HAL
open science

SUPPORT DES PROCESSUS STRATEGIQUES DANS UN ENVIRONNEMENT CENTRE PROCESSUS

Amen Ben Hadj Ali, Henda Ben Ghézala

To cite this version:

Amen Ben Hadj Ali, Henda Ben Ghézala. SUPPORT DES PROCESSUS STRATEGIQUES DANS UN ENVIRONNEMENT CENTRE PROCESSUS. 2006. sic_00001660

HAL Id: sic_00001660

https://archivesic.ccsd.cnrs.fr/sic_00001660

Submitted on 19 Jan 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SUPPORT DES PROCESSUS STRATEGIQUES DANS UN ENVIRONNEMENT CENTRE PROCESSUS

Amen Ben Hadj Ali
Doctorant en informatique
amen.benhadjali@fst.rnu.tn +216 71 706 164

Henda Ben Ghézala
Professeur des universités en informatique
henda.benghezala@riadi.rnu.tn +216 71 600 444

Résumé : Les environnements centrés processus (ECP) représentent la nouvelle génération des environnements supportant les activités de développement de logiciels. La majorité des ECP existants se focalise sur l'imposition et l'automatisation des activités du processus en se basant sur des modèles de processus prescriptifs complètement détaillés au préalable. Cependant l'expérience a montré que le support de processus serait plus efficace en offrant un guidage flexible au lieu d'imposer un ensemble d'étapes prédéfinies. Ce papier présente une solution pour le support des processus stratégiques dans un ECP en offrant un guidage flexible durant la phase d'exécution.

Summary : Process-centred Software Engineering Environments (PSEE) are the most recent generation of environments supporting software development activities. Most of PSEE are based on mechanisms promoting enforcement and automation of process activities. In this kind of mechanisms the process models are prescribed in a detailed and complete way. But the experience shows that supporting processes is more concerned with the flexibility of guidance offered during the process performance than with enforcement of a collection of predefined process models. In this paper, we present a solution to support strategic processes in a PSEE by providing a flexible guidance during process enactment.

Mots clés : Processus, modélisation de processus, processus stratégiques, flexibilité, guidage du processus, ECP.

Support des processus stratégiques dans un environnement centré processus

1 - INTRODUCTION

La standardisation du développement par l'application manuelle d'une méthode s'est avérée une tâche laborieuse. Ainsi, il en a résulté un important besoin d'aider les développeurs dans leurs activités, Lyytinen (1989).
Parmi les technologies proposées pour aider les développeurs dans leurs activités de développement, on trouve les outils CASE (Computer-Aided Software engineering). Ces outils sont généralement considérés comme un support pour les techniques structurées proposées par les méthodes et ce en offrant des éditeurs capables de faire des vérifications, de stocker des informations de générer de la documentation, de manipuler le produit et de générer du code, Henderson (1994).

La technologie CASE (que l'on traduit par le terme français Atelier de Génie Logiciel) a connu une grande évolution depuis son émergence, on retrouve au premier lieu les outils CASE qui sont spécifiques à chaque phase du développement, puis les environnements CASE ont vu le jour afin de couvrir plusieurs activités du cycle de vie du logiciel, ces environnements apportent un plus par rapport aux outils mais ils restent toutefois dédiés à une seule méthode, Fuggetta (1993). Depuis la fin des années 80 et avec la tendance de diversification des méthodes, une autre classe d'outils est apparue, il s'agit des environnements metaCASE, ces outils présentent l'avantage d'être multi-méthodes en permettant de modéliser les méthodes et en offrant les outils appropriés pour l'application de la méthode appropriée à une situation donnée, Koskinen (2000).
Les outils CASE se sont focalisés sur la production du «produit» du développement en offrant entre autres une multitude d'éditeurs graphiques, textuels, vérificateurs syntaxiques des spécifications. Cependant avec l'évolution des méthodes, il était impératif de prendre en compte le processus logiciel, d'où la naissance d'une autre catégorie d'outils qui sont apparus en réponse au besoin d'assister les ingénieurs dans la conduite du processus de développement et par voie de conséquence d'intégrer la conduite du processus dans un environnement d'assistance, il s'agit des *Environnements Centrés Processus* (ECP).
Les ECP sont donc le résultat du déplacement du centre d'intérêt des outils vers la maîtrise de «processus» de production des logiciels. Un processus étant défini comme un ensemble d'activités requises pour transformer les besoins de l'utilisateur en un logiciel fonctionnel appelé produit, Humphrey (1989). Les ECP sont basés sur une définition formelle et explicite du processus logiciel sous forme de *modèles de processus* qui ont pour objectif de guider les utilisateurs dans leurs activités de développement.
Dans le cadre de ce papier, notre objectif est d'étudier la possibilité d'offrir un support pour les processus qui sont de nature *stratégique* en prenant en compte la *flexibilité* qu'ils nécessitent lors de leur exécution au sein d'un ECP. Cet objectif sera traité en répondant à deux questions fondamentales: Comment modéliser les processus stratégiques? Et Comment exprimer les contraintes et les règles liées à l'exécution de tels modèles?
Ce papier est organisé comme suit. La deuxième section traite du support de processus dans les ECP. La section trois présente les concepts de base utilisés dans le cadre de la modélisation des processus. Ceci étant dans le but de définir la terminologie que nous allons utiliser par la suite. La quatrième section présente la solution proposée pour le support des processus. Dans cette section nous détaillons les choix que nous avons adopté pour représenter et implémenter les processus stratégiques. Enfin, nos travaux futurs sont donnés en guise de conclusion.

2 - LE SUPPORT DE PROCESSUS DANS LES ECP

Les ECP sont des outils d'aide au développement pouvant offrir différentes fonctionnalités selon les objectifs de leurs auteurs. On trouve dans Garg (1996) une synthèse des principales caractéristiques qui peuvent être offertes par un ECP.
Pendant ces dernières années, plusieurs groupes de recherche ont mis au point des

ECP, on peut citer par exemple Barghouti (1993), Bandinelli (1993), Kelly (1996), etc.. La majorité de ces environnements est basée sur la modélisation du processus par des modèles formels et *prescriptifs* exprimant ce qui est attendu d'un processus et comment il devrait être accompli. Ces modèles conviennent pour modéliser des processus dont le comportement peut être entièrement prévu à priori, leur exécution est souvent *automatique* (réalisée par une machine) et elle consiste à suivre le processus prescrit dans le modèle sans pouvoir le changer. Toutefois, l'expérience a montré qu'il est souvent difficile de rester conforme aux prescriptions du modèle de processus et que son exécution ne peut pas être complètement automatisée car l'intervention de l'utilisateur doit être prise en considération, koskinen (2000). Ces contraintes nous ont permis de constater qu'il y a plusieurs aspects du processus qui sont mal considérés ou négligés dans la plupart des environnements existants, ces aspects sont liés à la flexibilité du processus, à la dynamique de son exécution dans un ECP et au type de support de processus proposé par l'environnement logiciel :

- *La flexibilité des processus:* Lors de l'exécution du modèle de processus un choix doit être fait dynamiquement parmi les différentes solutions possibles pour réaliser une tâche koskinen (2000). Ce choix est influencé par l'historique du processus qui évolue constamment et par la situation du produit qui change sans cesse. La majorité des ECPs se base sur des modèles de processus orientés activités qui décrivent le processus sous forme d'une séquence d'activités bien définie à l'avance, mais ils ne supportent pas des processus offrant la possibilité de faire des choix lors de leur exécution et qui nécessitent une grande flexibilité durant leur définition ainsi que pour leur exécution.
- *La dynamique de l'exécution du processus:* Durant l'exécution du modèle de processus, l'utilisateur peut être amené à choisir entre les différentes activités qui sont offertes par ce modèle, Kobialka (1998). Par conséquent, le processus effectivement exécuté est construit d'une manière progressive et incrémentale durant l'exécution du modèle de processus. La complexité du choix entre les différentes exécutions augmente en fonction de la multiplicité des exécutions possibles.
- *Le guidage de processus:* Le guidage est une caractéristique fortement liée à l'aspect dynamique de l'exécution du processus. Le support de processus serait donc plus efficace en offrant à l'utilisateur durant l'exécution du modèle de processus un guidage de processus plutôt que par l'imposition des prescriptions présentes dans ce modèle. En effet le guidage est le type d'assistance qui convient le plus à l'exécution des modèles de processus flexibles, Si-Said (1999). Par ailleurs, il représente une caractéristique dont l'importance est reconnue par la communauté du génie logiciel, Dowson (1994), comme par celle de l'ingénierie des systèmes d'information, Olle (1988).

Pour qu'un modèle, soit capable de guider des processus, il doit pouvoir être interprété par des machines et des êtres humains. Cet objectif dépend fortement du niveau de *granularité* permis par le langage de modélisation de processus et de son niveau de formalité. Une description informelle est en effet difficile à exécuter automatiquement. Mais l'exécution automatique des modèles de processus comme celle de programmes, que préconisent les ECP semble cependant inadaptée aux processus où le facteur humain et sa participation sont prépondérants.

3 - LA MODELISATION DU PROCESSUS

Les ECP sont basés sur la notion de processus logiciel et un modèle explicite de ce processus à savoir le modèle de processus, Garg (1996). En effet, pour qu'il puisse être exécuté au sein d'un ECP, le processus doit être décrit totalement ou partiellement d'une façon formelle en se basant sur un langage de modélisation de processus (LMP). Un tel langage peut être défini comme étant une notation formelle utilisée pour la description des modèles de processus, Finkelstein (1994).

La modélisation du processus est un domaine de recherche où l'objectif de base consiste à capturer les aspects du processus d'une méthode spécifique dans une description abstraite appelée modèle de processus, Curtis (1992). Cette discipline a été mise en place dans le but de mieux comprendre le processus, l'évaluer et pouvoir supporter son exécution automatisée, McChesney, (1995). Ces objectifs attribués à la modélisation de processus la rendent une tâche complexe et incertaine, Koskinen (2000). En conséquence, les approches de modélisation de processus ont

connu une grande évolution dans le sens d'offrir des modèles de processus de plus en plus structurés et précis.

Les processus peuvent être de trois types différents, Rolland (1998):

- Les processus *stratégiques* sont ceux qui explorent et étudient les différentes solutions alternatives permettant de satisfaire un objectif et qui, éventuellement, produisent un plan pour l'atteindre. Ces processus sont critiques aussi bien dans la génération des solutions alternatives que dans le choix de l'alternative appropriée.
- Les processus tactiques sont ceux qui proposent une assistance lors de la poursuite d'un plan. Comme leur nom l'indique, ces processus sont plus concernés par la tactique à adopter pour exécuter un plan que par la production de ce plan.
- Les processus *d'implantation* sont les processus du niveau d'abstraction le plus bas. Ils sont directement concernés par les détails de *quoi* implanter et *comment* l'implanter.

Il est important de noter que la nature d'un processus (stratégique, tactique ou d'implantation) peut influencer le choix d'un modèle de représentation adéquat. En effet il existe dans la littérature qutres catégories de modèles de processus: *orientés-activité*, *orientés-produit*, *orientés-décision* Dowson (1993) et les modèles dits *contextuels*, Rolland (1994).

Ainsi les modèles orientés activités par exemple sont peu adéquats pour la représentation des processus stratégiques car ils n'incorporent pas le raisonnement sous-jacent aux processus et n'offrent ainsi pas les moyens qui permettraient de raisonner sur les différents choix et les différentes solutions offertes par le modèle de processus. Toutefois ces modèles conviennent pour modéliser des processus tactiques et d'implantation.

Ces modèles sont incapables de fournir un guidage à l'utilisateur, seule une stricte exécution du modèle de processus est possible. Les modèles de processus *orientés produit* se focalisent sur les activités mais offrent le plus de modéliser l'évolution du produit et de coupler l'état du produit aux activités correspondantes Humphrey, (1989), ce qui permet d'enrichir les traces des processus mais ils présentent les mêmes insuffisances que celles des modèles de processus orientés activités. Les modèles de processus *orientés décision* considèrent que les transformations successives des produits réalisées par un processus sont des conséquences de prises de décision. Ces modèles sont sémantiquement plus puissants car ils expliquent non seulement comment se déroule le processus mais aussi pourquoi des transformations ont eu lieu. La classe des modèles de processus contextuels a été définie dans le projet NATURE (Novel Approches to Theories Underlying Requirements Engineering), Rolland (1994). Ces modèles couplent la décision à son contexte c'est à dire à la situation dans laquelle se trouve le produit. Grâce à cette relation entre la décision et la situation, seules les décisions qui s'adaptent à la situation sont appliquées.

Par ailleurs, les approches de modélisation des processus peuvent avoir différents objectifs, une synthèse des propos recueillis dans le domaine de la modélisation des processus, Lonchamp (1993), Curtis (1992) précise trois usages principaux des modèles de processus :

- *descriptif*, pour garder une trace de ce qui se passe effectivement durant l'accomplissement du processus. Ce propos permet de décrire explicitement les processus exécutés afin d'en faciliter l'étude.
- *explicatif*, pour fournir des explications sur les processus et sur le raisonnement sous-jacent. Ce propos permet de garder une trace explicite du lien entre les étapes et les besoins auxquels elles répondent lors de l'exécution du processus.
- *prescriptif*, pour exprimer ce qui est attendu d'un processus. Les modèles prescriptifs tendent à définir à l'avance des règles de comportement qui, si elles sont suivies, devraient conduire au processus désiré. La prescription couvre un spectre à quatre niveaux de flexibilité qui va de *l'exécution stricte* (l'automatisation) des prescriptions au *guidage flexible,* Rolland (1998). L'*automatisation* consiste à prescrire en exécutant des éléments du modèle de processus de façon complètement automatique et sans intervention des agents humains. L'*imposition* consiste à imposer un ensemble d'étapes à suivre. L'ensemble des actions permises est contrôlé et est restreint par le modèle. L'*assistance* consiste à assister les ingénieurs en prenant en charge l'invocation d'outils, la communication entre les outils, la circulation des flux de documents entre les agents etc.. Enfin, le *guidage* représente le niveau le plus flexible dans le spectre d'assistance. Il vise à aider les

ingénieurs à satisfaire un but précis. Ce type d'assistance consiste généralement à suggérer un ensemble de solutions, d'étapes ou d'actions à accomplir. Il peut également se traduire par la détection et la suggestion des problèmes en suspend et des solutions éventuelles qui permettent de les résoudre.

4 - SUPPORT DES PROCESSUS STRATEGIQUES

Cette section présente la solution que nous avons adopté avec l'objectif principal d'offrir un support aux processus de nature stratégique. Nous allons décrire les choix que nous avons adopté pour la modélisation, l'implémentation et l'exécution des processus. Notre travail se focalise sur les processus stratégiques compte tenu de leur complexité engendrée par l'aspect non-déterministe qu'ils présentent. Dans le but de pouvoir représenter les processus stratégiques, nous avons opté pour une approche contextuelle qui présente l'avantage par rapport à l'approche orientée décision de coupler la décision à la situation du produit. Le méta-modèle de processus ainsi choisi est appelé NATURE. Ce modèle offre un moyen efficace pour la représentation des processus stratégiques, il s'agit du map. Dans la section suivante, nous détaillons les concepts de base de ce modèle.

4.1 - Le méta-modèle NATURE

NATURE (Novel Approches to Theories Underlying Requirements Engineering) Rolland (1994) est un méta-modèle (une représentation abstraite d'un ensemble de modèles) qui décrit le processus de développement comme des portions de modèles de processus permettant d'assister le concepteur au cours de la progression dans le processus de développement. Dans le cadre de ce méta-modèle, une méthode se base sur un (ou plusieurs) modèlele(s) de produit et un modèle de processus, le modèle de processus sera composé d'une carte (map) et des aides méthodologiques (directives) qui lui sont associées et qui peuvent être de trois types différents : stratégiques, tactiques ou informels.

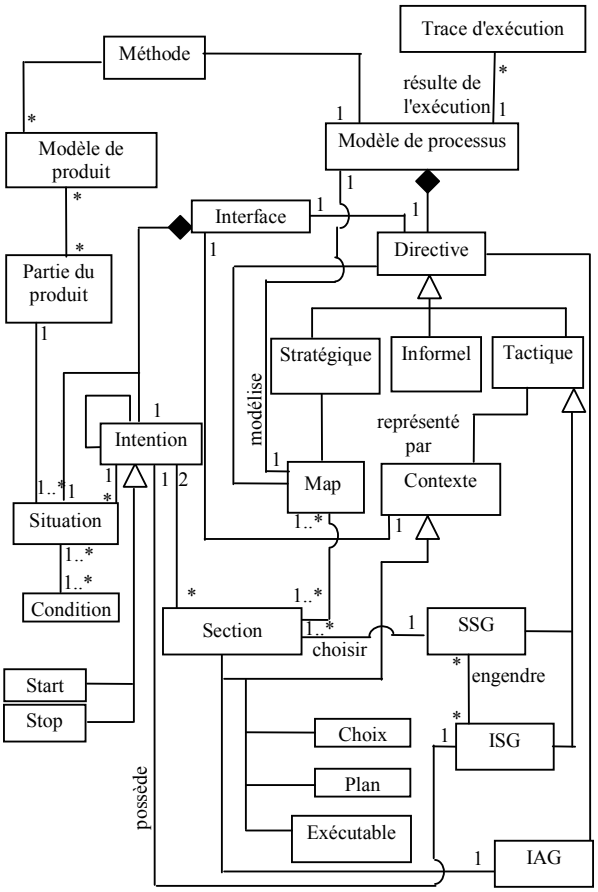

Figure 1. Le méta-modèle NATURE

Le méta-modèle NATURE s'inscrit dans le cadre de l'approche *contextuelle* dans laquelle le processus est vu comme une succession de décisions prises dans les situations spécifiques rencontrées au fur et à mesure du déroulement du processus. Le rôle de ce modèle est de guider cette prise de décisions selon la spécificité des situations rencontrées.
La Figure 1 résume les concepts de base que l'on retrouve dans le méta-modèle NATURE ainsi que les relations entre ces différents concepts. La notation utilisée sur cette figure est celle du langage UML.

4.1.1 - Le concept de contexte

Le concept de base de l'approche NATURE est celui du contexte qui représente l'association d'une situation et d'une décision que le concepteur prend sur cette situation : un contexte est représenté donc par un couple formé d'une situation et d'une intention. *Contexte : <situation, intention>*. La *situation* est constituée de toute partie du produit ou combinaison de parties de produit nécessaire à la satisfaction d'une intention. L'*intention*

permet de capturer l'objectif à atteindre à un stade donné du processus. Ce concept est l'expression d'un état d'avancement désiré sur le produit en cours de construction. Une *intention* est décrite par un nom qui exprime l'objectif à atteindre. Elle peut être *stratégique* ou *tactique* selon le niveau de granularité du processus auquel elle est associée.
On distingue trois types de contextes, Si-Said, (1999) qui sont:
- Les contextes plans : permettent d'implanter les intentions dont l'exécution nécessite un ordonnancement des actions sous forme d'un plan.
- Les contextes choix : permettent de mettre en œuvre une intention dont l'affinement revient à explorer différentes solutions alternatives en se basant sur des arguments de choix.
- Les contextes exécutables : permettent d'implémenter les actions élémentaires qui transforment le produit. Ce type de contextes permet de représenter les processus d'implantation.

La caractéristique majeure de l'approche NATURE est d'identifier une multiplicité de stratégies de prise de décisions en fonction des situations. De cette façon le modèle engendré contient plusieurs modèles de processus qui seront modélisés dans un map (une carte).

4.1.2 - Le concept de map (carte)
Le map est le modèle de processus dans lequel est inclus un ordonnancement non-déterministe *d'intentions* et de *stratégies*. La *stratégie* exprime une *manière de progresser* dans le processus. Son rôle est d'aider à savoir quelle intention choisir et à quelle étape du processus le faire, Rolland (1999). Le map permet de visualiser globalement l'ensemble des *intentions* et les *progressions* possibles entre ces intentions. Il est composé d'un ensemble de *nœuds* représentant des *intentions* et des liens orientés représentant les flux. Les *flux* ont des libellés qui sont les intitulés des *stratégies* de progression qui leur correspondent, Si-Said, (1999).
En effet, le map peut être vu comme étant la totalité des prescriptions possibles de processus, il s'agit alors de suivre une sélection dynamique pour trouver la prescription la mieux adéquate par rapport à la situation actuelle.
L'aspect dynamique est défini à travers l'utilisation de directives (guidelines) qui permettent de sélectionner une stratégie parmi plusieurs. Une *directive* est un ensemble d'indications décrivant la démarche à suivre afin d'atteindre une *intention*. Il ressort de cette définition qu'une directive encapsule l'intention à atteindre avec la procédure à suivre pour l'atteindre, SiSaid (1999). On distingue trois types de directives associées à un map :
- Intention Achievement Guideline (*IAG*) : Pour chaque section, il y a un IAG qui sert à définir les tactiques utilisées pour réaliser l'intention. L'IAG conseille l'ingénieur d'applications sur la manière avec laquelle il peut atteindre son objectif. Un IAG peut être tactique représenté sous forme de contextes, stratégique représenté par un map ou informel représenté par un langage naturel (voir Figure 1). Si un IAG est un guideline tactique alors il sera représenté par un contexte choix ou un contexte plan.
- Intention Selection Guideline (*ISG*) : L'ISG identifie l'ensemble des intentions qui peuvent être réalisées à l'étape suivante et il permet de sélectionner leur IAG et SSG correspondants. Un ISG est un guideline tactique représenté sous forme d'un contexte.
- Strategy Selection Guideline (*SSG*) : Ayant deux intentions, I_i (intention de départ), I_j (Intention d'arrivée) et un ensemble de stratégies S_{ij1}, S_{ij2}, S_{ij3},.., S_{ij} applicables à I_j. Le rôle du SSG est de guider la sélection d'un S_{ijk} puis de choisir l'IAG correspondant. Le SSG est un guideline tactique représenté par un contexte.

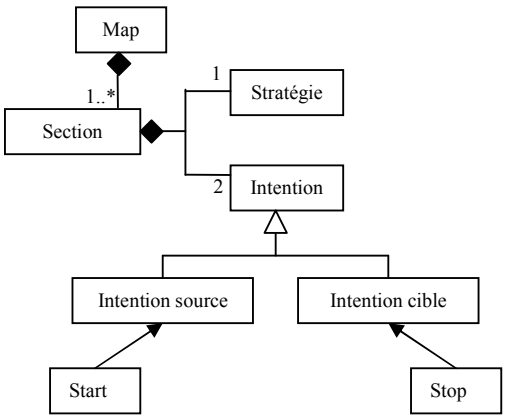

Figure 2. Méta-modèle du map

La carte se représente donc comme un graphe étiqueté et orienté (Figure 2). Elle est constituée d'un ensemble de sections. Chaque section d'une carte est composée d'un nœud source, d'un noeud cible et d'un arc qui les

relie. Les nœuds correspondent aux intentions que l'utilisateur veut réaliser. Les arcs décrivent les stratégies offertes pour progresser d'une intention à une autre. La navigation dans une carte permet d'obtenir un ordonnancement non déterministe d'intentions et de stratégies. La Figure 3 représente un exemple de map où on distingue les différentes sections composant le map ainsi que les guidelines qui lui sont associés.

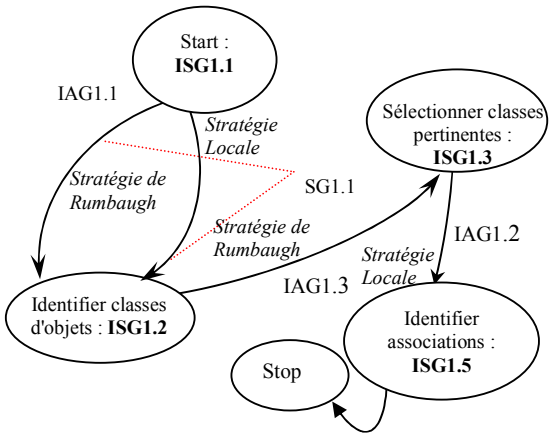

Figure 3. Exemple de map

4.1.3 - L'exécution d'un map

Un map peut être vu comme étant un ensemble de prescriptions de processus à partir duquel, par sélection dynamique, la prescription particulière qui est la mieux adaptée à la situation du produit rencontrée est sélectionnée. Dans ce sens, le map est un multi-modèles qui présente la capacité de modéliser dynamiquement le processus. Ce multi-modèles permet à l'ingénieur d'applications de déterminer un chemin qui lui prescrit une façon pour développer un produit et donc un modèle de processus. A priori, aucun des modèles n'est recommandé et c'est à l'utilisateur de choisir et de construire dynamiquement son chemin selon le contexte dans lequel il se trouve. D'où l'importance d'offrir un guidage à l'utilisateur dans le but de l'assister à chaque étape de l'exécution du modèle de processus et à tous les niveaux de détails de ce dernier. On parle ainsi de guidage de granularité fine ou guidage fin.

La Figure 4 montre qu'un map peut être défini comme étant une abstraction des différentes prescriptions de processus et que la trace de son exécution corresponde à la prescription réellement sélectionnée en fonction de la situation.

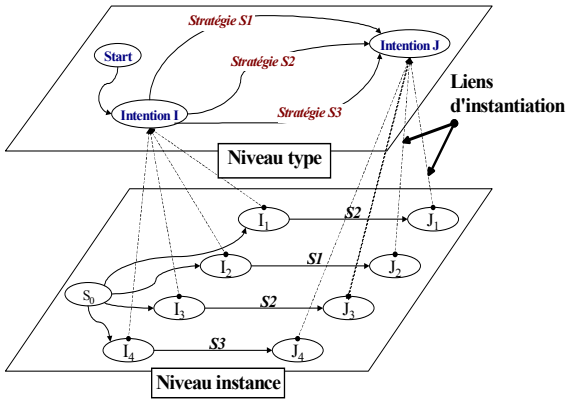

Figure 4. Les niveaux d'abstraction du map

La structure du map présente deux aspects dynamiques -Un map est de *nature* dynamique : il offre plusieurs stratégies pour réaliser des intentions à partir de la situation actuelle du produit (Figure 5A).

- Le *parcours* d'un map est dynamique : il existe plusieurs chemins possibles pour progresser dans un map. Ces chemins sont construits dynamiquement au cours de l'exécution du map (Figure 5B).

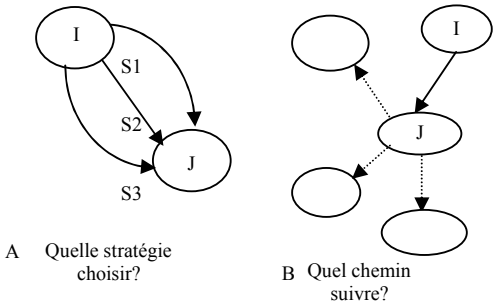

Figure 5. Les aspects dynamiques d'un map

La multiplicité des alternatives dans un map rend son exécution plus difficile que celle d'une suite de tâches ordonnées d'une manière déterministe. En effet le map est une structure navigable qui comporte un ordre non-déterministe d'intentions et de stratégies, ce non-déterminisme croit en fonction du nombre des intentions et celui des stratégies. Le choix entre les différentes intentions et stratégies est à la charge de l'utilisateur et ce en fonction de la situation courante du produit. Par conséquent, le langage d'implémentation d'un map doit prendre en compte le fait que le chemin à suivre au sein du map est entièrement construit par l'utilisateur d'une manière incrémentale lors de l'exécution. Cet aspect dynamique du map ne donne pas pour autant à l'utilisateur la

possibilité de parcourir n'importe quelle portion du map à n'importe quel moment.
Donc, le langage d'implémentation d'un modèle de processus stratégique représenté par un map doit d'une part, permettre l'instanciation d'un multi-modèles qui est le map pour retrouver le modèle réellement exécuté par l'utilisateur et d'une autre part, il doit contrôler le parcours du map exécuté afin de garantir à l'utilisateur un produit final cohérent.

4.2 - Formalisation des processus

Les modèles de processus à implémenter sont crées par instanciation des concepts du méta-modèle NATURE. Pour assurer cette instanciation nous avons choisi d'utiliser le paradigme orienté-objet comme paradigme de modélisation du processus.
Le paradigme ainsi choisi permet de construire des modèles de processus par instanciation des super-classes représentant les concepts du méta-modèle NATURE. Les modèles de processus issus de cette instanciation sont composés de fragments qui peuvent être de types différents : stratégiques, tactiques ou informels. Ces fragments seront à leur tour instanciés par un ECP pour créer des modèles de processus exécutables (voir Figure 6). Ces derniers seront exécutés par le moteur de guidage de l'ECP (le composant de l'ECP assurant l'exécution des processus).

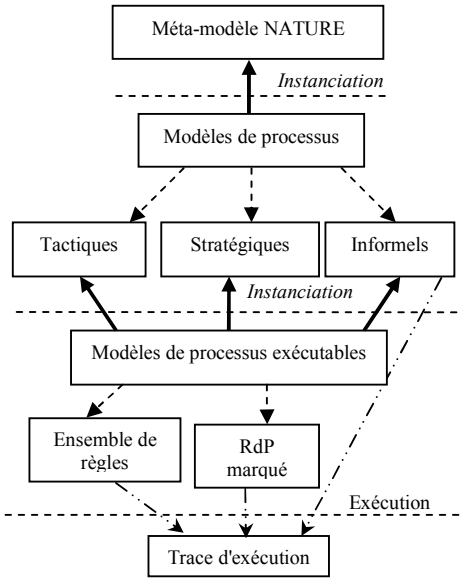

Figure 6. Evolution d'un modèle de processus dans un ECP

Les fragments informels sont décrits en langage naturel et leur exécution est entièrement laissée à la charge de l'utilisateur.
Les fragments tactiques sont représentés par une suite de contextes qui peuvent être de type plan, choix ou exécutable. La situation d'un contexte est représentée par une condition qui peut être élémentaire ou composée par un ensemble de conditions liées entre elles par les opérateurs logiques de la conjonction et la disjonction. L'exécution d'un contexte est réalisée donc en deux étapes : la première consiste à vérifier la condition définie par la situation de ce contexte ce qui revient à vérifier si le produit se trouve effectivement dans la situation du contexte à exécuter ou non. La deuxième étape consiste à réaliser l'intention décrite dans le contexte et ce en fonction du type de ce dernier. Si le contexte en question est de type plan, son exécution revient à exécuter les contextes composant le plan en respectant l'ordre qui figure dans le graphe de précédence qui lui est associé. S'il est de type choix, son exécution revient à choisir l'une des alternatives offertes par le modèle de processus. L'exécution d'un contexte exécutable est la plus simple, car dans ce cas il s'agit de transformer le produit en réalisant l'action associée à ce contexte.

4.2.1 - Les processus tactiques

Pour assurer l'implémentation des directives tactiques au sein du moteur de guidage, nous avons opté pour le paradigme basé-règles. Ainsi chaque contexte sera assimilé à une règle où la précondition correspond à la condition de la situation du contexte, et l'action à exécuter correspond à son intention. Dans ce sens, l'exécution des contextes est assurée par le mécanisme du chaînage avant. En effet, chaque fois que le moteur de guidage reçoit une demande d'exécution d'un guideline tactique représenté par un contexte, la condition de la règle associée à ce contexte sera comparée à la situation actuelle du produit, si elle est vérifiée la règle sera exécutée en réalisant l'intention du contexte. Les règles sont implémentées par des modules de programmes qui permettent de vérifier leurs conditions et d'exécuter les actions définies par ces règles. La Figure 7 représente un exemple de contexte plan ainsi que la règle associée au contexte C2.

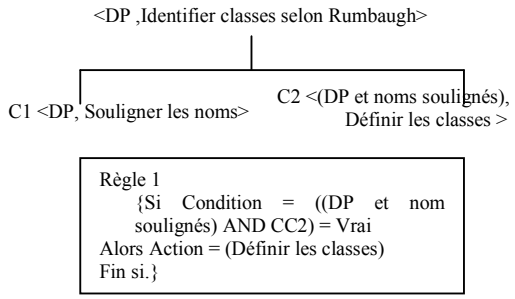

Figure 7. Exemple d'un contexte plan

Le choix du paradigme basé-règle est justifié par le fait qu'il offre un moyen efficace pour prendre en compte l'interaction avec l'utilisateur et ce en réagissant aux évènements déclenchés par ses demandes d'exécution des différentes parties du modèle de processus (Froelich, 1994). D'autre part les règles permettent d'assurer la modularité de la représentation du modèle de processus en associant chaque action à une règle. En conséquence, le modèle sera représenté par un ensemble de règles indépendantes.

4.2.2 - Les processus stratégiques

Le paradigme basé-règle est insuffisant pour répondre à un aspect primordial dans l'exécution d'un modèle de processus qui est l'ordre d'exécution des différentes activités du modèle (Froelich, 1994). L'implémentation de l'ordre d'exécution est d'autant plus importante lorsqu'il s'agit d'un modèle de processus stratégique car dans ce cas il faut déterminer un ordre sur un modèle qui est de nature non déterministe. Pour résoudre ce problème, nous avons proposé un formalisme qui a pour objectif de définir un ensemble de contraintes sur les différentes progressions possibles au sein d'un map. Les contraintes ainsi définies sont décrites par des règles qui doivent être respectées lors du guidage de processus. Le formalisme défini dans notre travail de recherche se base le paradigme orienté-réseaux qui permet d'une part, de représenter explicitement l'ordre d'exécution des activités d'un modèle de processus et d'autre part, il nous permet de représenter l'exécution d'un modèle d'une manière simple en faisant abstraction des détails, Froelich (1994). En effet, le mécanisme d'exécution d'un map est décrit par un réseau de Petri générique où la progression d'un état à un autre est régie par des règles de parcours qui doivent être vérifiées à chaque étape de l'exécution.

A chaque étape de l'exécution d'un modèle de processus, l'utilisateur a besoin de visualiser le modèle notamment dans le cas d'un processus stratégique, et ce pour pouvoir choisir son chemin parmi les différentes alternatives offertes. C'est à ce niveau que nous avons utilisé le paradigme graphique, ainsi les fragments de type stratégique sont représentés par des graphes orientés.

Dans la suite nous présentons les concepts de base utilisés dans le formalisme défini ainsi que les règles de parcours qu'il préconise.

L'exécution de la section <I, S, J> signifie la réalisation de l'intention J en utilisant la stratégie S et en partant de l'intention I.

Le Tableau 1 résume les notations utilisées dans la suite pour décrire les sections d'un map.

Notation	Signification	Signification
<I, S, J>	Une section quelconque dans le map	I, J sont des intentions et S une stratégie.
$<I_i, S, J_j>$	Une section exécutée	La section <I, S, J> est exécutée en partant du résultat de la réalisation de l'intention I et aboutissant à la réalisation de l'intention J. Les produits associés à cette étape sont indiqués par Les indices i et j.
$<I_i, S, J>$	Une section candidate	La section <I, S, J> peut être exécutée dans le futur en partant du résultat de la réalisation de l'intention I.

Tableau 1. Les notations formelles des sections d'un map

Le formalisme défini pour contrôler l'exécution d'un map offre plusieurs avantages, en effet il permet de :

- définir l'ensemble des sections constituant le map,
- identifier les sections qui sont déjà exécutées,
- identifier les sections qui sont candidates à l'exécution, et enfin
- décrire les contraintes qui permettent de contrôler l'exécution des sections.

Lors de l'exécution d'un map, à chaque fois qu'une section est exécutée, un nouvel ensemble de sections candidates (les sections

qui peuvent être exécutées à l'étape suivante) sera généré automatiquement. Pour ce faire, le moteur de guidage est doté d'une pile de sections qui sert à sauvegarder les différentes sections lors d'une session de travail. Cette pile contient les sections déjà exécutées ainsi que les sections candidates. Le fonctionnement de la pile de sections est décrit selon le paradigme orienté-réseaux par un réseau de Petri (RdP) qui regroupe les différents états par lesquels passe une section du map ainsi que les contraintes de parcours qui régissent la progression d'un état vers un autre. Ce RdP est représenté par la Figure 8.

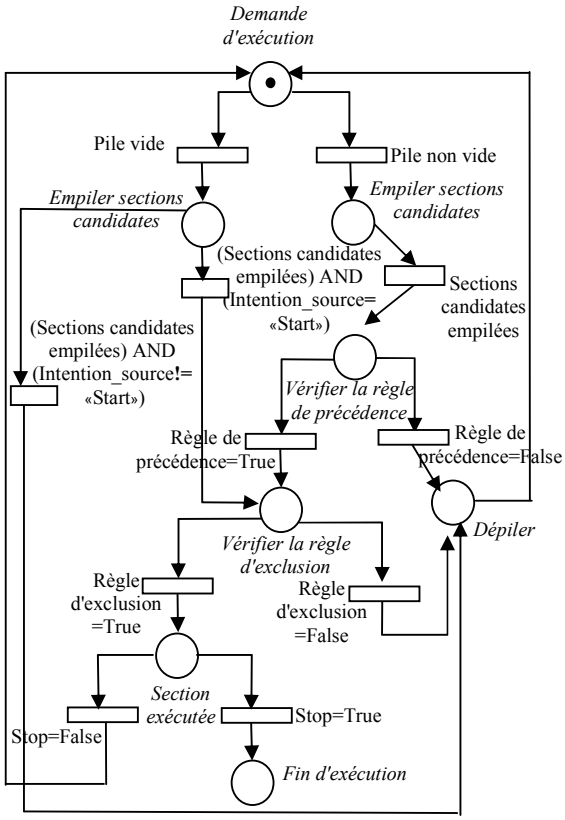

Figure 8. Fonctionnement de la pile de sections

La pile de sections fonctionne comme suit : chaque fois que le moteur de processus de guidage reçoit la demande d'exécution d'une section provenant de l'utilisateur, la pile de sections est automatiquement sollicitée. Si elle est vide, la première section demandée à être exécutée doit avoir pour intention source l'intention «Start», sinon la pile contiendrait toutes les sections déjà exécutées composant ainsi la trace d'exécution du map. Dans les deux cas les sections candidates sont empilées (c'est à dire les sections qui ont la même intention cible et la même intention source que celles de la section demandée à être exécutée). Dans le cas d'une pile vide, nous avons une seule règle qui doit être vérifiée, il s'agit de la règle d'exclusion, mais dans le cas contraire nous avons en plus la règle de précédence qui doit être satisfaite. Ces deux règles seront détaillées dans le paragraphe suivant. Après avoir été exécutée, la section en question fera partie de la trace d'exécution et le moteur de guidage peut ainsi recevoir une autre demande d'exécution ou non selon que l'intention «Stop» est déjà atteinte ou pas.

A chaque instant t de l'exécution, il existe des contraintes qui doivent être vérifiées dans n'importe quel map:
- une section parcourue du map est soit candidate, soit en cours d'exécution, soit exécutée
- l'intention «Start» doit être exécutée au moins une fois.

Ainsi, les différents états que peut avoir une section donnée du map sont les suivants : candidate à l'exécution, en cours d'exécution ou exécutée.
Ces états sont décrits par le graphe de transitions représenté sur la Figure 9.

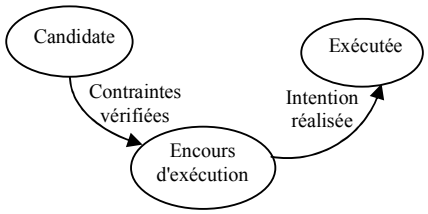

Figure 9. Les états d'une section du map

En plus des deux contraintes déjà décrites précédemment, nous avons définit d'autres règles de parcours qui doivent être respectées lors de la navigation dans le map. Ces règles peuvent être classées comme suit : La règle de précédence et la règle d'exclusion.

La règle de précédence **:** Elle consiste à vérifier un ordre dans le parcours des sections, elle est annoncée comme suit :
Si A=<I, S^1 ,J> et B=<J, S^2, K> sont deux sections du map M alors :
$\forall$ i, j, la section <J_j, S^2, K> (instance de la section B) ne peut être candidate si l'intention J_j n'est pas réalisée par l'exécution de la section A.

Cette règle correspond au cas où l'intention cible de la section A est la même que l'intention source de la section B (voir Figure 10).

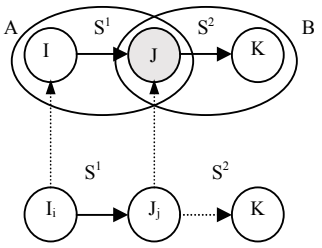

Figure 10. La règle de précédence

La Figure 11 représente un exemple de sections d'un map où la règle de précédence doit être vérifiée. Dans le cas de cette figure, si l'utilisateur demande l'exécution de la section S3 alors le moteur de guidage vérifie si la section S1 est déjà exécutée ou pas.

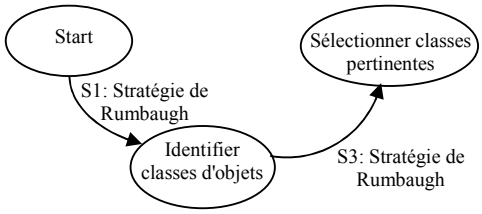

Figure 11. Exemple de la règle de précédence

La règle d'exclusion : Elle consiste à vérifier que des sections sont exécutées d'une manière exclusive dans le temps, elle est annoncée comme suit :
Si A=<I, S^1 ,J> et B=<J, S^2, K> sont deux sections du map M alors :
$\forall$ i, j, la section <J_j, S^2, K> (instance de la section B) ne peut être candidate si l'intention J_j est réalisée par l'exécution de la section A.

Cette règle correspond aux cas suivants :
- Les deux sections A et B ont la même intention source et des intentions cibles différentes (voir Figure 12A).
- Les deux sections A et B ont la même intention cible et des intentions sources différentes (voir Figure 12B).
- Les deux sections A et B ont la même intention source et la même intention cible (voir Figure 12C).

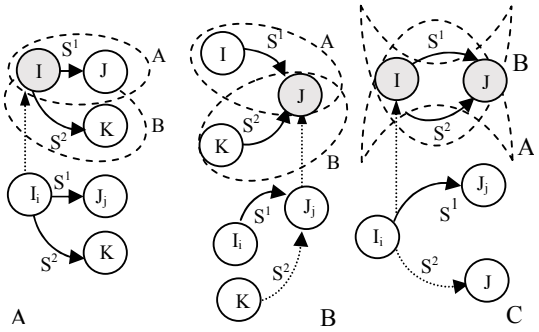

Figure 12. La règle d'exclusion

La Figure 13 représente un exemple de sections où la règle d'exclusion doit être vérifiée. Dans cet exemple, les deux sections S1 et S2 ont la même intention source d'où la nécessite de vérifier la règle d'exclusion avant d'exécuter l'une des deux. Ceci revient à dire que l'utilisateur n'a pas le droit d'exécuter S1 et S2 simultanément.

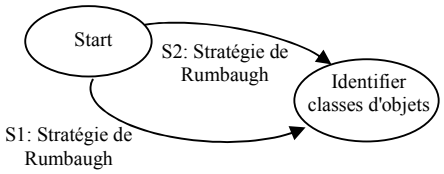

Figure 13. Exemple de la règle d'exclusion

Dans le cadre de notre recherche, un moteur de guidage appelé MPGTool (**M**odelling and **P**rocess **G**uidance **T**ool) a été développé pour tester l'implémentation des processus dans le but d'offrir un guidage flexible et fin à l'utilisateur. Ainsi la solution de support de processus que nous avons élaboré et le formalisme défini pour la modélisation des processus ont pu être testés.

5 - CONCLUSION

Dans ce papier, nous avons abordé le sujet du guidage flexible des processus de nature stratégique. Notre apport réside d'une part, dans la définition d'un formalisme de support des processus stratégiques permettant de régir l'exécution de ces modèles, d'une autre part dans l'implémentation du formalisme ainsi défini au sein d'un moteur de guidage qui supporte l'exécution de tels processus. Nous avons mis l'accent sur les processus de nature stratégique car ils présentent la difficulté majeure dans la phase d'implémentation du processus. En effet les processus stratégiques

sont ceux qui explorent et étudient les différentes solutions alternatives permettant de satisfaire un objectif et qui, éventuellement, produisent un plan pour l'atteindre. Ces processus sont créatifs et critiques aussi bien dans la génération des solutions alternatives que dans le choix de l'alternative appropriée, par conséquent ils sont difficiles à implémenter au niveau d'un environnement logiciel.

Nous avons opté pour un LMP qui se base sur le paradigme orienté-objet. Nous avons défini un formalisme qui permet de contrôler l'exécution des processus stratégiques en se basant sur le paradigme orienté-réseaux et celui basé-règle. Ce même paradigme nous a permis aussi de prendre en compte l'implémentation des processus tactiques et d'implantation en les décrivant sous formes de règles. Par ailleurs, un moteur de guidgae (MPGTool) a été implémeté dans le but de tester le support du guidage des processus stratégiques offert par le LMP défini au cours de notre travail. La visualisation de l'état d'exécution a été aussi considérée car elle représente un aspect primordial dans l'implémentation du guidage des processus qu'ils soient tactiques ou stratégiques. Cette fonctionnalité est assurée en se basant sur le paradigme graphique. Plusieurs directions de recherche peuvent être envisagées pour prolonger ce travail. La première direction consiste à l'extension du LMP défini dans le but de supporter le guidage du processus au niveau méta (guidage du méta-processus), il s'agit d'offrir un guidage à l'ingénieur de méthodes lors de la définition du modèle de processus. Cette extension peut être testée en enrichissant l'outil MPGTool par d'autres modules qui ont pour rôle d'assurer le guidage de l'ingénieur de méthodes. Une autre direction concerne l'amélioration du guidage en rajoutant un mécanisme de simulation au niveau de l'exécution du processus et du méta-processus. Les ingénieurs de méthodes pourraient utiliser ce mécanisme pour évaluer et valider les modèles de processus qu'ils décrivent. Ce même mécanisme, utilisé par les ingénieurs d'applications, permettrait de simuler les effets d'une étape du processus sur le processus en cours. Ceci offrirait une aide supplémentaire durant la construction dynamique du modèle de processus à utiliser. Il permettrait également de mieux gérer les changements dynamiques en offrant le moyen de visualiser les effets avant de provoquer le changement du modèle de processus et de tester l'effet du changement avant de l'appliquer sur le processus en cours. La dernière perspective concerne l'exploitation de la trace d'exécution du processus pour gérer le changement des modèles de processus. En effet nous avons proposé dans MPGTool un module de trace qui sert à garder trace de ce qui s'est réellement déroulé durant le guidage du processus. Cette trace pourrait être exploitée par un outil d'apprentissage dans le but d'améliorer les modèles de processus à partir de la connaissance que l'on peut extraire de la trace des processus.

BIBLIOGRAPHIE

Bandinelli S.C., Fuggetta A., Ghezzi C., (1993), "Software process Model Evolution in the SPADE Environment", In IEEE Transactions on Software Engineering, 19(12).

Barghouti N. et Krishnamurthy B., (1993), "An open environment for process modelling and enactment", Proceedings Of the 8th Int. Software Process Workshop: State of the Practice in Process Technology, Germany.

Curtis B., Kellner M. I., Over J., (1992), "Process Modelling", In the Communications of the ACM, 35(9).

Dowson M., (1993), "Software Process Themes and Issues, IEEE 2nd Int. Conf. on the Software Process, p. 28-40.

Dowson M., Fernstrom C., (1994), "Towards requirements for Enactment Mechanisms", Proceedings of the European Workshop on Software Process Technology, LNCS 772, p.90-106.

Finkelstein A., Kramer J., Nuseibeh B., (1994), "Software Process Modelling and Technology", Advanced software development series, Printed in Great Britain by SRP.

Froelich G., (1994), "Process modelling support in METAVIEW", Master of science thesis, College of Graduate Studies and Research, Université de Saskatchwan, Canada.

Fuggetta A., (1993) "A classification of CASE technology", IEEE computer, p. 25-38.

Garg, P. et Jazayeri M., (1996), "Process Centred Software Engineering Environments", John Wiley and Sons Ltd..

Henderson J.C., Cooprider J.C., (1994), "Dimensions of IS planning and design aids: a functional model of CASE Technology", Information Technology and the Corporation of 1990s, Oxford University Press, Oxford, p. 221-248.

Humphrey W. S., (1989), 2Managing the Software Process", Addison-Wesley.

Kelly S., Lyyttinen K., Rossi M., (1996), "Meta

Edit+: A fully configurable, multi-user and multi-tool CASE and CAME environment", Proceedings. CAiSE'96 Conf., Springer Verlag,

Kobialka HU., (1998), "Implementing Support for Software Process in a Process-Centred Software Engineering Environment", Thèse de doctorat, IAIS (Institute for Autonomous Intelligent Systems), Germany.

Koskinen M., (2000), "Process Metamodelling Conceptual Foundations and Application". Jyvaskyla University Printing house, Jyvaskyla and ER-Paino KY, Lievestuore.

Lonchamp J., (1993), "A structured Conceptual and Terminological Framework for Software Process Engineering", Proceedings of IEEE International Conference on Software Process.

Lyytinen K., (1989), "New Challenges of Systems Development: A Vision of the 90's". DATABASE 20(3), p. 1-12.

McChesney, (1995), "Towards a classification scheme for software process modelling approaches", Information and Software Technology, p 363-374.

Olle T. W., Hagelstein J., Macdonald I. G., Rolland C., Sol H. S., Assche F. J. V., Verrijn-stuart A. A., (1988), "Information systems design methodologies", Addison Wesley, Wokingham, England,

Rolland C., (1993), "Modelling the requirements engineering process", Information Modelling and Knowledge Bases. ISO Press.

Rolland C., Prakash N., (1994), "A Contextual Approach to modelling the Requirements Engineering Process", Proceedings of SEKE'94, 6th International Conference on Software Engineering and Knowledge Engineering, Vilnius, Lithuania.

Rolland C., (1998), "A Comprehensive View of Process Engineering", Proceeding of the 10th International Conference CAiSE'98, LNCS 1413, Springer Verlag Pernici, C. Thanos (Eds), Pisa, Italie, p. 1-24.

Rolland C., Prakash N. et Benjamen A., (1999), "A Multi-Model View of Process Modelling", Requirements Engineering Journal.

Si-Said S., (1999), "Proposition pour la modélisation et le guidage des processus d'analyse des systèmes d'information", Thèse de doctorat, Université Paris-1 La Sorbonne.