrages qui ne sont pas connectés au réseau : il les connecte, si besoin est, aux ouvrages du réseau adéquat.*
- 2.a. *L'ingénieur décèle des canalisations dont le sens d'écoulement est faux : il rétablit le sens d'écoulement.*
- 3.a. *L'ingénieur identifie des erreurs ou des omissions dans les données attributaires des ouvrages du réseau d'assainissement : il corrige ces erreurs et omissions.*
- 4.a. *L'ingénieur constate une erreur dans une cote : il corrige l'erreur.*
- 5.a. *L'ingénieur constate que deux sous-bassins versants d'assainissement se chevauchent : il modifie les limites des sous-bassins versants d'assainissement.*
- 5.b. *L'ingénieur constate qu'une zone qui devrait être couverte par un bassin versant d'assainissement ne l'est pas. Il délimite donc un nouveau sous-bassin versant d'assainissement.*

Cas d'utilisation :	Créer un [ouvrage de réseau d'assainissement]
Commentaire :	Le nom "Ouvrage de réseau d'assainissement" (SewersystemStructure) est un raccourci recouvrant les concepts suivants : "Chambre normée" (Manhole), "Ouvrage spécial" (Special structure), "Exutoire" (Outlet), "Installation d'infiltration" (Infiltration device), "Canalisation" (Pipe).
Numéro :	CE1
Date :	19.01.2000
Version :	1.1
Statut :	En révision
Niveau :	Système, développé
Objectif :	Introduire dans le système d'information une référence et des informations relatives à un nouvel ouvrage de réseau d'assainissement.
Acteur principal :	Employé chargé de la mise à jour des données (Opérateur)
Acteurs secondaires :	
Pré-condition :	L'employé chargé de la mise à jour des données est connecté au système.

Résultat en cas de succès :	L'information relative à l'ouvrage de réseau d'assainissement est capturée et enregistrée dans le système (y compris en ce qui concerne les composants de l'ouvrage d'assainissement). L'opérateur chargé de la mise à jour des données est averti du succès de l'opération.
Résultat en cas d'échec :	Aucune information relative à l'ouvrage de réseau d'assainissement n'est enregistrée dans le système. L'opérateur est averti de l'échec de l'opération.
Déclencheur :	L'opérateur chargé de la mise à jour des données déclare vouloir introduire des informations relatives à un nouvel ouvrage de réseau d'assainissement.

Scénario principal

1. *L'opérateur introduit dans le système l'information relative à l'ouvrage de réseau d'assainissement. Eventuellement, il ajoute un noeud de réseau (il doit avoir la possibilité de spécifier quels sont les tronçons amont et aval du noeud), ajoute un tronçon de réseau (il doit pouvoir spécifier quels sont les noeuds amont et aval du tronçon, ajoute un composant d'ouvrage (couverture, accès, ...), ajoute un composant hydraulique (pompe, valve, déversoir,...), définit le propriétaire et/ou le gestionnaire de l'ouvrage de réseau d'assainissement (ces différents cas d'utilisation peuvent être répétés plusieurs fois). L'opérateur doit pouvoir revenir en arrière s'il s'aperçoit qu'il a commis une erreur, et donc pouvoir supprimer un noeud de réseau, supprimer un tronçon de réseau, supprimer un composant hydraulique, enlever le propriétaire et/ou le gestionnaire de l'ouvrage de réseau d'assainissement. L'opérateur confirme vouloir enregistrer les données introduites dans le système.*
2. *Le système confirme l'enregistrement des données relatives à l'ouvrage d'assainissement.*

Déroulements alternatifs

- 1.a. *L'opérateur chargé de la mise à jour des données veut supprimer le noeud de réseau d'un ouvrage de type "Chambre standard", "Ouvrage spécial", "Exutoire", "Installation d'infiltration" qui ne possède qu'un noeud de réseau : le système n'accepte pas la demande et signale à l'opérateur que de tels ouvrages doivent comporter au moins un noeud.*
- 1.b. *L'opérateur veut supprimer le tronçon de réseau d'un ouvrage de type "Canalisation" qui n'en possède qu'un : le système n'accepte pas la demande et signale à l'opérateur qu'une canalisation doit comporter au moins un tronçon.*
- 1.c. *Le système détecte une ambiguïté quant à la position du composant que l'on veut ajouter à l'ouvrage de réseau d'assainissement :*
 - 1.c.1. *Il demande à l'opérateur sur quel noeud de réseau il faut placer le composant hydraulique.*
 - 1.c.2. *L'opérateur communique au système le noeud sur lequel placer le composant hydraulique.*
- 2.a. *L'information relative au composant ne respecte pas les règles de gestion relatives au composant : le système informe l'opérateur de la non-validité des données entrées et n'effectue pas l'opération d'enregistrement demandée.*
- 2.b. *L'enregistrement est impossible : le système informe l'employé du problème rencontré et renonce à effectuer l'opération d'enregistrement demandée.*

Cas d'utilisation :

Supprimer un [ouvrage de réseau d'assainissement]

Commentaire : Le nom "Ouvrage de réseau d'assainissement" (SewersystemStructure) est un raccourci recouvrant les concepts suivants : "Chambre normée" (Manhole), "Ouvrage spécial" (Special structure), "Exutoire" (Outlet), "Installation d'infiltration" (Infiltration device), "Canalisation" (Pipe).

Numéro : CE3

Date : 19.01.2000

Version :	1.0
Statut :	En révision
Niveau :	Système, développé
Objectif :	Effacer un ouvrage de réseau d'assainissement du système.
Acteur principal :	Employé chargé de la mise à jour des données (Opérateur)
Acteurs concernés :	
Pré-condition :	L'opérateur est connecté.
Résultat en cas de succès :	Toute information relative à l'ouvrage sélectionné est effacée du système (et en particulier les éventuels noeuds, tronçons, éléments d'ouvrage et composants hydrauliques appartenant à l'ouvrage sélectionné). L'opérateur chargé de la mise à jour des données est averti du succès de l'opération.
Résultat en cas d'échec :	L'ouvrage de réseau d'assainissement n'est pas effacé du système. L'ouvrage de réseau d'assainissement reste inchangé. L'opérateur chargé de la mise à jour de la base de données est averti de l'échec de l'opération.
Déclencheur :	L'opérateur chargé de la mise à jour des données déclare vouloir supprimer un ouvrage de réseau d'assainissement.

Scénario principal

1. *L'opérateur fixe la valeur des critères de recherche pour un ouvrage de réseau d'assainissement.*
2. *Le système retourne le ou les ouvrages de réseau d'assainissement correspondant au(x) critère(s) spécifié(s).*
3. *L'employé chargé de la mise à jour sélectionne l'ouvrage qu'il souhaite effacer et confirme vouloir l'effacer.*
4. *Le système confirme l'effacement de l'ouvrage sélectionné.*

Déroulements alternatifs

- 2.a. *Le système ne trouve pas d'ouvrage sur la base du ou des critère(s) spécifié(s) : il en informe l'utilisateur.*
- 4.b. *L'effacement est impossible : le système informe l'employé du problème rencontré et renonce à effectuer l'opération d'effacement demandée.*

MOTTIER Vincent

20, ch. Michée-Chauderon
CH-1203 Genève

Tél. prof. : +41 21 693 37 33
E-mail : vincent.mottier@epfl.ch

Né le 28 juin 1967
Originaire de Gy (GE)
Nationalités suisse
et française
Marié, un enfant

EXPERIENCE PROFESSIONNELLE

- Dès 1996 Ecole Polytechnique Fédérale de Lausanne, Département de génie rural, Institut d'aménagement des terres et des eaux. En qualité d'assistant-doctorant.
Développement d'une approche par composants logiciels des outils informatiques d'aide à la gestion intégrée des eaux, mise sur pied d'une procédure de réalisation de tels composants. Application au domaine de l'assainissement. Encadrement d'étudiants effectuant des travaux de séminaire ou de diplôme.
- 1994 - 1996 Institut Fédéral pour l'Aménagement, l'Epuración et la Protection des Eaux (EAWAG), Duebendorf (ZH). En qualité de collaborateur scientifique.
Participation à différents projets dans le domaine de l'infiltration artificielle des eaux pluviales : qualité des eaux de ruissellement, transport de composants chimiques dans les sols, dimensionnement d'installations d'infiltration. Réalisation d'une étude sur l'impact de l'infiltration artificielle sur le cycle hydrologique à l'échelle régionale.
- 1992 EAWAG.
Stage d'une durée de cinq mois dans le cadre de projets de recherche en hydrologie urbaine : expertise d'une installation de traitement des eaux, recherche bibliographique sur la composition chimique des eaux de ruissellement.
- 1985 - 1986 Disque Office S.A., importateur, Fribourg.
Durant 8 mois en qualité de chauffeur-livreur et collaborateur au stock.

FORMATION ET DIPLOMES

- 1997 - 2000 Certificat d'études supérieures en gestion de l'entreprise, Ecole des Hautes Etudes Commerciales, Université de Lausanne.
- 1992 - 1993 Diplôme d'Etudes Approfondies (DEA) en hydrologie, Université des Sciences et Techniques du Languedoc, Montpellier, France.
Travail pratique (sept mois): étude du fonctionnement et des performances d'une station d'épuration des eaux usées urbaines par infiltration percolation à Huelva (Espagne).
- 1986 - 1992 Diplôme d'ingénieur du génie rural, spécialisation environnement, Ecole Polytechnique Fédérale de Lausanne
Travail pratique (deux mois): étude de la biodégradabilité de deux composés chimiques en vue de la mise au point d'un système d'épuration d'effluents industriels.
- 1981 - 1985 Certificat de maturité de type C, Collège St-Michel, Fribourg, Suisse.

LANGUES

- Français : langue maternelle.
- Anglais : bonnes connaissances. Cambridge Certificate in Advanced English en 1996.
- Allemand : excellentes connaissances. Séjour professionnel de trois ans environ à Zurich.
- Espagnol : bonnes notions orales. Séjour professionnel de six mois à Huelva (Espagne).

INFORMATIQUE

- Méthodes de développement : MERISE, FUSION et RUP.
- CASE : Sybase PowerAMC et Rational Rose.
- SIG : MapInfo Professional, IDRISI, composant MapInfo MapX.
- Langages : VB 6, Pascal, Java, MapBasic, SQL.
- Bureautique : MS Powerpoint, Word, Access et Excel, Adobe FrameMaker.

PUBLICATIONS PRINCIPALES

- Seigneur C., Mottier V., Pulgarin P., Adler N., Péringer P., 1993, "*Biodegradation of xenobiotics in a fixed bed reactor*", Environmental Progress, Vol. 12, N°4, pp 306-311
- Boller M., Schwager A., Eugster J., Mottier V., 1993, "*Dynamic behaviour of intermittent buried filters*", Water Science and Technology, Vol. 28, N° 10, pp. 99-107
- Nieto P., Brissaud F., Mottier V., 1994, "*Depuración natural de ARU mediante su infiltración controlada. Un paso más.*", Boletín Geológico y minero, Vol. 105-3, pp 50-55
- Boller M., Mottier V., 1998, "*Wasserwirtschaftliche Bedeutung der Regenwasserversickerung am Beispiel einer Region*", Z. f. Kulturtechnik und Landentwicklung 39, pp 247-254
- Mottier V., Brissaud F., Nieto P., Alamy Z., 2000, "*Wastewater by infiltration percolation : a case study*", Water Science and Technology, Vol. 41, N° 1, pp. 77-84
- Zobrist J., Müller S.R., Ammann A., Bucheli T.D., Kobler D., Mottier V., Ochs M., Schoenenberger R., Eugster J., Boller M., 2000, "*Quality of roof runoff for groundwater infiltration*", Water Research, Vol. 34, N° 5, pp. 1455-1462
- Riedo M., Mottier V., à paraître, "*Langages de développement*", in "Traité IGAT : Information Géographique et Aménagement du Territoire. Interfaces homme/machine et langage", Hermès, Paris

ACTES DE CONFERENCES

- Mottier V., Bucheli T., Kobler D. et al, 1995, "*Qualitative aspects of roof runoff*", Proceedings of the 8th Junior European Workshop, 22-25 september, Deventer, the Netherlands
- Mottier V., Boller M., 1996, "*Bemessung und Leistung von Versickerungsanlagen*", VSA-Fortbildungskurs 1996 : "Standortbestimmung zum Vollzug von Art. 7 des Gewässerschutzgesetzes. Beseitigung von verschmutztem und nicht verschmutztem Abwasser", Engelberg
- Mottier V., Boller M., 1996, "*Quantitative und qualitative Aspekte des Dachwassers*", VSA-Fortbildungskurs 1996 : "Standortbestimmung zum Vollzug von Art. 7 des Gewässerschutzgesetzes. Beseitigung von verschmutztem und nicht verschmutztem Abwasser", Engelberg
- Rossi L., Mottier V., Baumgartner J., 1999, "*Nouvelles technologies au service de la gestion des eaux*", Actes, Congrès Infra '99, 22-24 novembre, Montréal, QC
- Mottier V., 2000, "*Vers une gestion dynamique des PGEE. Apport des technologies de l'information dans le contexte de l'assainissement des agglomérations*", Bulletin de l'ARPEA n°206, octobre 2000, pp. 13-15
- Rossi L., Mottier V., 2000, "*Conception de systèmes informatisés pour la gestion de l'assainissement. Exemples d'application en Suisse et au Canada*", 2ème conf. internationale sur l'Aide à la Décision dans le domaine Génie Civil et Urbain, 20-22 nov. 2000, Lyon, pp. 41-50

DIVERS

- Membre de la commission pour la structuration des données en hydrologie urbaine de l'Association suisse des professionnels de la protection des eaux (ASPEE, Zurich).