

Projet ENRC-DVD MIAGE, contexte

Alain VAILLY
Département Informatique - UFR de Sciences et Techniques
Université de Nantes

I) Avant-propos

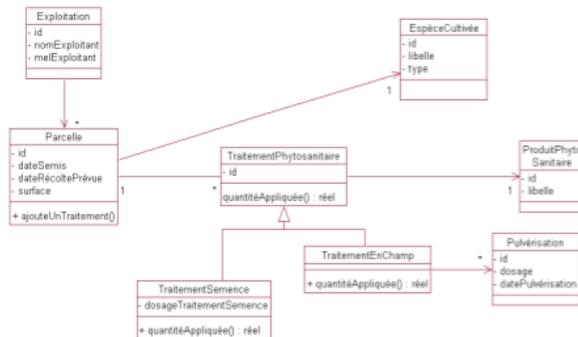
Avant de présenter le contexte dans lequel se situe ce module, plantons le décor. Pour cela, nous avons fait appel à des documents officiels extraits de sujets d'examen et de programmes tout aussi officiels.

a) extrait du cas Chambre de Commerce du Pas-de-Calais, sujet du Brevet de Technicien Supérieur en Informatique de Gestion, option Développeur d'applications, session 2008 :

« Proposer un schéma entité-association représentant les informations nécessaires... » p.2/16

« L'application permettra également de visualiser les informations saisies... ; le diagramme partiel des classes métier est présenté en annexe 4... » p.6/16

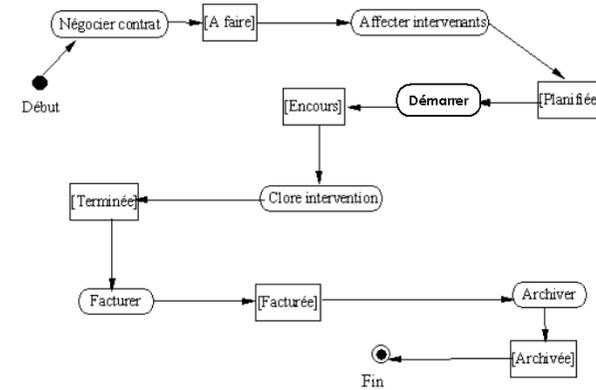
« Annexe 4 : Diagramme partiel des classes métiers » p.12/16



b) extrait du cas SYNAPSINFO, sujet du Brevet de Technicien Supérieur en Informatique de Gestion, option Développeur d'applications, session 2007 :

« Concevoir le schéma entité-association correspondant au domaine de gestion des prestations... » p.4/16

« Une partie du schéma représentant le cycle de vie d'une intervention a déjà été réalisée. Elle est présentée ci-dessous :



« À partir de l'énoncé et en vous aidant de l'annexe 1 qui précise le formalisme utilisé, reproduire et compléter le schéma fourni ci-dessus représentant le cycle de vie d'une intervention. » p.5/16

« Les annexes 3, 4 et 5 présentent les classes métiers et techniques utilisées. » p.10/16

c) extraits de l'arrêté définissant le Brevet de Technicien Supérieur en Informatique de Gestion, précisant le contenu de la formation et les compétences attendues :

« S32 : Analyse et conception de systèmes logiciels :

Méthodes et outils
Niveaux d'abstraction
Méthodes de conception
Modélisation des communications
Modélisation des données
Dictionnaire des données
Modèle entité-association
Extensions...
Modèle relationnel
[...]

« Contexte général de réalisation

C2- Concevoir et modéliser

C22 Représenter schématiquement les données d'un système d'information

Les schémas et algorithmes, à réaliser ou à interpréter, font référence aux modèles utilisés dans la profession (normes ou standards de fait) »

« C34 Développer tout ou partie d'une application à l'aide d'un langage de programmation à objets. »

« C35 Développer tout ou partie d'une application autour d'une base de données relationnelles. »

Nous pourrions multiplier les extraits. Ceux que nous avons présentés ci-dessus sont largement suffisants pour illustrer le contexte dans lequel sont plongés les étudiants des sections de techniciens supérieurs. Nous résumerons ce contexte de la façon suivante :

Une modélisation des données ; très peu (voire même pas du tout) de modélisation des traitements et du comportement ; une schématisation dichotomique (nous n'avons pas dit schizophrène...) à l'aide d'entités-associations et de classes ; une liberté –apparente– des formateurs pour le choix des notations étudiées en cours ; un classicisme des sujets d'examen (ils sont tous sur le même modèle) agissant comme un vrai patron de conception –et de formattage– très contraignant.

II) Analyse du point de départ

Les utilisateurs de notre module sont, tous, issus de la même formation ou d'une formation similaire. Nous pouvons caractériser celle-ci par :

- une pratique de la modélisation de données axée, pour l'essentiel, sur le modèle entités-associations-propriétés étendu –c'est une question récurrente à l'examen– complétée, si l'enseignant le souhaite, par une étude du modèle Objets (et plus précisément du diagramme de classes de type UML).
- une bonne maîtrise des notions sous-jacentes au modèle relationnel.
- une quasi-absence de tout autre modèle, qu'il s'agisse des modèles de traitements de Merise ou bien des diagrammes UML les plus usuels (scénarios, séquences, états-transitions).

Nous faisons donc le « pari » que nos étudiants (ceux à qui le DVD est destiné) ont tous suivi des enseignements de ce type. Pour ceux qui ne seraient pas dans ce cas, nous avons fourni les ressources pédagogiques permettant de comprendre de quoi il est question et notamment :

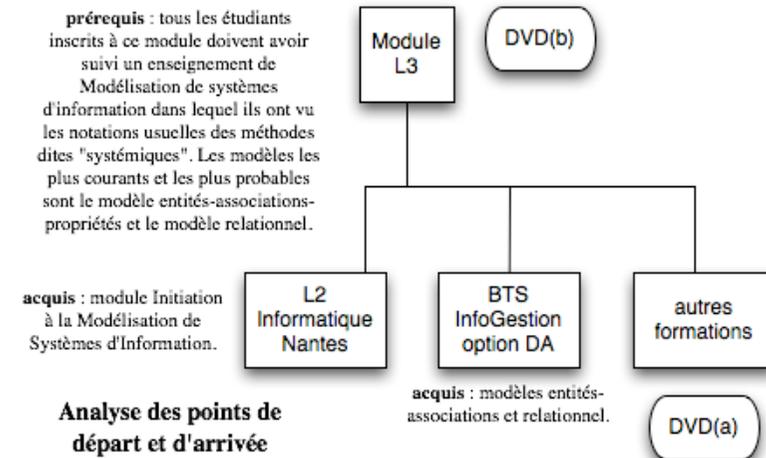
- les diapositives du chapitre *Modèle entités-associations-propriétés*,
- les diapositives du chapitre *Modèle relationnel*.

Ces deux documents sont tirés du module *Initiation à la Modélisation de Systèmes d'information* que suivent tous les étudiants de deuxième année de Licence Informatique.

Globalement, donc, nous supposons que les prérequis du Module de troisième année de Licence Informatique *Outils de modélisation UML* sont satisfaits :

- soit lors de la formation suivie dans l'établissement d'origine,
- soit grâce au DVD et à la fourniture des éléments de cours de deuxième année. Ceux-ci sont matérialisés sur la figure ci-après sous l'appellation DVD (b).

NB : ces éléments de cours sont dorénavant supposés **connus** et **maîtrisés**.



Si nous avons la conviction que ces prérequis sont effectivement satisfaits pour ce qui concerne les modèles, par contre, nous ne ferons aucune hypothèse pour ce qui est du volet « processus ». Nous avons enseigné deux années de suite en STS, il y a fort longtemps. Nous sommes également en contact avec nos collègues de ces sections. Nous savons donc parfaitement que chacun a une certaine liberté de choix des modèles et des limites qu'il peut assigner à son enseignement... pour autant que le programme de l'examen soit couvert. Nous ne pouvons donc pas avoir la garantie que le processus Merise, par exemple, soit compris et maîtrisé.

Nous avons rencontré des étudiants qui avaient été formés à UML. Nous en avons rencontré d'autres qui n'avaient eu aucun enseignement digne de ce nom sur les aspects traitements, dynamiques, leur professeur ayant axé son programme sur la modélisation des données.

Nous supposons donc que nos étudiants (lorsqu'ils arrivent en L3) ne savaient rien de la modélisation des traitements, de la prise en compte de la dynamique dans un système d'information. Nous supposons également qu'ils ne savent pas comment l'on aborde la création d'un plan logiciel, qu'ils n'ont aucune notion de *Unified Process* (UP), *Processus Unifié*.

III) Analyse du point d'arrivée souhaité

Le module *Outils de modélisation* (c'est le nom qu'il avait dans l'ancienne habilitation) fait partie d'une « chaîne » d'unités d'enseignement traitant de modélisation de systèmes d'information. Un recensement récent fait état de pas moins de vingt-et-un modules dans cette thématique, depuis la deuxième année de Licence Informatique jusqu'au Master deuxième année, en passant par des interventions en Biologie, en Master Sciences de la Terre et de l'Univers...

Il n'est donc pas possible d'ignorer le contexte dans lequel se plonge un module. Il a des exigences en matière de prérequis (voir point précédent), mais il a aussi des devoirs (par rapport aux années suivantes).

Il est le premier maillon d'une chaîne de trois, un par année, qui, autour de la notation unifiée UML, sensibilise les étudiants aux problèmes d'analyse et de conception de systèmes d'information. Le module qui suit immédiatement *Outils de modélisation* est un module de première année de Master Informatique intitulé *Analyse, Conception et Tests* (c'est l'intitulé de l'habilitation actuelle). Ce module est axé Conception. Il suppose que les étudiants maîtrisent les concepts Objets et qu'ils sont capables de lire les principaux diagrammes UML servant dans les niveaux 1 et 2 du processus (ceux qui correspondent à l'étude de l'existant et à l'analyse des besoins).

Nous avons fait le choix de limiter le nombre de diagrammes étudiés (UML est une notation très riche et près de quinze types de diagrammes sont disponibles) et de ne travailler **qu'au niveau Spécification**. Notre choix se justifie par le fait que le volume horaire associé au module (une cinquantaine d'heures) et le niveau des étudiants à l'entrée ne permettent pas d'être plus ambitieux.

Nous préférons travailler davantage en profondeur, sur un petit nombre de diagrammes, et nous focaliser sur la pratique professionnelle. Un diagramme en soi ne sert à rien si l'on ne sait pas dans quel contexte l'utiliser. Moyennant quoi, nous concentrons nos efforts sur les diagrammes suivants : **cas d'utilisation, scénarios, classes, séquences et collaborations, états-transitions et activités**. Nous avons là les diagrammes qui sont utilisés dans les phases dites d'analyse des besoins et d'analyse.

Nous avons aussi décidé de nous arrêter à la **version 1.5** de la notation UML. Il y a, à ce niveau d'évolution, les éléments essentiels des diagrammes que nous avons sélectionnés. Si nos étudiants arrivent à se servir correctement des notations de la version 1.5, ils n'auront aucune difficulté pour passer à la version 1.8 ou à la version 2.0, pas plus qu'ils n'en auront à maîtriser la prochaine évolution. Il est à noter que le module de Master est axé sur la 2.0. Le fait que nous ne voyions pas la toute dernière version n'est donc pas un handicap pour leur insertion professionnelle ultérieure. Ce qu'ils ne voient pas en Licence, ils le verront en Master.

IV) Quelques éléments entre points de départ et d'arrivée

La présentation des niveaux des étudiants à l'entrée et des objectifs visés par le cours en L3 a permis de mettre en évidence le fait qu'il y a peu de différences entre les prérequis souhaités et le niveau des étudiants en entrée.

Nous avons, rappelons-le, deux objectifs au moment où nous rédigeons les éléments constituant le module :

- le premier est de combler le trou existant entre entrants théoriques (les prérequis souhaités) et entrants réels (le niveau des étudiants en entrée) ;
- le second est de démystifier le cours de L3, notre message, pour l'essentiel, pouvant se résumer par la phrase suivante : « un module en fac, c'est pas plus difficile qu'un module en STS ou en IUT ».

Quoi mettre, alors, dans ce module ? Pour répondre à cette question, nous devons, au préalable, nous rappeler le titre de la thématique dans laquelle s'inscrit ce module : « passerelle dans et vers l'enseignement supérieur ». Nous avons donc décidé de fournir aux étudiants entrants des conseils, des éléments qui ne figurent pas dans le programme du module de L3, le contenu exact du cours de L3 (ce qui permet aux étudiants d'en disposer avant les autres et donc, quelque part, facilite le passage vers...).

Nous leur fournissons également une série d'exercices corrigés, les solutions de ces exercices étant rédigées de façon –tel est du moins notre souhait– autosuffisantes. Un

lecteur doit être capable de se débrouiller tout seul et, en lisant le texte des corrigés, comprendre ce qu'il fallait faire.

Nous avons aussi enregistré sur le DVD l'outil (StarUML) que nous préconisons, ce qui va permettre aux étudiants d'installer ce logiciel sur leur propre machine. Nous complétons cette partie Outils avec un mode d'emploi. Sachant que les corrigés des exercices sont, pour la plupart, réalisés avec cet outil, il y a là tout ce qu'il faut pour prendre en main très rapidement cet outil et donc disposer d'une aide supplémentaire pour maîtriser les techniques du cours. « C'est en forgeant que l'on devient forgeron », dit un proverbe. Nous pensons, pour notre part, qu'il faut pratiquer pour comprendre. L'outil et la manipulation de l'outil sont des ingrédients indispensables à tout apprentissage d'une méthode d'analyse et de conception en général et d'UML en particulier.

V) Quelques questions qui reviennent souvent (et la réponse à ces questions)

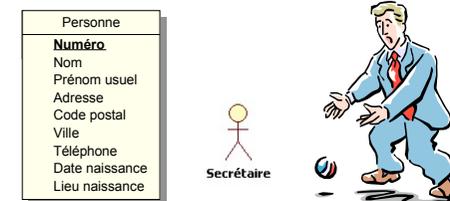
Nous regroupons ici quelques-unes des questions qui reviennent le plus souvent lors de nos discussions avec les étudiants :

- Modéliser, c'est quoi ?

Un modèle est une représentation souvent abstraite de quelque chose, faite dans un but précis par une personne donnée. Le modèle de A est la vision qu'a eu X à un moment donné lorsqu'il regardait B avec un objectif...

Un modèle est une représentation d'une réalité perçue par quelqu'un. Cette représentation est faite à l'aide d'une notation particulière, plus ou moins précise. Elle déforme plus ou moins.

Prenons quelques exemples simples (voire même simplistes) :



Toutes ces images sont des modèles de personnes. Ces modèles décrivent tous des personnes, à l'aide de notations, de formalismes plus ou moins précis, plus ou moins expressifs. Ils ont été faits dans un but précis, par des dessinateurs qui avaient un objectif tout aussi précis.

L'un des plus gros défis qu'auront à affronter les étudiants est celui de la modélisation. Il faut savoir modéliser et savoir utiliser des modèles pour raisonner. Lorsque vous discutez avec votre garagiste à propos de votre voiture, vous employez des termes correspondant à des pièces de la machine que vous n'avez (peut-être) jamais vues. Cela ne vous empêche pas d'en parler. Si le garagiste est pédagogue et s'il veut bien vous expliquer ce qui provoque la panne dont souffre votre véhicule, il se peut qu'il fasse un petit croquis. Ce croquis, les mots que vous employez, les photos de moteurs dans la notice de votre voiture, tout cela constitue des modèles. Les avoir et en discuter vous permet de comprendre le problème, de le résoudre.

Il faut faire la même chose lorsque l'on conçoit un logiciel quelqu'il soit.

- Par quoi on commence ?

On commence toujours par le début... Par cette réponse « idiote » à une question qui ne l'est pas, loin s'en faut, nous entendons mettre en évidence le fait qu'il ne faut pas mettre la charrue avant les bœufs et conduire l'étude en ayant déjà une idée de la solution logicielle qui sera implantée. Le logiciel que nous devons construire est destiné à un ensemble de personnes. C'est un outil que nous allons leur donner. Il faut donc, avant tout, se mettre à la place des personnes pour qui nous travaillons et essayer de voir comment elles vont se servir de cet outil.

Nous devons donc dans un premier temps tenter de comprendre quels sont leurs besoins. Pour cela, il nous faut discuter avec elles, maîtriser leurs tâches... pour mieux les reproduire à l'aide de l'ordinateur. L'informatique n'est guère mieux qu'un photocopieur. Le travail qui, auparavant, était fait à la main va être fait par une machine. Il ne changera pas fondamentalement de nature.

Une fois le problème bien assimilé (on a analysé l'existant), il faut mettre en évidence les besoins, tenter de les résoudre et peser les conséquences des solutions que nous allons esquisser sans pour autant qu'il nous faille les développer entièrement. « UMLement parlant », nous ferons appel à trois types de diagrammes, cas d'utilisation, scénarios et classes.

Cette première phase terminée, il nous faudra pousser un peu plus l'étude et commencer à construire, à concevoir une solution informatique respectant le cahier des charges défini durant l'étape précédente. La solution doit en effet être conforme à ce qui existe et aux objectifs qui ont été fixés. Chacun des points de ce cahier des charges devra être repris, approfondi en ayant sans cesse en tête deux « choses », le confort de l'utilisateur en priorité et la faisabilité informatique de l'autre. Les scénarios seront transformés en séquences (ou en collaborations), le diagramme de classes enrichi pour finir par préfigurer le futur système d'informations automatisé (la base de données, quoi...). Dans certains cas complexes, on complètera en produisant un diagramme états-transitions ou d'activités.

- En largeur ou en profondeur ?

Les deux ! Il faut travailler en largeur (tout voir) et en profondeur (de façon détaillée). Si l'on ne peut mener les deux de front, il faudra commencer par la largeur (pour couvrir tout l'espace du problème, voir tout ce que doit faire le logiciel), puis ensuite découper la proposition de solution en morceaux et étudier chacun d'entre eux de façon détaillée.

Le processus unifié (*Unified Process*) préconise, ne l'oublions pas, de travailler de façon incrémentale et itérative. Le symbole qui représente le mieux cette façon de travailler est la spirale. Les techniques Objets permettent ce genre d'approche. Il ne faut donc pas s'en priver. C'est en effet comme cela que nous pourrons rapidement livrer aux utilisateurs des briques de logiciel. Mises bout à bout, elles correspondront au logiciel dans sa totalité. Prises individuellement, et livrées individuellement, elles permettront aux clients de vivre la conception et de tester tranquillement ces morceaux un par un.

- A quoi servent ces schémas ?

On prête à Napoléon la paternité de la phrase « un bon croquis vaut mieux qu'un long discours ». Nous ne savons pas si cet empereur en est l'auteur ou non, mais nous partageons cette idée. Les schémas permettent d'exprimer le fonctionnement d'un processus, les données nécessaires... bien mieux qu'un texte.

Le fait de devoir faire des dessins nous permet de voir les choses autrement, de faire une sorte de travail de validation de nos idées en les paraphrasant. Pour ceux des schémas qui sont les plus naturels (nous pensons en particulier aux cas d'utilisation), ils sont suffisamment expressifs pour pouvoir être mis dans les mains de non-informaticiens et leur permettre de visualiser les échanges entre acteurs et systèmes.

Un dessin (même s'il est bon) ne pourra toutefois jamais remplacer complètement un discours (même s'il est long...). Que peut bien signifier, dans le schéma ci-dessous, *Gérer les AG* ? Qu'est-ce qu'une AG ? Le membre intervient de quelle façon dans la gestion des AG ? Autant de questions sans réponse... dans le schéma du moins.



C'est la raison pour laquelle nous recommandons à nos étudiants de faire l'effort de rédiger, de rédiger et de rédiger encore.

Un bon architecte est un bon technicien et aussi et surtout un bon communicant ! Il lui faut concevoir des maisons qui sont vivables et aussi le faire savoir. Il doit expliquer à ses clients à quoi correspondent les plans qu'il leur montre. Ce qu'il est valable pour une maison l'est pour un logiciel.

Les dessins, les schémas participent à ce travail d'explication.

- Puis-je avoir un exemple, SVP ?

Nous en donnons plusieurs. Le premier est fourni avec l'outil StarUML et sert de mode d'emploi à cet outil. Le second est triple. Il s'agit d'une série de trois dossiers de spécification d'un logiciel de gestion de clientèle développé pour une maison de vente par correspondance de timbres de collection. Nous présentons, en effet, un dossier que nous jugeons mauvais (nous l'avons noté 7/20), un dossier correct (il a obtenu une note de 13,5/20) et un dossier excellent (auquel nous avons donné une note de 17,5/20).

Nous pouvons également recommandé la lecture attentive des solutions des devoirs et exercices qui, d'une certaine façon, peuvent aussi servir d'exemples.

VI) En guise de conclusion

Nous avons mis dans ce DVD tout ce qui nous paraît nécessaire. Tous les cours sont fournis, qu'il s'agisse de ceux du cours de L3 ou de ceux qui correspondent aux cours de L2 vus en STS ou en IUT. Une série de quarante exercices permet de faire des gammes sur chacun des chapitres clés du cours. TOUS ces exercices sont corrigés, les solutions étant fournies dans le DVD. Quatre devoirs (avec leurs corrigés) sont également disponibles, permettant ainsi aux apprenants de s'étalonner. Un logiciel gratuit complète ce « package ».

Cela n'est assurément pas suffisant. Possession ne vaut pas compréhension. Nous attendons donc des étudiants qu'ils prennent en main leur formation. Sans travail personnel... important... de LEUR part, cela ne marchera pas. Ils doivent être acteurs de leur propre formation, car c'est bien pour eux qu'ils travaillent. Ils doivent aussi être autonomes, même si cette autonomie pourra être assistée si le besoin s'en fait sentir. Il ne faut en effet pas perdre de vue qu'un tuteur se tient à disposition de l'autre côté d'Internet.

Nous attendons aussi des étudiants un feedback. Ce que nous avons mis à disposition des entrants fait partie des éléments pédagogiques que nous offrons à nos étudiants. Ils sont en principe éprouvés et nous n'avons guère de doute sur leur efficacité. Ce module, dans sa forme ENRC, est toutefois une première. Nous ne savons pas si ce que nous offrons est suffisant pour que les objectifs soient atteints.

Nous comptons donc très fortement sur tous ceux qui liront un jour ce module pour nous faire part de leurs réactions. Qu'ils n'hésitent pas à nous signaler les fautes de frappe (il y en a toujours), les erreurs (on ne sait jamais...), leurs incompréhensions (cela peut être dû à eux, mais aussi peut être à nous). Ce n'est que par ce moyen que la qualité globale du contenu de ce module pourra s'élever et que les défauts seront éliminés.

Par avance, nous les en remercions.
