



HAL
open science

Une Bibliothèque de Fragments de Méthodes pour la Construction de Méthodes Situationnelles

Sihem Ben Sassi, Henda Hadjami Ben Ghézala, Colette Rolland

► **To cite this version:**

Sihem Ben Sassi, Henda Hadjami Ben Ghézala, Colette Rolland. Une Bibliothèque de Fragments de Méthodes pour la Construction de Méthodes Situationnelles. Génie Logiciel et Ingénierie de Systèmes et leurs Applications, 2000, France. pp.1-10. hal-00707548

HAL Id: hal-00707548

<https://hal.science/hal-00707548>

Submitted on 16 Jun 2012

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.

Une Bibliothèque de Fragments de Méthodes pour la Construction de Méthodes Situationnelles

Sihem Ben Sassi*, Henda Hadjami Ben Ghézala*, Colette Rolland**

*Ecole Nationale des Sciences de l'Informatique
Campus Universitaire la Manouba
1025 Manouba, Tunisie
Tél : +216 1 600 444 – Télécopie : +216 1 600 449

**Université de Paris 1 Sorbonne
90 Rue de Tolbiac, 75013, Paris, France.

Mél : Sihem.BenSassi@ensi.rnu.tn
Henda.BenGhezala@isd.rnu.tn
Rolland@univ-paris1.fr

Résumé :

La pratique de l'utilisation des méthodes montre que ces dernières ne sont presque jamais suivies convenablement du fait qu'il n'existe aucune méthode qui satisfasse toutes les situations. Développer une nouvelle méthode pour toute nouvelle situation n'est pas judicieux puisque le coût, en terme de temps, sera très cher. L'approche situationnelle vise à définir des méthodes qui sont construites par assemblage de fragments de méthodes déjà existantes de façon à bien s'adapter à la situation du projet et à gagner en temps de développement. Parmi les approches proposées pour supporter la construction des méthodes situationnelles celle de Colette Rolland [6], qui permet de construire et d'exécuter dynamiquement et d'une manière flexible les modèles de processus. Cette approche est basée sur le concept de carte et sur les aides méthodologiques qui lui sont associées.

Notre travail consiste en le développement d'une bibliothèque de fragments de méthodes aidant dans la construction de méthodes situationnelles et mettant en œuvre l'approche en question. Cette bibliothèque fournit une base où sont décrites les deux facettes des fragments : produits et processus, ainsi que trois modules permettant de gérer la base, de consulter son contenu et de rechercher des fragments selon des critères bien déterminés.

Mots clés : ingénierie des méthodes, méthodes situationnelles, fragments de méthodes, descripteur de fragment, sélection de fragments, recherche de fragments.

Abstract:

The methods engineering practice shows that methods are not conveniently applied because there is no method that fits all situations. Developing a new method for each new situation will not be judicious, it needs a lot of time. The situational approach aims at defining information systems development methods by reusing and assembling different existing method fragments. Among the proposed approaches for situational methods construction, the one proposed by Rolland [6]. This approach allows the construction and the dynamical execution of the process model in a flexible manner. It is based on the map concept and its associated guidelines. Our contribution consists on developing fragments methods repository supporting the construction of situational methods based on Rolland's approach. This repository provides a base where are described the two fragments facets: products and process, as well as three modules that manage the base, consult it and search for the fragments according to defined criteria.

Keywords: *Methods engineering, situational methods, method fragment, fragment descriptor, fragment selection, fragment search.*

1. INTRODUCTION

L'utilisation d'une méthode durant le cycle de vie du développement logiciel permet de structurer le travail, de guider le développeur et de faciliter et organiser son travail. Mais en réalité, la pratique d'utilisation d'une méthode dépend de la situation du projet. En effet, dans certaines situations, certaines étapes de la méthode peuvent ne pas être nécessaires à exécuter, d'autres ne sont pas convenablement suivies, etc... Pour cela, une nouvelle stratégie de développement de nouvelles méthodes, dites *situationnelles*, répondant plus aux spécificités du projet est proposée.

Une méthode est définie par ses produits livrables et son processus représentant les étapes ou les activités à réaliser. Elle peut être vue comme une collection de fragments, où chaque fragment peut être selon [1] un fragment produit s'il représente une partie du produit ou un fragment processus s'il représente une partie du processus, et selon [2] une description intégrant à la fois une partie de produit et la partie de processus permettant de la réaliser. L'approche situationnelle [3] se base sur les fragments de méthodes comme bloc de construction de base en définissant des méthodes de développement de systèmes d'information par réutilisation et assemblage des fragments de méthodes déjà existantes permettant de construire ainsi les nouvelles méthodes situationnelles, de façon modulaire, plus appropriées et adaptées à la situation du projet [4].

Dans ce sens, plusieurs approches d'assemblage de méthodes sont proposées. Harmsen [1] présente l'assemblage de méthodes par des relations entre des types différents de fragments de méthodes; cet assemblage doit être régi par un certain nombre de règles afin d'assurer l'exactitude et la complétude de la méthode construite. L'approche Euromethod [5] tourne autour de la formalisation des méthodes de manière à pouvoir identifier les fragments et les caractériser pour faciliter leurs usages dans des contextes situationnels; cette formalisation est souvent faite au moyen de méta-modèles. Ces approches ne sont pas assez souples pour aboutir à une méthode qui s'adapte bien à toute situation, quelque soit son genre; par exemple Euromethod est classée comme semi-rigide et ne s'applique qu'aux projets faisant intervenir des clients et des fournisseurs. C'est pourquoi Rolland [6] a proposé une autre approche, que nous adoptons, permettant de construire et d'exécuter dynamiquement et d'une manière flexible les modèles de processus. Cette approche se base d'une part sur le concept de carte, qui est un graphe orienté d'intentions et étiqueté par des stratégies, et d'autre part sur les aides méthodologiques qui lui sont associées. Une intention capture la notion de tâche que l'ingénieur application projette de réaliser tandis que la stratégie est la manière avec laquelle l'intention peut être accomplie. La carte contient un groupe de prescriptions de processus à partir desquels, par sélection dynamique, le plus approprié à la situation courante est sélectionné. Elle est, par conséquent, considérée comme un multi-modèle avec des capacités de modélisation dynamique du processus [7] s'adaptant bien à la construction de méthodes situationnelles.

Dans ce travail, nous nous sommes proposés de développer un environnement mettant en œuvre les concepts proposés par Rolland pour supporter la construction de méthodes situationnelles, en mettant à la disposition de l'ingénieur d'application une base de méthodes de nature différentes fragmentées, ainsi qu'un outil de consultation et de recherche pour pouvoir sélectionner le ou les fragments répondant à la situation en cours.

Le reste de l'article est organisé globalement en deux parties; la première partie, structurée en trois sections, s'intéresse aux concepts de la bibliothèque : la présentation des fragments et la manière de les décrire. La première section est dédiée au méta-modèle du fragment ; la deuxième section est consacrée à la carte et finalement la troisième section présente le descripteur du fragment. Dans la seconde partie, nous présentons l'environnement que nous avons développé pour supporter la recherche des fragments pour la construction de méthodes situationnelles. Cet environnement est détaillé dans trois sections, chacune présente un module: alimentation de la base, le générateur de requêtes et l'outil de consultation. Cet article est clôturé par une conclusion qui résume les principaux volets de notre travail.

2. LES CONCEPTS DE LA BIBLIOTHEQUE

La bibliothèque de fragments de méthodes, dont l'objectif est d'apporter une aide dans la construction de méthodes situationnelles, s'articule autour d'une base où sont décrites dans un même descripteur les deux facettes des fragments : produits et processus. A un fragment, est associée une aide méthodologique qui peut nécessiter une formalisation à l'aide d'une carte. Cette dernière permet à l'ingénieur d'application d'atteindre un but parmi plusieurs, étant donnée la situation dans laquelle il se trouve. La description des fragments, le concept de carte et de descripteur seront détaillés dans les prochaines sections ; dans cette description nous utilisons le formalisme Entité/Association étendu, où un rectangle représente une entité et un lien étiqueté représente une relation.

2-2 Le Méta-Modèle du fragment:

Dans ce qui suit, nous présentons le méta-modèle du fragment. Ce dernier, illustré par la figure 1 ci-dessous, montre que chaque fragment comprend un modèle de produit et un modèle de processus. Le modèle de produit

représente la classe de produits obtenus comme résultats de l'utilisation du fragment ; tandis que le modèle de processus représente le processus de développement du produit et est supporté par des aides méthodologiques. Une aide méthodologique inclut des connaissances méthodologiques pour aider l'ingénieur application dans la réalisation d'une intention pour une situation donnée. Elle est formée d'une interface et d'un corps. L'interface est définie par un couple <situation, intention> qui caractérise les conditions de son applicabilité : la situation courante qui est l'entrée pour le fragment, et l'intention ou le but que le fragment réalise. Le corps de l'aide méthodologique détaille comment appliquer le fragment pour réaliser l'intention. Il peut être représenté graphiquement ou décrit informellement selon le type de l'aide : stratégique, tactique ou informelle. Les aides méthodologiques informelles fournissent des suppositions générales et des explications informelles décrivant comment procéder pour obtenir le produit visé.

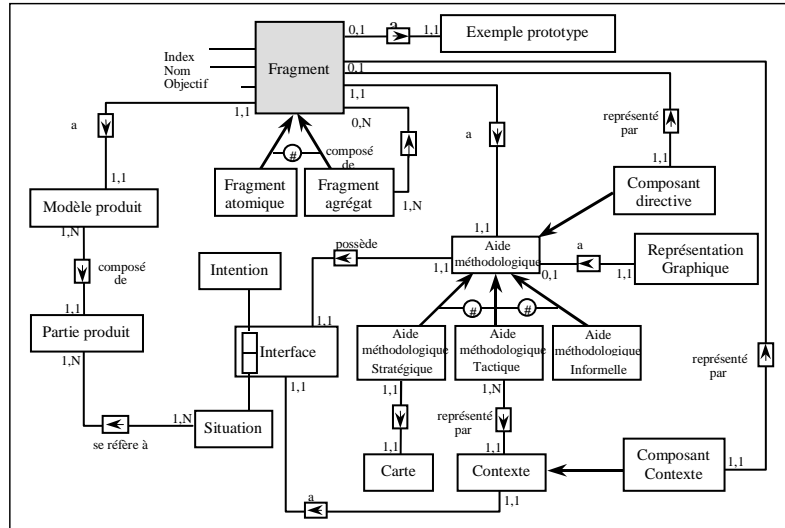


Figure 1 : le méta-modèle du fragment de méthode.

Les aides méthodologiques tactiques proposent un processus expérimenté pour produire le produit correspondant. Ils sont représentés par un arbre de contextes selon le formalisme du modèle de processus NATURE [8] [9]. Enfin, les aides méthodologiques stratégiques fournissent une vue stratégique du processus de développement indiquant quelle intention peut être accomplie en suivant quelle stratégie; elle est représentée par une carte, objet de la section suivante.

2-3 La carte:

La carte, à laquelle est associée une aide méthodologique stratégique, est un modèle de processus défini par un graphe orienté et étiqueté avec les intentions comme nœuds et les stratégies comme liens entre les intentions [7]. Comme le montre la figure 2, une carte consiste en un nombre de sections chacune est composée d'une intention source, une intention but et une stratégie décrivant comment atteindre l'intention but à partir de la source. Il y a deux intentions particulières appelées *Début* et *Fin* qui représentent respectivement l'intention qui permet de commencer la navigation dans la carte et l'intention qui arrête la navigation.

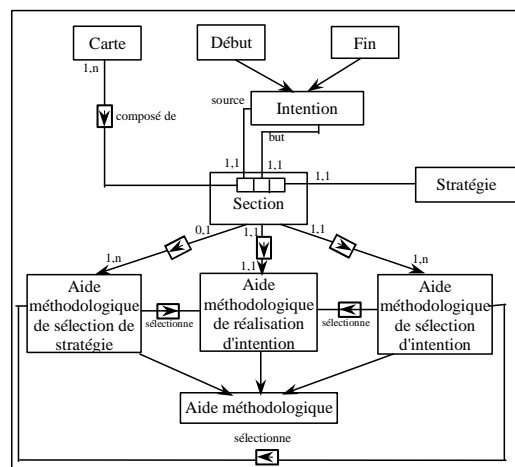


Figure 2 : La carte et ses relations avec les aides méthodologiques.

Une intention est exprimée comme une formulation en langage naturel comprenant un *verbe* et plusieurs paramètres [10] [6]. Chaque paramètre joue un rôle différent par rapport au verbe. La structure navigationnelle de la carte permet à l'ingénieur application de déterminer le chemin entre les deux intentions *Début* et *Fin*. Elle contient un nombre fini de chemins, chacun prescrit une manière de développer le produit.

A chaque carte, comme illustré dans la figure 2, un ensemble d'aides méthodologiques est associé pour guider l'ingénieur application à atteindre une intention pour une situation donnée.

Les aides méthodologiques associées à une section sont:

- 'Aide méthodologique de Réalisation d'Intention' (ARI) : elle a pour objectif d'aider l'ingénieur d'application à accomplir l'intention but d'une section donnée selon la stratégie de cette section. Elle fournit des moyens opérationnels pour accomplir l'intention,
- 'Aide méthodologique de Sélection d'Intention' (ASI) : pour une intention donnée, une ASI identifie l'ensemble des intentions qui peuvent être atteintes à l'étape suivante et sélectionne l'ensemble des ARI ou ASS correspondants, et
- zéro ou une 'Aide méthodologique de Sélection de Stratégie' (ASS) : elle a pour but de guider dans la sélection d'une stratégie de façon à mener à la sélection des ARI correspondants.

Comme l'illustre la figure 1, l'interface de l'aide méthodologique est définie par un couple <situation, intention>.

La figure 3 illustre un exemple d'une carte de construction du modèle objet de la méthode OMT [11].

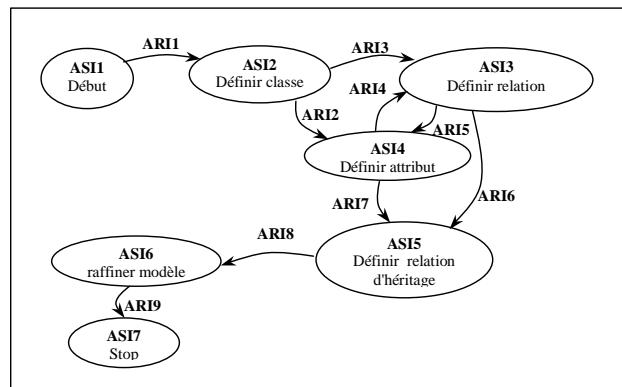


Figure 3: La carte de construction du modèle objet de la méthode OMT

Deux types d'aides méthodologiques sont utilisés dans la carte pour aider l'ingénieur application à construire ce modèle objet selon ses intentions. Le tableau de la figure 4 présente les interfaces des aides méthodologiques de la carte.

<i>Aides méthodologiques de Réalisation d'Intention (ARI)</i>	
<(problème description), définir classe selon Rumbaugh>	ARI1
<(classes définies), définir attributs selon Rumbaugh>	ARI2
<(classes définies), définir associations selon Rumbaugh>	ARI3
<(attributs définis), définir associations selon Rumbaugh>	ARI4
<(associations définies), définir attributs selon Rumbaugh>	ARI5
<(associations définies), définir relations d'héritage selon Rumbaugh>	ARI6
<(attributs définis), définir relations d'héritage selon Rumbaugh>	ARI7
<(relation d'héritage), modèle objet, raffiner modèle>	ARI8
<(modèle raffiné), arrêter la définition du modèle objet OMT>	ARI9
<i>Aides méthodologiques de Sélection d'Intention (ASI)</i>	
<(problème description), progresser à partir du début>	ASI1
<(classes définies), progresser à partir de définir classes>	ASI2
<(associations définies), progresser à partir de définir associations>	ASI3
<(attributs définis), progresser à partir de définir attributs>	ASI4
<(relations d'héritage définies), progresser à partir de définir relations d'héritage>	ASI5
<(modèle raffiné), progresser à partir de raffiner modèle>	ASI6
<(modèle objet OMT raffiné), arrêter>	ASI7

Figure 4: Les aides méthodologiques du modèle objet de la méthode OMT.

Il y a une aide méthodologique de réalisation d'intention pour chacune des sept sections de la carte de la figure 3. Le choix d'un chemin par l'ingénieur application est guidé par les ARI et les ASI de la carte. Par exemple, l'AR2 fournit des règles pour définir les attributs selon la stratégie Rumbaugh (ce qui correspond à l'intention) des classes déjà définies (ce qui correspond à la situation).

Le méta-modèle du fragment (figure 1) n'est pas suffisant pour décrire les situations pour lesquelles le fragment a été réutilisé, ni pour donner des informations précises pour aider l'ingénieur application dans la recherche et la sélection d'un fragment qui répond à une situation bien déterminée. Pour cela, des travaux effectués dans [12] ont montré que le méta-modèle du fragment doit être étendu en ajoutant un descripteur de fragment; celui-ci sera détaillé dans la prochaine section.

2-4 Le Descripteur du Fragment

Dans notre approche, le descripteur capture les connaissances du contexte de réutilisation du fragment [12]. Comme le montre la figure 5, il définit la situation dans laquelle le fragment peut être utilisé et l'intention qui peut être accomplie par le fragment. La situation du descripteur inclut deux aspects : les domaines d'application dans lesquels le fragment peut être appliqué et les étapes du cycle de vie dans lesquelles le scénario du fragment a un sens. L'intention du descripteur exprime comment le scénario de l'approche encapsulé dans les fragments participe dans la réalisation des étapes du cycle de vie. Le nom de la méthode dans laquelle le fragment a été identifié, son auteur et les références dans la littérature sont aussi pris en compte par le descripteur. Le descripteur spécifie aussi le type du fragment (agrégat ou atomique) ; si le fragment est agrégat, son descripteur est lié aux descripteurs de ses fils qui le composent. Si le fragment est un composant d'un ou plusieurs fragments agrégats, son descripteur est relié aux descripteurs de ses pères. L'intention du descripteur incorpore récursivement l'intention du fragment correspondant.

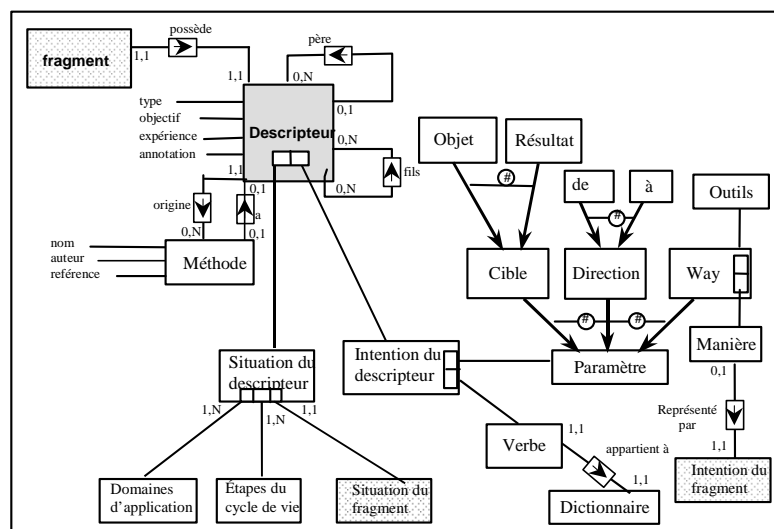


Figure 5: Le méta-modèle du descripteur.

3. ENVIRONNEMENT DE REALISATION DE LA BIBLIOTHEQUE

Du fait que le principe des méthodes situationnelles est d'avoir des méthodes ajustées à des contextes spécifiques permettant alors la prise de décision, leur construction est faite d'une manière dynamique en réutilisant des méthodes existantes. En effet, durant la sélection, les fragments de méthodes correspondants aux caractérisations du projet sont sélectionnés à partir de la base des fragments de méthodes décrite dans la partie précédente, puis assemblés en une méthode situationnelle.

Dans ce qui suit, nous présentons l'environnement correspondant aux différentes implémentations réalisées pour aider l'ingénieur méthode à collecter les fragments nécessaires.

Cet environnement propose trois modules:

- le module d'alimentation de la base permettant d'introduire des informations sur les fragments,
- le module du générateur de requêtes permettant de trouver les fragments satisfaisants des caractéristiques des descripteurs,
- et le module de consultation des informations sauvegardées dans la base de méthodes.

Pour développer cet environnement, nous avons utilisé ;

- la syntaxe SQL+ pour créer notre base de données Oracle,
- Developer 2000 , le produit de Oracle, pour réaliser l'interface d'alimentation de la base et le générateur de requête,
- le produit Oracle Web Server pour développer le module de consultation.

3-1 Module d'alimentation de la base:

Ce module consiste en un ensemble d'interfaces reliées entre elles par des boutons, ou peuvent être accédées indépendamment à travers le menu. En effet, la barre des menus présente un ensemble de menus en relation avec la gestion de notre base oracle, exemple: *Action*, *Enregistrement*, *Champ* et *Table*. Nous notons deux menus particuliers *Composant* et *Général* ; le premier permet d'ajouter des éléments des composants comme *Descripteur*, *Fragment*, *Intention* ou *Situation*, le second permet d'ajouter des informations d'ordre général comme *domaines d'application*, *étapes du cycle de vie* ou *verbe*.

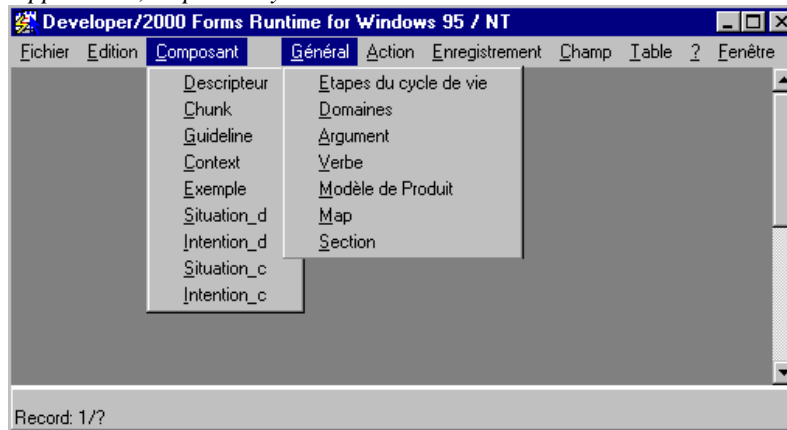


Figure 6: le menu du module de l'alimentation de la base.

Parmi les interfaces principales permettant d'introduire les informations selon la structure déjà pré-définie, certaines sont illustrées dans la suite de cette section.

Pour ajouter un descripteur, nous devons introduire des informations tels que son objectif, l'expérience, le nom de la méthode et celui de son auteur comme le montre la figure 7. Puis, nous pouvons soit continuer avec un autre descripteur soit ajouter les détails relatifs à la situation, l'intention du descripteur et le fragment correspondants au descripteur que nous venons d'insérer.



Figure 7: l'interface d'ajout du descripteur.

Pour créer la situation du descripteur, nous utilisons l'interface illustrée par la figure 8; nous devons spécifier en particulier la description de la situation, les étapes de cycle de vie et les domaines d'application du fragment correspondant. Les étapes du cycle de vie et les domaines d'application possibles sont extraits à partir des tables.

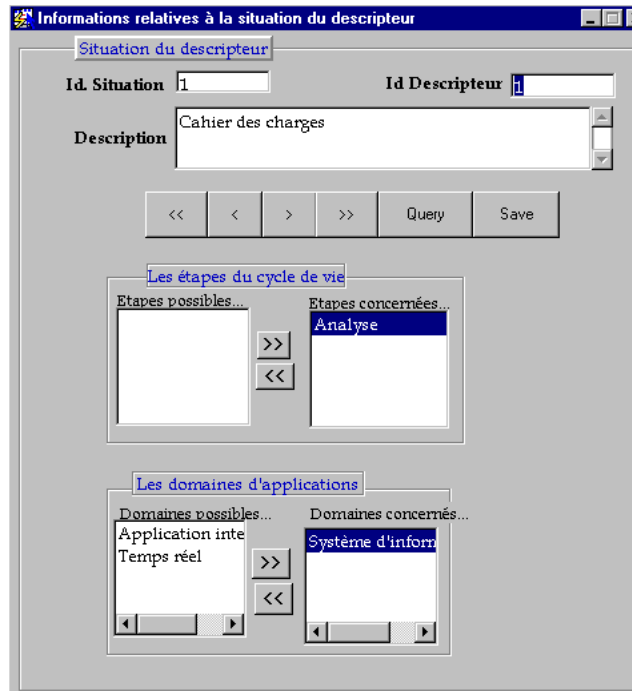


Figure 8: interface d'introduction de la situation du descripteur

La figure 9 montre comment insérer l'intention du descripteur. Tout d'abord, il faut spécifier si l'intention cible est objet ou résultat. L'intention cible est objet si l'intention permet de créer un objet (créer classe, définir relation, etc.), et elle est résultat si l'intention permet de gérer un objet déjà créé (raffiner classe, valider relation, etc.). Par la suite, le verbe de l'intention doit être introduit en le choisissant parmi la première liste; la seconde liste contient les synonymes du verbe sélectionné dans la première. Finalement, la direction de l'intention est précisée en spécifiant l'intention précédente.

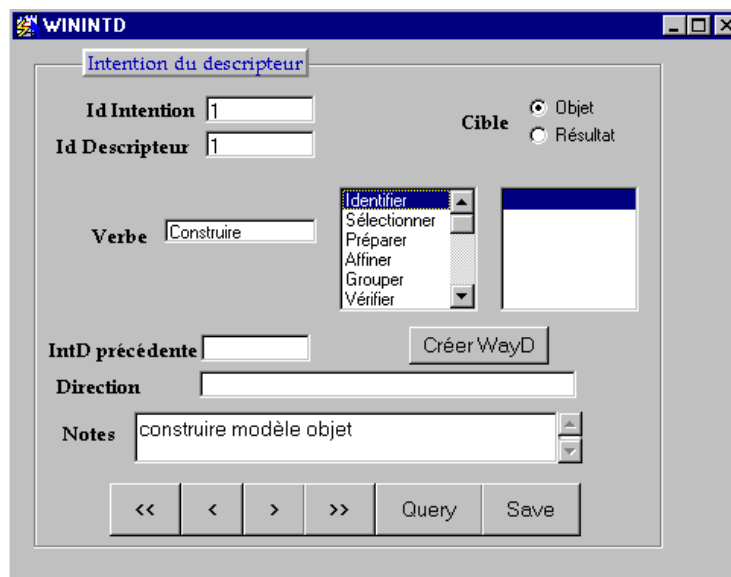


Figure 9: interface d'ajout de l'intention du descripteur.

L'interface d'ajout d'un nouveau fragment est présentée dans la figure 10 où il faut spécifier si le fragment à insérer est atomique ou agrégat, et s'il est un composant contexte et/ou composant aide méthodologique.

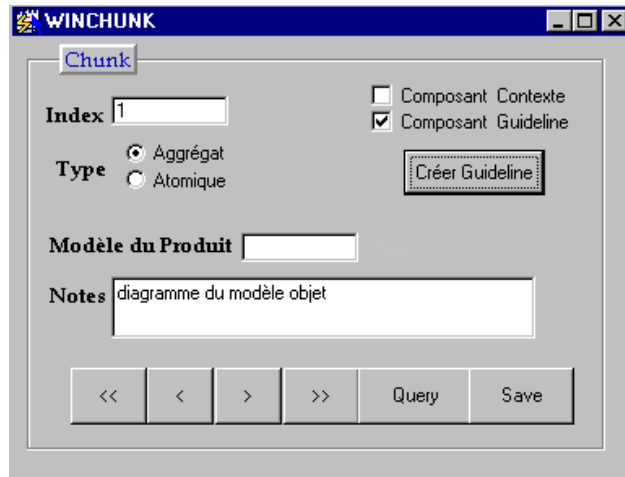


Figure 10 : interface d'ajout du fragment.

3-2 Générateur de requêtes

Le module de recherche est formé de deux fenêtres illustrées par la figure 11. La première correspond aux critères de recherche de base comme les étapes du cycle de vie, les domaines d'application, le nom de la méthode ou encore l'auteur de la méthode. La seconde correspond aux critères avancés comme le type du fragment (agrégat, atomique), le type de la directive (stratégique, tactique ou informelle), le verbe de l'intention, l'intention source et cible, et enfin si l'intention cible est objet ou résultat. Si l'utilisateur veut inclure un critère parmi ceux présentés, il n'a qu'à cocher la case correspondant au critère choisi et spécifier sa valeur à partir de la liste des valeurs disponibles. Une fois que l'utilisateur termine la spécification de ses critères, il clique sur le bouton Rechercher pour lancer la recherche dans la base de fragments, le résultat apparaît alors dans une autre fenêtre

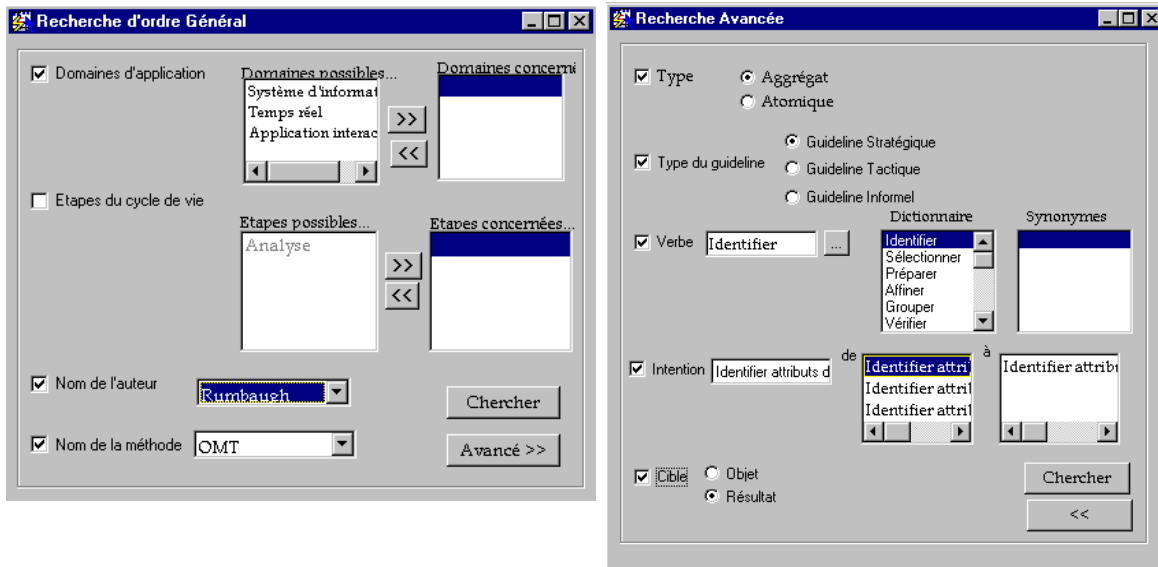


Figure 11 : critères de recherche de base et avancés.

illustrée par la figure 12 qui comporte les objectifs des descripteurs de fragments qui satisfont les critères.

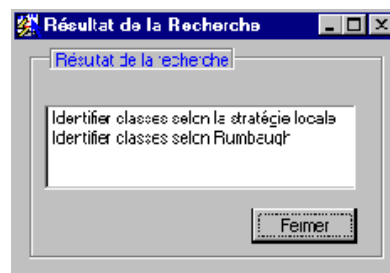


Figure 12 : résultat de la recherche.

3-3 Module de consultation

Le module de consultation développé avec Oracle Web Server, le produit de Oracle qui permet de générer des pages HTML à partir du contenu de la base Oracle, nous permet de consulter le contenu de la base d'une manière assez simple; en effet, à partir de la première page où sont présentés les fragments par leurs identificateurs, noms et objectifs, l'utilisateur peut naviguer dans les détails de son choix par simple clic grâce aux liens HTML. Le menu de droite lui permet d'afficher les détails décrivant le fragment en question, entre autres la situation et l'intention, et de consulter le descripteur en question dans ses moindres détails.

3-4 Cas d'application

Pour illustrer le fonctionnement de notre environnement, nous pouvons considérer la description de la fragmentation du modèle objet de la méthode OMT. En effet, le descripteur, ayant pour objectif la construction du modèle objet, décrit un fragment de nom cmo1 relatif à la méthode OMT dont l'auteur est Rumbaugh et est du type agrégat. A ce descripteur sont associées d'une part une situation qui est l'élaboration du cahier des charges et d'autre part une intention définie par le verbe « construire » et dont l'intention cible est du type objet. Il note que le fragment ne s'applique que dans l'étape d'analyse du cycle de vie et qu'il ne concerne que les systèmes d'information. La description du fragment correspondant indique qu'il est un composant aide méthodologique de type agrégat et qu'il délivre comme produit le diagramme du modèle objet. La figure 13 ci-dessous illustre la description de ce fragment ainsi que sa représentation graphique.

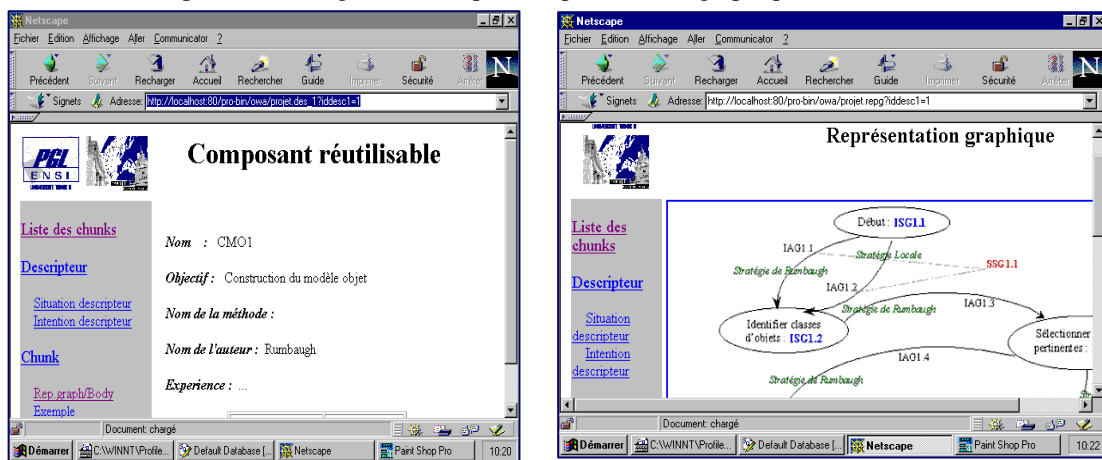


Figure 13 : Description du fragment de construction du modèle objet de la méthode OMT et sa représentation graphique.

4. Conclusion

Dans cet article, nous avons vu que les méthodes situationnelles, visant à satisfaire une situation particulière d'un projet, sont construites à partir d'un ensemble de fragments de méthodes pré-définis validés et accessibles dans une base de méthodes. Pour aider à la construction de ce type de méthodes, nous avons proposé une bibliothèque constituée d'une base contenant des fragments de méthodes de natures différentes et de trois modules. Le premier module est un module qui permet l'alimentation de la base. Le module de génération de requête permet de rechercher des fragments selon des critères bien spécifiques, et enfin le module de consultation permettant de visualiser le contenu de la base grâce à une structure navigationnelle.

Ce travail s'inscrit dans le cadre d'une approche de développement à partir de composants où un composant n'est autre que le fragment de méthode avec son descripteur. Cette approche commence à émerger d'autant plus que plusieurs travaux se font sur des composants de granularités différentes tels que les progiciels de gestion intégrée : ERP (Entreprise Resource Planning), les composants sur étagères COTS (Commercial Off-The-Shelf) et les CCOTS (Complex COTS). Dans notre travail futur, nous projettons de développer un environnement similaire à celui présenté dans cet article pour supporter la sélection et l'assemblage de ces composants.

5. Bibliographie

- [1] F. Harmsen : Method Engineering ; Moret Ernst & Young Management Consultants, 1997.
- [2] C. Rolland : A proposal for Context-Specific Method Engineering ; IFIP TC8 Working Conference on Method Engineering, Atlanta, Georgie, USA, 1996.
- [3] F. Harmsen : Situational Method Engineering for Information System Projects ; Methods and Associated Tools for Information Systems Life Cycle, Proceeding of the IFIP WG8.1 Working Conference CRIS'94, pp. 169-194, North-Holland, Amsterdam, 1994.

- [4] S. Brinkkemper : Assembly Techniques for Method Engineering ; Proceedings of the 10th Conference on Advanced Information Systems Engineering, CASE'98, Pisa Italy, 8-12 june, 1998.
- [5] Euromethod Architecture, Euromethod project deliverable Work Package 3, 1994.
- [6] C Rolland : Guiding Goal Modelling Uusing Scenarios ; IEEE Transactions on Software Engineering, Special Issue on Scenario Management, 1998.
- [7] C. Rolland : A Multi-Model View of Process Modelling ; Requirement Engineering Journal, 1999.
- [8] G. Grosz : Modelling and Engineering the Requirements Engineering Process : un overview of the NATURE approch ; Requirements Engineering Journal 2, pp. 115-131, 1997.
- [9] C. Rolland : An Approach for Defining Ways-of-Working ; Information System Journal, Vol. 20, No 4, pp. 337-359, 1995.
- [10] N. Prat : Goal Formalisation and Classification for Requirements Engineering ; Proccedings of the third international workshop on requirement engineering: foundation of software quality (REFSQ'97), Barcelona, Spain, pp. 145-156, June 1997.
- [11] J. Rumbaugh : Object Oriented Modelling and Design ; Prentice Hall International, 1991.
- [12] I. Gam : Proposition d'un descripteur de composants réutilisables pour les méthodes situationnelles ; Mémoire de DEA en Informatique, Institut Supérieur de Gestion, Tunis, Tunisie, 1999.