

L'ingénierie des méthodes : une visite guidée

Colette Rolland

*Université Paris1 Panthéon Sorbonne CRI, Centre PMF. 90 rue de Tolbiac, 75013 Paris
rolland@univ-paris1.fr*

Résumé

Ce chapitre présente le domaine de l'Ingénierie des Méthodes (IM) et en fait un état de l'art. L'IM se définit comme l'application d'un ensemble de techniques d'ingénierie à la construction du produit que constitue une méthode. L'IM trouve sa justification dans la nécessité de construire une méthode de manière industrielle, systématique et efficace et éventuellement «à la volée», pour l'adapter aux conditions ou exigences particulières d'un projet. Cet article définit l'IM, présente les motivations et les justifications de l'IM et introduit les principes et techniques clés qui la sous-tendent. Il développe certaines des techniques clés et notamment la représentation modulaire des méthodes sous forme de composants de méthodes et la construction de méthode par assemblage de composants.

Mots-clés

Méthode, composant de méthode, modèle de produit, modèle de processus, ingénierie de méthodes

1. Introduction

De plus en plus d'activités de notre société s'appuient sur des systèmes d'information (SI) dont la complexité ne cesse de croître. En conséquence, leur développement devient de plus en plus complexe, coûteux et difficile et il est admis que la maîtrise de leur développement passe par l'emploi de méthodes d'ingénierie. Cependant, si de nombreuses méthodes existent sur le marché, la diversification des domaines dans lesquels on a recours à des SI ainsi que la l'accroissement de leur complexité imposent de nouvelles demandes relatives aux méthodes. En outre, l'analyse de la pratique des méthodes met en évidence des failles et des limites qu'il est nécessaire de combler (Siau 1998), (Stamper 2003), (EMMSAD). Enfin, de nombreuses enquêtes montrent que les méthodes ne sont jamais appliquées comme elles le devraient et qu'elles doivent s'adapter aux conditions de leur usage (Wijers 1990), (Russo 1995).

Ce sont à ces besoins et à ces exigences que tente de répondre l'ingénierie des méthodes (Rossi 2000). Le domaine de l'ingénierie des méthodes (IM) a émergé en réponse à cette sensation croissante que les méthodes ne répondent pas aux besoins de leurs utilisateurs, aux conditions de leur usage et aux critères de qualité qui leur sont imposés. L'IM est donc avant tout une tentative de réponse aux difficultés rencontrées dans la mise en pratique des méthodes. L'IM peut être vue comme une discipline, celle de la conceptualisation, de la construction et de l'adaptation de méthodes qui répondent aux exigences particulières d'une situation d'entreprise ou de projet. Cette perspective implique le dynamisme de la production d'une méthode que l'on doit construire «à la volée». Pour y parvenir, l'IM fait appel à la réutilisation de composants de méthodes issus de méthodes existantes, mis à disposition dans une base de composants de méthodes et accessibles selon une large panoplie de critères de sélection.

Cet article introduit le domaine de l'Ingénierie des Méthodes, détaille les motivations et les justifications de cette discipline et propose quatre principes qui sous-tendent selon l'auteur, les différentes approches de l'ingénierie des méthodes. L'article présente quelques-unes des techniques qui sont au cœur de l'IM, (notamment la représentation modulaire des méthodes qui est un pré-requis de la production de composants de méthodes réutilisables) ainsi que la production d'une nouvelle méthode par composition de composants issus d'autres méthodes. L'article est organisé comme suit. La section 2 est un bref rappel de la notion de méthode d'ingénierie et des éléments qui la composent. La section 3 définit la discipline qu'est l'Ingénierie des Méthodes (IM), présente les motivations et les justifications de cette discipline, propose quatre principes fondateurs ainsi le cycle de l'IM. Les sections 4 et 5 se focalisent sur l'état de l'art des techniques qui sous-tendent les deux étapes principales du cycle: la ré ingénierie des méthodes sous forme modulaire et l'ingénierie des méthodes situationnelles par composition de modules réutilisables. La notion de base de composants de méthodes est proposée à la section 6 suivie enfin des conclusions et des perspectives de l'IM discutées à la section 7.

2. Rappel sur la notion de méthode d'ingénierie de SI

Le terme méthode vient du grec *methodos* qui signifie «moyen d'investigation». Harmsen (1997) détermine ce que sont ces moyens d'investigation : « une collection de procédures, de techniques, de descriptions de produit et d'outils pour le support effectif, efficace et consistant du processus d'ingénierie d'un SI ».

D'autres définitions de la notion de méthode ont été proposées dans (Kronlof, 1993 ; Smolander et al., 1991 ; Seligmann, Wijers et Sol, 1989 ; Prakash, 1997 ; Brinkkemper, 1996). La plupart convergent vers l'idée qu'une méthode apporte les concepts pour décrire le

produit et les règles de conduite méthodologique pour façonner un produit de qualité avec une efficacité raisonnable. Cette acception est synthétisée par Booch (1991) qui définit une méthode comme « un processus rigoureux permettant de générer un ensemble de modèles qui décrit divers aspects d'un logiciel en cours de construction en utilisant une certaine notation bien définie ». En d'autres termes, une méthode traite les deux aspects de l'ingénierie, le produit et le processus, et comporte deux éléments : un ou plusieurs modèles de produit et un ou plusieurs modèles de processus.

2.1 Modèle de produit

Le produit est le résultat d'application d'une méthode. Comme le dit Olle (1992), le produit est la cible désirée d'un développement de SI. Un schéma conceptuel de base de données est un exemple de produit, la base de données d'une application spécifique est un autre exemple. Un produit est exprimé dans les termes d'un modèle de produit.

Le modèle de produit prescrit ce que sont les caractéristiques attendues des produits fabriqués. Il est le moule de production des produits. Une méthode peut comporter plusieurs modèles de produit permettant de modéliser différentes facettes d'un SI (statique, dynamique, fonctionnel) (Olle et al., 1992), à différents niveaux de détail (par exemple : classe, paquetage, sous-système) et à différents niveaux d'abstraction (par exemple : objet, classe, méta-classe). Des modèles pour représenter le contexte organisationnel du SI et les exigences à son égard ont introduit de nouveaux concepts tels que but, acteur, scénario, rôle (Rolland, Souveyet et Ben Achour, 1998 ; Lamsweerde, 2000 ; Yu, 1997). Parallèlement, les travaux dans le domaine de l'ingénierie des besoins ont introduits la distinction entre les besoins fonctionnels et les besoins non fonctionnels (Robertson et Robertson, 1999 ; Chung et al., 2000). Ces extensions du champ modélisation dans l'ingénierie d'un SI ont conduit à une nouvelle typologie des modèles permettant de classer les aspects du SI en fonctionnel, non fonctionnel et intentionnel (Rolland, 1998).

La figure 1 donne l'exemple du modèle de produit de la méthode d'analyse de Coad et Yourdon (1 de 991) présenté selon une liste de cinq éléments types : liste d'objets et de classes, relations entre classes, groupes de classes, attributs et opérations. Tout produit conforme à ce modèle se compose d'un ensemble d'éléments des cinq types identifiés par le modèle.

Modèle de processus :	Modèle de produit :
1. Identifier les objets	1. Liste des objets et des classes
2. Identifier les structures	2. Relations entre les classes
3. Identifier les sujets	3. Groupes de classes
4. Définir les attributs	4. Attributs
5. Définir les services	5. Opérations (services)

Figure1. Exemple des modèles de la méthode Coad et Yourdon (1991)

2.2 Modèle de processus

Le processus est «la route à suivre» pour atteindre la cible que constituent les produits (Olle et al., 1992). Il s'exprime le plus souvent comme un ensemble d'activités reliées entre elles et menées dans le but de définir un produit. Il est exprimé dans les termes d'un modèle de processus. Le texte décrivant la séquence des activités qui ont mené au schéma Entité/Relation d'une application spécifique est un exemple de trace de processus.

Un modèle de processus prescrit une manière de faire, une démarche méthodologique pour atteindre la cible souhaitée. Il décrit à un niveau abstrait et idéal, la façon d'organiser la

production du produit : les étapes, les activités qu'elles comprennent, leur ordonnancement, et parfois les critères pour passer d'une étape à une autre. Il joue le rôle de moule des processus d'ingénierie.

La figure 1 présente le modèle de processus de la méthode Coad et Yourdon (1991) qui est le corollaire du modèle de produit. Il se présente sous la forme d'une séquence de cinq activités-types : Identifier les objets, Identifier les structures, Identifier les sujets, Définir les attributs et Définir les services. Tout processus conforme à ce modèle est une séquence de cinq activités de chacun des cinq types identifiés par le modèle.

D'une part, le modèle de processus n'a de sens que s'il est explicitement mis en relation avec le (ou les) modèle(s) de produit associé(s), mais d'autre part, la qualité du produit dépend fortement de celle du processus mis en œuvre pour l'obtenir. Cependant, jusqu'à la fin des années 80, les concepteurs de méthodes ont concentré leurs travaux sur la définition de modèles de produit capables de prendre en compte de plus en plus d'aspects du monde réel. Depuis le début des années 90, on observe en revanche un déplacement du centre d'intérêt vers la modélisation des processus d'ingénierie.

La première classification des modèles de processus faite par Dowson (1988) comporte trois types : orienté activité, orienté produit, orienté décision.

Les modèles orientés activité se concentrent sur les activités exécutées pour produire un produit et sur leur ordonnancement. Dans cette catégorie, se trouvent les modèles tels que la Cascade (Royce, 1970), la Spirale (Boehm, 1988) et le RUP (Jacobson, Booch et Rumbaugh, 1999).

Les modèles orientés produit couplent l'état du produit à l'activité qui génère cet état. Ils visualisent le processus comme un diagramme de transitions d'état. Le modèle d'Humphrey (1989) et celui des ViewPoints (Finkelstein, Kramer et Goedicke, 1990) appartiennent à cette catégorie.

Les modèles orientés décision perçoivent les transformations successives du produit comme résultats de décisions. Ces modèles mettent l'accent sur la décision à prendre et sur le contexte de délibération autour de la décision à prendre (alternatives, arguments). Les activités ne sont plus au centre du modèle mais apparaissent comme les conséquences des décisions. Le modèle DAIDA (Jarke et Pohl, 1992) ainsi que le modèle IBIS (Potts, 1989) sont classés dans cette catégorie.

Nous ajoutons à cette typologie deux classes : orienté contexte et orienté stratégie.

Les modèles orientés contexte comme celui de la théorie NATURE (Jarke et al., 1999), du projet F3 (Rolland et Prakash, 1994) ou de l'environnement PRIME (Pohl et al., 1999) sont inspirés du paradigme de planification en Intelligence Artificielle utilisé dans GRAPPLE (Huff et Lessor, 1989). Ils définissent le processus par la combinaison de situations observables avec un certain nombre d'intentions (aboutissant à des décisions) spécifiques. Le travail à accomplir est décrit dans le processus comme étant dépendant à la fois de la situation et de l'intention ; en d'autres termes, il dépend du contexte de réalisation.

Le modèle orienté stratégie (Rolland, Prakash et Benjamin, 1999) est une extension des modèles précédents qui vise à représenter plusieurs démarches dans le même modèle de processus. Il est donc multi-démarches et prévoit plusieurs chemins possibles pour élaborer le produit. Il est basé sur les notions d'intention d'ingénierie et de stratégie à suivre pour réaliser ces intentions.

Après ce rappel de la notion de méthode et de ses deux éléments fondateurs, nous présentons le domaine de l'ingénierie des méthodes dans la section suivante.

3. Le domaine de l'ingénierie des méthodes

Le domaine de l'Ingénierie des Méthodes (IM) traite la définition de nouvelles méthodes d'ingénierie des SI. Brinkkemper (1996) définit l'IM comme «une discipline de conceptualisation, de construction et d'adaptation de méthodes, de techniques et d'outils pour le développement des systèmes d'information ».

Plusieurs autres définitions restreignent la notion d'IM à la construction de nouvelles méthodes à partir de celles déjà existantes. Par exemple, Punter et Lemmen (1996) définissent l'IM comme «une approche de construction de méthodes combinant différentes (parties de) méthodes pour développer une solution optimale au regard d'un problème posé». Kumar et Welke (1992) au contraire, proposent une définition plus générale selon laquelle l'IM est « une proposition pour la conception et le développement d'une méta-méthodologie destinée à la conception des méthodes de développement des systèmes d'information ».

Dans ce chapitre, nous définissons l'IM comme la discipline visant à construire et à adapter une méthode de développement de SI aux exigences particulières d'une situation d'entreprise. Notre définition présente l'IM comme une activité répondant à un besoin, celui de construire une méthode adaptée au contexte particulier d'un projet d'entreprise. Elle s'apparente en cela à ce qui a été qualifié par Kumar et Welke (1992) d'ingénierie de méthodes situationnelles qui vise à «adapter une méthode de développement de SI et les outils associés à chacun des projets spécifiques auxquels elle est appliquée».

3.1 Motivations pour l'IM

Une méthode qui a fait ses preuves dans un domaine n'est pas nécessairement adaptée à un autre domaine. Par ailleurs, la situation d'ingénierie de chaque projet de SI pour un domaine donné est spécifique de ce projet, du moins en partie. La notion de méthode universelle (i.e. MERISE) a montré ses limites. Des enquêtes sur l'expérience des méthodes dévoilent (Russo, 1995) que les méthodes ne sont pratiquement jamais utilisées «à la lettre» et que les ingénieurs d'application doivent les adapter aux contextes des projets auxquels ils participent (Hidding, 1994). Il est donc nécessaire de pouvoir adapter une méthode à la situation spécifique du projet auquel on l'applique.

Inversement, il existe un nombre important de méthodes disponibles pour le développement des SI et il n'est pas raisonnable de réinventer une méthode à chaque nouvelle situation ou pour chaque nouveau domaine. La réutilisation de parties de méthodes semble être une solution au besoin d'adaptabilité permettant d'éviter la construction « à partir de rien » (from scratch).

Le caractère universel des méthodes est pénalisant. Une étude de Ernst & Young effectuée sur une période de trois ans sur la pratique des méthodes dans des projets de développement de SI a montré qu'une large partie des 35% des efforts gaspillés dans les projets de SI est due à l'utilisation de méthodes standards de développement. Parkinson (1996) va dans le même sens : «Les méthodologies ont tendance à traiter tous les projets comme si c'était les mêmes, alors qu'en pratique, chaque projet est différent. En traitant tous les projets de la même façon, les méthodologies conduisent les chefs de projet, à créer des plans de travail incluant du travail non nécessaire, ou absent de valeur supplémentaire pour un projet particulier». Ceci suggère qu'une méthode doit être suffisamment flexible pour être adaptée à chaque situation spécifique d'usage.

Une des limites des méthodes classiques est l'insuffisance des démarches qu'elles proposent : celles-ci sont souvent informelles et peu précisément définies. Elles sont souvent trop générales et mal adaptées aux problèmes rencontrés, et difficiles à faire évoluer pour prendre en compte l'expérience résultant de leur utilisation. Elles se bornent pour la plupart, à suggérer une organisation du cycle de vie en étapes globales, et ne permettent pas un guidage

fin des activités de développement. Elles ne prennent pas en compte les connaissances heuristiques accumulées auparavant par les ingénieurs d'application. Pour remédier à cet état de faits, il faut avoir recours à des modèles de processus autres que les prescriptions de simples séquences d'activités.

Le domaine de l'ingénierie des méthodes a donc émergé en réponse à cette sensation croissante que les méthodes ne répondent pas aux besoins de leurs utilisateurs, aux conditions de leur usage et aux critères de qualité qui leur sont imposés. L'IM est donc avant tout une tentative de réponse aux difficultés rencontrées dans la pratique des méthodes. En synthèse, on peut dire que l'IM est justifiée par le double constat suivant:

- la systématisation du développement de SI de plus en plus complexes qui doivent être aussi développé à moindre coût requiert l'usage d'une méthode,
- l'inaptitude des méthodes existantes à répondre aux besoins diversifiés et sans cesse changeants de développement de SI justifie que l'on repense la façon de définir, construire et adapter les méthodes.

3.2 Les Principes de l'IM

Pour répondre aux exigences introduites à la section précédente et en réponse aux limites constatées des méthodes, l'IM met en œuvre quatre principes : la méta modélisation, la réutilisation, la modularité et la flexibilité.

3.2.1 Principe de méta modélisation

La méta modélisation est le principe régissant la description des méthodes. Par méta modélisation, on entend la modélisation des modèles qui composent une méthode. Il y a donc, en général, deux méta modèles interconnectés pour modéliser une méthode : le méta modèle de produit (qui représente le modèle de produit) et le méta modèle de processus (qui représente le modèle de processus). Les liens entre les deux méta modèles modélisent l'impact des activités du processus sur le produit. Ce principe est appliqué par exemple au cas de la méthode Coad et Yourdon introduite à la section 1.1 (figure 2).

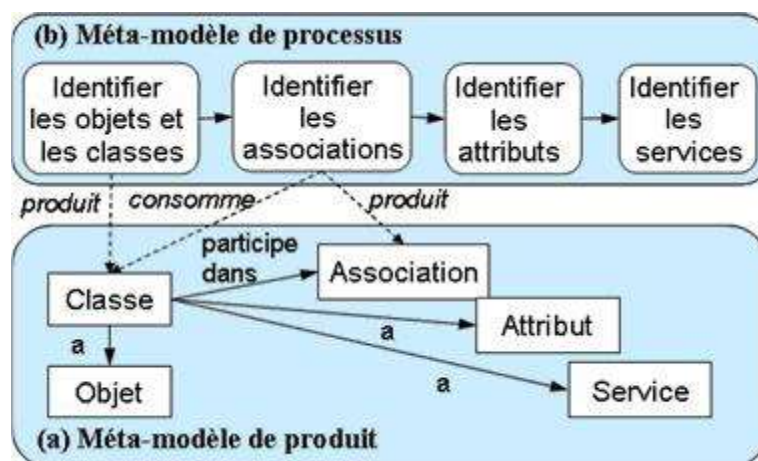


Figure 2. Méta modèles de produit et processus de la méthode de Coad et Yourdon

Le méta modèle de produit est représentée par un schéma de classes UML (figure 2/a) qui met en évidence que le modèle de produit de Coad et Yourdon s'appuie sur :

- cinq concepts, classe, objet, association, attribut et service,
- trois relations « a » qui lient toute classe respectivement aux objets, attributs et aux services qui la composent,

- une relation participe dans qui relie les classes via les associations.

Le méta modèle de processus, présenté par un schéma d'activités UML, montre que la procédure de Coad et Yourdon comporte quatre activités exécutables en séquence pour : identifier les classes et les objets, identifier les attributs, identifier les attributs et identifier les services (figure 2/b). Les liens produit et consomme entre les activités du modèle de processus et les éléments du modèle de produit assurent la relation produit/processus. Ces liens permettent de s'assurer que l'ensemble {modèle de produit, modèle de processus} est complet et cohérent : toute activité du processus doit agir sur un élément du produit et inversement il n'y a pas d'élément du produit qui ne soit manipulé par le processus.

Le principe de méta modélisation est justifié par la nature de l'artéfact que constitue une méthode. En effet, les modèles de produit et de processus qui composent toute méthode sont en essence, des structures d'éléments qu'il est approprié de représenter par un méta modèle.

Notons enfin que ce principe a son inverse, l'instanciation, par laquelle tout produit est une instance du modèle de produit de la méthode pratiquée pour le construire et tout processus est une instance (plus ou moins exacte) de son modèle de processus. En ce sens, on peut dire que les modèles sont des moules de production des instances, les produits comme les processus.

La méta modélisation est non seulement utile à la construction des méthodes mais aussi à la formalisation des méthodes mal définies (Tolvanen et Lyytinen, 1993), à la comparaison des méthodes (Hong et al., 1993 ; Rossi et Brinkkemper, 1996), à la standardisation des méthodes (Booch, Jacobson et Rumbaugh, 1998) et à la définition des liens entre les méthodes d'ingénierie et les langages de programmation.

3.2.2 Principe de réutilisation

La réutilisation en ingénierie des méthodes est inspirée de la réutilisation dans le monde du logiciel où elle est définie comme une approche de développement selon laquelle il est possible de construire un système à partir de composants existants, produits à l'occasion de développements antérieurs. Le principe de réutilisation logicielle est appliqué aujourd'hui à toutes les étapes du cycle de développement d'un logiciel. Initialement introduite pour améliorer la productivité de la programmation, la réutilisation intervient dans les activités d'analyse et de conception ainsi qu'en ingénierie des besoins.

L'IM applique le principe de réutilisation pour construire de nouvelles méthodes d'ingénierie des SI en assemblant différents composants de méthodes qui ont déjà fait leurs preuves. Les composants de méthode sont des descriptions réutilisables des parties des modèles de produit et des modèles de processus qui constituent une méthode, c'est-à-dire des fragments de leurs méta modèles. Ces descriptions constituent les blocs de construction réutilisables qui permettent de définir des méthodes de manière modulaire. Les méthodes ainsi construites sont elles-mêmes modulaires et peuvent être modifiées et étendues facilement.

3.2.3 Principe de modularité

La mise en œuvre de la réutilisation en IM nécessite que l'on divise une méthode en blocs réutilisables et donc d'introduire la modularité comme principe dans la description des méthodes. Selon ce principe, une méthode est vue comme une collection de composants réutilisables. Par analogie avec la notion de module en génie logiciel, un composant de méthode doit avoir un certain nombre de qualités telles que la cohésion, l'autonomie et l'interopérabilité. Il est cohésif à condition que son contenu constitue un tout cohérent et autonome s'il peut être utilisé seul pour résoudre un problème d'ingénierie de système d'information. Il doit être inter-opérable afin de permettre son assemblage avec d'autres composants dans le processus de construction d'une nouvelle méthode.

3.2.4 Principe de flexibilité

Harmsen, Brinkkemper et Oei (1994) proposent un spectre des méthodes d'ingénierie (figure 3) qui organise les approches d'IM selon le degré de flexibilité de la méthode au regard de la situation rencontrée. Les approches sont placées sur une échelle de flexibilité variant d'une flexibilité « faible » à une flexibilité « élevée ».

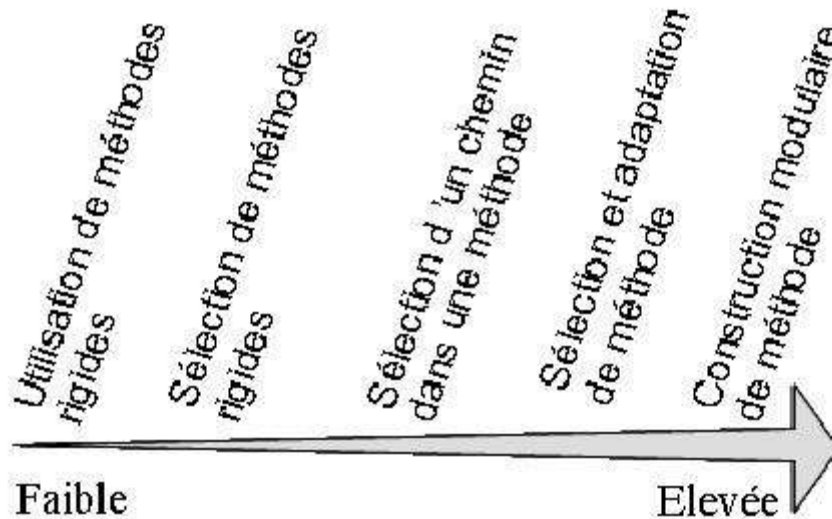


Figure 3. Spectre des approches d'ingénierie des méthodes

Au niveau « faible » de ce spectre se situent les méthodes rigides tandis qu'au niveau « élevée », on situe la construction modulaire de méthode. Les méthodes rigides sont complètement prédéfinies et laissent peu de possibilité pour s'adapter aux situations rencontrées. A l'opposé, les méthodes modulaires peuvent être modifiées et améliorées pour s'adapter à une situation donnée. La sélection de méthodes rigides permet de choisir la méthode la plus adaptée à un projet à partir d'un panel de méthodes rigides prédéfinies, tandis que la sélection d'un chemin dans une méthode consiste à sélectionner le chemin approprié à la situation rencontrée. Enfin, la sélection et l'adaptation d'une méthode permettent à chaque projet de sélectionner des méthodes parmi différentes approches et de les accorder aux besoins du projet.

Le principe de flexibilité est au cœur de l'IM dont la problématique centrale est celle de l'adaptation des méthodes ; adaptation selon l'acception de Harmsen, Brinkkemper et Oei (1994), c'est-à-dire aux contingences d'un projet, adaptation aux besoins spécifiques d'un groupe d'utilisateurs, adaptation dynamique dans le contexte du processus d'IM lui-même. Le principe de flexibilité influence le produit de l'ingénierie, c'est-à-dire la nouvelle méthode mais aussi le processus d'IM. Le spectre d'Harmsen s'attache au premier aspect et conclut que les méthodes modulaires sont les plus flexibles ; le second aspect, implicite dans le spectre de la figure 3, conduit à la conclusion que la construction d'une méthode « à la volée » est la plus flexible.

On comprend que l'application de ces principes milite pour une représentation modulaire des méthodes qu'elles soient existantes ou nouvellement construites « à la volée » en réponse aux besoins des utilisateurs et par assemblage de composants réutilisables.

3.3 Le processus d'IM

Le processus d'IM comporte deux étapes principales : la ré-ingénierie des méthodes existantes sous forme modulaire et l'ingénierie des méthodes situationnelles par composition (figure 4).

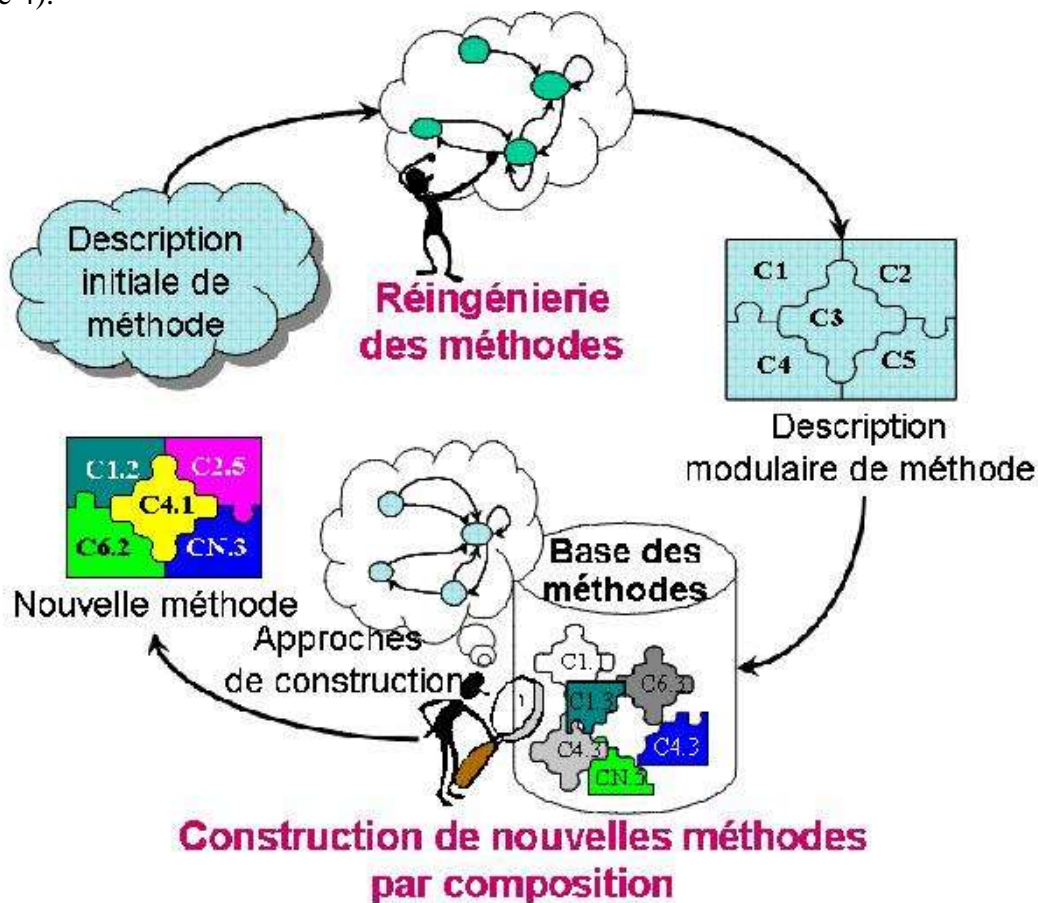


Figure 4. Le cycle d'ingénierie des méthodes

L'étape de ré-ingénierie des méthodes vise à redéfinir les méthodes existantes sous forme de modules réutilisables aussi appelés les composants de méthodes. Ce sont des briques de base pour la construction de nouvelles méthodes ou l'adaptation de méthodes existantes. De plus, une méthode modulaire est plus facile à adapter, compléter ou configurer. L'objectif d'ingénierie de méthodes situationnelles par composition est de proposer des techniques de construction et d'adaptation de méthodes constitués de ces composants réutilisables. Les méthodes obtenues en utilisant ces techniques sont également modulaires.

Les deux étapes sont organisées autour de l'élément central qui est la base des méthodes, en fait la base des composants réutilisables de méthodes. On commente chacun de ces trois aspects dans les trois sections suivantes.

4. Ré-ingénierie des méthodes sous forme modulaire

L'alimentation de la base des méthodes requiert la ré-ingénierie des méthodes existantes dont tout ou partie a fait ses preuves. Elle se fonde sur le découpage modulaire des méthodes en composants réutilisables.

4.1 Notion de composant réutilisable de méthode

Il n'y a pas de définition standard d'un composant de méthode mais selon Brinkkemper et al. (1999), trois dimensions sont à prendre en compte : la perspective, l'abstraction et la granularité. Un composant peut se décliner à différents niveaux de granularité : modèle de produit tout entier (le modèle E/R), heuristique méthodologique fine (enlever la cardinalité zéro dans un schéma E/R) et à différents niveaux d'abstraction (niveau d'un modèle spécifique tels que le modèle E/R, d'un modèle générique tels que le modèle de processus « en fontaine » ou celui d'un méta-méta-modèle tels que le MOF).

Enfin, dans la mesure où une méthode est composée d'un modèle de produit et d'un modèle de processus, trois perspectives peuvent être considérées : celles d'un composant de méthode ' produit ' ou ' processus ' (aussi appelé fragment de produit et fragment de processus) (Brinkkemper et al., 1999 ; Punter et Lemmen, 1996 ; Saeki, 2003 ; Henderson-Sellers, 2000) et celle qui voit un composant comme un mixte des deux (Rolland et Prakash, 1996 ; Ralyté et Rolland, 2001a,b ; Prakash, 1999 ; Wistrand et Karlsson, 2004). Par ailleurs, Van Slooten et Brinkkemper (1993) combinent les fragments de méthodes en itinéraires. Un itinéraire complet représente une méthode de développement de systèmes.

En plus de la notion de composant de méthode, la notion de patron de conception est également utilisée dans l'IM. Les patrons de conception génériques (Rolland et Plihon, 1996 ; Rolland et Prakash, 1996) définissent des règles génériques régissant la construction de méthodes différentes mais similaires tandis que les patrons de conception spécifiques au domaine (Deneckère et Souveyet, 1998) permettent d'étendre les méthodes avec de nouveaux concepts spécifiques.

La figure 5 présente un méta modèle de représentation modulaire des méthodes (Ralyté et Rolland, 2001a). Selon ce méta modèle, une méthode est vue comme un ensemble de composants de différents niveaux de granularité, la méthode elle-même pouvant être un composant du plus haut niveau de granularité. La partie processus du composant est nommée directive. Elle est couplée aux parties de produit nécessaires à l'exécution de la démarche qu'elle préconise. Afin de s'adapter aux différents modèles de processus des méthodes existantes, le méta modèle prévoit trois types de directives : simple, tactique ou stratégique. Une directive simple est atomique, elle ne se décompose pas et propose une ou plusieurs actions à exécuter, manuellement ou avec l'aide d'un outil. Par exemple, le composant de méthode destiné à la découverte des acteurs de SI extrait du modèle des cas d'utilisation (Jacobson et al., 1992) et suggère de manière informelle de poser des questions, telles que : « Quels sont les acteurs que le système est censé aider ? Qui va superviser et maintenir le système ?... ».

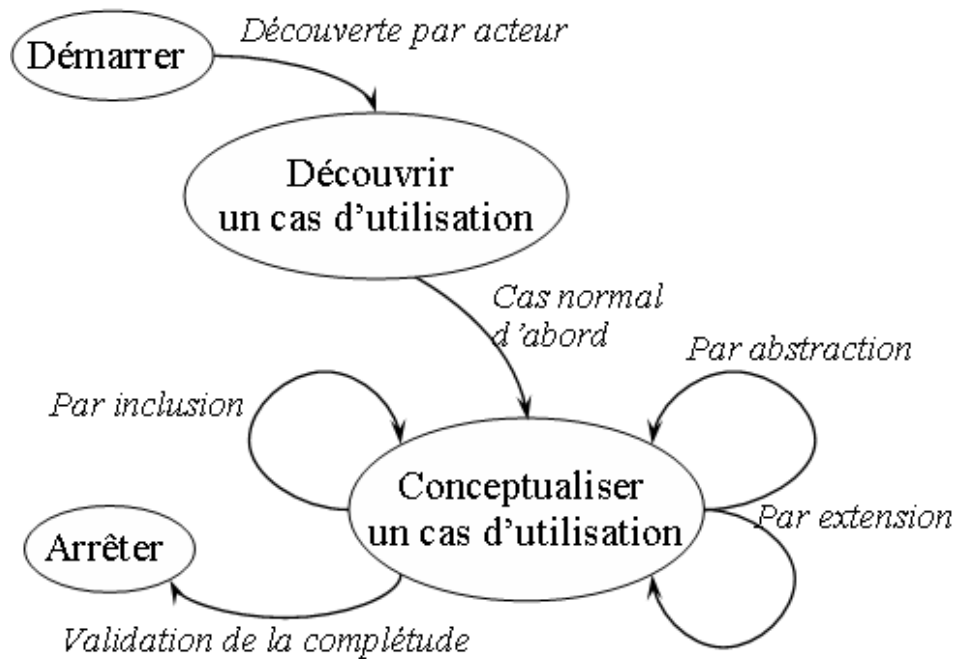


Figure 6. Directive stratégique pour la construction de modèle des cas d'utilisation

Le contexte d'application d'un composant est défini dans sa signature formalisée par un couple <Situation, Intention> (Figure 5). La situation caractérise le point d'entrée dans le processus du composant tandis que l'intention définit l'objectif d'ingénierie qu'il aide à atteindre. Construire un modèle des cas d'utilisation est un exemple d'intention associée au composant de méthode. Description du problème représente une situation nécessaire pour appliquer ce composant.

En outre, un descripteur est associé à chaque composant de méthode (figure 5). Il étend la vue contextuelle capturée dans la signature du composant afin de définir son contexte de réutilisation.

4.2 Processus ré-ingénierie de méthodes

Il y a peu de travaux sur la ré-ingénierie des méthodes sous forme de composants réutilisables. La plupart des approches (Brinkkemper, Saeki et Harmsen, 1999 ; Punter et Lemmen, 1996 ; Song, 1997) considèrent comme réutilisables les différents modèles ou diagrammes de méthodes existantes. Parfois, le niveau de granularité est celui d'un concept (Brinkkemper, Saeki et Harmsen, 1999), d'une propriété, d'un critère, d'une directive, d'une notation ou d'une action (Song, 1997). Toutefois, ces approches n'expriment pas clairement quand, une étape, un modèle, un diagramme ou un concept peut être considéré comme un module réutilisable dans la construction de nouvelles méthodes.

D'après Ralyté et Rolland (2001a), la ré-ingénierie des méthodes sous forme de composants réutilisables est centrée sur la décomposition du modèle de processus de la méthode en directives autonomes et réutilisables en dehors de la méthode d'origine. On trouvera dans BenAyed, (2005) une expérience industrielle de ré-ingénierie basée sur le modèle de représentation modulaire de méthodes précédemment introduit.

4.3 Langages de spécification de composants de méthodes

Même si tout langage de modélisation conceptuelle peut être utilisé pour la spécification des méthodes et des composants de méthodes, leur pouvoir d'expression varie considérablement. Certains langages sont plus adaptés à la spécification des méthodes que d'autres. L'étude des

exigences faite par Marttiin et al. (1995) souligne le besoin de constructions sémantiques spécifiques de la modélisation des méthodes que n'offrent pas les langages standards tels que UML. Dans cette optique, un langage spécifique COCOA été développé et utilisé dans l'environnement MVIEW (Grundy et Venable, 1996) pour la spécification des méthodes.

Harmsen et Saeki (1996) distinguent quatre écoles de langages de représentation et de manipulation de méthodes. La première opte pour une approche orientée données qui met l'accent sur la représentation de l'aspect produit des méthodes. Elle inclue des langages tels que GOPRR (Kelly, Lyytinen et Rossi, 1996), PSM-LISA/D (Hofstede, 1993), les modèles sémantiques de données (Sowa et Zachmen, 1992) et ADSM (Heym et Österle, 1992). Une deuxième école adopte l'approche orientée objet. Les langages appartenant à cette école sont Telos (Mylopoulos et al., 1990), Mataview (Sorenson, Tremblay et McAllister, 1988) et ObjectZ (Saeki et Wen-yin, 1994). La troisième école met l'accent sur l'aspect processus des méthodes et propose des langages tels que les diagrammes de Structure de Tâches (Verhoef et Ter Hofstede, 1995), HFSP (Song et Osterweil, 1992), ALF (Benali et al., 1989) et Merlin (Emmerich et al., 1991). La dernière école propose des langages hybrides, réellement définis pour l'IM tels que MEL (Harmsen et Brinkkemper, 1995).

5. Ingénierie des méthodes par composition

La deuxième étape du cycle d'IM présenté à la figure 4 considère différentes techniques d'IM à base de composants réutilisables issus de la base des méthodes. La plus connue est la technique de construction d'une nouvelle méthode par assemblage de composants. Elle fait l'objet de cette section que l'on termine par une introduction aux outils logiciels de support de l'IM.

5.1 Construction de méthode par composition

On trouve différentes variantes de composition chez les auteurs Brinkkemper et al. (1999), Saeki (2003), Punter et Lemmen (1996), Song (1997) ainsi que Ralyté et Rolland (2001b). Ces techniques consistent à définir les besoins méthodologiques pour une situation en cours, à sélectionner les composants de méthodes satisfaisant ces besoins et à assembler les composants sélectionnés. Deux types d'assemblage, nommés par association et par intégration, sont identifiés dans (Ralyté et Rolland, 2001b). Dans l'assemblage par association, les composants issus de méthodes différentes (M1 et M2) sont disjoints. Ils sont en général complémentaires et l'association consiste à établir des liens entre M1 et M2. Dans l'assemblage par intégration, les composants se recouvrent et la construction de la nouvelle méthode requiert un travail d'assemblage plus complexe qui consiste à intégrer les concepts de M1 à ceux de M2 au moyen d'opérateurs appropriés (Ralyté et Rolland, 2001b).

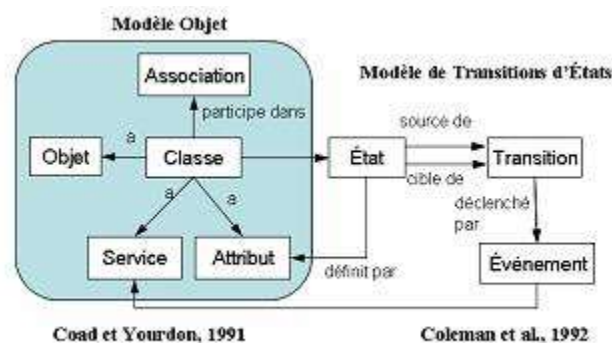


Figure 7. Exemple d'association de composants

L'exemple de la figure 7 illustre l'association des concepts du modèle objet de Coad et Yourdon (1991) à ceux des diagrammes de transitions de Coleman, Hayes et Bear (1992). À l'évidence, l'association enrichit la méthode initiale (Coad et Yourdon, 1991) en lui apportant un jeu de concepts qui permet de compléter la représentation des aspects statiques des objets par leurs aspects dynamiques (Coleman, Hayes et Bear, 1992).

La démarche de construction de modèles des cas d'utilisation proposée par Jacobson et al. (1992) est enrichie par intégration des directives de classification et de rédaction des cas d'utilisation proposées par Cocburn (2001) (figure 8).

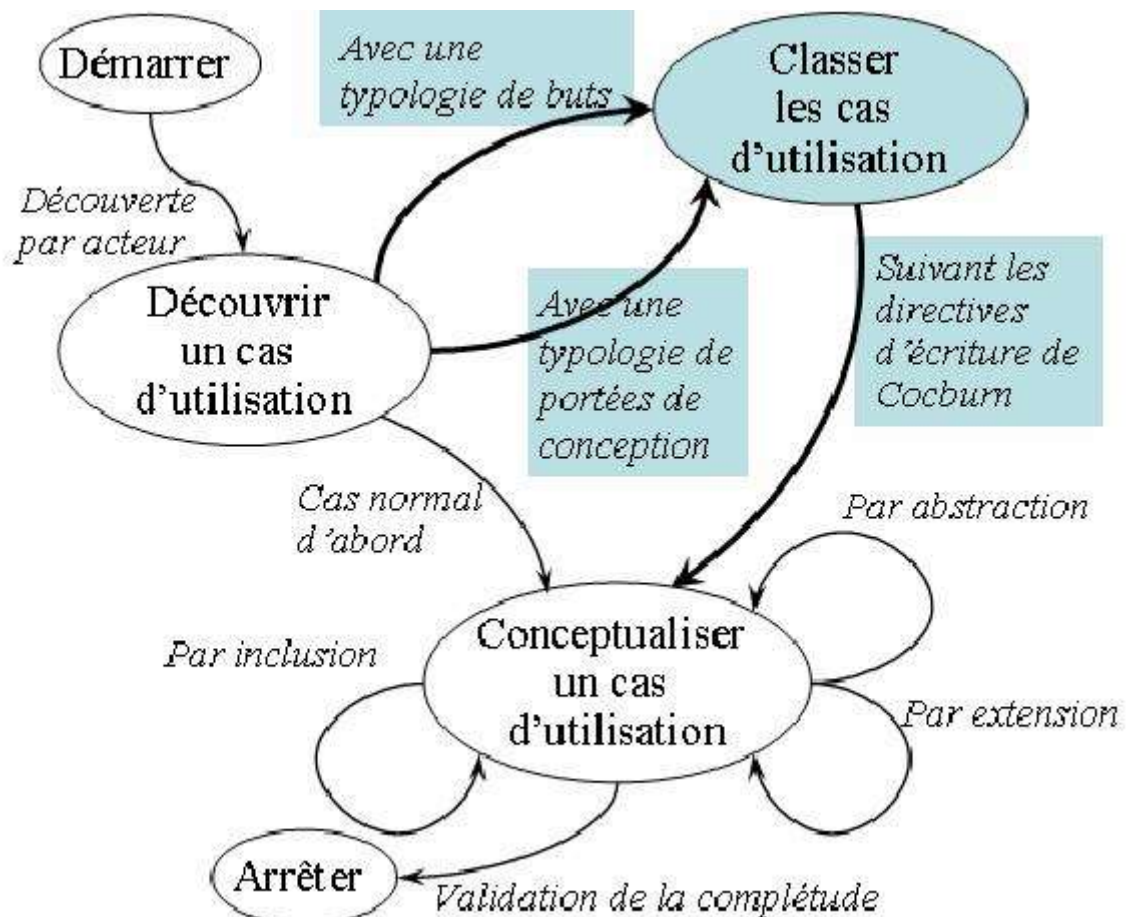


Figure 8. Exemple d'intégration des composants

Par ailleurs, si l'assemblage de composants a été la première approche de construction proposée, d'autres approches ont vu le jour. Sur la base de l'analyse de la littérature en la matière Ralyté, Rolland et Deneckère (2004) proposent une classification des approches d'IM en quatre catégories (figure 9).

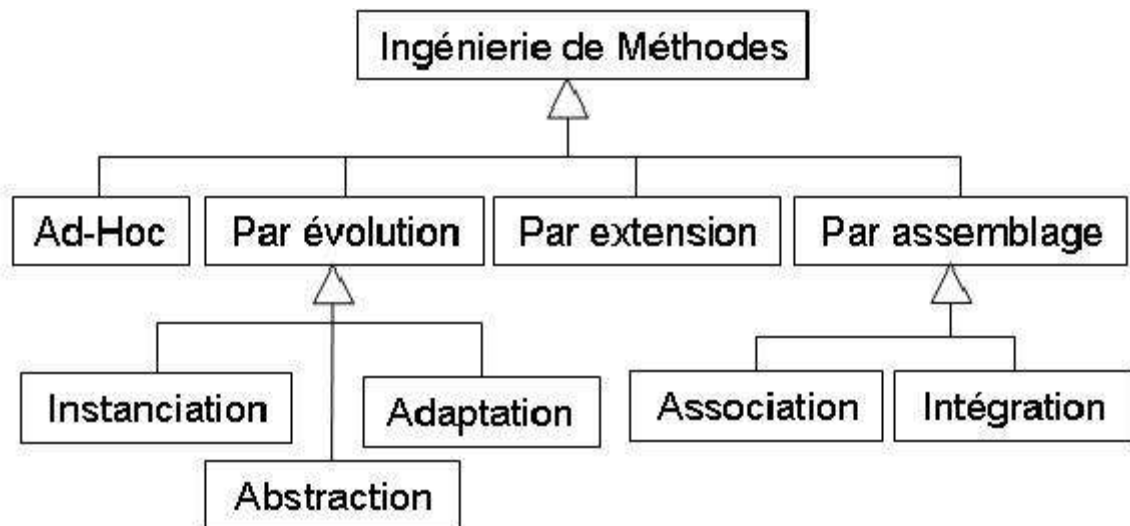


Figure 9. Typologie des approches d'ingénierie de méthodes par composition

Les approches « Ad-Hoc » correspondent à la construction d'une méthode « à partir de rien » et de manière intuitive. Il y a plusieurs raisons qui peuvent conduire à prendre la décision de construire une méthode entièrement nouvelle. L'apparition d'un domaine d'application nouveau est un exemple, la construction d'une méthode basée sur la capitalisation d'expérience en est un autre.

Les approches « Par évolution » (Ralyté, Deneckère et Rolland, 2003) utilisent un modèle ou méta modèle initial (As-Is) comme base d'évolution pour aboutir au modèle souhaité (To-Be) par abstraction (Ralyté, Rolland et Ben Ayed, 2004), instanciation (Gupta et Prakash, 2001) ou adaptation (Tolvanen, 1998) en tenant compte des objectifs d'évolution ou des conditions spécifiques d'un projet.

L'objectif des approches « Par extension » est d'enrichir une méthode existante par de nouveaux concepts et propriétés (Deneckère et Souveyet, 1998) pour la modélisation d'un domaine particulier. Par exemple, une méthode statique telle que celle pour la construction de schémas E/R peut être étendue pour représenter « du temps » de manière systématique par l'introduction d'un calendrier de points temporels, d'intervalles etc. ainsi que celle des aspects temporels tels que l'historisation des entités.

5.2 Support logiciel

Le cycle de l'IM peut difficilement se concevoir sans aide logicielle. L'IM a donc fait émerger une nouvelle génération d'ateliers logiciels désignés par l'acronyme anglais CAME (Computer Aided Method Engineering). Par analogie avec les ateliers de génie logiciel dont le propos est d'apporter une aide à la conduite du développement des SI, un atelier méthode vise à guider la construction des méthodes. Il doit donc offrir des fonctionnalités d'aide à l'accomplissement des activités du cycle de l'IM, soit schématiquement :

- la ré-ingénierie d'une méthode existante ;
- le stockage des composants de méthodes ;
- l'extraction de composants répondant à certains critères ;
- la construction de méthodes selon différentes stratégies (assemblage, adaptation, évolution, « à partir de rien » etc.) ;
- la vérification et la validation de la méthode construite ;
- l'amélioration dynamique de la méthode obtenue.

Harmsen, Brinkkemper et Oei (1994) ont été les premiers à définir les exigences d'un environnement CAME dont certaines ont été implémentées dans le prototype Decamerone (Harmsen et Brinkkemper, 1995). Un certain nombre de prototypes ont vu le jour tels que MetaEdit+ (Kelly, Lyytinen et Rossi, 1996), Meet (Heym et Osterle, 1993), Meru (Prakash, 1999 ; Gupta et Prakash, 2001) et Mentor (Si-said, Grosz et Rolland, 1996). Mais, seul MetaEdit+ est devenu un produit commercial.

6. Base des composants de méthodes

Au cœur du processus d'IM se trouve la base des composants réutilisables de méthodes. Elle capitalise des connaissances sur les méthodes encapsulées dans des composants réutilisables. Un composant peut, par exemple, apporter des heuristiques d'écriture d'un scénario d'interaction (partie processus) et comporter la description conceptuelle d'un tel type de scénario (partie produit). La connaissance qu'il fournit est réutilisable dans tout projet où la capture des besoins se fait par écriture de scénarios.

Toutefois, afin de permettre la réutilisation efficace des composants, la base des méthodes doit comporter aussi des méta connaissances, c'est-à-dire des connaissances sur la connaissance. La méta connaissance peut prendre des formes variées : descripteur (Rolland et Prakash, 1996, Ralyté et Rolland, 2001a), méta classe (Saeki et al., 1993), liens hypertexte (Brinkkemper, 2000). Elle vise à satisfaire deux objectifs, d'une part comprendre la nature de la connaissance apportée par le composant et d'autre part caractériser les situations de sa réutilisation. Dans cette optique, Wistrand et Karlsson (2004) parlent de la vue interne et externe d'un composant de méthode. Van Slooten et Hodes (1996) associe les fragments de méthodes (les composants) à des valeurs d'un ensemble prédéfini de facteurs de contingence, caractéristiques des situations de projets.

Une autre manière de présenter les composants de méthodes pour qu'ils soient accessibles à un large public consiste à les proposer sous forme de bibliothèques électroniques accessibles par Internet. Brinkkemper (2000) souligne que la technologie Internet apporte des nouvelles perspectives dans la création et l'utilisation des méthodes.

Henderson-Sellers (2000) commercialise la base de méthodes OPEN dont le contenu s'enrichit au cours du temps. Les composants sont classés en cinq catégories :

- les produits de travail qui représentent les résultats obtenus lors du processus de développement ;
- les producteurs qui sont responsables de la création, de l'évaluation et de la maintenance des produits de travail ;
- les unités de travail qui définissent les opérations réalisées par les producteurs pour construire les produits de travail ;
- les langages qui sont les moyens de documentation des produits de travail ;
- les étapes qui représentent des périodes dans le processus de développement permettant d'obtenir des résultats.

7. Conclusion et perspectives

La complexité des systèmes d'information et la diversité croissante des domaines d'application des TIC rendent incontournable l'emploi d'une méthode. On a montré dans cet article qu'une méthode d'ingénierie de SI est un guide pour le développement du système d'information. Ce guide apporte d'une part un ensemble de modèles qui sont des moules de représentation du produit SI selon différentes perspectives et à différents niveaux d'abstraction et d'autre part une ou plusieurs démarches, c'est-à-dire des façons de procéder pour aboutir à un produit de qualité. L'article a introduit le domaine de l'ingénierie des

méthodes comme une réponse au besoin de méthodes adaptées aux situations spécifiques des projets de développement et à l'échec des méthodes dites « universelles ».

Bien que le besoin de méthodes spécifiques aux projets auxquelles elles s'appliquent soit clairement identifié, le développement des supports pour la construction de telles méthodes est encore peu avancé. L'article a montré que des résultats significatifs ont été obtenus dans la représentation modulaire des méthodes et dans la construction de méthodes par composition. Il est cependant évident que toute démarche de construction ou d'adaptation de méthodes situationnelles doit être simple, rapide et assistée par des outils logiciels adéquats. Ces outils, les outils CAME permettant de stocker les composants de méthodes et de les assembler en fonction des besoins d'un projet en cours sont à l'état balbutiant.

Finalement, l'émergence perpétuelle de nouveaux domaines d'application de SI renforce la discipline d'IM qui est, plus que jamais d'actualité, tant sur le plan de la recherche que sur celui de la pratique professionnelle.

Références

Benali, K., Boudjlida, N., Charoy, F. Derniame, J. C., Godart, C., Griffiths, Ph., Gruhn, V., Jamart, Ph., Oldfield, D., Oquendo, F. (1989), « Presentation of the ALF project », Proceedings of the International Conference on System Development Environments and Factories.

BenAyed M., (2005) «Une approche pour l'ingénierie d'évolution des méthodes», Thèse de Doctorat de l' Université Paris1Panthéon Sorbonne, Paris, France.

Boehm, B. (1988), « A Spiral Model of Software Development and Enhancement », IEEE Computer, Vol. 21 (5).

Booch, G. (1991), Object Oriented Analysis and Design with Application, Benjamin/Cummings.

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

Brinkkemper, S. (1996), « Method Engineering: Engineering of Information Systems Development Methods and Tools », Information and Software Technology, Vol. 38 (4), pp.275-280.

Brinkkemper, S., Saeki, M. et Harmsen, F. (1999), « Meta-Modelling Based Assembly Techniques for Situational Method Engineering », Information System, Vol. 24 (3), pp. 209-228.

Brinkkemper, S. (2000), « Method Engineering with Web-enabled methods », dans Brinkkemper, S. Lindencrona, E. et Solvberg, A. (Eds.), Information Systems Engineering: State of the Art and Research Themes, Springer, pp. 124-133.

Chung, L., Nixon, B.A, Yu, E. et Mylopoulos, J. (2000), Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers.

Coad, P. et Yourdon, E. (1991), Object-Oriented Analysis, Yourdon Press.

Coleman, D., Hayes, F. et Bear, S. (1992), « Introducing Objectcharts or How to Use Statecharts in Object-Oriented Design », IEEE Transactions Software Engineering, 18(1), pp. 9-18.

Deneckère, R. et Souveyet, C. (1998), « Patterns for Extending an OO Model with Temporal Features », Proceedings of the International Conference on Object Oriented Information Systems (OOIS), Paris, France.

Dowson, M. (1988), « Iteration in the Software Process », Proceedings of the 9th International Conference on Software Engineering (ICSE'98).

- Emmerich, W., Junkermann, G. Peuschel, B. Schäfer, W. et Wolf, S. (1991), « MERLIN: Knowledge based process modeling », 1st European Workshop on Software Process Modeling. Associazione Italiana per l'Informatica ed il Calcolo Automatico, Milan, Italie, pp. 181- 187.
- Finkelstein, A., Kramer, J. et Goedicke, M. (1990), « ViewPoint Oriented Software Development », Actes de Conférence Le Génie Logiciel et ses Applications, Toulouse, pp. 337-351.
- Grundy, J.C. et Venable, J.R. (1996), « Towards an integrated environment for method engineering », Proceedings of the IFIP WG 8.1 Conference on Method Engineering, Chapman and Hall, pp. 45-62.
- Gupta, D. et Prakash, N. (2001), « Engineering Methods from Method Requirements Specifications », Requirements Engineering Journal, Vol.6, pp.135-160.
- Harmsen, A.F., Brinkkemper, S. et Oei, H. (1994), « Situational Method Engineering for Information System Projects » dans Olle T. W. and A. A. Verrijn Stuart (Eds.), Methods and Associated Tools for the Information Systems Life Cycle, Proceedings of the IFIP WG8.1 Working Conference CRIS'94, North-Holland, Amsterdam, pp. 169-194.
- Harmsen, F. et Brinkkemper, S. (1995), « Design and Implementation of a Method Base Management System for Situational CASE Environment », Proceedings of the 2nd APSEC Conference, IEEE Computer Society Press, pp 430-438.
- Harmsen, F. et Saeki, M. (1996), « Comparison of four method engineering languages », IFIP 8.1 Conference on Method Engineering, Chapman and Hall, pp. 209-231.
- Harmsen, A.F. (1997), Situational Method Engineering. Moret Ernst & Young.
- Henderson-Sellers, B. (2000), « The OPEN Framework for Enhancing Productivity », IEEE Software, Vol. 17 (2).
- Heym, M. et Österle, H. (1992), « A Reference Model of Information Systems Development », dans Kendall, K.E., Lyytinen K. et DeGross J.I. (Eds.), The Impact of Computer Supported Technologies on Information Systems Development, Amsterdam, North-Holland, pp. 215-240.
- Heym, M. et Osterle, H. (1993), « Computer-aided Methodology Engineering », Information and Software Technology, Vol. 35 (6/7), June/July, pp. 345-354.
- Hidding, G.J. (1994), « Methodology Information: who uses it and why not? », Proceedings of WITS-94, Vancouver, Canada.
- Hong, S. van der Goor, G. et Brinkkemper, S. (1993), « A Formal Approach to the Comparison of Object-Oriented Analysis and Design Methodologies », Proceedings of the 26th Hawaii International Conference on Systems Science, J. Nunamaker, R. Sprague (Eds.), 4, IEEE Computer Society Press.
- Huff, K. E. et Lessor, V. R. (1989), « A Plan-based Intelligent Assistant that Supports the Software Development Process », Proceeding of the 3rd Software Engineering Symposium on Practical Software Development Environments, Software Engineering Notes, Vol. 13 (5), pp.97-106.
- Humphrey, W. S. (1989), Managing the Software Process, Addison Wesley.
- Jacobson, I., Booch, G. et Rumbaugh, J. (1999), The Unified Process A Software Engineering Process Using the Unified Modelling Language, Addison-Wesley Longman, Incorporated.
- Jarke, M. et Pohl, K. (1992), « Information systems quality and quality information systems ». Proceedings of the IFIP 8.2 Working Conference on the impact of computer-supported techniques on information systems development, Minneapolis, NM.
- Jarke, M., Rolland, C., Sutcliffe, A. et Domges, R. (1999), The NATURE requirements Engineering. Shaker Verlag.
- Kelly, S., Lyytinen, K. et Rossi, M. (1996), « Meta Edit+: A fully configurable, multi-user and multi-tool CASE and CAME environment », Proceedings of the International Conference

- on Information Systems Engineering (CAiSE'96), May, Heraklion, Crete, Greece, Springer-Verlag.
- Kronlof, K. (1993), « Method Integration, Concepts and Case studies ». Wiley series in software based systems, John Wiley & Sons Ltd.
- Kumar, K. et Welke, R.J., (1992), « Methodology engineering: a proposal for situation-specific methodology construction » dans Cotterman, W.W. et Senn J.A. (Eds.), Challenges and Strategies for Research in Systems Development, John Wiley & Sons Ltd, pp. 257-269.
- Lamsweerde, van, A. (2000), « Requirements Engineering in the Year 00: A Research Perspective », Keynote paper, Proceedings of International Conference on Software Engineering (ICSE'2000), ACM Press.
- Marttiin, P., Lyytinen, K., Rossi, M. Tahvanainen, V-P., et Tolvanen, J-P. (1995), « Modeling requirements for future CASE: issues and implementation considerations », Information Resources Management Journal, Vol. 8 (1), pp. 15-25.
- Mylopoulos, J., Borgida, A., Jarke, M. et Koubarakis, M. (1990), « Telos: Representing Knowledge About Information Systems », ACM Transactions on Information Systems, Vol. 8 (4), pp. 325-362.
- Olle, T. W., Hagelstein, J., MacDonald, I.G., Rolland, C., Sol, H.G., Van Assche, F.J.M. et Verrijn-Stuart, A.A. (1992), Information Systems Methodology: a Framework for Understanding, Addison-Wesley.
- Parkinson, (1996), 60 Minute Software-Strategies for Accelerating the Information Systems Delivery Process, John Wiley & Sons, New York, 1996.
- Pohl, K., Weidenhaupt, K., Dömges, R., Haumer, P., Jarke, M. et Klamma, R. (1999), « PRIME - Toward Process-Integrated Modeling Environments », ACM Transactions on Software Engineering and Methodology, Vol. 8 (4), pp. 343-410.
- Potts, C. (1989), « A Generic Model for Representing Design Methods ». Proceedings of the 11th International Conference on Software Engineering (ICSE'89).
- Prakash, N. (1997), « Towards a formal definition of methods », Requirements Engineering Journal, Vol 2 (1).
- Prakash, N. (1999), « On Method Statics and Dynamics », Information Systems, Vol.34 (8), pp 613-637.
- Punter, H.T. et Lemmen, K. (1996), « The MEMA model: Towards a new approach for Method Engineering », Information and Software Technology, Vol. 38 (4), pp.295-305.
- Ralyté, J., Deneckère, R. et Rolland, C. (2003), « Towards a Generic Model for Situational Method Engineering », Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAISE'03), Springer-Verlag, LNCS 2681, pp. 95-110.
- Ralyté, J. et Rolland, C. (2001a), « An Approach for Method Reengineering ». Proceedings of the 20th International Conference on Conceptual Modeling (ER2001), Springer-Verlag, LNCS 2224, pp.471-484.
- Ralyté, J. et Rolland, C. (2001b). « An Assembly Process Model for Method Engineering », Proceedings of the 13th Conference on Advanced Information Systems Engineering (CAISE'01),Springer-Verlag, LNCS 2068, pp. 267-283.
- Ralyté, J., Rolland, C. et Ben Ayed, M. (2004), « An Approach for Evolution Driven Method Engineering », dans Krogstie, J., Halpin, T. et Siau, K. (Eds.) Information Modeling Methods and Methodologies, IDEA Group.
- Ralyté, J., Rolland, C. et Deneckère, R. (2004), « Towards a Meta-Tool for Change-Centric Method Engineering: a Typology of Generic Operators », Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAISE'04), Springer-Verlag, LNCS 3084, pp.202-218.
- Robertson, S. et Robertson, J. (1999), Mastering the Requirements Process, Addison-Wesley Professional edition.

- Rolland, C. (1998), « A Comprehensive View of Process Engineering », Proceedings of the 10th International Conference on Advanced Information Systems Engineering (CAISE'98), Pisa, Italy, Springer-Verlag, LNCS 1413.
- Rolland, C. et Plihon, V. (1996), « Using Generic Chunks to Generate Process Models Fragments », Proceedings of 2nd IEEE International Conference on Requirements Engineering (ICRE'96), Colorado Spring.
- Rolland, C. et Prakash, N. (1994), « A Contextual Approach to modelling the Requirements Engineering Process », Proceedings of the 6th International Conference on Software Engineering and Knowledge Engineering (SEKE'94), Vilnius, Lithuania.
- Rolland, C. et Prakash, N. (1996), « A Proposal for Context-Specific Method Engineering », Proceedings of the IFIP WG 8.1 Conference on Method Engineering, Atlanta, Georgia, USA, Chapman and Hall, pp 191-208.
- Rolland, C., Prakash, N. et Benjamin, A. (1999), « A Multi-model View of Process Modelling », Requirements Engineering Journal, pp. 169-187.
- Rolland, C., Souveyet, C. et Ben Achour, C. (1998), « Guiding Goal Modelling Using Scenarios », IEEE Transactions on Software Engineering, Special Issue on Scenario Management, Vol. 24 (12), pp. 1055-1071.
- Rossi, M. et Brinkkemper, S. (1996), « Complexity Matrix for Systems Development Methods and Techniques ». Information Systems, 21 (2), pp. 209-227.
- Royce, W. W. (1970), « Managing the development of large software systems », Proceedings of the IEEE WESCON, August.
- Russo et al, (1995), « The Use and Adaptation of System Development Methodologies », Proceeding of the International Resources Management. Association (IRMA) Conference, Atlanta.
- Saeki, M., Iguchi, K., Wen-yin, K., Shinohara, M. (1993), « A Meta-model for Representing Software Specification & Design Methods », Proceeding of the IFIP'WG8.1 Conference on Information Systems Development Process, Come, pp. 149-166.
- Saeki, M. et Wen-yin, K. (1994), « Specifying Software Specification and Design Methods », Proceedings of Conference on Advanced Information Systems Engineering (CAISE'94), Springer-Verlag, LNCS 811, pp. 353-366.
- Saeki, M. (2003), « Embedding Metrics into Information Systems Development Methods: An Application of Method Engineering Technique », Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAISE'03), Velden, Austria, Springer-Verlag, LNCS 2681, pp. 374-389.
- Si-said, S., Grosz, G. et Rolland, C. (1996), « Mentor, A computer Aided Requirements Engineering Environment », Proceedings of the 8th International Conference on Information Systems Engineering (CAISE'96), May, Heraklion, Crete, Greece, Springer-Verlag.
- Seligmann, P. S., Wijers, G. M. et Sol, H. G. (1989), « Analyzing the Structure of I. S. methodologies, An Alternative Approach », Proceeding of the 1st Dutch Conference on Information Systems, The Netherlands.
- Smolander, K., Lyytinen, K., Tahvanainen, V. et Marttiin, P. (1991), « MetaEdit – A Flexible Graphical Environment for Methodology Modelling », Proceedings of the 3rd International Conference in Advanced Information Systems Engineering (CAISE'91), Trondheim, Norvège, May.
- Song, X. et Osterweil, L.J. (1992), « Towards objective, systematic, design-method comparison », IEEE Software, Vol. 34 (5), May, pp. 43-53.
- Song, X. (1997), « Systematic Integration of Design Methods », IEEE Software.
- Sorenson, P.G. Tremblay, J.P. et McAllister, A.J. (1988), « The Metaview System for Many Specification Environments », IEEE Software, pp. 30-38.

- Sowa, J.F. et Zachmen, J.A. (1992), « Extending and Formalising the Framework for Information Systems Architecture », *IBM Systems Journal*, Vol. 31 (3), pp. 590-616.
- Tolvanen, J.-P. (1998), *Incremental Mehtod Engineering with Modeling Tools: Theoretical Principles and Empirical Evidence*, PhD Dissertation, University of Jyväskylä, Finland.
- Tolvanen, J.P. et Lyytinen, K. (1993), « Flexible method adaptation in CASE environements – The metamodeling approach », *Skandinavian Jurnal of Information Systems*, Vol. 5 (1), pp. 51-77.
- Van Slooten, K. et Hodes, B. (1996), « Characterising IS development project », *Proceedings of the IFIP WG 8.1 Conference on Method Engineering*, Chapman and Hall, pp. 29-44.
- Van Slooten, K et Brinkkemper, S. (1993), « A Method Engineering Approach to Information Systems Developmen », dans Prakash, N., Rolland, C. et Pernici, B. (Eds.), *Information Systems Development Process*, Elsevier Science Publishers B.V. (North-Holand), Amsterdam.
- Verhoef, T.F. et Ter Hofstede, A.H.M. (1995), « Feasibility of Flexible Information Modelling Support », dans Iivari, J., Lyytinen K., et Rossi, M. (Eds.), *Advanced Information Systems Engineering*, Springer-Verlag, pp. 168-185.
- Wistrand, K. et Karlsson, F. (2004), « Method Components – Rationale Revealed », *Proceedings of the 16th International Conference on Advanced Information System Engineering (CAISE'04)*, Riga, Latvia, June, Springer-Verlag, LNCS 3084, pp. 189-201.
- Yu, E. (1997), « Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering », *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97)*, Washington D.C., USA. pp. 226-235