ENSICAEN
6, bd maréchal Juin
F-14050 Caen cedex 4

Spécialité Informatique - 1re année

Livrable : Etude comparative sur les méthodes de modélisation

Comparatif UML - Merise

Mohamed Yassir Bouhaddaoui
Mohamed Tarek El Allam

2010 - 2011

Table des matières

1.Introduction.....3
2.Systemes d'information.....3
3.Merise….....4
4.UML….....5
5.Comparaison.....6
6.Conclusion…….....9
7.Réferences….....10

1. Introduction

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer.
La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Ce type de méthode est appelé *analyse*. Il existe plusieurs méthodes d'analyse, cette étude fera l'objet d'une comparaison entre le langage UML et la méthode Merise.

2. Systèmes d'information

[Le système d'information doit décrire (on dit encore représenter) le plus fidèlement possible le fonctionnement du système. Pour ce faire, il doit intégrer une base d'information dans laquelle seront mémorisés la description des objets, des règles et des contraintes du système opérant. Cette base étant sujette à des évolutions, le système d'information doit être doté d'un mécanisme (appelé processeur d'information) destiné à piloter et à contrôler ces changements. Le schéma suivant synthétise l'architecture d'un système d'information.

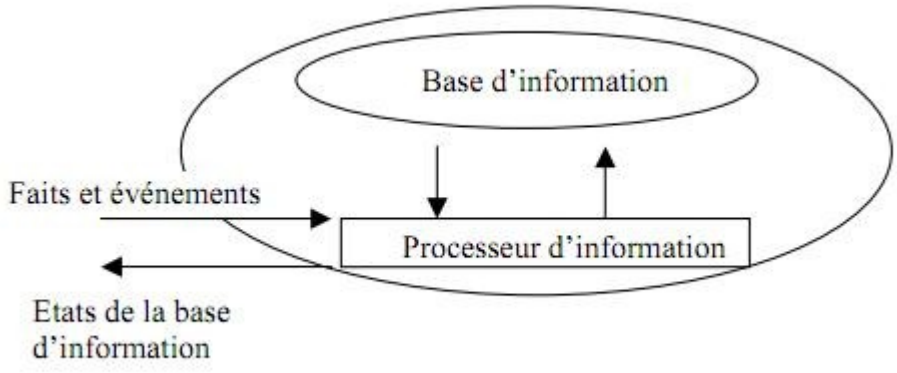

figure 1 : Architecture d'un systeme d'information

Le processeur d'information produit des changements dans la base d'information à la réception d'un message. Un message contient des informations et exprime une commande décrivant l'action à entreprendre dans la base d'information. Le processeur d'information interprète la commande et effectue le changement en respectant les contraintes et les règles.
Si le message exprime une recherche sur le contenu de la base d'information, le processeur interprète la commande et émet un message rendant compte du contenu actuel de la base d'information. Dans tous les cas, l'environnement a besoin de connaître si la commande a été acceptée ou refusée. Le processeur émet, à cet effet, un message vers l'environnement. Relativement à la conception d'un système d'information, l'architecture présentée ci-dessus induit une double conception :

- celle de la base d'information (aspect statique)
- celle du processeur de traitement (aspect dynamique)] [1]

Base de données biométriques

La biométrique est un domaine émergeant où la technologie améliore notre capacité à identifier une personne. La protection des consommateurs contre la fraude ou le vol est un des buts de la biométrie. L'avantage de l'identification biométrique est que chaque individu a ses propres caractéristiques physiques qui ne peuvent être changées, perdues ou volées. La méthode d'identification biométrique peut aussi être utilisée en complément ou remplacement de mots de passe.
Pour réaliser un système d'information capable de gérer ces données biométriques, l'utilisation d'une base de données est donc nécessaire.
Cependant, la conception d'une base de données biométrique n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Ce type de méthode est appelé **analyse**. Il existe plusieurs méthodes d'analyse, notamment le Merise et l'UML.

3. Merise

Pour aider le concepteur dans ces deux tâches, la méthode Merise - méthode d'analyse, de conception et de réalisation de systèmes d'informations informatisés - propose un ensemble de formalismes et de règles destinées à modéliser de manière indépendante les données et les traitements du système d'information. Ces modèles ne sont qu'une base de réflexion pour le concepteur et un moyen de communication entre les divers acteurs du système d'information . Seul la validation de l'ensemble se fera en commun.

[Parmi les informations qui appartiennent au système d'information, certaines doivent ou peuvent faire l'objet d'un traitement automatisé grâce aux outils informatiques. Pour assurer la cohérence du système d'information, la méthode Merise propose une démarche d'informatisation comportant les étapes suivantes :

- **Le schéma directeur** : dont le rôle est de définir, de manière globale, la politique d'organisation et d'automatisation du système d'information. Pour ce faire, il est nécessaire de répertorier l'ensemble des applications informatiques existantes à modifier et à développer. Pour rendre contrôlable et modulable ce développement, il est nécessaire de découper le système d'information en sous-ensembles homogènes et relativement indépendant. Ces sousensembles sont appelés domaines. Par exemple, on peut trouver le domaine « Approvisionnement », le domaine « Personnel ». Les résultats attendus à la fin de cette étape sont une définition précise des domaines, une planification du développement de chaque domaine et un plan détaillé, année par année, des applications qui doivent être réalisées.

- **L'étude préalable par domaine** : qui doit aboutir à une présentation générale du futur système de gestion (modèles des données et des traitements) en indiquant les principales novations par rapport au système actuel, les moyens matériels à mettre en œuvre, les bilans coût – avantage. Cette étude est réalisée en 4 phases :

  - une phase de recueil qui a pour objectif d'analyser l'existant afin de cerner les dysfonctionnements et les obsolescences les plus frappantes du système actuel.

- une phase de conception qui a pour objectif de formaliser et hiérarchiser les orientations nouvelles en fonction des critiques formulées sur le système actuel et d'autre part des politiques et des objectifs de la direction générale. Cela revient à modéliser le futur système avec une vue pertinente de l'ensemble.
- une phase d'organisation dont l'objectif est de définir le système futur au niveau organisationnel: qui fait quoi ?
- une phase d'appréciation dont le rôle est d'établir les coûts et les délais des solutions définies ainsi que d'organiser la mise en œuvre de la réalisation. A cet effet un découpage en projets est effectué.

- L'étude détaillée par projet qui consiste d'une part à affiner les solutions conçues lors de l'étude préalable et d'autre part à rédiger, pour chaque procédure à mettre en œuvre, un dossier de spécifications détaillé décrivant les supports (maquettes d'états ou d'écran) ainsi que les algorithmes associés aux règles de gestion… A l'issue de cette étude, il est possible de définir le cahier des charges utilisateurs qui constitue la base de l'engagement que prend le concepteur vis à vis des utilisateurs. Le fonctionnement détaillé du futur système, du point de vue de l'utilisateur, y est entièrement spécifié.

- La réalisation dont l'objectif est l'obtention des programmes fonctionnant sur un jeu d'essais approuvés par les utilisateurs.

- La mise en œuvre qui se traduit par un changement de responsabilité : l'équipe de réalisation va en effet transférer la responsabilité du produit à l'utilisateur. Cette étape intègre en particulier la formation des utilisateurs. Après une période d'exploitation de quelques mois, la recette définitive de l'application est prononcée.

- La maintenance qui consiste à faire évoluer les applications en fonction des besoins des utilisateurs, de l'environnement et des progrès technologiques.

Le schéma suivant, reprend les étapes décrites ci-dessus.

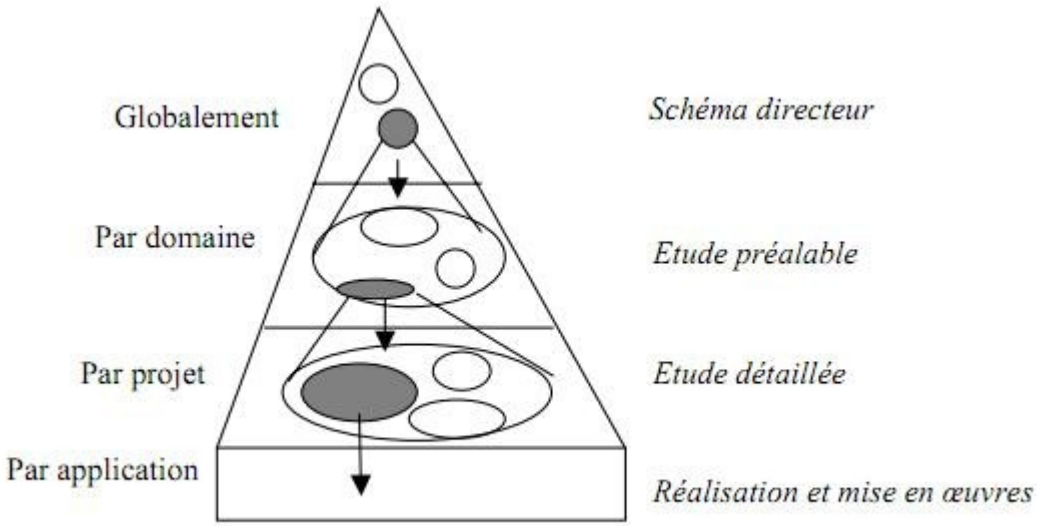

figure 2 : Etapes de modélisation Merise [4]

Cette démarche lourde et parfois complexe est adaptée à l'automatisation de « gros

systèmes d'information ». Pour des informatisations plus modestes, elle peut être perçue comme un carcan, et il convient donc de l'adapter afin de retenir uniquement les concepts et/ou les étapes appropriées aux besoins.] [1]

4. UML

UML (en anlais ***Unified Modeling Language*** ou « langage de modélisation unifié ») est un language de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la «conception orientée objet». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

UML est l'accomplissement de la fusion de précédents langages de modélisation objet : **Booch, OMT**, **OOSE**. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jackobson, UML est à présent un standard défini par l'**Objet Management Group** (OMG). La dernière version diffusée par l'OMG est UML 2.3 depuis mai 2010.

[UML est avant tout **un support de communication performant**, qui facilite la représentation et la compréhension de solutions objet :

- Sa notation graphique permet d'**exprimer visuellement une solution objet**, ce qui **facilite la comparaison et l'évaluation** de solutions.
- L'aspect formel de sa notation, **limite les ambiguïtés** et les incompréhensions.
- Son **indépendance** par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel.

Petit aparté :

> La notation graphique d'UML n'est que le support du langage. La véritable force d'UML, c'est qu'il repose sur un métamodèle.
> En d'autres termes : la puissance et l'intérêt d'UML, c'est qu'il normalise sémantique des concepts qu'il véhicule !
>
> Qu'une association d'héritage entre deux classes soit représentée par une flèche terminée par un triangle ou un cercle, n'a que peu d'importance par rapport au sens que cela donne à votre modèle. La notation graphique est essentiellement guidée par des considérations esthétiques, même si elle a été pensée dans ses moindres détails.
>
> Par contre, utiliser une relation d'héritage, reflète l'intention de donner à votre modèle un sens particulier. Un "bon" langage de modélisation doit permettre à n'importe qui de déchiffrer cette intention de manière non équivoque ! Il est donc primordial de s'accorder sur la sémantique des éléments de modélisation, bien avant de s'intéresser à la manière de les représenter.
> Le métamodèle UML apporte une solution à ce problème fondamental.
>
> UML est donc bien plus qu'un simple outil qui permet de "dessiner" des représentations mentales... Il permet de parler un langage commun, normalisé mais accessible, car visuel. Il représente un juste milieu entre langage

mathématique et naturel, pas trop complexe mais suffisamment rigoureux, car basé sur un métamodèle.

UML comme cadre d'une analyse objet:

Une autre caractéristique importante d'UML, est qu'il cadre l'analyse. UML permet de représenter un système selon différentes vues complémentaires : les diagrammes. Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle.

Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis) et véhicule une sémantique précise (il offre toujours la même vue d'un système).

Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système. Les diagrammes permettent donc d'inspecter un modèle selon différentes perspectives et guident l'utilisation des éléments de modélisation (les concepts objet), car ils possèdent une structure.

Une caractéristique importante des diagrammes UML, est qu'ils **supportent l'abstraction**. Cela permet de mieux contrôler la complexité dans l'expression et l'élaboration des solutions objet.

UML opte en effet pour **l'élaboration des modèles**, plutôt que pour une approche qui impose une barrière stricte entre analyse et conception. Les modèles d'analyse et de conception ne diffèrent que par leur niveau de détail, il n'y a pas de différence dans les concepts utilisés. UML n'introduit pas d'éléments de modélisation propres à une activité (analyse, conception...) ; le langage reste le même à tous les niveaux d'abstraction.

Cette approche simplificatrice facilite le passage entre les niveaux d'abstraction. L'élaboration encourage une approche non linéaire, les "retours en arrière" entre niveaux d'abstraction différents sont facilités et la traçabilité entre modèles de niveaux différents est assurée par l'unicité du langage.

UML favorise donc le prototypage, et c'est là une de ses forces. En effet, modéliser une application n'est pas une activité linéaire. Il s'agit d'une tâche très complexe, qui nécessite une approche itérative, car il est plus efficace de construire et valider par étapes, ce qui est difficile à cerner et maîtriser.

UML permet donc non seulement de représenter et de manipuler les concepts objet, il sous-entend une démarche d'analyse qui permet de concevoir une solution objet de manière itérative, grâce aux diagrammes, qui supportent l'abstraction.

Comme UML n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente. UML est une sorte de boîte à outils, qui permet d'améliorer progressivement vos méthodes de travail, tout en préservant vos modes de fonctionnement.

Intégrer UML par étapes dans un processus, de manière pragmatique, est tout à fait possible. La faculté d'UML de se fondre dans le processus courant, tout en véhiculant une démarche méthodologique, facilite son intégration et limite de nombreux risques (rejet des utilisateurs, coûts...).] [3]

5. Comparaison

5.1. Niveaux d'abstraction

[L'approche Merise **:** Le cycle d'abstraction permet de sérier les niveaux de préoccupations lors de la description ou de l'analyse du système. Les trois niveaux retenus correspondent à des degrés de stabilitéet d'invariance de moins en moins élevés.

–Le niveau conceptuel,
–le niveau logique,
–le niveau physique.

L'approche UML propose différentes notions (cas d'utilisation, paquetage, classe, composant, noeud) et différents diagrammes pour modéliser le systèmes aux différents niveaux d'abstraction.

5.2. Approche fonctionnelle

L'approche Merise propose une approche descendante oùle système réel est décomposéen activités, elles-mêmes déclinées en fonctions. Les fonctions sont composées de règles de gestion, elles-mêmes regroupées en opérations. Ces règles de gestion au niveau conceptuel génèrent des modules décomposés en modules plus simples et ainsi de suite jusqu'àobtenir des modules élémentaires... Les limites d'une telle approche résident dans le fait que les modules sont difficilement extensibles et exploitables pour de nouveaux systèmes.

L'approche UML: Les fonctions cèdent la place aux cas d'utilisation qui permettent de situer les besoins de l'utilisateur dans le contexte réel. A chaque scénario correspond des diagrammes d'interaction entre les objets du système et non pas un diagramme de fonction...

5.3. Dualité des données -traitements

L'approche Merise propose de considérer le système réel selon deux points de vue: un point de vue statique (les données), un point de vue dynamique (les traitements). Il s'agit d'avoir une vision duale du système réel pour bénéficier de l'impression de relief qui en résulte, et donc consolider et valider le système final.
L'approche UML: L'approche objet associe les informations et les traitements. De cette façon, elle assure un certain niveau de cohérence.] [2]

Tableau 1 : tableau comparatif UML - Merise

Merise	UML
méthode d'analyse et de conception de système d'information	langage de représentation d'un système d'information.
méthode de modélisation de données et traitements orienté bases de données relationnelles.	système de notation orienté objet.
relationnel	objet.
Franco-français	International
schéma directeur, étude préalable, étude détaillée et la réalisation.	langage de modélisation des systèmes standard, qui utilise des diagrammes pour représenter chaque aspect d'un systèmes ie: statique, dynamique,....en s'appuyant sur la notion d'orienté objet
plus adapté à une approche théorique	plus orientée vers la conception
du "bottom up" de la base de donnée vers le code	du "top down" du modèle vers la base de donnée.

6. Conclusion

En conclusion, Merise et UML sont techniquement **complémentaires**.Si l'on considère que le SI est modélisable comme deux sous-systèmes inclus l'un dans l'autre: le SIO (système d'information organisationnel) représentant le métier, englobant le SII (système d'information informatisé) représentant l'application informatique associée.Merise est adaptée au SIO, UML au SII. Maintenant, selon le positionnement de chacun (consultant, analyste métier,concepteur de logiciel, développeur), on s'appuiera plus ou moins sur chaque partie de méthode.

Selon l'étude effectuée, la méthode la plus convenable pour la modélisation des bases de données biométriques, serait la méthode Merise. Car elle apporte une formalisation éclairant les choix à effectuer. Aussi, elle se caractérise comme étant un langage commun de référence centré sur le système d'information et non sur l'informatique appliquée. Enfin, grâce à Merise la création d'un base de données biométriques devient claire et documentée, tout en atteignant des objectifs de fiabilités des données et d'évolution des applications.

7. Références bibliographiques

[1] Méthodologie des systèmes d'information – MERISE par DI GALLO Frédéric

[2] Méthodologies de développement de logiciels de gestion par P.-A. Sunier

[3] UML en français site web : http://uml.free.fr/

[4] La méthode Merise