

# Etude comparative de méthodes du Génie Logiciel en vue du développement de systèmes interactifs dans les processus administratifs complexes.

Emmanuel ADAM<sup>1</sup>(\*)(\*\*), Christophe KOLSKI (\*\*)

(\*) SOLVAY, Rue de Ransbeek, 310, B - 1120 Bruxelles, BELGIUM, Doctorant au LAMIH  
E-mail : emmanuel.adam@solway.com

(\*\*) LAMIH - URA CNRS 1775, Université de Valenciennes et du Hainaut Cambrésis,  
Le Mont Houy - B.P. 311 - 59304 Valenciennes Cedex, FRANCE  
E-mail : kolski@univ-valenciennes.fr

**Résumé.** Dans cet article, plusieurs méthodes issues du Génie logiciel sont comparées. Les critères de comparaison étendent ceux proposés dans une étude de Pascot et Bernadas en 1993. Des discussions sont systématiquement menées par rapport à la prise en compte des facteurs humains. Le domaine d'application sous-jacent à cette étude est le développement de systèmes interactifs dans les processus administratifs complexes de type workflow, où la notion d'organisation humaine est particulièrement importante.

**Mots-clés.** Systèmes complexes, modélisation, système interactif, génie logiciel, organisation humaine, confrontation de méthodes

## INTRODUCTION

Chaque année, des dizaines de milliers de logiciels interactifs sont développés dans les entreprises. Dans le meilleur des cas, l'équipe de développement se base sur une méthode rigoureuse de développement issue du Génie Logiciel. Cependant, il est fréquent de constater que les systèmes interactifs posent en général de nombreux problèmes d'utilisabilité, ne répondent pas toujours aux besoins des utilisateurs, sont mal adaptés à l'organisation du travail, etc ([30, 26, 20, 16]).

Il est important à ce sujet d'étudier et de comparer les différentes classes de méthodes disponibles et de voir dans quelle mesure celles-ci peuvent inciter à une meilleure prise en compte des facteurs humains. Dans ce but, et sans souci d'exhaustivité, cet article s'intéresse à plusieurs méthodes, représentatives de différentes classes :

- MERISE [41, 42] : représentative des méthodes systémiques (au sens de Le Moigne [23]), et dont la finalité est l'analyse et la conception de systèmes d'information (connectés à une base de données) ;
- OMT (Object Modeling Technique ; [37]) : représentative des méthodes d'analyse et de conception orientées objets ; bien entendu, d'autres méthodes bien connues auraient pu également être considérées, telles les méthodes proposées par Coad et Yourdon [9], Jacobson [18] ou Booch [7].
- UML (Unified Modeling Language; [29, 38]) : nouvelle méthode de modélisation et de conception unifiant les principales méthodes orientées objets [33]; une comparaison sommaire avec OMT permettra de constater si UML est mieux adaptée à la modélisation d'organisations humaines qu'OMT.
- 3AR (Agent Acteur Ressource Abstrait ; [27]) : représentative des méthodes d'analyse et de spécification de système d'information orientée agents (ou acteurs) qui commencent à apparaître

---

<sup>1</sup> La thèse d'Emmanuel ADAM est cofinancée à 50 % par SOLVAY et à 50 % par la Région Nord-Pas de Calais (France).

dans la littérature, s'inspirant de concepts provenant de l'Intelligence Artificielle Distribuée ;

- SADT (Structured Analysis and Design Technique ; [17]) : inspirée des concepts proposés par Ross [36], et représentative des méthodes dites structurées, telle SA [10], orientées vers les étapes d'analyse et de spécification,
- OSSAD (Office Support System Analysis and Design; [11]) : méthode d'analyse et de spécification de système d'information centrée sur l'organisation du travail,
- CISAD (Cooperative Information System Analysis and Design ; [31]) : méthode étendant OSSAD, tout en accordant énormément d'importance aux notions de coopération et de communication,
- MKSM (Methodology for Knowledge Systems Management ; [13]) : méthode de gestion de la connaissance dans l'entreprise. Nous avons ajouté cette méthode non classique (et plus orientée sur une approche de l'Intelligence Artificielle) dans la mesure où celle-ci peut également être source d'idées pour l'analyse et la conception de systèmes interactifs.

Toutefois, il faut noter qu'il existe de nombreux domaines d'application (logiciels de bureautique, systèmes d'information, systèmes de supervision des procédés industriels, systèmes grand public, etc). Ces domaines possèdent tous leur spécificité.

Notre objectif n'est pas une étude comparative générique valable pour n'importe quel domaine d'application. Notre objectif est plus modeste et réaliste : certaines de nos discussions seront influencées par un domaine d'application particulier : l'analyse et la conception de systèmes interactifs dédiés aux processus administratifs complexes, de type workflow [39]. Nous appelons processus administratif complexe : un processus de créations, d'échanges et de modifications de documents entre plusieurs personnes, réparties dans des espaces différents, n'ayant pas le même niveau hiérarchique ni la même responsabilité à l'égard des documents. Nous visons à ce que le système final ("intelligent" ou non au sens de l'Intelligence Artificielle) assure une veille pour guider les acteurs du système et coopérer avec eux. Ce système interactif doit contribuer à diminuer les circulations de documents, et donc le risque d'erreurs de traitement de ces documents [35]; en conséquence, la durée des procédures associées aux documents devrait également diminuer, de même que la charge de travail des différents intervenants humains.

Cette étude comparative reprend les critères proposés par Pascot et Bernadas en 1993 [32], et les étend avec pour toile de fond le domaine d'application cité ci-dessus. Ces auteurs, dans leur article, avaient proposé un cadre de référence, sorte de méta-modèle, permettant de comparer des méthodes de conception de systèmes d'information informatisés.

## CRITERES DE CONFRONTATION ETENDANT CEUX DE PASCOT ET BERNADAS

Dans ce paragraphe sont résumés les critères utilisés pour la confrontation des méthodes. Ces critères sont fortement inspirés de Pascot et Bernadas, mais ont été adaptés pour répondre à notre problématique : l'analyse et la conception de systèmes interactifs dédiés aux systèmes administratifs complexes, où la coopération joue un rôle important. Pour plus de détail sur ces critères et leurs valeurs possibles, il est bien entendu conseillé de se référer à ces auteurs.

Après chaque définition de critères, se trouvent quelques lignes *en italique* expliquant les modifications apportées aux critères de Pascot et Bernadas. Aux 4 dimensions ("représentation", "organisation", "méthodologie" et "technologie") du cadre de référence proposé par ces auteurs a été ajoutée une cinquième dimension appelée "coopération".

Les méthodes d'analyse et de conception existantes n'ont pas été définies pour être employées sur tous les domaines d'application, mais partent en général d'un besoin de méthode dans un domaine particulier (par exemple : base de données pour la méthode MERISE, conception de systèmes orientés objets pour la méthode OMT, gestion des connaissances pour la méthode MKSM), ou d'une nécessité d'approfondir une méthode (par exemple : la méthode CISAD étend la méthode OSSAD; UML fusionne des concepts objets de plusieurs méthodes : OMT, la méthode de BOOCH, et les cas d'utilisation provenant de Jacobson [18]).

Le point commun de toutes ces méthodes est qu'elles cherchent à représenter dans un formalisme adapté les données issues d'une analyse du système. A ce sujet, nous associerons à une dimension appelée "représentation", quatre critères.

Si toutes les méthodes décrivent des modes de représentation du système, toutes ne précisent pas quelle structure doit avoir ce système (c'est le cas de plusieurs méthodes visant la conception, telles que OMT ou UML), et quelles peuvent être les caractéristiques de l'environnement dans lequel évolue le système. Cette notion est reprise dans les trois critères de la dimension "structure".

Le point le plus important d'une méthode est la description du cadre méthodologique qui la sous-tend. Ce point est plus ou moins détaillé selon l'objectif de la méthode : il l'est plus dans le cas de méthodes tournées vers la conception logicielle (UML et MERISE par exemple), il l'est moins dans les méthodes visant l'optimisation des prises de décisions dans les organisations (telle que OSSAD et CISAD). Cinq critères s'intéressent au cycle de développement dans la dimension "méthodologie".

Les différentes méthodes étudiées visent en général différents objectifs, concrétisés par des résultats logiciels. Trois critères permettent de caractériser ces résultats dans la dimension "technologie".

Enfin, suite aux évolutions récentes telles que l'explosion d'Internet et des Intranets et la numérisation de plus en plus poussée des données, de nouveaux besoins ressortent au niveau des groupes de travail, ce qui oblige les concepteurs à considérer des facteurs humains orientés sur le travail coopératif. A cet effet, quatre critères, dans la dimension "coopération", visent à prendre en compte ces aspects.

Le cadre de référence pour la confrontation des méthodes est donc composé de 5 dimensions : la dimension "représentation", la dimension "structure", la dimension "méthodologie", la dimension "technologie" et la dimension "coopération".

**La dimension représentation** est associée à quatre critères :

↳ La position de l'analyse (façon dont les auteurs de la méthode "attaquent" le système) :

- *totale* : analyse de l'ensemble du fonctionnement système,
- *partielle* : analyse du système centrée sur les points critiques.

↳ Le principe de construction : Il s'organise autour de quatre axes : *les données, les traitements, l'activité* (dans le cas de processus administratifs, ce sont les flux de communication) et *la dynamique*. Il est possible d'indiquer dans quel ordre ils sont traités. Ainsi, des axes peuvent être traités en parallèle (par exemple, lorsque les données et les traitements sont définis en parallèle). Des opérations de confrontation et d'assemblage peuvent ensuite être effectuées sur les quatre axes.

↳ Le découpage du système (moyen pour regrouper/séparer les éléments décelés dans l'analyse) : celui-ci est réalisé par niveaux d'abstraction (lorsqu'une organisation est étudiée, on peut partir du comportement des personnes interrogées, puis abstraire la personne en l'associant à un rôle, puis à un groupe de travail, dans le cas d'une approche ascendante), généralisation - spécialisation (toujours dans le cas d'organisations humaines, on peut regarder le comportement d'une personne par la description de son rôle, puis étudier plus en détail son comportement spécifique), type - occurrence ou stratégie - tactique (dans le cas d'un département, il s'agit d'étudier ses stratégies à long terme, le pourquoi de l'activité, ainsi que les tactiques, le comment).

↳ Le formalisme (schémas, concepts et règles) : on s'intéresse ici aux formalismes de *données, d'activités, de traitements* et de *la dynamique* utilisés par la méthode.

*Dans la dimension "représentation", par rapport à Pascot et Bernardas, nous avons ajouté la prise en considération de l'activité du système (au sens des flux de données). Cette donnée est très importante dans les processus administratifs car elle correspond en quelque sorte à la traduction directe (photographie) du système étudié. Un autre point, très important aussi dans l'analyse et la modélisation d'organisations*

*humaines, est la prise en compte de la dynamique inhérente aux systèmes coopératifs (tels les interruptions, le quasi parallélisme, la synchronisation).*

**La dimension organisation** est associée à trois critères :

↳ Images d'organisation : c'est la possibilité que la méthode a de traduire ou de représenter les organisations de type *boîte noire, hiérarchie, communauté, système cybernétique* (l'organisation est connue par ses échanges et ses actions sur les données), *système cybernétique ouvert* (l'organisation est considérée comme un système d'échange et les stratégies du système sont étudiées), *système holonique* (proposé par Koestler en 1968 [19], c'est un système hiérarchique ouvert, c'est-à-dire que les différents niveaux disposent d'une autonomie, d'une liberté d'action pour faire face à des contraintes).

↳ Nature de l'environnement : ce sont les caractéristiques de l'environnement du système sur lequel s'applique la méthode. Celles-ci comprennent le *degré de structuration*, de *stabilité* et de *certitude* (en fait certitude des connaissances sur l'environnement).

↳ Typologie de la connaissance traitée : *qualitative, quantitative, pertinente* (prise en compte du sens des données).

*Ici, de nouvelles images d'organisation ont été ajoutées, plus proches des organisations humaines (en particulier les structures hiérarchique, communautaire et holonique). De même, le critère de typologie de la connaissance a été ajouté, car il est important de connaître les types de documents manipulés dans les systèmes administratifs complexes.*

**La dimension méthodologie** est associée à cinq critères :

↳ Cycle de développement : celui-ci peut se baser sur les modèles *cascade ou spirale* (au sens de Boehm, [6]), en *V*,  $\nabla$  [20]

↳ Étapes concernées : on doit y retrouver des étapes classiques en Génie Logiciel : *Analyse, Modélisation, Spécification, Conception, Validation technique*. Il est important d'y retrouver d'autres étapes importantes au sens des facteurs humains: *Evaluations ergonomiques a priori* (avant la réalisation logicielle), *Evaluations ergonomiques a posteriori* (après la réalisation logicielle) [4, 14].

↳ Approche de développement : celle-ci peut être *descendante* (top-down), *ascendante* (bottom-up), ou *évolutive* (middle-out).

↳ Degré d'implication de l'utilisateur : ce degré va de '*pas d'implication*' à '*implication essentielle*'.

↳ Moment d'implication de l'utilisateur : celui-ci se situe au *début, au milieu et/ou en fin de cycle*.

*Les cycles de développement considérés étaient peu nombreux, il manquait le cycle en V, pourtant largement répandu, et le cycle  $\nabla$  proposé récemment par Kolski (cf. annexe 2), et dédié à la conception et à*

*l'évaluation de systèmes interactifs. Le critère reprenant les étapes du cycle de vie concernées par la méthode a aussi été ajouté, car les méthodes étudiées ne visent pas toutes la réalisation d'un logiciel.*

**La dimension technologie** est associée à trois critères :

- ↳ **Type de traitement** : ce sont les types de fonctionnement possibles des logiciels visés (*batch, interactif, client-serveur, synchrone, asynchrone, distribué*),
- ↳ **Type d'Interface homme-machine** : il existe différents types possibles d'interfaces : classique, adaptable (ou flexible, au sens de Williges [44]), adaptative (au sens de Schneider-Hufschmidt [40]), assistante (au sens de Boy [8]),
- ↳ **Programmation** : indique si elle doit être structurée, orientée base de données, orientée objet, orientée multi-agent (ou acteurs, au sens de l'Intelligence Artificielle Distribuée).

*Le critère "mémoire externe" (concernant le type d'accès à la mémoire : séquentiel pour les bandes et direct pour les disques) initialement prévue par Pascot et Bernardas a été abandonné car jugé obsolète et sans importance ici. Par contre, le critère "interface homme-machine" a été enrichi de nouveaux types possibles en rapport avec le domaine des interfaces dites "intelligentes" [22]. Le critère "programmation" prend en compte maintenant l'approche des systèmes multi-agents, faisant actuellement l'objet d'un courant de recherche.*

**La dimension coopération** est associée à quatre critères :

- ↳ **Communication** : indique la prise en compte des moyens de communication ou du *type de communication* (directe (envoi de message), indirecte (tableau noir), synchrone (par téléphone par exemple), asynchrone (par exemple : la messagerie)), la possibilité de représenter la

*négociation* (qui devient la forme de communication de plus en plus courante, les entreprises se tournant de plus en plus vers un management par projets) et l'utilisation d'un *modèle* permettant de formaliser les communications (celui-ci étant le plus souvent basé sur des théories de la linguistique),

- ↳ **Coordination** : indique la considération de la coordination des acteurs (plus généralement, la capacité à représenter la synchronisation entre les acteurs),
- ↳ **Relations** : indique la prise en compte du type de relation entre les acteurs (de hiérarchie, de responsabilité),
- ↳ **Individualité** : indique si la méthode permet de représenter des caractéristiques propres aux acteurs du groupe comme l'autonomie et les problèmes de confiance que pose le travail coopératif [43], c'est-à-dire si les acteurs peuvent être définis plus précisément que par leur appartenance à un groupe.

*Cette dimension a été ajoutée par rapport au cadre initial de Pascot et Bernardas. Elle est indispensable à partir du moment où une organisation humaine est concernée.*

La partie suivante propose une comparaison des méthodes MERISE, 3AR, OMT, UML SADT, OSSAD, CISAD et MKSM au travers de tableaux reprenant ces nouveaux critères.

## COMPARAISON DES DIFFERENTES METHODES

### ETUDE COMPARATIVE AUTOUR DE LA DIMENSION "REPRESENTATION"

Le tableau 1 montre globalement comment les différentes méthodes permettent de représenter le système à étudier, c'est-à-dire à l'aide d'un ensemble de formalismes, avec un ordre d'apparition dans la méthode.

| Représentation           |                                | MERISE | OMT         | UML           | 3AR | SADT   | OSSAD              | CISAD                          | MKSM     |
|--------------------------|--------------------------------|--------|-------------|---------------|-----|--------|--------------------|--------------------------------|----------|
| position de l'analyse    | Totale                         |        | *           |               |     | *      | *                  | *                              | *        |
|                          | Partielle                      | *      |             | *             | *   |        |                    |                                |          |
| principe de Construction | Données                        | ①      | ①           | ①③            | ①   | ①      |                    |                                | ③        |
|                          | Activités                      | ①      | ③           | ②④            | ①   | ①      | ①                  | ①                              | ②        |
|                          | Traitements                    | ②      |             | ⑥             |     |        | ②                  | ②                              | ①④       |
|                          | Dynamique                      |        | ②           | ⑤             | ①   |        |                    |                                |          |
| principe d'assemblage    | Type / Occurrence              | *      | *           | *             | *   |        |                    |                                |          |
|                          | Niveaux d'abstraction          | *      | *           | *             | *   | *      | *                  | *                              | *        |
|                          | Généralisation / spécification | *      | *           | *             |     | *      | *                  | *                              | *        |
|                          | Stratégique / tactique         |        |             |               |     |        |                    |                                |          |
| Formalisme               | Données                        | MCD    | MO          | DC, DO, Dcol, | MCD | Datag. | Fiches OSSAD       | cf OSSAD                       | MdD, MdC |
|                          | Activités                      | MCC    | DFD         | CU, Dseq,     | DFD | Actig. | Diag. A1, A2,D1,D2 | cf OSSAD + MC, MdAr, MdCv, D1' | MdA      |
|                          | Traitements                    | MOT    |             | DA            |     |        | Diag. D3, D4 et D5 | cf OSSAD                       | MdT      |
|                          | Dynamique                      |        | D.E. évolué | D.E. évolué   | RdP |        |                    |                                |          |

Tableau 1. Le système vu des différentes méthodes

**MCD** : (de MERISE), Modèle Conceptuel des Données

**MOT** : (de MERISE), Modèle Organisationnel des Traitements

**MCC** : (de MERISE), Modèle Conceptuel de Communication

**MO** : (de OMT), Modèle Objet

**DE évolué** : Diagramme d'Etat adapté ici pour OMT

**DC** : (de UML), Diagramme de Classe,

**DO** : (de UML), Diagramme d'Objet,

**Dcol** : (de UML), Diagramme de Collaboration,

**CU** : (de UML), Cas d'Utilisation,,

**DSeq** : (de UML), Diagramme de Séquence,

**DA** : (de UML), Diagramme d'Activités,

**DE** : (de UML), Diagramme d'Etat,

**DFD** : Diagramme de Flot de Données

**RdP** : Réseau de Petri

**Datag.** : (de SADT), Datagramme

**Actig.** : (de SADT), Actigramme

**Fiches OSSAD** : Les données ne sont pas représentées. Les fiches distinguent ici, les rôles, acteurs, unités, tâches, opérations, procédures, ressources et outils.

**Diagrammes de OSSAD** :

A1 : relations entre fonctions et sous-fonctions aux sein d'un organisme, A2 : Matrice Activité / Rôle

D1 : Relations entre Rôles, D2 : Relations entre tâches, D3 : Diagramme d'une tâche (1 rôle) (basé sur les RdP),

D4 : Diagramme d'une procédure (plusieurs rôles), D5 :

Détail d'une opération ou d'une tâche (basé sur les actigrammes)

**Diagrammes de CISAD** :

MC : modèle de communication inter-rôles sous la forme d'une matrice de Grudin,

MdAr : Modèle d'argumentation (raisonnement = question,

options, critères), MdCv : Modèle de conversation,

D1' : modèle de rôle de OSSAD modifié.

**MdD** : Modèle du Domaine

**MdA** : Modèle d'Activité, basé sur les actigrammes

**MdC** : Modèle des Concepts

**MdT** : Modèle des Tâches

[Ces formalismes sont représentés en annexe 1.]

A partir du tableau 1, différentes remarques s'imposent :

- Tout d'abord, dans MERISE, la représentation a une très grande importance. D'ailleurs, les formalismes y sont essentiels, même si la méthode est quelquefois injustement résumée par de nombreux développeurs à ses formalismes principaux : le Modèle Conceptuel des Données, et à un degré moindre le Modèle Organisationnel des Traitements. Le découpage en niveaux d'abstraction est très marqué. En général, on retrouve dans les différentes versions, les niveaux conceptuel, organisationnel et physique. Le système est d'abord abordé par les flux de communication, puis les données et les traitements sont analysés en parallèle pour ensuite être confrontés. En principe, les moyens fournis pourraient apporter un plus pour la collaboration entre concepteurs techniques et spécialistes de l'ergonomie (cf. à ce sujet les travaux de Barthelet, [5]).

- Pour OMT, les données ont bien sûr une très grande importance et le formalisme des données est très complet (il permet de représenter l'héritage, l'agrégation et les contraintes). Au niveau des formalismes, OMT est intéressante car elle n'en utilise que trois (le modèle Objet, le Diagramme d'Etat adapté et le Diagramme des Flots de Données) pour le cycle de vie du projet qui se déroule en quatre phases : l'analyse, la conception du système, la conception des données et l'implémentation. Cependant, on ne retrouve pas véritablement de modèle suggérant (comme par exemple le modèle Organisationnel des Traitements de la méthode MERISE) un rapprochement entre concepteurs et ergonomes, pour la prise en compte des besoins des utilisateurs.
- UML provient d'une fusion des principales méthodes objets. En cela, elle reprend et améliore leurs formalismes. Pour ce qui est des données, UML

simplifie l'écriture du modèle de données d'OMT. UML utilise beaucoup plus de modèles qu'OMT, ce qui peut être déroutant, même si ces modèles sont basés sur des formalismes semblables (comme les diagrammes de classe et les diagrammes d'objets par exemple). Les diagrammes utilisés sont : le diagramme de classe, le diagramme d'objet, le diagramme de collaboration, les cas d'utilisation, les diagrammes de séquences, le diagramme d'activité et les diagrammes d'état. L'utilisation des cas d'utilisation de Jacobson oblige d'intégrer ou tout au moins de prendre en compte l'utilisateur, ce qui constitue un progrès très important par rapport à OMT en particulier.

- 3AR propose de réunir les trois aspects données, activités et comportements dans un seul modèle d'acteurs. Ces acteurs sont ensuite découpés en niveaux d'abstraction afin d'accroître la réactivité du système d'information à concevoir. Cette méthode traite donc les trois aspects données, traitements et activités de façon unitaire et s'arrête à l'analyse et à la représentation des connaissances. 3AR reprend les formalismes existant (le MCD de MERISE, les Réseaux de Petri et les Diagrammes de Flots de Données). A notre avis, les concepts sont prometteurs mais encore mal exploités par les auteurs de la méthode 3AR vis-à-vis de la prise en compte des facteurs humains.
- SADT permet l'analyse et la modélisation de systèmes à l'aide de deux formalismes : les datagrammes et les actigrammes. Cette méthode est très utilisée pour décrire de façon structurée des problèmes et leurs solutions. Elle comprend 7 concepts dont celui de la dualité activités-données. Ce qui entraîne un test de correspondance entre les deux modèles par une analyse de liens Activités/Données et de liens Données/Activités. Elle procède par une démarche Top-Down qui peut être interprétée en terme de niveaux d'abstraction ou en terme de généralisation/spécification. Notons que les moyens de représentation de la méthode SADT a été exploitée par Abed et Angue [1] pour la modélisation statique des tâches et/ou des activités humaines, et complémentée par les réseaux de Petri pour la composante dynamique.
- OSSAD se compose de trois modèles : le modèle Abstrait qui comprend 2 graphes (A1 : relations entre fonctions ou activités, A2 : matrice activité/Rôle), le modèle descriptif qui comprend 5 graphes (D1 : relations entre rôles, D2 : relations entre tâches, D3 : diagramme d'une tâche (1 rôle), D4 : diagramme

d'une procédure (plusieurs rôles), D5 : description détaillée d'une opération). Cette méthode ne décrit pas les données, mais uniquement les activités, les tâches et les acteurs de ces activités essentiellement de type bureautique. En principe, les méthodes de représentation disponibles dans cette méthode sont centrées sur l'analyse du travail humain, et nous paraissent donc prometteuses au sens d'une coopération entre concepteurs techniques et ergonomes.

- CISAD est une amélioration de la méthode OSSAD. Elle reprend les modèles de OSSAD auxquels elle ajoute des composantes permettant de représenter la communication entre le personnel coopérant d'un système. Ces composantes sont la matrice de Grudin [15] pour la modélisation des communications, le modèle d'argumentation de la méthode IBIS qui représente le raisonnement par une représentation "Question - Options - Critères" et le modèle de conversation d'après le modèle orienté coordination Action. Le modèle de rôle D1 est adapté afin de représenter la responsabilité des rôles dans les différentes activités. De même que OSSAD, cette méthode ne décrit pas les données, mais décrit en plus les liens de communications et de responsabilités. Encore plus que pour OSSAD, plusieurs moyens sont disponibles pour faciliter la prise en compte des facteurs humains.
- MKSM découpe le système selon trois axes : la syntaxe, la pragmatique et la sémantique. Chacun de ces axes est découpé en trois points de vue : la structure, la fonction et l'évolution. Cette méthode commence d'abord par identifier les processus par le modèle du domaine, puis le modèle d'activité décrit les flux de données par actigrammes. Les données sont ensuite représentées par le modèle des concepts, et enfin les tâches sont représentées par le modèle des tâches. Là aussi, la plupart des concepts de base manipulés peuvent aller dans le sens de la prise en compte des facteurs humains.

#### **ETUDE COMPARATIVE AUTOUR DE LA DIMENSION "ORGANISATION"**

La manière dont l'organisation homme-machine est considérée dans une méthode du Génie Logiciel est essentielle : elle peut avoir des conséquences importantes sur le logiciel résultant qui répondra plus ou moins bien aux besoins de ses différents utilisateurs, et surtout de la manière dont ils sont organisés. Le tableau 2 apporte un premier éclairage à ce sujet.

| Structure                 |                             | MERISE | OMT | UML | 3AR | SADT | OSSAD | CISAD | MKSM |
|---------------------------|-----------------------------|--------|-----|-----|-----|------|-------|-------|------|
| images d'organisation     | Boîte noire                 |        |     |     |     |      |       |       |      |
|                           | Hierarchie                  |        |     |     |     |      |       |       |      |
|                           | Communauté                  |        |     |     |     |      |       |       |      |
|                           | Système holonique           |        |     |     |     |      |       |       |      |
|                           | Système cybernétique        | x      | x   | x   |     | x    |       |       |      |
|                           | Système cybernétique ouvert |        |     | x   | x   |      | x     | x     | x    |
| nature de l'environnement | Structuré                   | x      | x   | x   | x   | x    | x     | x     | x    |
|                           | Semi-structuré              |        | x   | x   | x   |      | x     | x     | x    |
|                           | Non structuré               |        |     |     |     |      |       |       | x    |
|                           | Stable                      | x      | x   | x   | x   | x    | x     | x     | x    |
|                           | Instable                    |        | x   | x   | x   |      |       |       |      |
|                           | Certain                     | x      | x   | x   | x   | x    | x     | x     | x    |
|                           | Incertain                   |        |     |     |     |      |       |       |      |
| Typologie des Données     | Qualité                     | x      | x   | x   | x   | x    | x     | x     | x    |
|                           | Quantité                    | x      | x   | x   |     |      |       |       |      |
|                           | Pertinence                  |        | x   | x   |     |      |       |       | x    |

Tableau 2. L'organisation vue des différentes méthodes

Rappelons d'abord que pour le *système cybernétique*, l'organisation est connue par ses échanges et ses actions, et que pour le *système ouvert finalisé*, elle est considérée comme un système d'échange et les actions effectuées sont connues. Dans ce cadre, les conclusions suivantes ressortent :

- MERISE considère l'organisation par ses échanges et ses actions (par le MCD et le MOT). Cette méthode est destinée aux systèmes structurés, stables et certains... ce qui n'est pas le cas dans différents domaines d'application (ou entreprises). De plus, sa finalité orientée base de données permet d'avoir une information qualitative et quantitative des données.
- Pour OMT, le système en général (on ne parle pas vraiment d'organisation dans OMT, ce qui constitue à notre sens une limitation) est aussi connu par ses échanges et ses actions, mais cette méthode, par sa conception objet, peut s'appliquer à des systèmes semi-structurés et instables dans une mesure prédéfinie. En principe, l'approche objet permet d'obtenir toutes les informations nécessaires sur les données.
- UML étant une méthode objet, les remarques précédentes s'y appliquent. UML par contre peut aussi s'appliquer à un système cybernétique ouvert, c'est-à-dire un système basé sur la communication dont les stratégies sont connues. En effet, par l'utilisation des diagrammes d'état et de séquence, il est possible de modéliser les comportements des objets comme ceux plus globaux des classes.
- La méthode 3AR, orientée système d'information, s'applique aux systèmes ouverts finalisés, et comme OMT, peut s'appliquer à des systèmes semi-structurés et instables. Pour représenter les données, elle utilise le formalisme MCD de MERISE qui n'apporte qu'une information qualitative. Notons que des recherches prometteuses sont menées en IAD concernant l'analyse et la modélisation de différentes organisations [24, 34]. Des travaux restent à faire

concernant l'intégration de tels concepts dans des méthodes du Génie Logiciel orientées agents.

- La méthode SADT s'applique à l'analyse fonctionnelle de systèmes et donc aux systèmes cybernétiques. De par son analyse descendante, hiérarchique et structurée, elle ne peut s'appliquer qu'à des environnements stables, certains et structurés. Seul l'aspect qualitatif des données est pris en compte. Un aspect important de la méthode est qu'elle suggère une analyse prenant en compte le point de vue des différents intervenants humains.
- OSSAD, destinée aux systèmes d'information, voit l'organisation uniquement comme un système ouvert finalisé. Son modèle abstrait (représentant ce qui doit être fait et pourquoi) nécessite un système stable et certain. Son modèle descriptif (représentant qui fait quoi et comment) apporte les solutions. OSSAD peut donc s'appliquer à un environnement semi-structuré. Au travers de ses diagrammes, OSSAD fournit des informations relatives à la pertinence et la qualité des données.
- CISAD étant une adaptation de OSSAD, les commentaires valables pour OSSAD le sont aussi pour CISAD.
- MKSM, présentée comme une méthode de gestion des connaissances, s'applique aux systèmes ouverts finalisés. Cette méthode permet de modéliser le patrimoine de connaissances par raffinements successifs et ne nécessite pas un environnement structuré. Par contre, son analyse de processus et d'activités nécessite un environnement stable et certain. Son modèle de processus et son modèle de concepts permettent d'avoir également des informations sur la pertinence et la qualité des données.

#### ETUDE COMPARATIVE AUTOUR DE LA DIMENSION "METHODOLOGIE"

Le tableau 3 donne un aperçu de la manière dont s'enchaînent les étapes dans les différentes méthodes, et

leur portée dans le cycle de vie du projet. Cette dimension montre aussi globalement comment les

méthodes intègrent les utilisateurs.

| Méthodologie                          |                         | MERISE | OMT | UML | 3AR | SADT | OSSAD | CISAD | MKSM |
|---------------------------------------|-------------------------|--------|-----|-----|-----|------|-------|-------|------|
| Cycle de Développement                | Cascade                 | x      |     |     | x   | x    |       |       |      |
|                                       | Spirale                 |        | x   | x   |     |      |       |       |      |
|                                       | V                       |        |     |     |     |      | x     | x     | x    |
|                                       | ∇                       |        |     |     |     |      |       |       |      |
| Étapes concernées                     | Analyse                 | x      | x   | x   | x   | x    | x     | x     | x    |
|                                       | Modélisation            | x      | x   | x   | x   | x    | x     | x     | x    |
|                                       | Spécification           | x      | x   | x   | x   |      | x     | x     |      |
|                                       | Conception              | x      | x   | x   |     |      |       |       |      |
|                                       | Evaluation à priori     |        |     |     |     |      |       |       |      |
|                                       | Evaluation à posteriori |        | x   | x   |     |      |       |       |      |
| Approche                              | Descendante             | x      | x   |     |     | x    | x     | x     | x    |
|                                       | Ascendante              |        |     |     | x   |      |       |       |      |
|                                       | Évolutive               |        |     | x   |     |      |       |       |      |
| Degré d'implication de l'utilisateur  | Pas                     |        |     |     | x   |      |       |       |      |
|                                       | Peu                     | x      | x   |     |     | x    |       |       | x    |
|                                       | Beaucoup                |        |     | x   |     |      |       |       |      |
|                                       | Essentiel               |        |     |     |     |      | x     | x     |      |
| Moment d'implication de l'utilisateur | Début                   | x      |     | x   |     | x    | x     | x     | x    |
|                                       | Milieu                  |        |     | x   |     |      | x     | x     |      |
|                                       | fin                     |        | x   | x   |     | x    | x     | x     |      |

Tableau 3. Aspects méthodologiques abordée par les méthodes

Le tableau 3 suggère des remarques significatives au sens de la prise en compte des facteurs humains dans les méthodes du Génie Logiciel.

- MERISE couvre le cycle de vie total du projet, supporte le développement en cascade et possède une approche descendante. L'utilisateur est peu impliqué dans le projet et sa contribution se situe surtout au début, au niveau des modèles conceptuels et organisationnels. Soulignons à nouveau les travaux de Barthet [5] visant à mieux intégrer la prise en compte de l'utilisateur dans la méthode MERISE.
- OMT couvre presque entièrement le cycle de vie du projet (jusqu'aux phases de validation, mais non de maintenance). Le cycle de développement utilisé est le cycle en spirale (suggérant en principe le prototypage), mais celui-ci est uniquement centré sur la modélisation de l'application, et son raffinement successif. Son approche est également descendante, l'utilisateur est un peu impliqué à la fin du projet. On trouve dans la plupart des méthodes orientées objets de telles limitations : l'utilisateur y est quasi inexistant.
- UML utilise aussi un cycle de développement en spirale (présenté comme développement itératif), et couvre le cycle de vie du logiciel jusqu'aux phases de validation. L'approche peut être définie comme évolutive. En effet, les classes sont d'abord définies, puis les liens entre ces classes (on obtient un schéma général du système), puis on descend au niveau des objets et de leur utilisation et de leurs comportements. L'utilisateur est plus impliqué que dans OMT grâce aux cas d'utilisation et au développement itératif.

- La méthode 3AR ne couvre pas l'ensemble du cycle de vie du logiciel mais seulement l'analyse et la spécification. Elle utilise le cycle de développement en cascade et n'intègre pas l'utilisateur dans le projet. De nombreuses extensions sont donc nécessaires à cette méthode et aux méthodes de ce type inspirées de concepts de l'Intelligence Artificielle Distribuée.
- SADT facilite surtout l'analyse et la modélisation du système (même si ses auteurs suggèrent d'aller jusqu'à la validation du système) et suit le cycle en cascade. Son approche est bien sûr descendante, et l'utilisateur est surtout impliqué lors de cette phase d'analyse.
- OSSAD et CISAD, ont pour but la réorganisation et sa simulation. Elles s'intéressent à l'analyse et à la spécification et laissent le choix à d'autres méthodes pour ce qui est de l'implémentation. Elles utilisent le cycle de vie en V et suivent une démarche descendante. L'avis de l'utilisateur est essentiel tout au long du projet.
- MKSM, en tant que méthode de gestion des connaissances, ne permet que l'analyse du système. La spécification et la conception sont réalisées en s'appuyant sur la méthode OMT. Elle utilise le cycle de développement en V et une approche descendante. L'utilisateur s'avère peu impliqué durant le projet.

#### ÉTUDE COMPARATIVE AUTOUR DE LA DIMENSION "TECHNOLOGIE"

Le tableau 4 permet d'appréhender le résultat final que l'on peut attendre des différentes méthodes, du point de vue logiciel.



| Technologie                    |                      | MERISE | OMT | UML | 3AR | SADT | OSSAD | CISAD | MKSM |
|--------------------------------|----------------------|--------|-----|-----|-----|------|-------|-------|------|
| Mode de Traitement             | Batch                | x      |     |     |     |      |       |       |      |
|                                | Interactif           | x      | x   | x   | x   | x    |       |       | x    |
|                                | Client-serveur       | x      | x   | x   | x   | x    |       |       |      |
|                                | Synchrone            | x      | x   | x   | x   | x    |       |       |      |
|                                | Asynchrone           |        |     |     |     | x    |       |       | x    |
|                                | Distribué            |        |     | x   |     |      |       |       | x    |
| Type d'Interface Homme Machine | Mode de construction |        |     |     |     |      |       |       |      |
|                                | Classique            | x      | x   | x   | x   | x    | x     | x     | x    |
|                                | Adaptable            |        | x   | x   |     |      |       |       |      |
|                                | Adaptative           |        |     | x   |     |      |       |       |      |
|                                | Assistante           |        |     |     |     |      |       |       |      |
| Programmation                  | Structurée           | x      | x   | x   | x   | x    | x     | x     | x    |
|                                | Base de données      | x      |     |     |     |      | x     | x     |      |
|                                | Objet                |        | x   | x   | x   |      |       |       | x    |
|                                | Multi-agents         |        |     | x   | x   |      |       |       |      |

Tableau 4. Logiciels réalisables par les méthodes

Plusieurs remarques découlent du tableau 4.

- MERISE est destinée à la conception de bases de données qui peuvent supporter un mode de traitement batch, interactif, synchrone et client-serveur (suivant la version de MERISE). Elle peut aboutir à une programmation structurée, mais surtout à la conception d'une base de données éventuellement orientée objet (grâce aux extensions actuelles de MERISE; cf. par exemple [28]).
- OMT est une méthode orientée objet. Elle permet de concevoir des programmes qui peuvent être interactifs, basés sur une architecture client-serveur, synchrones et/ou distribués. La programmation issue d'une spécification OMT peut bien sûr être orientée objet, mais aussi structurée comme l'expliquent les auteurs de la méthode.
- UML étant aussi une méthode orientée objet, elle permet de concevoir les programmes du même type que OMT. Mais ses autres diagrammes (diagrammes de séquence, diagramme de collaboration, diagrammes d'activités) permettent la conception de logiciels issus d'une programmation distribuée, et pourraient servir de base à la conception d'interfaces adaptatives par les diagrammes des cas d'utilisation, mais des recherches sont encore à mener à ce sujet.
- 3AR est orientée agent et permet d'obtenir des programmes interactifs, pouvant reposer sur une architecture client-serveur, fonctionnant de manière synchrone ou asynchrone ou ayant des modules distribués. Le type de programmation issu d'une spécification 3AR peut être structuré, orienté objet ou orienté multi-agents (acteurs).
- SADT, OSSAD et CISAD sont des méthodes d'analyse et de modélisation. En principe, elles ne sont pas destinées à la conception d'applications. A partir d'une analyse SADT, il est possible de concevoir des applications à l'aide d'une programmation structurée. Il en est de même pour OSSAD et CISAD qui peuvent, par leur modélisation des rôles, aboutir en plus à une programmation orientée objet.

- MKSM, méthode de gestion des connaissances repose sur OMT pour ce qui est de la spécification d'application. Les critères technologiques reprennent donc ceux de OMT.

Il est très important de souligner qu'aucune méthode ne prend en considération véritablement l'interface utilisateur. C'est pourquoi le critère "classique" apparaît dans chacune d'elle. Aucune suggestion n'est faite dans ces méthodes concernant les approches d'interfaces dites "intelligentes" qui sont pourtant apparues dans la littérature depuis le début des années 80 (cf. par exemple [12]). De nombreuses améliorations devraient donc pouvoir leur être apportées à ce sujet.

#### ETUDE COMPARATIVE AUTOUR DE LA DIMENSION "COOPERATION"

La coopération n'est considérée que dans OSSAD et CISAD, le tableau suivant met donc l'accent sur ces deux méthodes.

Plusieurs remarques peuvent être formulées au vu du tableau 5.

Toutes les méthodes permettent de représenter les flux de données d'une organisation. Mais seules quatre d'entre elles permettent de représenter la coordination (ou synchronisation) au sein de l'organisation. En effet, les modèles de traitement de MERISE de OSSAD et de CISAD ainsi que le réseau de Petri utilisé par 3AR permettent de représenter la coordination au sein d'une organisation. Tandis que les modèles dynamiques de OMT et de UML ne s'appliquent qu'à la définition des opérations à l'intérieur des objets. Par contre, la modélisation objet des acteurs permet à OMT et UML de représenter leur éventuelle autonomie, OSSAD et CISAD utilisant les fiches acteurs.

| Relation interpersonnel |                        | MERISE | OMT | UML | 3AR | SADT | OSSAD | CISAD                                      | MKSM |
|-------------------------|------------------------|--------|-----|-----|-----|------|-------|--|------|
| Communication           | De données             | x      | x   | x   | x   | x    | x     | x  | x    |
|                         | Modes de communication |        |     |     |     |      |       | Synchrone/<br>Asynchrone                   |      |
|                         | Modèle du langage      |        |     |     |     |      |       | Modèles de conversation et d'argumentation |      |
|                         | Négociation            |        |     |     |     |      |       | x  |      |
| Relations               | Hiérarchie             |        |     |     |     |      | x     | x  |      |
|                         | Responsabilité         |        |     |     |     |      |       | x  |      |
| Coordination            |                        | x      |     |     | x   |      | x     | x  |      |
| Individualité           | Confiance              |        |     |     |     |      |       |  |      |
|                         | Autonomie              |        | x   | x   |     |      | x     | x  |      |

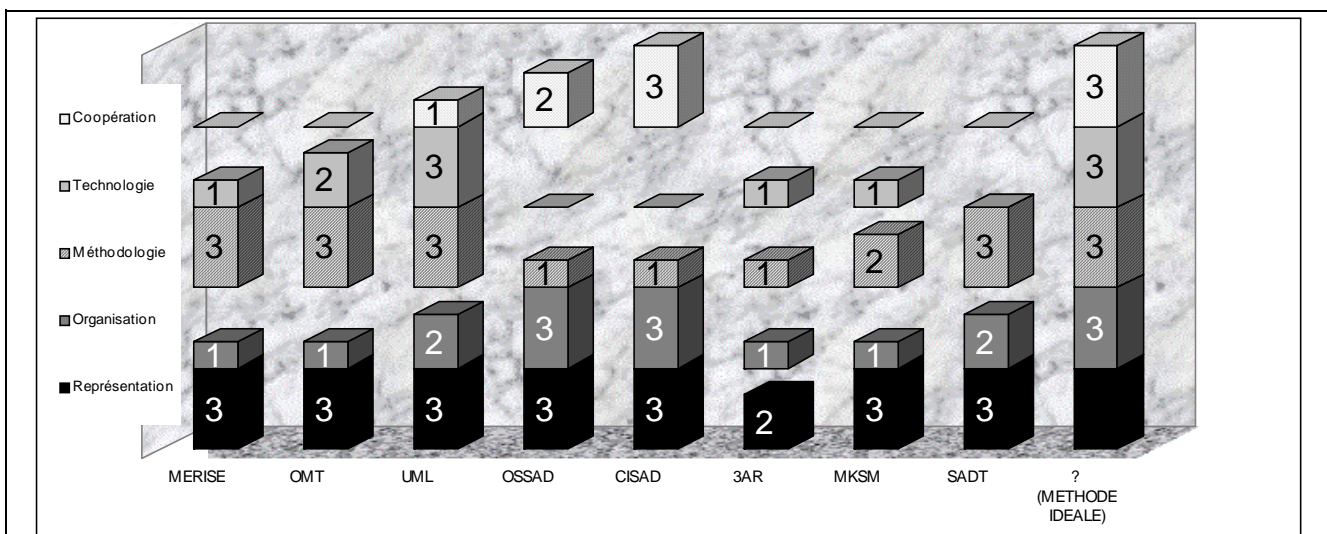
Tableau 5. Prise en compte de la coopération dans les méthodes

CISAD reprend les points forts de OSSAD et les étend, notamment en prenant en compte la notion de responsabilité. CISAD offre la possibilité de définir le type de communications et de les détailler par les modèles de conversation et d'argumentation, ce qui permet de représenter les phases de négociation entre acteurs, principales structures des communications dans les organisations actuelles, qui ont de plus en plus une structure par projet plutôt que pyramidale.

En revanche, aucune méthode ne formalise la confiance que porte un rôle envers un autre rôle; pourtant cette notion peut être très importante dans certaines situations [25].

### RESULTAT GLOBAL DES CONFRONTATIONS ET DISCUSSION PAR RAPPORT AUX PROCESSUS INDUSTRIELS COMPLEXES

Le diagramme suivant représente une synthèse des différentes confrontations selon les cinq dimensions. La méthode idéale devrait couvrir toutes ces dimensions. Mais, si certaines méthodes sont tout a fait pertinentes pour quelques dimensions, elles ne répondent jamais à toutes les dimensions. OMT, par exemple, possède des formalismes bien décrits permettant de représenter clairement un problème, mais ne peut pas s'appliquer à différents types d'organisation, et ne prend pas en compte la coopération dans un système informatique multi-utilisateur. De même, si OSSAD s'intéresse à l'aspect coopération et organisation, elle ne prend pas en compte l'aspect technologique.



légende : 1 = préoccupation peu importante, 2 = préoccupation importante, 3 = préoccupation très importante.

Figure 1 .confrontation des 7 méthodes à l'aide de diagrammes

La figure 1 permet de faire ressortir plusieurs points importants. On remarque d'abord qu'aucune méthode

ne permet de gérer à la fois l'aspect coopération et l'aspect technologique. De plus, on peut obtenir une

colonne presque complète en couplant MERISE, UML, CISAD et SADT.

Peut-on déduire de cette confrontation une méthode répondant à notre préoccupation : c'est-à-dire l'analyse et la modélisation de systèmes administratifs complexes en vue de la conception d'un système interactif destiné à un ensemble d'acteurs ? Il est utile à ce sujet de reprendre certains points relatifs aux différentes dimensions.

- Pour ce qui est de la dimension **représentation**, le but étant d'obtenir une méthode d'analyse et de modélisation de systèmes administratifs complexes, il serait préférable d'utiliser une partie de la méthode UML pour ce qui est de la représentation des données. Pour la représentation des activités, CISAD ne propose pas un formalisme assez strict. L'utilisation de la méthode SADT, et surtout de ses actigrammes est à envisager. Les actigrammes de SADT ne permettant pas de représenter les interruptions de travail, ni les tâches parallèles ; les réseaux de Pétri peuvent apporter leur contribution, comme ils le font dans 3AR et dans OSSAD. Pour ce qui est des traitements, le MOT de MERISE est pertinent, mais OSSAD ajoute à ce formalisme la notion de rôle et d'équipe de travail.

L'approche doit être orientée par les problèmes, afin de réduire le nombre de cas à envisager. L'activité doit être étudiée en premier lieu, puis de cette analyse d'activité est déduite la représentation des données et des traitements. Ceci de façon à avoir un modèle le plus proche possible de la réalité car déduit de trois points de vue différents : l'activité, les données et les traitements. Une représentation, nécessaire, de la dynamique doit ensuite être effectuée afin d'obtenir tous les détails du système.

- Concernant la dimension **organisation**, la méthode visée est destinée aux systèmes administratifs, et donc aux systèmes ouverts finalisés. Elle doit être capable de représenter les systèmes structurés ou semi-structurés. Comme la finalité de la méthode est la spécification d'une solution logicielle interactive diminuant la charge de travail et le risque d'erreur dans le groupe, l'environnement ne peut pas être instable ni surtout incertain en ce qui concerne les activités et rôles des acteurs du système.

Il est bien sûr nécessaire d'avoir toutes les informations concernant les données qui circulent dans le système, c'est-à-dire les informations sur la quantité, la qualité et la pertinence des données.

- Concernant la dimension **méthodologie**, le cycle  $\nabla$  ([20,21]) dédié à la conception de systèmes interactifs, intègre beaucoup le futur utilisateur dans le cycle de vie du projet, procède à une approche descendante et suggère l'utilisation d'un modèle réel et un modèle de référence (le modèle visé). L'implication de l'utilisateur est bien sûr essentielle dans le cas d'interface hommes-machines. Nous chercherons à utiliser, voire adapter ce modèle.

- Concernant la dimension **technologie**, le mode de traitement sera bien sûr interactif, distribué car les utilisateurs sont répartis géographiquement dans des bureaux, synchrone ou asynchrone en ce qui concerne la transmission de données. L'interface sera adaptable, et devra être adaptative, sinon assistante. L'adaptativité est nécessaire dans un système où les personnes ne jouent pas le même rôle. La programmation pourra être structurée, orientée objet ou basée sur une architecture multi-agent.

- Pour la prise en compte de la dimension **coopération**, la méthode doit intégrer les aspects essentiels de communication et de relations inter-personnels telles que la hiérarchie, la responsabilité et prendre en compte certains problèmes de confiance dans l'organisation.

## CONCLUSION

Il existe maintenant un nombre important de méthodes du Génie Logiciel qui se sont développées en suivant l'évolution des langages et des techniques (MERISE et UML par exemple). Mais ces méthodes n'apportent pas la solution à tous les problèmes, elles disposent de certaines qualités propres au traitement de problèmes spécifiques mais ont des lacunes au regard d'autres problèmes. Elles sont en général très limitées lorsque le système visé est hautement interactif, et ont toutes besoin d'améliorations à ce sujet. Pourtant, et comme nous l'avons remarqué dans l'étude comparative de la dimension représentation, des moyens de représentation (en rapport avec les notions de tâches, procédures...) pourraient faire office de support de dialogue entre concepteurs techniques et ergonomes, dans plusieurs des méthodes étudiées.

La confrontation que nous avons effectuée a permis d'étudier en détail 8 méthodes représentatives de leur classe en fonction de notre besoin, relatif aux systèmes industriels complexes, et ceci selon 5 dimensions : représentation, structure, méthodologie, technologie et coopération. Cependant, il est clair que cette confrontation a été influencée par notre perception du domaine d'application ; de plus, pour certains critères, l'avis n'est pas toujours en réalité aussi binaire que le remplissage des tableaux le suggère. D'autres auteurs auraient pu remplir certaines cases de manière sensiblement différente.

Cette confrontation nous facilite tout de même la définition d'un cahier des charges global d'une méthode répondant à un besoin particulier. Dans notre cas, elle nous a permis de définir les principaux axes d'une méthode d'analyse et de modélisation de systèmes administratifs complexes en vue de la spécification d'un système interactif répondant à la problématique de tels systèmes (coopération, circulation de données, prise en compte de l'utilisateur dans le cycle de vie du projet). Cette méthode est en cours d'étude [2,3].

## Remerciements.

Les auteurs remercient Emmanuel Vergison pour ses remarques avisées. Ils remercient également D.Pascot et C.Bernardas qui ont été pour eux une source très importante d'inspiration.

## RÉFÉRENCES

- [1] ABED M., ANGUE J.C. (1994). A new method for conception, realisation and evaluation of man-machine interfaces. *Proceedings IEEE "Systems, man and cybernetics*, San Antonio 2-5/10.
- [2] ADAM E., VERGISON E., KOLSKI C., MANDIAU R. (1997). Holonic user driven methodologies and tools for simulating human organizations. *Proceedings European Simulation Symposium ESS'97*, University of Passau, Germany, October 19-23, 1997.
- [3] ADAM E., KOLSKI C., VERGISON E. (1998). Méthode adaptable basée sur la modélisation de processus pour l'analyse et l'optimisation de systèmes coopératifs dans l'entreprise. *Actes du Colloque International ERGO-IA'98*, Biarritz, 3-6 novembre.
- [4] BALBO S. (1994). *Evaluation ergonomique des interfaces utilisateur : un pas vers l'automatisation*. Thèse de doctorat, Université de Grenoble I, septembre.
- [5] BARTHET M.F. (1995). The DIANE Method and its connection with MERISE Method. *Proceedings IEA World Conference "Ergonomic design, interfaces, products, Information"*, October 16-20, Rio de Janeiro, Brazil, pp. 106-110.
- [6] BOEHM B.W. (1988). A spiral model of Software Development and enhancement. *Computer*, May.
- [7] BOOCH G. (1994). *Conception orientée objets et applications*. 2ème édition, Addison-Wesley.
- [8] BOY G. (1991). *Intelligent Assistant Systems*. New York: Academic Press.
- [9] COAD P., YOURDON E. (1991). *Object-oriented design*. Prentice Hall.
- [10] DeMARCO T. (1979). *Structured analysis and system specification*. Youdon Press, NY. Englewood Cliffs - Prentice Hall.
- [11] DUMAS P., CHARBONNEL G. (1990), *La méthode OSSAD, pour maîtriser les technologies de l'information. Tome 1 : principes*. Les éditions d'organisation, Paris.
- [12] EDMONDS E.A. (1981). Adaptative Man-Computer Interfaces. In : M.J. Coombs and J.L. Alty (eds.), *Computing skills and the user interface*, London, Academic press.
- [13] ERMINE J-L. (1995), *MKSM, méthode de gestion des connaissances*. CEA DIST/SMTI.
- [14] GRISLIN M., KOLSKI C. (1996). Evaluation des interfaces homme-machine lors du développement de système interactif. *Technique et Science Informatiques (TSI)*, 15 (3), mars, pp. 265-296.
- [15] GRUDIN J. (1994), *Eight Challenges for Developers*, Communications of the ACM, 37(1), January, pp.93-105
- [16] HELANDER M., LANDAUER T.K., PRABHU P. (Eds.) (1997). *Handbook of Human-Computer Interaction*. Elsevier Science Publishers B.V., North-Holland.
- [17] I.G.L. Technology (1989). *SADT, un langage pour communiquer*. Eyrolles, Paris.
- [18] JACOBSON I., CHRISTENSON M., JONSSON P., OVERGAARD G. (1993). *Le génie logiciel orienté objet, une approche fondée sur les cas d'utilisation*. Addison-Wesley, ACM Press.
- [19] KOESTLER A. (1969). *The Ghost in the Machine*. Arkana Books, London.
- [20] KOLSKI C. (1997). *Interfaces Homme-Machine, application aux systèmes industriels complexes*. Editions HERMES, Paris.
- [21] KOLSKI C. (1998). A "call for answers" around the proposition of an HCI-enriched model. *ACM SIGSOFT Software Engineering Notes*, 23 (3), pp. 93-96.
- [22] KOLSKI C., LE STRUGEON E. (1998). A review of "intelligent" human-machine interfaces in the light of the ARCH Model. *International Journal of Human-Computer Interaction*, 10 (3), pp.193-231.
- [23] LE MOIGNE J.L. (1977). *La théorie du système général - théorie de la modélisation*. Presses Universitaires de France.
- [24] LE STRUGEON E., MANDIAU R., MILLOT P. (1994). Representation for an organization in a multi-agent world. *Proceedings IFAC Conference on Integrated Systems Engineering*, Baden-Baden, Germany, 27-29 Septembre 1994, pp. 41-46.
- [25] LEE J.D., MORAY N. (1992). Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, 35, pp. 1243-1270.
- [26] LIM K.Y., LONG J.B. (1994). *The MUSE method for usability engineering*. Cambridge series on Human-Computer Interaction, Cambridge University Press.
- [27] MORAND B. (1993), 3AR : Un modèle Orienté Représentation des Connaissances pour les Systèmes d'Information. *Proceedings INFORSID'93 «Systèmes d'information, systèmes à base de connaissances»*, Lille, 11-14 Mai.
- [28] MOREJON J. (1994). *MERISE, vers une modélisation orientée objet*. Paris, Les Editions d'Organisation.
- [29] MULLER P.A. (1997). *Modélisation objet avec UML*. Paris, Eyrolles.
- [30] NIELSEN J. (1993). *Usability engineering*. Academic Press.
- [31] NURCAN S. (1996) Analyse et conception de systèmes d'information coopératifs. *Technique et Science Informatiques*. Vol 15, n°9, pp. 1287-1315.
- [32] PASCOT D., BERNADAS C. (1993), *L'Essence des Méthodes : Etude Comparative de Six Méthodes de Conception de Systèmes d'Information Informatisés*. INFORSID'93

- «Systèmes d'information, systèmes à base de connaissances», Lille, 11-14 Mai.
- [33] PERRIN P. (1997). La guerre des standards des méthodes orientées objet n'aura pas lieu ! *Ingénierie des Systèmes d'Information*, 5 (5), pp. 533-552.
- [34] PRIETULA M., CARLEY K., GASSER L. (1998). *Simulating organizations, computational models of institutions and groups*. Boston, MIT Press.
- [35] REASON J., 90. *Human Error*. Cambridge : Cambridge University Press, 1990.
- [36] ROSS D.T. (1977). Structured analysis (SA) : a language for communicating ideas. *IEEE Transactions on Software engineering*, vol.SE-3, 1, January.
- [37] RUMBAUGH J., BLAHA M., PREMERLANI W., EDDY F., LORENSEN W. (1991). *Object-oriented modeling and design*. Prentice-Hall.
- [38] RUMBAUGH J., JACOBSON I., BOOCH G. (1998). *Unified Modeling Reference Manual*. Addison-Wesley.
- [39] SHAEL T. (1997). *Théorie et pratique du Workflow, des processus métiers renouvelés*. Ed. Springer.
- [40] SCHNEIDER-HUFSCHMIDT M., KÜHME T., MALINKOWSKI U. (Eds.) (1993). *Adaptive User Interfaces*. North Holland.
- [41] TARDIEU H., ROCHFELD O., COLLETI R. (1991). *La méthode Merise, principes et outils, 2ème édition*. Editions d'Organisation (tome 1), Paris.
- [42] TARDIEU H., ROCHFELD O., COLLETI R., PANET G., VAHEE G. (1985). *La méthode Merise, démarche et pratiques*. Editions d'Organisation (tome 2), Paris.
- [43] THIMBLEBY H., MARSH S., JONES S., COCKBURN A.(1994). *Trust in CSCW, CSCW : The multimedia and networking paradigm*, UNICOM, ed. S.A.R. Scrivener
- [44] WILLIGES R.C., WILLIGES B.H., ELKERTON J. (1987). Software Interface Design. In G. Salvendy (ed.), *Handbook of Human factors*, pp. 1416-1449, New york, Wiley.