Clés d'accès à UML

Chantal Morley

Introduction : Maître d'ouvrage cherche méthode pour exprimer ses besoins !

Le rôle de la maîtrise d'ouvrage va croissant depuis quelques années et exige des connaissances et un savoir-faire spécifiques. L'art de produire un cahier des charges porte même un nom, l'ingénierie des besoins (*requirements engineering*). Il y a vingt ans, la méthode Merise était parmi les premières à prendre en compte le point de vue des gestionnaires, en proposant des modèles pour représenter les aspects conceptuels (le sens des informations, la structuration du système de gestion) et organisationnels (choix d'informatisation, choix de répartition du travail…). Au début des années 90, les méthodes objets ont cherché à s'implanter. Succès mitigé : après avoir sérieusement investi sur une méthodologie, les entreprises freinent tout changement qui n'est pas solidement justifié. La donne commence à changer et UML (*Unified Modeling Language*, Langage de modélisation unifié) se présente aujourd'hui comme une alternative sérieuse.

1 Un peu d'histoire pour mieux comprendre

UML est le résultat de la fusion de trois de méthodes d'analyse orientées objet : OOD, OMT et OOSE.

La méthode OOD, *Object Oriented Design*, de G.Booch a été conçue à la demande du Ministère de la Défense des USA. L'objectif était de préparer de façon rigoureuse la structuration des programmes écrits en langage ADA ou C++.

La méthode OMT, *Object Modeling Technique*, a été mise au point à General Electric. Ses auteurs ont puisé leur inspiration d'une part dans les langages à objets pour des applications d'informatique industrielle (automates, contrôle de processus...), d'autre part dans les techniques de modélisation conceptuelle des méthodes d'analyse des années 80. OMT représente un système comme un assemblage d'éléments auxquels on attache des comportements, c'est-à-dire des opérations pouvant être déclenchées à la réception d'un message envoyé par d'autres composants.

La méthode OOSE, *Object Oriented Software Engineering*, est d'origine universitaire (informatique temps réel) et industrielle (Ericsson). Son originalité consiste à faire reposer l'analyse sur une expression par l'utilisateur de la façon dont il pense utiliser le futur système.

Devant l'attentisme du marché face aux méthodes et aux AGL objets, la société Rational Software a réuni les auteurs principaux de ces trois méthodes pour qu'ils se mettent d'accord sur un langage de modélisation dans l'espoir qu'il devienne une référence. Sa réussite fut d'être retenu comme norme de modélisation par l'OMG[1], après avoir reçu le soutien de plusieurs grands constructeurs informatiques et éditeurs de logiciels[2]. Ce langage a passé par différents stades et est encore en évolution (Fig. 1).

---

[1] *Object Management Group*, organisme de normalisation pour le monde objet : www.omg.com

[2] Notamment DEC, HP, i-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft Oracle, TI et Unisys. Aucun d'eux n'était jusque-là un acteur public dans le domaine méthodologique.

Date	Stade d'UML	Acteurs	Action
1991		G.Booch J.Rumbaugh	Elaboration de méthodes orientées objet
1994		Rational Software	Rapprochement de G.Booch et J.Rumbaugh
1995	Méthode unifiée 0.8	G.Booch J.Rumbaugh *rassemblés par Rational Software*	
1996	UML 0.9	G.Booch J.Rumbaugh I.Jacobson *rassemblés par Rational Software*	
Fin 1996		OMG	Appel d'offres pour une méthodologie
Début 1997	UML 1.0	G.Booch J.Rumbaugh I.Jacobson *rassemblés par Rational Software* ainsi que d'autres partenaires (Digital, HP, Microsoft…)	Soumission à l'OMG *Remarque : l'OMG a reçu une offre concurrente d'UML*
Fin 1997	UML 1.1	Les auteurs d'UML 1.0 et ceux de l'offre concurrente (dont Softeam)	Rapprochement et standardisation de la nouvelle version par l'OMG
Mars 1999	UML 1.3	Task Force constituée par l'OMG	Amélioration de la version 1.1
En cours	UML 2.0	Revision Task Force, constituée par l'OMG	Amélioration de la version 1.3

Figure 1 : Les stades de constitution d'UML

2 UML et la maîtrise d'ouvrage

A priori, UML n'est destiné pas à la maîtrise d'ouvrage. Plusieurs raisons conduisent toutefois à préconiser son utilisation pour définir un système d'information et élaborer le cahier des charges correspondant.

La première est sa normalisation par l'OMG. L'histoire des méthodes a montré que la profusion des notations est préjudiciable aux entreprises clientes comme à leurs fournisseurs. Toute norme doit donc être considérée avec le plus grand sérieux.

La seconde raison est l'intérêt montré par les informaticiens pour ce langage de modélisation. Même si l'utilisation qu'ils en font est tournée vers la description de composants logiciels, il est intéressant, pour faciliter le dialogue entre maître d'ouvrage et maître d'œuvre, de disposer de modèles communs.

La troisième raison est la possibilité d'utiliser le même atelier de génie logiciel, depuis l'expression des besoins jusqu'à la génération de tout ou partie du code.

La dernière raison, mais non la moindre, est de s'appuyer sur des principes et concepts objets pour enrichir la démarche d'analyse du système d'information. On en attend des améliorations quant à la richesse, la modularité, la cohérence et la rigueur du système ainsi spécifié.

3 Que trouve-t-on dans UML ?

UML est un ensemble de modèles. L'emploi du terme «langage» signifie que l'accord a porté sur les modèles, ainsi que sur leur représentation sous forme de diagrammes. Cependant, UML ne propose ni mode d'utilisation de ces modèles, ni démarche.

Les concepts de base sont très larges.

Un **objet** est un élément matériel ou immatériel dans la réalité étudiée. Ce peut être aussi bien un acteur, un composant logiciel, un élément matériel, un élément informationnel ou la réalisation d'un processus.

Une **classe** est une abstraction d'un ensemble d'objets sur lesquels on peut reconnaître des similitudes dans le champ de l'étude. Ces similitudes portent sur la façon de les identifier, les types d'états qu'ils peuvent prendre et le rôle qu'ils jouent.

UML comprend sept diagrammes qui peuvent présenter un intérêt pour la maîtrise d'ouvrage.

- Le diagramme d'objets permet de représenter un ensemble d'objets et de mettre en évidence des liens entre ces objets.
- Le diagramme de classes permet de modéliser un ensemble de classes, de même nature ou de natures différentes, qui sont en relation d'une façon ou d'une autre.
- Le diagramme d'états-transitions est associé à une classe : il permet de représenter tous les états possibles d'un objet de la classe, ainsi que les événements provoquant un changement d'état.
- Le diagramme d'activités permet de décrire les traitements en schématisant leur déroulement. Il peut comporter des synchronisations pour représenter les déroulements parallèles. Avec la notion de couloir d'activité, on peut décrire la répartition des responsabilités entre acteurs opérationnels.
- Le diagramme de collaboration permet de mettre en évidence les interactions entre différents objets du système étudié, ainsi que les messages qu'ils s'échangent.
- Le diagramme de séquence permet de visualiser une séquence de messages entre des objets qui collaborent.
- Les cas d'utilisation sont une technique de description du système étudié privilégiant le point de vue de l'utilisateur. Un cas d'utilisation est une façon spécifique d'utiliser le système.

Devant cet ensemble de modèles, que doit faire l'analyste chargé d'exprimer les besoins sous la forme d'un cahier des charges précis ? Faut-il utiliser tous les diagrammes ? Dans quel ordre ? Pour représenter quels objets et quelles classes ? Autant de questions qu'il est légitime de se poser.

UML n'apportant pas de réponse, le maître d'ouvrage ne peut pas pour autant se transformer en méthodologue chevronné ! Pour pouvoir utiliser UML efficacement, il doit donc disposer de principes qui guident sa pensée et d'une démarche qui oriente son travail.

3 Des principes de base

Tout est classe, oui mais… il n'est pas utile d'avoir de la réalité une vision indifférenciée. Au contraire, des typologies qui ordonnent le réel perçu peuvent s'avérer des plus profitables. Ainsi, nous proposons de distinguer dans l'ensemble des classes trois sous-ensembles disjoints :

- *Entité*
  Les classes entités permettent de modéliser toutes les informations que l'on veut gérer.
- *Processus*
  Les classes processus permettent de répertorier les réponses organisées pour accomplir les missions.
- *Acteur*
  Les classes acteurs représentent les rôles attribués.

Les entités

Une entité est un concept global d'information, qui peut être décrit par une structure regroupant les types d'informations élémentaires nécessaires à sa gestion.

De plus, il est utile de distinguer trois sortes d'entité, selon le rôle joué dans un système d'information. Cette distinction apporte une aide à l'analyste et peut économiser des efforts de modélisation inutiles. La typologie est la suivante (Fig. 2) :

- Une *entité de gestion* est une entité pour laquelle on a choisi de gérer une transformation. Elle passera donc par des états de gestion successifs.
- Une *entité de référence* est le support d'informations stables, sur lesquelles on s'appuie pour effectuer les activités opérationnelles.
- Une *entité de reporting* porte des informations, souvent calculées ou agrégées, qui apportent une aide pour le pilotage des activités à court ou moyen terme.

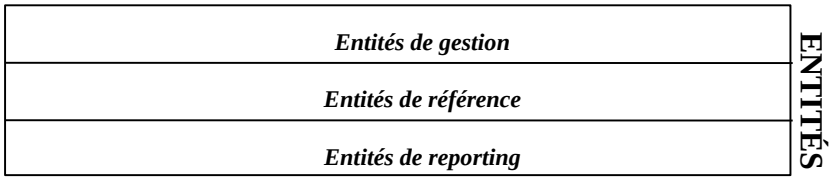

Figure 2 : La typologie des entités

Les processus

Un processus est un ensemble finalisé d'activités effectuées par des acteurs et mettant en jeu des entités. Il est généralement transverse à plusieurs fonctions dans l'entreprise.

Quand on fait l'analyse d'un système d'information, il est utile de distinguer trois types de processus.

1. Le *processus métier* a pour but d'accomplir une mission du domaine. Des acteurs externes au domaine ont une visibilité sur ce type de processus.
2. Le *processus support* n'est pas au cœur du métier : l'accomplissement de son but n'est pas la mission du domaine. Mais le résultat produit est nécessaire pour les processus métier.
3. Le *processus de pilotage* a pour but est d'organiser les activités de pilotage à l'intérieur d'un domaine.

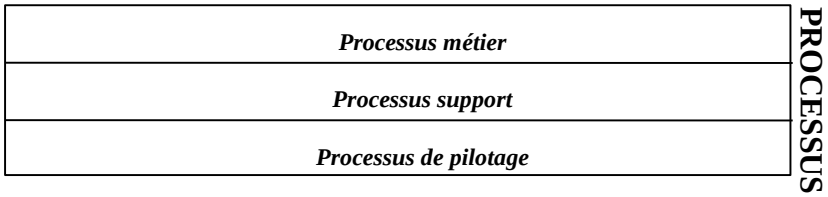

Figure 3 : La typologie des processus

4 Un mode d'utilisation

La modélisation d'un système d'information avec UML s'appuie sur plusieurs types de diagrammes, qui rendent compte chacun d'un point de vue particulier. Nous proposons l'utilisation suivante.

Le *diagramme de classes* expose la structuration des entités et des opérations qu'elles portent. Ce diagramme est également utilisé pour représenter les acteurs impliqués.

Le *diagramme de collaboration* montre les interfaces entre le futur système d'information et les autres domaines.

Le *diagramme de cas d'utilisation* montre l'ensemble des processus du domaine d'étude. Chaque processus, ou plus précisément chaque variante de processus, sera modélisé au moyen d'un diagramme d'états-transitions et/ou d'un diagramme de séquence et/ou d'un diagramme d'activités.

Le *diagramme d'états-transitions* a une double utilisation. D'une part, il met en évidence les différents états d'une entité de gestion. D'autre part, il permet de représenter les différentes étapes d'un processus métier, ainsi que les événements ou décisions faisant passer d'une étape à l'autre.

Le *diagramme de séquence* fait apparaître les échanges des acteurs entre eux ou avec le système informatique, ainsi que leur ordonnancement dans le temps. On peut visualiser le déroulement temporel d'un scénario d'un processus métier, ainsi que les points d'utilisation du système informatique. Ces derniers se retrouveront sur le diagramme d'activités, dans des activités mettant en jeu une entité.

Le *diagramme d'activités* représente la répartition des rôles entre les acteurs et permet de décrire les activités d'un processus métier ou d'un processus support. Une activité peut mettre en jeu une ou plusieurs entités. Dans ce cas, elle fait appel à une opération de l'entité. Si l'entité fait partie du domaine d'étude, l'opération doit se retrouver sur le diagramme d'états-transitions de cette entité. Si l'entité ne fait pas partie du domaine d'étude, cela signifie que l'activité fait appel à un service d'un autre domaine. On doit retrouver cet appel sur le diagramme de collaboration entre domaines.

5 Une démarche pour s'orienter

Le schéma général d'une démarche d'élaboration d'un cahier des charges est représenté à la figure 4. Les trois premières étapes permettent de porter un diagnostic sur le système existant et de poser des principes de reconfiguration du système d'information. Les deux dernières étapes représentent la plus grande partie du travail.

Pour chaque étape, on s'appuie sur certains des diagrammes UML pour représenter les objets et les classes.

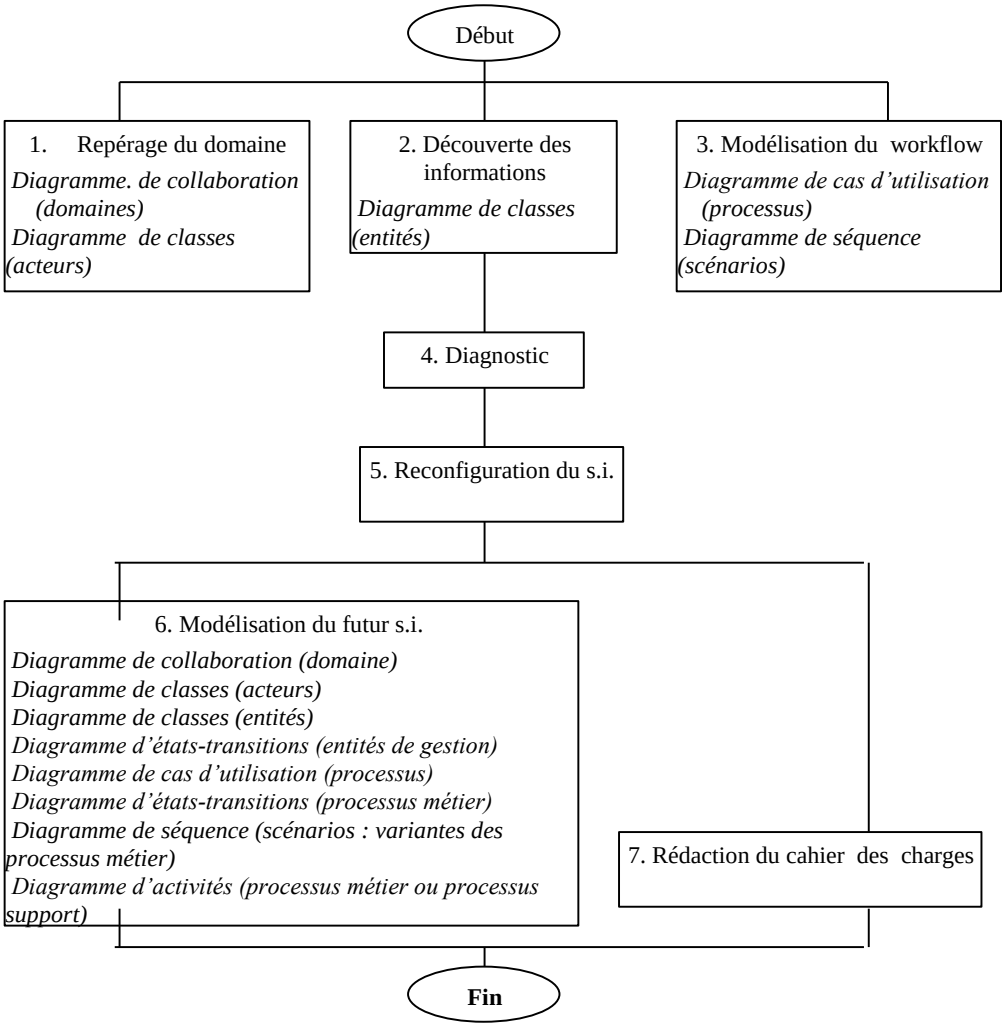

Figure 4 : La démarche d'élaboration du cahier des charges

Quelques recommandations pour finir :

- Éviter la sophistication : toutes les possibilités du langage ne sont pas utiles au maître d'ouvrage.
- Capitaliser sur des concepts éprouvés, en particulier ceux qui touchent à l'information et à la communication : entité, processus, workflow…
- Bannir la redondance dans la modélisation : l'utilisation d'un diagramme doit apporter des précisions et/ou permettre de réfléchir sur les choix à faire.
- Rester souple dans la démarche pour prendre en compte des particularités du projet, mais s'appuyer sur un cadre rigoureux.

Bibliographie :

Booch G., Rumbaugh J. et Jacobson I., *The Unified Modeling Language User Guide*, Addison Wesley, 1998.

Ingénierie des systèmes d'information, numéro spécial «L'ingénierie des besoins», Vol.4, n°6, 1994.

Kettani N., Mignet D., Paré P. et Rosenthal-Sabroux C., *De Merise à UML*, Eyrolles, 1998.

Lopez N., Migueis J. et Pichon E., *Intégrer UML dans vos projets*, Eyrolles, 1998.

Morley C., Hugues J. et Leblanc B., *UML pour l'analyse d'un système d'information -le cahier des charges du maître d'ouvrage*, Dunod, 2000.

Site Internet de Rational Corporation www.rational.com