

Réflexions en vue d'une meilleure prise en compte des experts et utilisateurs dans le développement de SIAD : apports des patrons et de la composition d'IHM

Thoughts in order to better taking account of experts and users in DSS development: contribution of patterns and HCI composition

Sophie LEPREUX

LAMIH, Université de Valenciennes et du Hainaut-Cambrésis, France
sophie.lepreux@univ-valenciennes.fr

Résumé. Cet article propose une méthodologie de développement de SIAD, centrée sur la connaissance des experts et les besoins des utilisateurs. L'objectif est de capitaliser la connaissance métier à l'aide de patrons. Ces patrons sont utilisés également pour la présentation des outils mis à la disposition des utilisateurs, afin de les guider dans le choix des composants qui leur apporteront une aide à la décision. En effet, dans certains domaines, la prise de décision nécessite un ensemble d'analyses pouvant être réalisées de manière plus ou moins indépendante à l'aide de composants logiciels ; le choix des analyses à mener est donc associé au choix des composants. La méthodologie générale est d'abord présentée, puis appliquée sur une étude de cas dans le domaine ferroviaire. Cette étude de cas permet de valider globalement la méthodologie et l'apport des patrons et met en évidence la problématique de la composition des IHM. En effet, les résultats issus de diverses analyses doivent être assemblés afin d'apporter une présentation globale facilitant la décision. Les IHM des composants doivent donc être composées pour être présentées ensemble à l'utilisateur.

Mots-clés. Aide à la décision, méthodologie, connaissance, guidage, composition d'IHM.

Abstract. This article proposes a development methodology of DSS, centered on expert knowledge and users needs. The goal is to capitalize the business knowledge with patterns. These patterns are used to present tools, which are proposed to the users in order to guide them in their tools choice. These tools give them help to take their decisions. In several domains, the decision making needs analyses which can be performed with different units (as components). The choice of analyses to complete is associated to the choice of components. The global methodology is first presented and then applied on a case study in railway domain. The case study allows

validating the methodology and the patterns role. It opens a new issue relative to the HCI composition which is treated before to conclude.

Keywords. Decision support, methodology, knowledge, guiding, HCI composition.

1 Introduction

Les systèmes interactifs d'aide à la décision (SIAD) sont souvent des systèmes « décideurs » ; dans ce cas c'est le système qui fournit la décision à prendre. Pour nous, *a contrario*, ces systèmes doivent être un support à l'utilisateur en lui permettant de mesurer les conséquences des décisions tout en essayant de ne pas l'induire en erreur ; c'est donc au décideur de faire des choix et non pas au système. Pour cela, l'utilisateur doit pouvoir comparer les diverses options pour pouvoir faire le choix le plus approprié. Des outils d'analyse lui sont alors nécessaires. Ces outils peuvent être de natures très différentes (simulateur, système expert, recherche opérationnelle,...) et apportent tous un indicateur de décision. Le système doit permettre des interactions avec l'utilisateur afin d'utiliser l'ensemble de ces outils et d'analyser l'ensemble des résultats en vue de lui permettre de prendre une décision. Nous nous situons ici dans le processus de décision basé sur la connaissance dans le modèle de Rasmussen (1983).

Les IHM dans les systèmes d'aide à la décision ont déjà fait l'objet de nombreuses recherches. Par exemple, Chen (1995) a cherché à améliorer les IHM pour les systèmes experts en connectant chaque composant du système expert avec des interfaces homme-machine pour apporter une meilleure information à l'utilisateur. Pour cela, il propose un composant, nommé « wrapper » d'interface, qui est une couche associée au composant « fonctionnel » contenant l'interface utilisateur spécifique à ce composant. Chaque composant fonctionnel est donc « accompagné » d'une interface utilisateur. Les utilisateurs experts n'ont pas nécessairement besoin des interactions fournissant plus de précision sur le fonctionnement tandis que les novices pourront les utiliser pour mieux comprendre le fonctionnement du système. De cette manière, l'utilisateur peut suivre le processus mis en œuvre par le système et donc avoir plus confiance dans ses résultats. Ces interfaces peuvent également permettre aux utilisateurs de mieux vérifier les bases de connaissance et identifier les incompatibilités. Le principe adopté par Chen est intéressant car il montre l'intérêt d'associer des interfaces utilisateur à chaque partie de noyau fonctionnel même si il ne se pose pas la question du style des interfaces proposées.

Le domaine des composants s'est développé en informatique, comme dans d'autres disciplines (matériaux, électroniques...) auparavant, pour faciliter la réutilisation. La définition la plus souvent utilisée est celle de Szyperski (1998) : « A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third party ». Les composants logiciels peuvent avoir des granularités variables. Cette granularité peut être mise en relation avec le niveau d'abstraction dans lequel le composant apparaît, et par conséquent selon l'étape du cycle de développement où il apparaît. Par exemple les composants de code sont impliqués entre les étapes de conception détaillée et l'étape de développement de code, tandis que les composants métier (qui correspondent à un métier ou domaine) sont introduits à l'étape d'analyse des besoins pour être mis en œuvre lors de la spécification. De nombreux travaux traitent de ces composants métiers en Génie logiciel (Dao et Dony, 2001), dans le

domaine des systèmes d'information (Cauvet et al., 2001), (Oussalah et Rieu, 2004), (Tamzalit et Anioré, 2005), etc. Les composants sont également utilisés dans le domaine des IHM pour tenter de réaliser des compositions d'applications interactives par exemple dans le cadre d'applications collaboratives (Lewandowski *et al.*, 2007b). Dans cet article, nous nous focalisons sur les composants métier. Nous nous basons sur deux définitions de Cauvet et Semmak (1999) et d'Hassine, Rieu et leurs collègues (2002). La définition de Cauvet et Semmak (1999) considère les types de composants suivant leur étape d'implication dans le cycle de développement. Ils définissent alors les composants de domaine comme exprimant des connaissances réutilisables dans le développement de tous les systèmes d'un champ d'application. Ces composants sont utilisables et réutilisables dès la phase d'analyse du processus de conception car ils expriment souvent une connaissance de haut niveau, orientée sur la définition de problèmes, de besoins. Ces composants métier capitalisent une certaine connaissance du domaine. Cette connaissance est donc capitalisée et peut être réutilisée lors d'autres développements associés au même domaine. Pour nous, les composants de domaine sont des composants métier. La seconde définition est celle de Hassine, Rieu et leurs collègues (2002) pour qui il existe deux types de composants : les composants métier et les composants techniques. Pour eux, « les composants métier couvrent les besoins du domaine. Ces composants sont divisés en composants verticaux qui sont spécifiques à un domaine (par exemple les modèles d'objets métier financiers) et en composants horizontaux qui représentent des éléments récurrents dans des applications appartenant à différents domaines, comme les services d'authentification ou l'édition de rapport ». Les composants métier sur lesquels nous allons nous focaliser dans le cadre des SIAD sont de type vertical, spécifiques aux métiers associés à la décision. Dans cet article, nous considérons qu'un composant métier peut être un composant composite comprenant à la fois un noyau fonctionnel mais aussi des composants associés aux IHM adaptées. En suivant le principe adopté par Chen (1995), nous souhaitons de cette manière coupler les composants fonctionnels et leurs IHM dans une entité globale spécifique à un métier. Ceci permet également de séparer les composants logiciels et les IHM comme recommandé par les architectures PAC ou MVC. On pourrait utiliser ce type de composants comprenant plusieurs composants d'interface à utiliser selon le contexte afin d'intégrer de la plasticité dans les applications (Calvary et al., 2006 ; Calvary et Coutaz, 2007).

Les patrons (pattern en anglais) ont été adaptés à la réutilisation de code informatique par Gamma (Gamma *et al.*, 1994) qui a proposé un catalogue de patrons de conception. Front-Conte et ses collègues (1999) utilisent des patrons orientés objet pour intégrer la réutilisation durant les premières phases du cycle de développement d'un logiciel. Ils distinguent les patrons d'analyse, les patrons de conception et les patrons d'implémentation. D'après les mêmes auteurs, un patron permet de donner la solution d'un problème ou de pointer vers d'autres problèmes ayant si possible eux-mêmes des patrons. Un patron utilise d'autres patrons lorsque ces derniers peuvent résoudre en partie ces problèmes : "un patron X utilise un patron Y, si une partie des problèmes posés par X peuvent être résolus en partie ou complètement par Y ". Borchers (2001) et Welie et van der Veer (2003) en ont proposés pour les interactions homme-machine. Pour nous les patrons se situent à un niveau plus en amont que ceux de Gamma. Ils ne sont pas utilisés en conception mais plus tôt dans le cycle de développement, c'est-à-dire en phase d'analyse. Ils permettent de capitaliser les problèmes du domaine ainsi que les solutions si elles existent et/ou les liens entre ces problèmes (Par exemple, pour trouver une solution à tel problème, il faut étudier tel aspect et tel autre). Une solution peut être mise en

évidence par une étude qui sera réalisée à l'aide de composants métier. Les patrons sont à la fois un moyen de capitaliser et de permettre la réutilisation des connaissances du domaine. Ils sont utilisés pour guider l'utilisateur dans son processus de prise de décision car ils permettent de faire le lien entre la connaissance et les outils (composants métier).

Nos travaux précédents ont consisté à proposer une amélioration à une démarche de développement de SIAD permettant de mieux prendre en compte l'utilisateur en phase d'analyse (Lepreux *et al.*, 2003). Cette méthode nous semblait encore insuffisante et a amené à une démarche plus complète pour intégrer l'ensemble des acteurs dans le cycle de développement (Lepreux, 2006). Nous souhaitons renforcer l'aide aux utilisateurs de SIAD en leur fournissant des patrons les guidant dans le choix des composants métier en fonction de leur(s) problème(s) et du contexte de la décision.

La section suivante présente un ensemble de modèles de développement issus des domaines des SIAD, du génie logiciel, des interactions homme-machine et de la gestion des connaissances. Une approche spécifique aux SIAD est proposée en troisième section. Le modèle, les rôles des acteurs et la méthode sont présentés de manière à préciser les moyens qui sont mis en œuvre pour prendre en compte les experts et les utilisateurs dans le cycle de manière à supporter la décision (les décideurs). Une étude de cas permet d'illustrer l'ensemble des propositions en quatrième section. Elle permet également de montrer que la partie fusion des résultats issus des composants métier et nécessaire à la décision est difficile à traiter. Ainsi, la cinquième section introduit la problématique relative à la composition d'IHM ainsi qu'une proposition. Une conclusion et des perspectives terminent l'article.

2 Les modèles de développement existants

Malgré l'importance des SIAD, il n'existe pas d'approche de développement générique (i.e. indépendante du domaine d'application) et spécifique aux SIAD (i.e. prenant en compte les spécificités des SIAD) (Gachet *et al.*, 2007). Les méthodes existantes dans ce domaine sont souvent spécifiques à un domaine d'application (Palma-dos-Reis et Zahedi, 1999) ou centrées sur un moyen de calcul (RO, IA, simulation...) (Lévine et Pomerol, 1989).

Nous nous sommes donc intéressés aux méthodes « génériques » issues d'autres domaines. Le modèle en V est le plus connu des modèles du génie logiciel (Jaulent, 1994). Il structure les étapes en deux phases : une descendante pour la conception et une ascendante pour l'intégration et l'évaluation. Les moyens d'évaluation sont définis dans la phase descendante du cycle de vie. Ce modèle n'est pas suffisamment précis sur le rôle des utilisateurs et experts pour permettre un développement efficace d'un SIAD. Ce modèle est un pilier du génie logiciel ; il a souvent été réutilisé et adapté dans d'autres domaines, comme nous le verrons dans la fin de cette section. Il existe de nouvelles méthodes dites « agiles » proposées dans le génie logiciel telles que RUP et eXtreme Programming (XP). RUP est guidé par les cas d'utilisation, centré sur l'architecture, itératif et incrémental. Ce modèle recommande un développement basé sur les composants. Il utilise le langage de modélisation UML. Il comprend cinq étapes classiques telles que l'étude de faisabilité, l'analyse et la spécification des besoins, la conception et l'évaluation qui sont plus ou moins approfondies selon l'avancement dans le processus découpé en quatre phases : l'initialisation (ou l' inception), l'élaboration, la construction, et la transition. Ce processus est donc transversal aux cycles « habituels » tels que le cycle

en V. Il propose des principes qui sont adaptables à de nombreux cycles. XP est strictement itérative ; les développements s'effectuent selon un système d'itérations imbriquées de courtes durées comprenant les itérations de développement et les itérations de livraison (Cloux, 2003). Certaines pratiques sont très intéressantes comme « assurer un feedback constant » avec l'inconvénient d'amener des principes tels que le développement piloté par les tests. Nous prenons le point de vue que ce principe est un inconvénient dans le cas des SIAD. En effet, il est difficile de valider un SIAD en partie seulement pour vérifier qu'il amène bien une aide à la décision pour un type de problèmes. Un autre défaut, à notre avis, de cette méthode pour le développement de SIAD vient de sa particularité qui consiste à refuser toute rédaction de documentation d'analyse et de conception.

Les rôles des utilisateurs et des experts sont primordiaux dans le développement de SIAD. Nous nous intéressons donc aux modèles issus du domaine des IHM qui a pour objectif de mieux prendre en compte l'utilisateur. Le cycle en V a été utilisé dans le domaine des IHM puisqu'il a inspiré le modèle en U (Abed, 2001), (Lepreux et al., 2003). Ce dernier modèle intègre des étapes centrées sur les facteurs humains et a par la suite été enrichi d'étapes permettant de mieux intégrer les experts dans la phase descendante. Ce modèle pourrait être utilisé dans notre étude car il prend en compte les utilisateurs et les experts mais il n'est pas assez complet pour la réutilisation ni en terme de gestion des connaissances. En effet, il ne permet pas de capitaliser et de réutiliser ces connaissances. Un autre modèle impliquant les utilisateurs qui n'est pas basé sur le modèle en V est le modèle en étoile (Hartson et Hix, 1989). Il a l'intérêt de placer l'évaluation au centre du processus. Cela permet de réellement intégrer les utilisateurs au centre du processus par le biais des évaluations. Cependant, ce cycle n'est pas suffisamment complet pour un développement logiciel ; nous n'en retiendrons que son principal intérêt. Le domaine intégrant au mieux l'expert est le domaine de la gestion des connaissances. Dans ce domaine, le modèle MODESTI couple les principes des IHM et l'intégration des experts (Duribreux-Cocquebert, 1995). Il propose trois phases descendantes en parallèle de manière à distinguer les tâches de l'expert, de l'utilisateur, et des autres intervenants de la conception comportant les étapes classiques du génie logiciel. Il permet de cette manière d'intégrer les besoins de l'utilisateur, les connaissances de l'expert dans le cycle « classique » de développement. L'inconvénient de ce modèle est qu'il ne prend pas en compte la réutilisabilité.

Les modèles visant à améliorer la réutilisation sont également étudiés. Le cycle en V a été enrichi pour intégrer la réutilisation en lui ajoutant des modules de gestion des composants. Cet enrichissement a abouti au cycle en X qui intègre des étapes d'utilisation et de fabrication et stockage des composants (Coulange, 1996). Boehm (2006) présente les tendances des prochaines années et les problématiques semblables à celles de cet article. Il propose une adaptation du modèle en spirale nommé « Scalable Spiral Process model » proposant des incréments successives intégrant une équipe de développement agile, une équipe de construction et une équipe de vérification et validation. Ce modèle pourrait être utilisé mais reste cependant trop général pour être appliqué concrètement à un système de type SIAD.

L'ensemble des modèles offre des principes intéressants : démarche centrée utilisateur pour les modèles issus du domaine des IHM, utilisation de composants afin de faciliter la réutilisation, prise en compte de l'expert et utilisation de patrons pour la capitalisation des connaissances du domaine... Cependant ils ne sont pas individuellement suffisants pour le développement d'un SIAD. C'est pourquoi la

section suivante présente un modèle que nous proposons pour le développement de SIAD.

3 Approche de développement de SIAD

L'Approche de Développement de SIAD (ADESIAD) proposée se compose d'un modèle (au sens cycle de développement), d'une méthode et d'une architecture. Le modèle est une synthèse issue des cycles en V, en U et en étoile, en X et MODESTI. Chaque étape comprend des activités qui peuvent être issues du GL, des IHM, en lien avec la réutilisation et les systèmes à base de connaissances. Ce cycle se veut également itératif. Le cycle proposé, vu en Figure 1, permet d'impliquer les utilisateurs et experts lors des phases (a) descendante, dite de conception, (b) ascendante qui comprend l'intégration et les évaluations techniques et terminales (ou sommatives) et (c) centrale dite d'évaluations formatives. Nous remarquons qu'il y a des liens permanents avec l'étape d'évaluation technique et d'évaluation par des utilisateurs. L'objectif de cette étape est d'anticiper au maximum les évaluations et d'impliquer au maximum les utilisateurs. Elle rend plus agile le cycle de développement car elle permet de faire des sauts dans le cycle aussi bien en avant si les propositions sont validées qu'en arrière si des erreurs ou problèmes sont détectés.

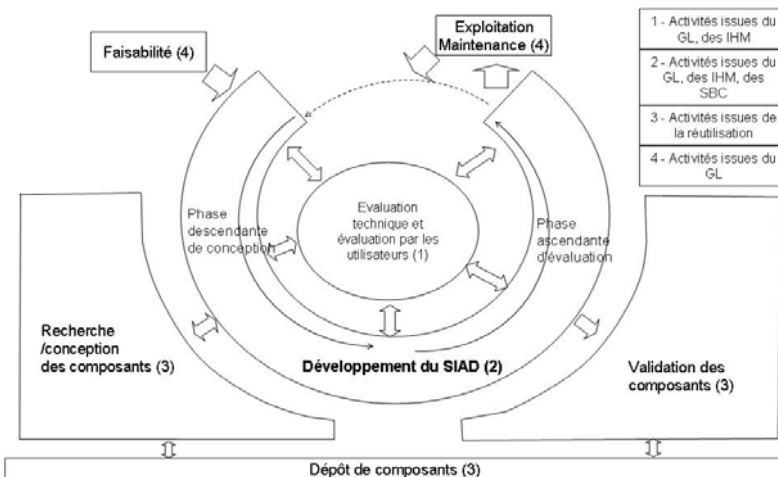


Figure 1. Le modèle d'ADESIAD, vue globale

Une présentation plus détaillée du modèle est donnée en Figure 2. Dans cette vue, les relations entre la conception d'un système interactif et la réutilisation sont exprimées par les trois liens : entre la spécification et la recherche de composants métier, entre la conception préliminaire et la recherche de composants de conception (couramment associés aux designs patterns) et entre la conception détaillée et la recherche de composants de code. Ces liens se retrouvent également de manière symétrique lors de la phase ascendante d'évaluation permettant de valider les choix en terme de composants à toutes les étapes de tests. Ces étapes de validation n'intègrent pas la vérification du composant car celui-ci a, par nature, subi des tests unitaires par ailleurs. Pour plus de détails concernant ce modèle, le lecteur peut consulter (Lepreux, 2005). Nous verrons dans la suite (cf. § 3.2)

comment la réutilisation influe sur la conception et l'utilisation du SIAD et comment les patrons seront utilisés pour faciliter l'utilisation de ces composants.

L'originalité de ce cycle est qu'il couple les principes (enchaînement des activités, implication des acteurs, prévision des tests, types d'évaluation, positionnement des composants) issus d'autres cycles (V, X, étoile) venant de divers domaines. A ce stade il pourrait être valable pour tout système interactif souhaitant intégrer de la réutilisation. L'implication des acteurs et la méthode permettent de personnaliser le développement pour un système interactif permettant de gérer de la connaissance.

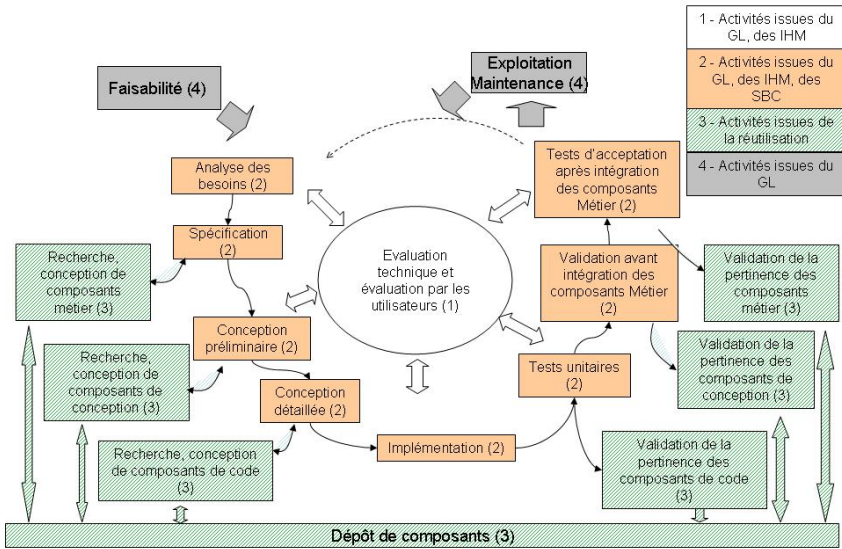


Figure 2. Le modèle d'ADESLAD, vue détaillée

3.1 Les acteurs

Les acteurs agissant dans ce modèle sont présentés en Figure 3. Ce schéma est issu des travaux de (Caulier, 1997) et (Duribreux-Cocquebert, 1995) auquel nous avons ajouté le bibliothécaire. Les acteurs sont regroupés en trois groupes. Le groupe de pilotage (ou comité directeur) qui définit les finalités et arrête des stratégies pour l'ensemble des aspects du projet. Ce groupe a une fonction de contrôle de projet d'un point de vue administratif et de coordination d'un point de vue technique. Le directeur fait nécessairement partie de ce groupe. Le groupe de développement est composé d'un cogniticien, d'un informaticien, d'un ergonome et d'un bibliothécaire. Le cogniticien recueille et analyse les connaissances expertes, élabore les modèles. La réalisation est à la charge de l'informaticien qui définit les orientations pour la réalisation technique et concrétise le système. Il a un rôle majeur dans cette étape mais reste en relation avec le cogniticien, l'ergonome, l'expert et l'utilisateur. L'ergonome analyse les besoins de l'utilisateur. Le bibliothécaire est chargé de fournir les documents aux autres intervenants et de classer les nouveaux documents établis de manière à avoir un suivi et un archivage de toutes les décisions et actions. Il n'a pas pour objectif de contrôler les spécifications des composants. Un informaticien est chargé de cette tâche technique ; il doit vérifier que les composants sont compatibles avec les autres composants de la base avant d'y être ajoutés. Enfin le groupe d'application englobe

l'expert et l'utilisateur. L'expert est le détenteur du savoir alors que l'utilisateur est la personne qui va utiliser ce savoir. Cette répartition des tâches est une proposition qui n'est pas figée. Elle peut être aménagée suivant les habitudes des entreprises. Pour nous, l'important est de ne pas oublier de faire intervenir l'utilisateur et l'expert dans certaines phases et de les tenir informés de l'évolution des autres phases.

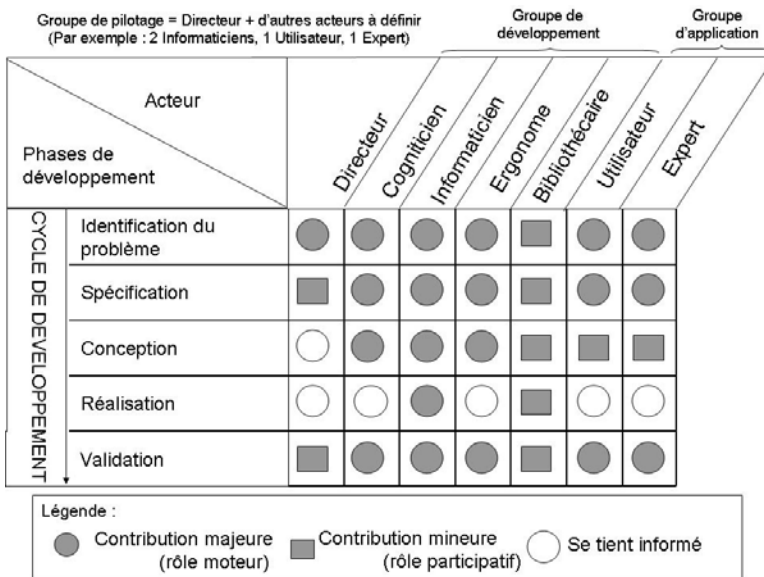


Figure 3. Indication du rôle des acteurs dans le développement

3.2 Les patrons pour la capitalisation de connaissance métier

La méthode a pour but de détailler l'ensemble des étapes du modèle. Elle introduit la notion de patron dans chacune des étapes de la phase descendante de conception. Ces patrons sont utilisés pour la capitalisation des connaissances, à la fois expertes (métier ou du domaine) et informatiques. Ils permettent de faire le lien entre les experts (du domaine de décision ou du domaine informatique) et les composants métier. Ils sont un des moyens introduits dans le cycle pour impliquer les utilisateurs et décideurs et pour permettre de capitaliser les connaissances relatives au problème. L'étape d'analyse des besoins est présentée en Figure 4. Les activités à réaliser sont présentées dans les boîtes accompagnées entre parenthèses des moyens qui permettent de les réaliser. Par exemple, la modélisation des tâches peut être utilisée lors de l'activité d'analyse de l'existant. Les patrons sont proposés pour faciliter et modéliser la décomposition du problème de décision. A partir de cette modélisation, il sera possible d'identifier les composants métier. Ces patrons sont un moyen permettant de faire exprimer la connaissance des utilisateurs et de capitaliser ces connaissances relativement au problème de décision. Ils permettent également aux utilisateurs d'exprimer leur besoin et leur contexte de travail. Par exemple, un utilisateur sait qu'il possédera ou pas une donnée (donnée critique) pour résoudre le problème. Le patron capitalise les données qui sont nécessaires à l'analyse du problème relatif, il est nommé patron de problème. Ce support facilite la discussion entre les différents acteurs et permet d'explicitier les connaissances implicites.

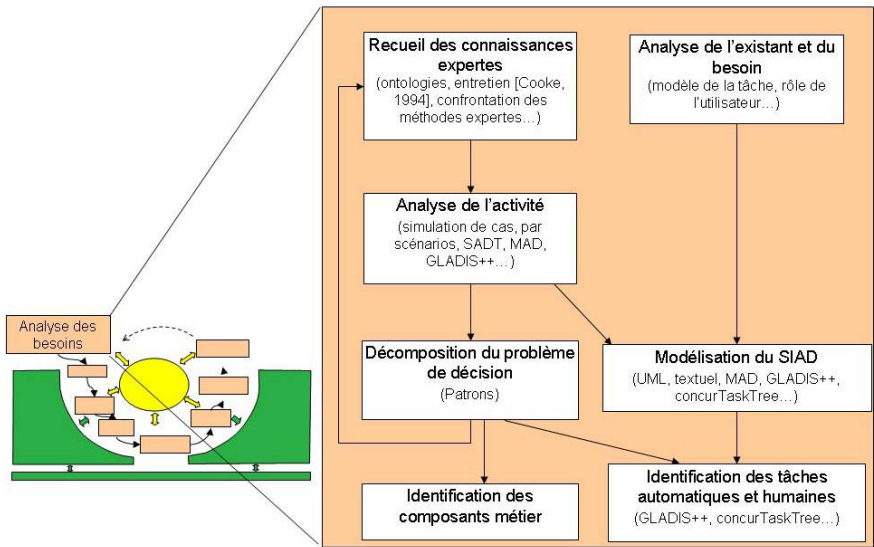


Figure 4. Etape de l'analyse des besoins

Nom du problème	
Définition du problème	
Attributs	
Problème(s) père	
Sous problème(s)	
Problèmes connexes	
Analyse	
Solutions	

Figure 5. Patron pour la description d'un problème ou patron de problème

A l'étape d'analyse les patrons sont complétés majoritairement par des définitions textuelles afin de faciliter les échanges entre les acteurs. Dans les étapes suivantes, ils peuvent être intégrés à une bibliothèque afin de faciliter leur accès.

Nous proposons le patron fourni en Figure 5 qui sert de support aux experts, utilisateurs et ergonome(s) pour la définition des problèmes métier relatifs au SIAD. Sa structure est inspirée des patrons proposés par (Gamma, 1994). De manière à utiliser et partager cette connaissance, chaque problème doit porter un nom. Une définition doit y être associée pour permettre à d'autres personnes non initiées de comprendre le problème. Le problème est toujours lié à des entités qui sont précisées dans le champ « attribut ». Les patrons doivent pouvoir avoir des liens entre eux ; pour cela les champs « problème père » et « sous-problème » assurent la navigabilité. Enfin, d'autres relations peuvent exister entre deux problèmes ; elles sont figurées dans le champ « problèmes connexes » qui est optionnel. Pour terminer, le champ « analyse » permet de renseigner les méthodes de traitement qui permettent d'analyser ou de résoudre le problème. Ces méthodes de traitements seront intégrées dans des composants métier ; c'est ainsi que le patron guide le choix de composants métier. Si des solutions sont d'ores et déjà connues, elles figurent dans le champ « solution ». Le patron (vierge) fournit ainsi un support réutilisable par tous et une base de discussion. Il est générique et peut être adapté en fonction des attentes et des remarques des experts et du domaine. Il est utile également à la présentation des informations relatives à un problème.

Nom du problème	Recherche d'un moyen de transport privé
Définition du problème	Consiste à rechercher un moyen de transport issu d'une entreprise et à usage individuel
Attributs	Origine, Destination
Problème(s) père	Problème de déplacement
Sous problème(s)	Recherche de moyen de transport privé et personnel, Recherche de moyen de transport privé non personnel
Problèmes connexes	Recherche de covoiturage Recherche de moyen de transport public
Analyse	
Solutions	Utilisation du véhicule personnel Partage d'un véhicule avec une autre personne Solution de covoiturage

Figure 6. Patron pour le problème de recherche de moyen de transport privé

Un exemple est donné pour le problème de recherche d'un moyen de transport privé en Figure 6. Ce problème prend comme attributs les points d'origine et de destination. Une décomposition complète est montrée en Figure 7 correspondant au problème de choix de déplacement. Les sous-problèmes sont reliés au problème père par de simples flèches. Cependant, il est possible d'enrichir le patron en ajoutant des opérateurs logiques comme le fait Buisine (1999) relativement à la décomposition de tâches. Par exemple, pour traiter ce problème A, il faut auparavant traiter le problème B et (le problème C ou le problème D).

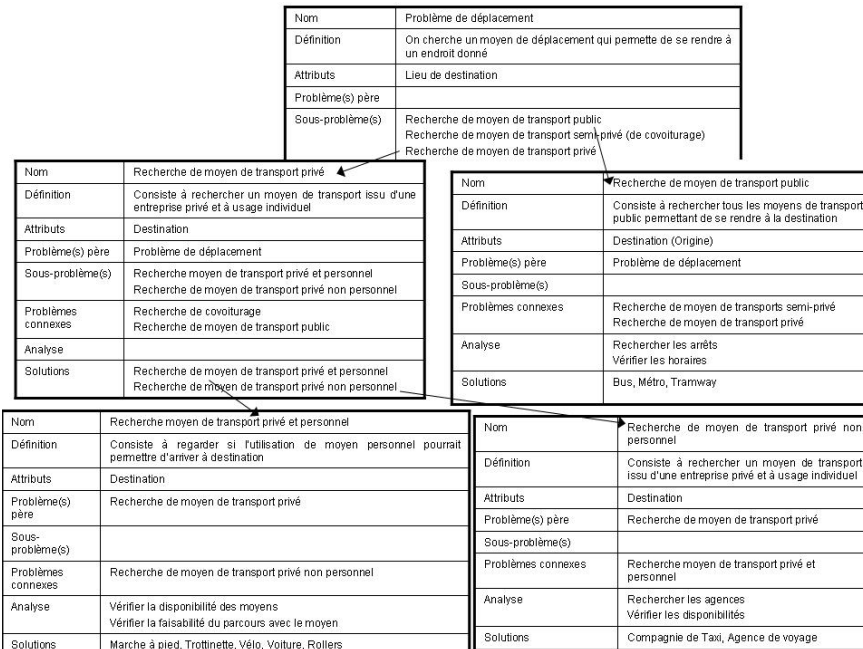


Figure 7. Exemple de décomposition d'un problème de déplacement, mise sous la forme d'une hiérarchie de patrons de problème

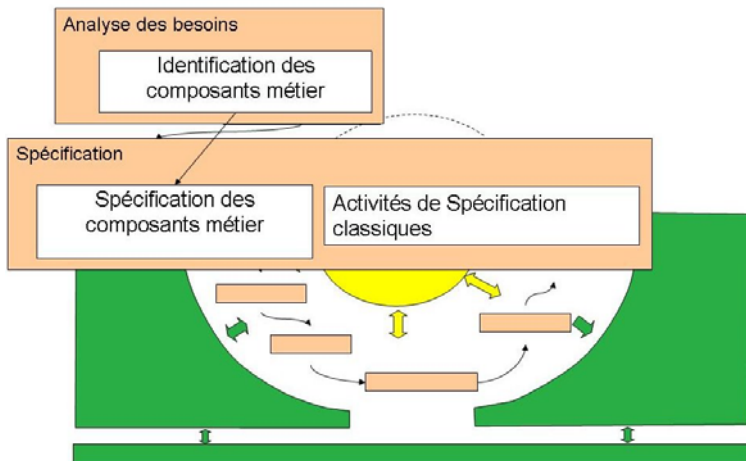


Figure 8. Etape de Spécification

Une fois que les besoins en composants métier sont identifiés par les patrons définis en étape d'analyse, l'étape de spécification peut débuter. L'étape de spécification est séparée en deux parties : une partie relative à la spécification des composants métier est détaillée par la suite ; l'autre partie nommée « activités de spécification classiques », qui regroupe les activités habituelles visant le développement du système lui-même (telles que la modélisation des tâches interactives et de l'utilisateur, les besoins en assistance et le choix des modules d'aide, la spécification des interfaces homme-machine du SIAD), ne fait pas l'objet de cet article. La spécification des composants métier permet de faire le lien entre les problèmes définis lors de l'analyse et les composants métier (cf. Figure 8). La Figure 10 illustre à la fois la relation entre le champ « analyse » du patron et les patrons associés aux composants métier. Elle présente les spécifications des composants métier qui devraient/pourraient être utilisés pour le problème de la recherche d'un moyen de transport privé et personnel. Les composants métiers sont spécifiés à l'aide d'un autre patron nommé patron de composant car il correspond à un composant (cf. Figure 9). Pour situer le rôle de ce patron, il est possible de voir ce patron comme l'interface du composant au sens du concept d'interfaces d'objets. On retrouve dans ce patron un champ « nom », l'« intention » qui correspond à l'action que devra accomplir le composant, le « contexte » dans lequel il s'applique et les problèmes qu'il résout, les « données d'entrée obligatoires » qui devront être renseignées par l'utilisateur, les « données d'entrée facultatives » qui ne sont pas indispensables au fonctionnement du composant métier (au sens de l'analyse qu'il permet de réaliser) mais qui peuvent permettre de fournir plus de précision, les « données de sortie » correspondent au résultat attendu suite à l'utilisation de ce composant. Par exemple, l'utilisateur recherche une mise en relation de plusieurs données sous la forme d'un pourcentage. Il consultera la « rubrique » résultat pour trouver l'outil (composant métier) adapté. Enfin le champ « solution » permet de fournir une description succincte ou un algorithme plus précis. Ce champ permet de préciser comment le composant métier procède (i.e. quelle est la méthode utilisée). Par exemple, si il y a plusieurs méthodes pour calculer le pourcentage en question et si l'utilisateur préfère une méthode à une autre, il choisira son outil (composant métier) en fonction de ses préférences. L'ensemble des données renseignées est utile

Ces composants métier sont composés eux-mêmes d'un noyau fonctionnel et de composants d'IHM (au sens de structure d'IHM et pas de simple widget). Il peut y avoir un ou plusieurs composants IHM car ils peuvent être spécifiques à un contexte donné de manière à faciliter la plasticité (Hariri *et al.*, 2006). Les composants métier sont utilisés pour mener des analyses et fournissent chacun des résultats doivent être synthétisés. Or chaque composant métier intègre lui-même au moins une IHM. La problématique concernant la composition des IHM est apparue lorsqu'il a fallu permettre de fusionner des composants IHM intégrés dans des composants métier. Cette fusion doit être appliquée aux IHM dédiées à la présentation des résultats. La composition de composants techniques ou fonctionnels, ou de composants d'IHM (de type widget) est réalisable alors que la composition d'IHM complète est rarement abordée. Nos perspectives se dirigent dans cette direction. Etant donné qu'elles ne sont pas prises en compte dans l'étude de cas qui est présentée dans la section suivante, elles sont présentées en §5.

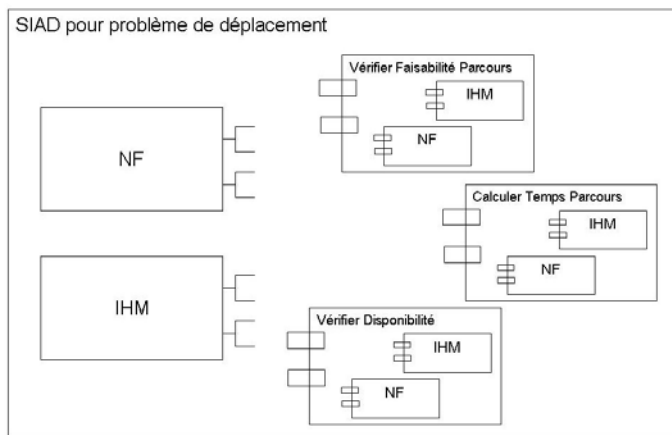


Figure 11. Architecture d'un SIAD pour le problème de déplacement

4 Application de la méthode dans le cadre des investissements ferroviaires

4.1 Phase de conception

L'approche ADESIAD proposée dans la section précédente a été appliquée au développement d'un SIAD dans le domaine ferroviaire. Le SIAD a pour objectif d'aider la prise de décision dans les investissements dans les infrastructures ferroviaires. Le développement du SIAD a suivi le cycle proposé par l'approche. Dans la phase descendante (dite de conception) et conformément à la méthode, les patrons pour l'investissement en infrastructure ont été définis. La décomposition du problème principal a été validée après trois tentatives ; les remises en cause ont été facilitées par les évaluations formatives placées au centre du modèle et faisant intervenir l'ensemble des acteurs. Le problème d'investissement a été scindé en trois sous-problèmes dont un est relatif à l'analyse de l'infrastructure par rapport au trafic ferroviaire, un à l'analyse de l'infrastructure seule et un à l'analyse du trafic seul. Ceux-ci ont été décomposés à leur tour pour donner l'arborescence de patrons de problème vue en Figure 12 (seuls les noms de problèmes sont indiqués, les patrons de composants ne figurent pas sur la figure).

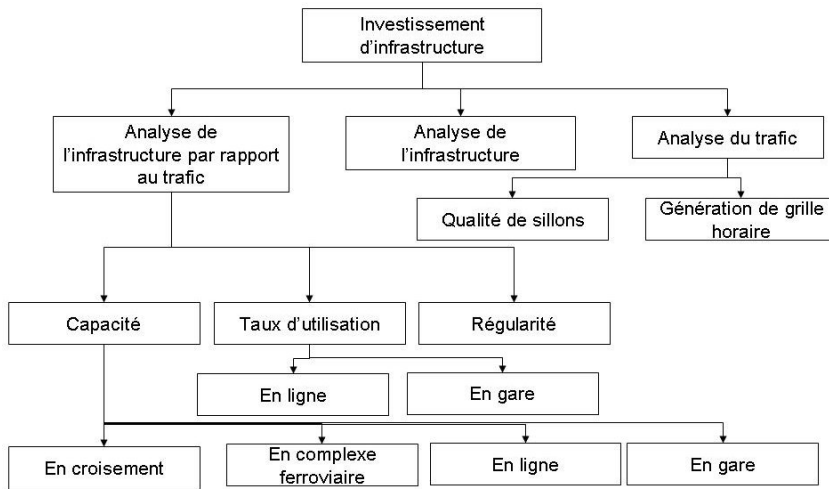


Figure 12. Décomposition du problème d'investissement dans les infrastructures ferroviaires

Les patrons du problème principal et d'un de ses sous-problèmes feuilles sont visibles de manière plus complète en Figure 13 et Figure 14. On remarque que le problème de capacité en ligne, vu en Figure 14, indique quatre analyses possibles : l'insertion d'un type de sillon¹, d'un type de sillon cadencé, d'un ensemble de sillons hétérogènes ou l'analyse de l'implantation de la signalisation.

Un composant, associé à une de ces analyses et qui consiste à insérer un type de sillon, est présenté en Figure 15. Ce composant n'existe pas encore ; il devra donc être développé, intégré à la base de composants et pourra ensuite être intégré lors de la phase de validation, après validation du SIAD avant intégration des composants métier.

Nom du problème	Investissement d'infrastructures
Définition du problème	Adaptation de l'infrastructure à la demande d'utilisation
Attributs	Infrastructure de base, besoin (demande)
Problème(s) père	Non
Sous problème(s)	Analyse de l'infrastructure par rapport au trafic Analyse de l'infrastructure Analyse du trafic
Problèmes connexes	Non
Analyse	Non définie
Solutions	Non définies

Figure 13. Patron associé au problème d'investissement

¹ Un sillon est la capacité d'infrastructure requise pour faire circuler un train donné entre deux points d'un réseau ferré pendant une période de temps donnée.

Nom du problème	Capacité en ligne
Définition du problème	La capacité ferroviaire correspond au nombre maximum de sillons qui peuvent s'intégrer sur une portion d'infrastructure durant un intervalle de temps donné.
Attributs	Infrastructure, grille horaire, besoin : quantité et type de sillons
Problème(s) père	Capacité
Sous problème(s)	Non
Problèmes connexes	Capacité en croisement sur un linéaire (intersection), en gare (quai), en nœud (cisaillement)
Analyse	Insertion d'un type de sillon Insertion d'un type de sillon avec cadencement Insertion d'un ensemble de sillons hétérogènes Analyse de l'implantation de la signalisation
Solutions	Modification du système d'espacement (la modification peut s'effectuer entre un block manuel, un block automatique lumineux ou la planification des signaux). Installation d'une voie de dépassement. Accroissement de la vitesse

Figure 14. Patron du (sous)-problème de capacité en ligne

Nom = Insertion_d'un_type_de_sillon	
Intention	Insérer un sillon de type donné dans une grille horaire
Contexte	Analyse de la capacité résiduelle
Données d'entrée obligatoires	Infrastructure, grille horaire, type de sillon à insérer, temps d'espacement
Données d'entrée facultatives	Cantons pénalisants
Données de sortie	Planifications, sillons, timetable, horaires, trafic, graphique, capacité, réserve, BAL, espacement
Solution	Algorithme grossier ou description de la méthode d'insertion

Figure 15. Patron du composant métier d'insertion d'un sillon dans une grille horaire

Indépendamment, les profils utilisateurs sont définis. En nous inspirant de (Robert, 2003), trois types d'utilisateurs ont été définis pour cette application : le décideur, le chargé d'études ferroviaires intermédiaires et le chargé d'études ferroviaires expert. Un des profils types correspondant est présenté en Tableau 1. L'hypothèse de départ est que tous les utilisateurs doivent être initiés au monde ferroviaire, ils doivent connaître le vocabulaire utilisé (dans le profil, cette information se situe dans "travail et expérience" : "connaissance de la tâche"). Cette hypothèse est due au fait que le métier est très spécifique et qu'un système n'est pas destiné à des débutants dans ce métier. Le niveau de compétence dans le domaine des études ferroviaires varie selon l'expérience de l'utilisateur et influe sur l'utilisation du système.

Une partie des interactions homme-machine a été modélisée par un diagramme états-transitions de UML, cf. Figure 16. Cette modélisation, réalisée en présence d'experts, d'utilisateurs, de spécialistes en IHM et d'informaticiens, a permis de définir le nombre d'écrans à utiliser, l'enchaînement des vues, les modalités de dialogue homme-machine. Pour chacune des vues, les modes de présentation des informations sont définis en collaboration avec l'expert, l'utilisateur et l'informaticien. Les modes d'activation des différents outils d'aide sont déterminés. Les tâches du processus de décision ont été synthétisées en trois ensembles d'activités qui sont le choix des outils, l'utilisation des outils et l'analyse des résultats.

Le premier type d'activité (choix des outils) est basé sur les patrons de problème qui ont été définis. Le choix des outils intègre trois types d'aide traduits par différents modes de présentation des outils. Le mode « par liste » présente l'ensemble des outils. Le mode « par mots-clés » permet une sélection selon des mots-clés proposés. Le troisième mode d'affichage suit la décomposition du problème faite en analyse et est nommé « par thème ». Désormais, les composants métiers sont vus comme des outils pour l'utilisateur. L'utilisation des outils correspond donc à l'utilisation des composants métier. L'analyse des résultats consiste théoriquement à composer une interface avec les résultats hétérogènes issus des divers composants métier. Cette étape n'a pas été réalisée lors de ce développement car elle dépend des recherches en cours sur la composition des IHM (cf. §5).

Déficiences : NON
Formation et habiletés : <ul style="list-style-type: none"> • Niveau de formation : variable • Vitesse de lecture : moyenne • Niveau d'habileté au clavier : moyen.
Travail et expérience : <ul style="list-style-type: none"> • Catégorie d'emploi : chargé d'études, • Connaissance de la tâche : expert • Lieu de travail : bureau.
Connaissance et utilisation du système : <ul style="list-style-type: none"> • Connaissance du système informatique : intermédiaire • Connaissance d'autres systèmes semblables à celui que l'on développe : bonne, • Type d'utilisation du système : discrétionnaire, • Fréquence d'utilisation du système : non définie.
Connaissances informatiques
<ul style="list-style-type: none"> • Générales : faibles, • Système d'exploitation : Windows, • Outils pour Internet : courriels, engins de recherche, • Famille de logiciels : bureautique.
Aspects psychologiques : <ul style="list-style-type: none"> • Attitudes : neutres, • Motivation : élevée.

Tableau 1. Profil type des utilisateurs "chargé d'études ferroviaires experts"

A partir de cette analyse, le maquettage puis le prototypage sont réalisés (pour plus d'informations sur les principes et outils disponibles à ce sujet, le lecteur peut se référer à l'article de Vanderdonck et Coyette (2007)). Ils permettent de définir la structure de la page que souhaite rencontrer l'utilisateur. Dans le cas de cette application, il y a deux fenêtres principales. La première permet le choix des outils permettant d'aboutir à une décision (cf. Figure 19). Dans la fenêtre de présentation des outils, on remarque sur la partie gauche, l'arborescence des problèmes. Quand l'utilisateur sélectionne un problème, les outils qui lui sont associés sont affichés dans la partie droite. L'affichage se fait sous la forme d'un tableau présentant le nom, la définition et éventuellement l'état de disponibilité de l'outil (par exemple dans le cas de la programmation par composant, des outils peuvent être prévus mais pas encore intégrés). Ceci donne, à la fin du premier cycle, l'interface montrée en Figure 17. Dans cette interface, l'ensemble des champs des patrons de problème a

été présenté à l'utilisateur pour le guider dans ses choix. La seconde fenêtre concerne la présentation des résultats (cf. Figure 18). Elle permet de montrer que l'objectif des utilisateurs (experts) est de pouvoir comparer les résultats d'analyses effectuées avec les mêmes outils mais sur des études différentes. En effet, dans cette étude de cas, le problème de décision, c'est-à-dire d'investissement, est souvent accompagné de plusieurs propositions ; la société doit alors vérifier l'opportunité de ces propositions (d'où le nom « étude »). L'expert doit mener les études et présenter l'ensemble des résultats aux décideurs.

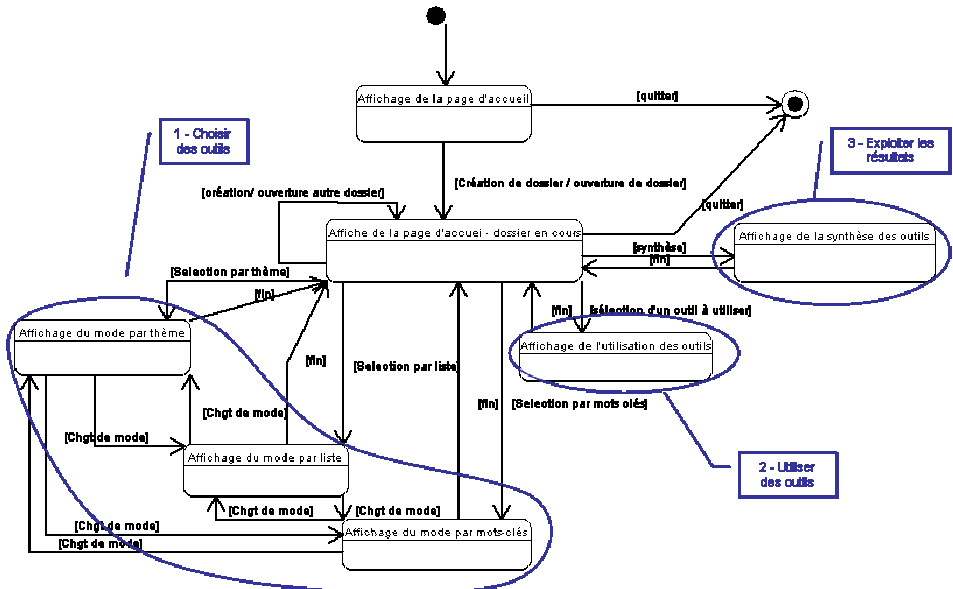


Figure 16. Diagramme états-transitions pour la spécification des IHM

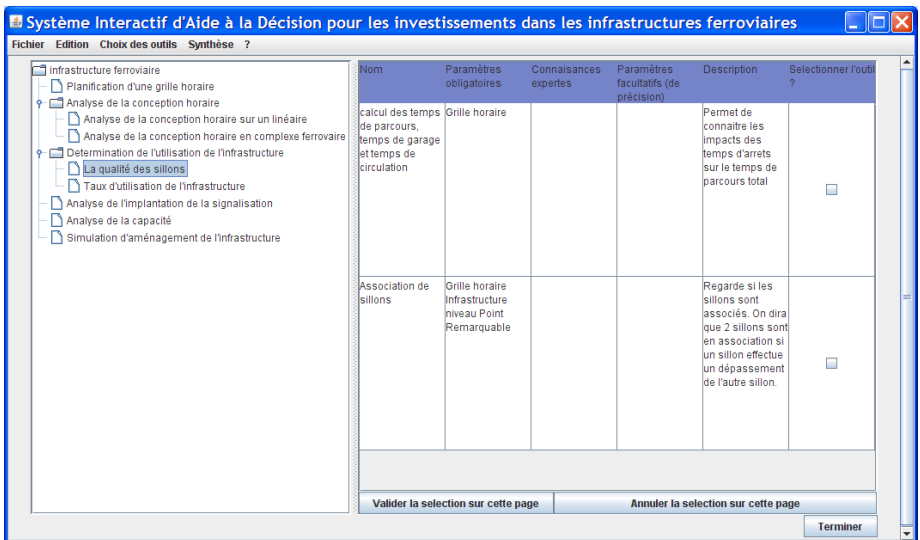


Figure 17. Interface de choix des outils (ou composants métier)

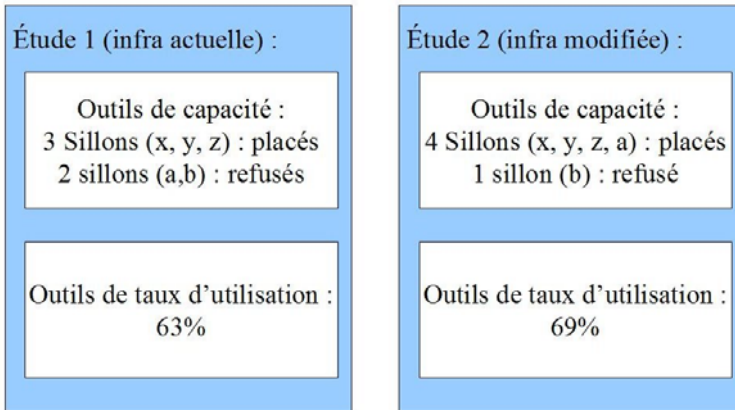


Figure 18. Maquettage de la présentation des résultats

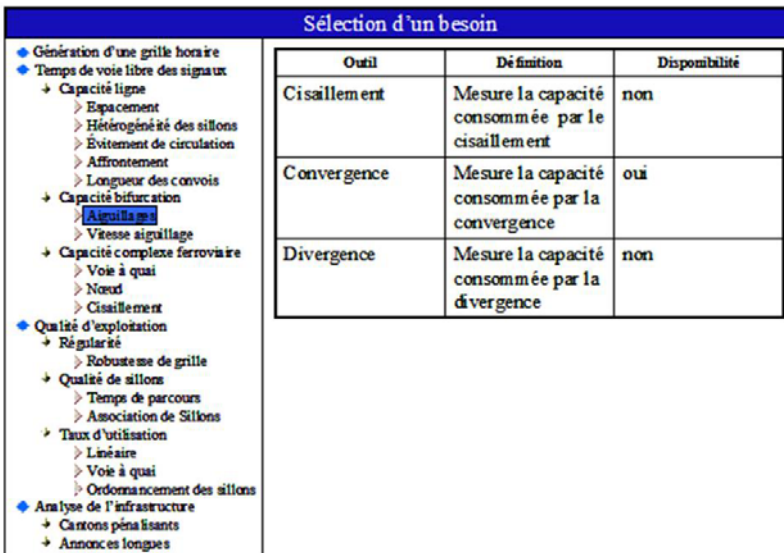


Figure 19. Maquettage de la fenêtre de choix des outils

4.2 Phase ascendante d'évaluation, estimation par les utilisateurs

Les évaluations qui ont été menées consistent à valider l'utilisation des patrons associés pour l'aide à la prise de décision. Selon le cycle de développement, la première étape de validation concerne le système indépendamment des composants métier. L'objectif de cette étape est de valider l'intérêt des supports pour l'aide à la décision ; seule l'activité de choix des outils visant l'utilisation de patrons est présentée.

Le protocole est centré sur l'évaluation de l'interaction et l'influence des modes de présentation des outils sur les utilisateurs. Pour mener à bien cette évaluation, des futurs utilisateurs ont été mis en situation la plus réelle possible. L'évaluation s'est déroulée en quatre étapes : (1) la présentation du système et des modes de choix qui lui sont proposés, (2) la familiarisation de l'utilisateur avec le système, (3)

une première mise en situation avec un problème issu du domaine. L'utilisateur a dû utiliser le SIAD pour choisir les outils adaptés à son problème et a répété cette action trois fois pour chacun des modes d'aide au choix. A la fin de chaque sélection, l'utilisateur a rempli un questionnaire sur son impression par rapport au mode utilisé et les résultats obtenus. A la fin, il a rempli un questionnaire permettant d'évaluer le SIAD selon les critères d'utilisabilité de Bastien et Scapin (2001). (4) L'utilisateur a reproduit le même scénario avec deux autres problèmes caractéristiques du domaine mais en utilisant les modes selon un ordre préétabli. Les documents recueillis sont les questionnaires sur les problèmes d'investissement ferroviaires, le questionnaire d'évaluation ergonomique de l'IHM et l'enregistrement vidéo (image et son) contenant l'ensemble des actions exécutées par l'utilisateur. Celui-ci a été enregistré à partir de la première mise en situation jusqu'à la fin de l'expérimentation. Trois utilisateurs de profils différents ont participé à ces évaluations. L'utilisateur dit « novice » a des notions issues du domaine ferroviaire mais ne travaille pas directement dans le domaine de l'expertise. L'utilisateur débutant est un chargé d'étude ferroviaire ayant une expérience de deux ans et l'utilisateur dit « expert » est un chargé d'étude ferroviaire de trente ans d'expérience. Nous n'avons pas souhaité faire participer plus de monde car ces évaluations ne sont pas les dernières de la phase d'évaluation.

On constate que l'utilisateur considéré comme expert a préféré commencer l'étude avec le mode de recherche par liste car il ne lui semblait pas utile d'être aidé par les autres mécanismes. L'utilisateur débutant a choisi le mode thème car il a apprécié la structure arborescente le guidant dans le choix des outils en fonction du problème ; d'après cet utilisateur, ce mode a l'avantage de lui éviter d'oublier des outils. Le troisième utilisateur, dit novice, a utilisé le mode mots-clés car il lui semblait lui permettre de trouver une sélection d'outils adaptés à son problème. A la fin des études, l'utilisateur expert gardera sa préférence pour le mode Liste, le débutant celle pour le mode Thème et l'utilisateur novice celle pour les modes Liste et Mots-clés. On notera que l'utilisateur novice a été gêné lors de l'utilisation du SIAD avec le mode Thème car des outils ont été proposés plusieurs fois dans des thèmes différents. Les autres utilisateurs n'ont pas été gênés et ont été globalement en accord avec les propositions de ce mode.

Après chaque utilisation concernant un mode donné, il a été demandé aux utilisateurs de quantifier leur satisfaction par rapport à la sélection d'outils faite. Il en ressort que les trois utilisateurs ont jugé les résultats sortants du mode « recherche par liste » comme sensiblement meilleurs par rapport aux deux autres modes. On remarque par ailleurs que les utilisateurs expert et débutant n'ont pas apprécié les résultats sortants de l'usage du mode par Mots-clés. Or pour l'utilisateur expert les outils sélectionnés dans le mode Mots-clés sont plus nombreux que dans les autres modes. Ce mode Mots-clés a donc correctement proposé les outils (sans manque). Concernant l'utilisateur débutant, la sélection est rigoureusement la même entre le mode Liste et le mode Mots-clés. Pour cet utilisateur, la différence de sélection entre deux outils est due à un changement d'avis concernant la stratégie à suivre pour résoudre le problème, indépendant de l'utilisation du système. La seule différence qui est due au mode de recherche se situe au niveau d'un outil non sélectionné lors de l'utilisation du mode Thème car celui-ci n'était pas proposé. A contrario, un des outils n'a pas été sélectionné par l'utilisateur expert lors de l'utilisation du mode Thème alors que ce mode le proposait ; on peut supposer que le problème est de lisibilité. L'outil ne devait pas être présent là où l'utilisateur s'attendait à le trouver. C'est principalement pour cette raison que l'utilisateur expert est moins satisfait des résultats obtenus à l'aide du mode Thème.

D'un point de vue utilisabilité, le principal reproche concernait le manque de retour d'information particulièrement dans le mode 'Thème ; ce mode proposait les outils dans plusieurs tableaux. Des outils pouvaient se retrouver dans plusieurs thèmes. Le manque de retour d'information obligeait les utilisateurs à se souvenir (ou à prendre des notes) des outils au fur et à mesure de la sélection. Cela s'est répercuté sur leur notation concernant le guidage. Une amélioration est d'ores et déjà prévue pour ce point à l'itération suivante.

En conclusion, on peut dire que les utilisateurs ont apprécié l'organisation de l'interface. Le tableau présentant l'ensemble des outils a été très largement utilisé pour leur choix de composants. Le mode thème utilisant la décomposition du problème supportée par les patrons pour la présentation des outils, a été majoritairement apprécié des utilisateurs novices. L'objectif de capitalisation et de réutilisation des connaissances expertes a donc été atteint. Les étapes suivantes dans le processus de décision sont l'utilisation des outils et l'analyse des résultats obtenus par les différentes analyses avec les outils sélectionnés. Pour faire cette analyse, les résultats variés doivent être présentés aux utilisateurs. Cette présentation implique la composition des résultats. Cette problématique de composition et les premiers résultats sont présentés dans la partie suivante.

5 Interfaces des composants métiers et de leurs résultats

Nous nous intéressons à la composition d'IHM issues de divers composants métier (Lepreux et al., 2004). Les travaux relatifs à l'intégration de composants de Grundy et Hosking (2002) ont montré qu'assembler des composants avec interface utilisateur engendrait des interfaces inconsistantes et des complications. Ils montrent donc le besoin d'interface utilisateur adaptable et le besoin d'introspection et de compréhension des autres interfaces utilisateur.

Grundy et Hosking (2002) proposent quatre sortes d'adaptation :

1. *Extension* : un composant interface intègre les éléments des autres en les contrôlant et pour respecter le même style (respect du critère d'utilisabilité de Bastien et Scapin (2001) de cohérence)
2. *Composition* : les informations de plusieurs composants sont composées dans une interface. Elle doit éviter d'utiliser plusieurs fenêtres.
3. *Reconfiguration* : d'autres composants logiciels peuvent avoir besoin de reconfigurer l'interface d'un composant.
4. *Adaptation à l'utilisateur, au rôle et à la sous-tâche* : les utilisateurs spécifient leur préférence sur les éléments ou interfaces alternatives qu'ils veulent utiliser et quelles approches d'adaptation ils préfèrent.

Dans le cas du SIAD, l'adaptation par extension peut être utilisée lors du second type d'activité (cf. Figure 16) concernant l'utilisation des outils (composants métier). De cette manière c'est le composant d'interface du SIAD qui intègre les interfaces des composants métier et les contrôle. La vue des modules composant le SIAD (cf. Figure 20) met en avant le module de présentation des composants qui est utilisé pour cette étape. On remarque que les composants métier sont placés à l'extérieur du SIAD de manière à montrer l'indépendance du SIAD par rapport à ces composants. Par contre le module « présentation des résultats » doit pouvoir présenter l'ensemble des résultats associés aux études soit par étude soit par outils ou les deux à la fois. Dans ce cas l'adaptation par composition doit être utilisée. On peut également laisser le choix à l'utilisateur pour qu'il choisisse le mode d'affichage

le plus approprié à sa tâche de décision et à ses préférences personnelles. Dans ce cas, la quatrième adaptation proposée par Grundy et Hosking peut être utilisée.

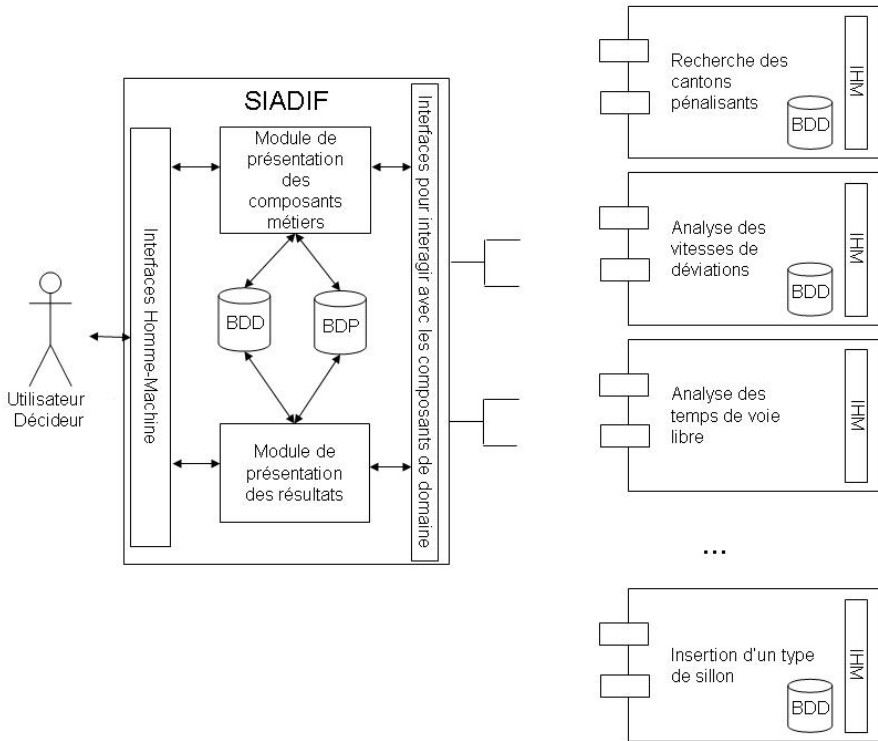


Figure 20. Description des modules du SLAD et positionnement des composants métiers

Pour réaliser l'objectif de composition des interfaces, nous proposons d'utiliser des opérateurs ensemblistes utilisables sur des ensembles logiques. Dans un premier temps, nous nous sommes focalisés sur la composition d'interfaces graphiques à « la conception » (Lepreux et Vanderdonck, 2006). A partir d'un langage de description d'interfaces utilisateur (UIDL²) en XML, des opérateurs ensemblistes permettent de manipuler/composer/décomposer des interfaces graphiques. Ces travaux s'inscrivent dans une démarche d'ingénierie dirigée par les modèles et respectent le cadre de référence CAMELEON (Calvary et al., 2003) pour des interfaces utilisateur multi-cibles. Nous avons utilisé le langage UsiXML (User Interface eXtensible Markup Language – <http://www.usixml.org>) (Limbourg et al., 2005) car il suit cette démarche mais d'autres UIDL auraient pu être utilisés (UIML (Ali et al., 2004), XIML (Eisenstein et al., 2001), SUNML (Dery-Pinna et al., 2003)...). Les travaux relatifs au projet RAINBOW (Dery-Pinna et al., 2003) visent le même objectif à partir de spécification de l'IHM en SUNML.

Le langage que nous avons choisi permet de définir l'interface à plusieurs niveaux d'abstraction. Le niveau tâche et domaine est le plus élevé. Ensuite il peut être dérivé au niveau de l'interface abstraite, c'est-à-dire indépendante de la modalité et de la plateforme. Le modèle de l'interface concrète est quant à lui spécifique à la modalité (graphique, vocale...) alors que l'interface finale est spécifique à la

² En anglais : User Interface Description Language

modalité et à la plateforme cible. Pour la composition d'interface, nous proposons un ensemble d'opérateurs utilisables sur des fichiers en XML adaptés au domaine des IHM et plus particulièrement aux interfaces utilisateurs concrètes (niveau CUI). Un exemple de définition d'une interface au niveau CUI est montré en Figure 21 sous la forme d'un arbre. On remarque que les éléments sont répartis en groupe logique (box). L'objectif de ces opérateurs est de manipuler un ensemble logique d'éléments en préservant la structure initiale de l'IHM.

Par exemple, l'opérateur de sélection prend comme argument un arbre XML et la structure à sélectionner dans l'arbre. La Figure 22 montre une interface graphique (au niveau concret) et la structure qui permet de sélectionner tous les éléments de type output dans l'arbre. Le résultat graphique est présenté ainsi que sa définition en UsiXML. L'intérêt de cet opérateur est qu'il permet de sélectionner des structures d'arbres et donc, par ce biais, permet de manipuler/préserver la structure logique de l'IHM et pas seulement les éléments sans ordre ni logique. D'une part, cela permet de satisfaire le critère de cohérence au sein du composant métier.

D'autre part, cela permet de conserver les ensembles d'informations par exemple relatives à une tâche. Dans ce cas pour extraire les éléments de l'interface relative à une tâche, il suffit de sélectionner le sous-arbre de l'IHM relatif à cette tâche.

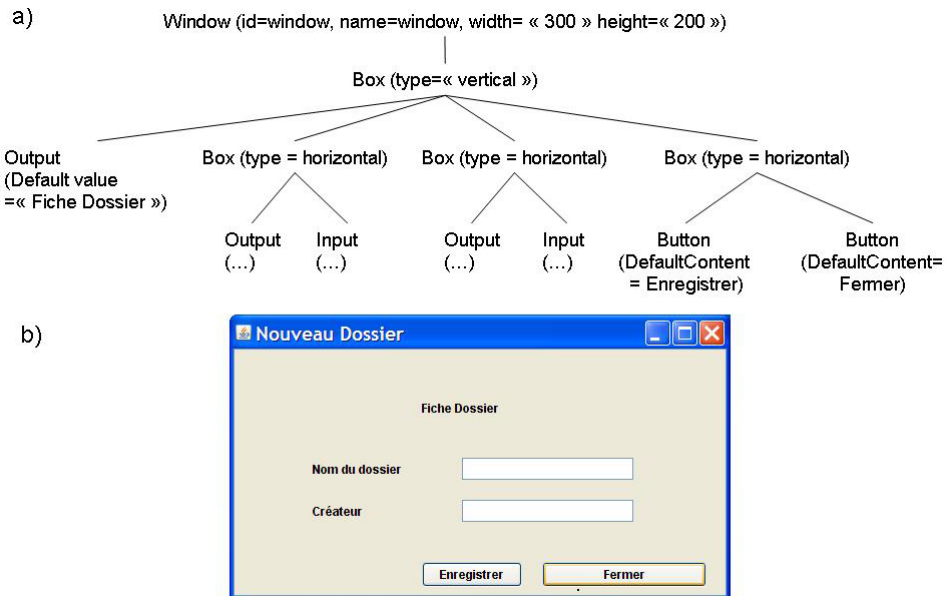


Figure 21. Arbre (a) du niveau CUI d'UsiXML associé à une interface utilisateur créée à l'aide de GrafiXML (b)

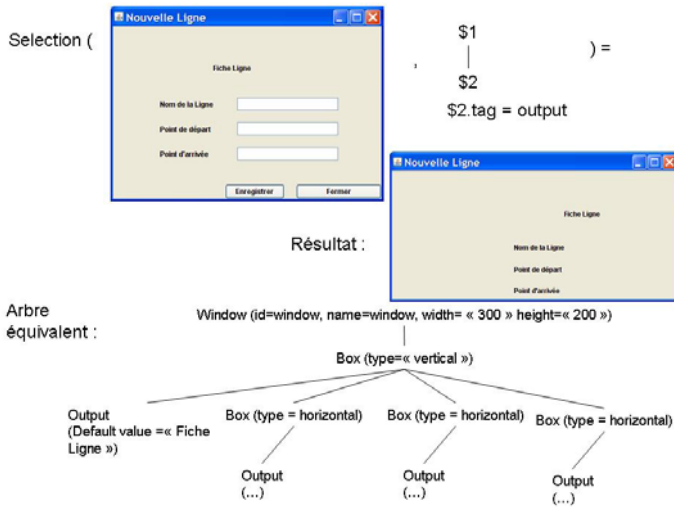


Figure 22. Représentation graphique de l'opérateur de sélection

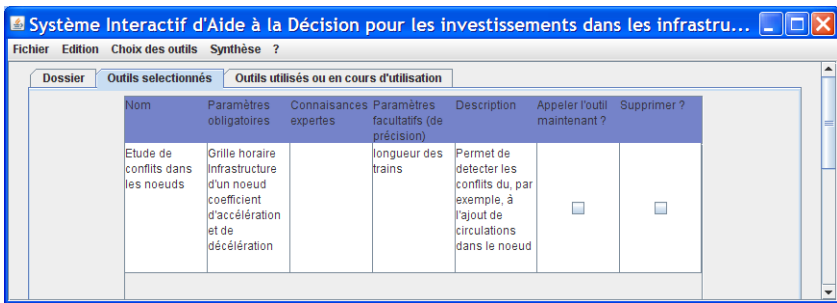


Figure 23. Interface de SIAD concernant le choix d'un outil

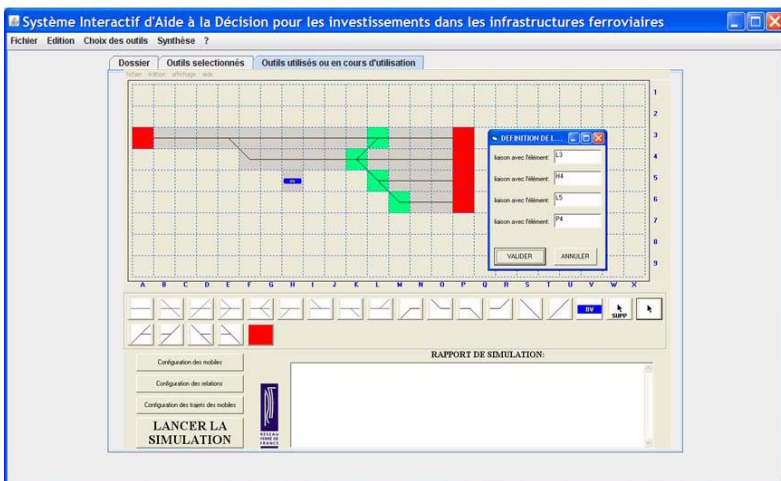


Figure 24. Interface de SIAD lors de l'utilisation de l'outil « étude de conflits dans les nœuds »

Cet opérateur permet de faire la composition par extension. Dans ce cas l'ensemble des éléments (ou une partie) de l'interface du (des) composant(s) métier sont sélectionnés puis intégrés dans l'interface du SIAD. Par exemple, l'outil d'analyse des conflits dans les nœuds, sélectionné par l'utilisateur dans l'interface de présentation des outils du SIAD (Figure 23), est intégré dans le SIAD (Figure 24). Il permet la configuration des voies du complexe ferroviaire (i.e. nœud). Il est possible de ne pas sélectionner la partie « rapport de simulation » de manière à ne l'afficher qu'au moment de la synthèse des résultats si l'utilisateur le souhaite. Pour ce cas, nous avons choisi d'afficher les résultats de manière individuelle en plus de la synthèse récapitulative donc nous avons choisi de laisser la possibilité à l'utilisateur d'afficher le rapport de simulation.

L'opérateur « union » est particulièrement utile pour la composition d'interfaces. Il prend en argument deux interfaces et retourne l'interface qui les compose. Des options sont disponibles quant à la stratégie de traitement des doublons : répéter les doublons, ne pas répéter les doublons, supprimer les doublons. Des options permettent également de préciser le sens de la composition : horizontale ou verticale. Un exemple est appliqué à deux interfaces de saisies, cf. Figure 25. Dans cet exemple l'utilisateur (ici le concepteur de l'IHM) souhaite faire l'union sans répétition des doublons et souhaite avoir une union horizontale. Les opérations qui sont menées par l'opérateur d'union sont illustrées sur l'arbre résultat en Figure 26. L'algorithme fusionne les deux interfaces suivant le souhait (horizontal et vertical) du concepteur. La taille de la fenêtre résultante est alors ajustée. Ensuite la recherche des doublons est effectuée.

Pour le moment cette recherche est basée sur le type et les intitulés des champs. Il est possible de faire des recoupements avec les actions qui sont associées aux éléments. Dans cet exemple, l'algorithme détecte les boutons comme étant des doublons. Le choix du concepteur étant de ne pas dupliquer les doublons, une partie de l'arbre est alors sectionnée. Le résultat graphique est visible en Figure 25 partie inférieure.

Cet opérateur est donc utilisable pour la composition des interfaces résultats des outils. En effet, nous avons vu que l'utilisateur souhaite obtenir une interface regroupant l'ensemble des résultats pertinents à son problème et associé aux analyses menées (cf. Figure 18). Dans un premier temps les opérateurs de sélection (permettant de cibler les informations nécessaires) et union (permettant de les regrouper) sont utilisés. Un exemple de résultat attendu est montré en Figure 27. Cette interface permettra au décideur d'évaluer les bonnes ou mauvaises solutions d'investissement et donc de prendre une décision. Pour le moment, ce type de composition n'est pas encore réalisable car les interfaces complexes vues en solutions dans la Figure 27 ne sont pas descriptibles en UsiXML (ni en XML). Une des perspectives à ces travaux est donc d'étendre les opérateurs de composition à des interfaces plus complexes.

De plus, pour réaliser l'objectif d'adapter les IHM en fonctions des tâches, une autre approche vise à intégrer l'arbre des tâches dans le composant métier (Lewandowski et al., 2006, 2007b) : à ce sujet, nous avons commencé à traiter la perspective de composer les interfaces au niveau des tâches (Lewandowski, Lepreux et Bourguin, 2007a).

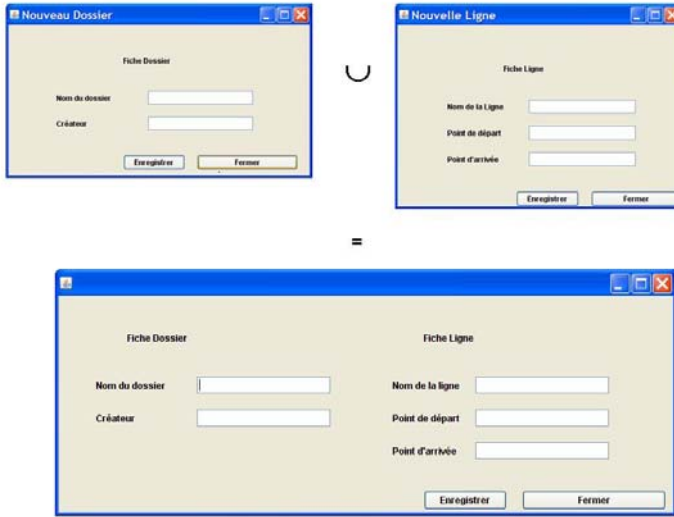


Figure 25. Représentation graphique de l'opérateur « union » (non répétition des doublons et union horizontale)

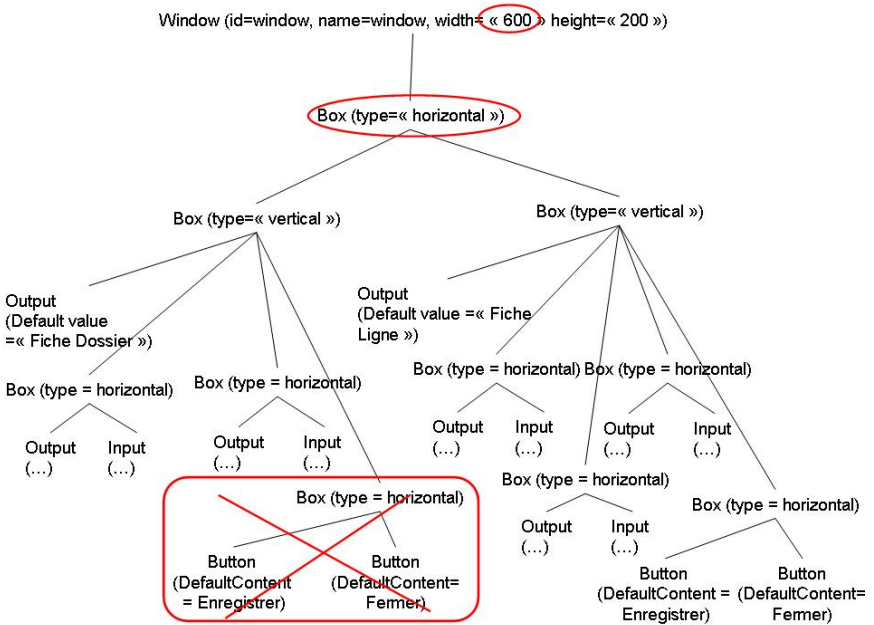


Figure 26. Arbre résultat de l'opérateur d'Union relatif à l'exemple de la Figure 25

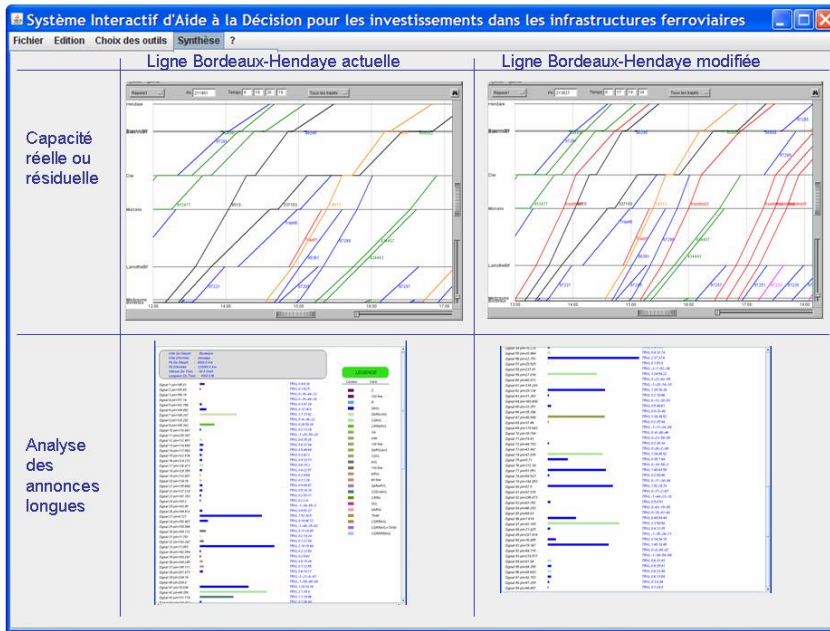


Figure 27. Résultat attendu des opérateurs Sélection et Union pour la synthèse des résultats

6 Conclusion

Les principaux cycles de développement issus de divers domaines en lien avec les SIAD ont été présentés. Pour nous le SIAD est un moyen d'assistance proposant un ensemble d'outils et une aide pour faciliter l'ensemble du processus de décision. Les modèles de développement existants ont montré des avantages mais aussi souvent des insuffisances pour le développement des SIAD. Un cycle et une méthode couplant des principes tels que la prise en compte de l'utilisateur, la gestion des connaissances, la réutilisation issus de ces cycles ont été présentés. Des supports à la prise en compte des utilisateurs et des experts pour le développement et la réutilisation ont été proposés. La structure des patrons proposée est adaptable aux cas d'application des SIAD. L'étude de cas dans le domaine ferroviaire a montré l'intérêt de ces supports (patrons et composants métier) pour l'aide à la décision et notamment dans un premier temps dans la définition et le choix des outils. L'approche qui a été proposée pourrait encore être affinée et approfondie. Une plateforme la supportant serait une réelle aide dans le cadre du développement, notamment pour faciliter la gestion des patrons de domaine et de composants introduits lors d'un développement. Le SIAD tel qu'il est développé à l'aide de l'approche permet de manipuler un certain nombre d'outils. L'étape finale du processus de décision (la synthèse des analyses de manière à aboutir à une décision) dépend de la problématique de composition d'IHM à laquelle s'ajoute la problématique de la présentation des résultats pour aider à une décision. La problématique de la composition des IHM, due à l'intégration de composants métier possédant des IHM a été introduite. Des réflexions ont été présentées mais les solutions proposées ne sont pas encore suffisantes pour permettre la composition d'IHM intégrées à des composants métier et devant « fusionner » dans un système unique tel que le SIAD. D'une part la composition telle qu'elle a été

proposée n'est valable que pour des IHM spécifiées avec un langage structuré ; or les interfaces fournies par des outils associés à des SIAD peuvent être de toute nature et peuvent sortir du cadre des langages de spécification d'IHM. D'autre part, les solutions proposées sont pour le moment valables à la conception ; or il faudrait que la composition se fasse à la volée (à l'exécution) lorsque le décideur choisit de voir la synthèse des outils qu'il a utilisés pour ses études. Une perspective liée à d'autres travaux intégrant les arbres de tâches dans les composants métier a été abordée et nous semble très importante en perspective de nos recherches.

Remerciements

L'auteur tient à remercier les experts de Réseau Ferré de France pour leur participation dans ce travail et la région Nord - Pas-de-Calais qui a participé au cofinancement de la thèse avec RFF. Ce travail bénéficie aussi d'idées provenant des projets TACT MIAOU et EUCUE supportés par la région Nord- Pas-de-Calais et le FEDER et du groupe de travail CESAME. Je remercie également Jean Vanderdonck qui a participé aux travaux sur la composition d'IHM ainsi que les relecteurs qui ont permis de considérablement améliorer l'article.

7 Références

- Abed, M. (2001). *Méthodes et Modèles formels et semi-formels de conception et évaluation des systèmes homme-machine*. Mémoire d'HDR de l'Université de Valenciennes et du Hainaut-Cambrésis.
- Ali, M.F., Pérez-Quiñones, M.A., Abrams, M. (2004). Building Multi-Platform User Interfaces with UIML. In *Multiple User Interfaces: Engineering and Application Framework*, Seffah, A., Javahery, H. (eds.). John Wiley, Chichester, 95–118.
- Bastien, J., Scapin, D. (2001). Evaluation des systèmes d'information et critères ergonomiques. In *Environnements évolués et évaluation de l'IHM. Interaction Homme Machine pour les SI*, Kolski, C (ed.), Éditions Hermès, Paris, 53-79.
- Boehm, B. (2006). Some Future Trends and Implications for Systems and Software Engineering Processes. *Systems Engineering*, Vol. 9, No. 1, 1-19.
- Borchers, J. (2001). *A pattern approach to interaction design*. J. Wiley and Sons, Chichester.
- Buisine A. (1999). *Vers une démarche industrielle pour le développement d'Interfaces Homme-Machine, De l'analyse de l'activité à la génération du code*. Thèse de l'Université de Rouen.
- Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonck, J. (2003). A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers*, Vol. 15, num. 3, 289–308.
- Calvary, G., Coutaz, J., Daassi, O., Ganneau, V., Balme, L., Demeure, A., Sottet, JS. (2006). Métamorphose des IHM et Plasticité : Article de synthèse. In *10ème conférence ERGO-LA, L'humain comme facteur de performance des systèmes complexes*, Brangier, E., Kolski, C., Ruault, J.R. (eds). ESTIA & ESTIA.INNOVATION, Biarritz, France, Octobre, 79-86.
- Calvary, G., Coutaz, J. (2007). Métamorphose des IHM et Plasticité. *Revue d'Interaction Homme-Machine*, Vol. 8(1), 35-59.

- Caulier P. (1997) *Méthodologie de capitalisation et de réutilisation de connaissances pour l'aide à la supervision des procédés automatisés complexes - Application à la supervision du trafic téléphonique de l'Île de France*. Thèse de l'Université de Valenciennes et du Hainaut-Cambrésis.
- Cauvet, C., Rieu, D., Front-Conte, A., Ramadour, P. (2001) Réutilisation dans l'ingénierie des systèmes d'information. *Revue Ingénierie des Systèmes d'Information (ISI)*, Cauvet C. et Rosenthal-Sabroux C., Hermès Science Publication, Paris, 115-148.
- Cauvet, C., Semmak, F. (1999). La réutilisation dans l'ingénierie des systèmes d'information. In *Génie Objet : Analyse et Conception de l'évolution*, Hermès, Paris. 25-55.
- Chen, Z. (1995). Interacting with software system components. *Decision Support Systems*, Vol. 14, num. 4, 349-357.
- Cloux, P.-Y. (2003). *RUP, XP Architectures et Outils - industrialiser le processus de développement*. Dunod, Paris.
- Coulangue, B. (1996). *Réutilisation du logiciel*. MASSON, Paris.
- Dao, M., Dony, D. (2001). La réutilisation. *Technique et science informatiques (TSI)*, Vol. 20, num. 4, 461-487.
- Dery-Pinna, A.-M., Fierstone, J., Picard, E. (2003). Component Model and Programming: a First Step to Manage Human-Computer Interaction Adaptation. In *Proc. of 5th Int. Symposium on Human-Computer Interaction with Mobile Devices and Services MobileHCI'2003* (Udine, September 8-11, 2003), Lecture Notes in Computer Science, Vol. 2795, Springer-Verlag, Berlin, 456-460.
- Duribreux-Cocquebert M. (1995). *MODESTI : vers une méthodologie interactive de développement de Systèmes à Base de Connaissances*. Thèse de l'Université de Valenciennes et du Hainaut-Cambrésis.
- Eisenstein, J., Vanderdonckt, J., Puerta, A. (2001). Model-Based User-Interface Development Techniques for Mobile Computing. In *Proc. of 5th ACM Int. Conf. on Intelligent User Interfaces IUI'2001*, Lester, J. (ed.), Santa Fe, January 14-17, ACM Press, New York, 69-76.
- Front-Conte, A., Giraudin, J. P., Rieu, D. Saint-Marcel, C. (1999). Réutilisation et Patrons d'Ingénierie. In C. Oussalah (ed.), *Génie Objet : Analyse et Conception de l'évolution*, Hermès, Paris, 91-136.
- Gachet, A., Haettenschwiler, P. (2007). Development process of intelligent decision-making support systems: Review and perspective. In *Decision Engineering, Intelligent Decision-making Support Systems Foundations, Applications and Challenges*, G. A. F. Jatinder N. D. Gupta, M. M. T. (Eds.), Springer, London, , 97-121.
- Gamma E., Hem R., Johnson R., Vlissides J. (1994). *Design Patterns: Elements of reusable Object-Oriented Software*. Addison Wesley, Massachusetts.
- Grundy, J., Hosking, J. (2002). Developing adaptable user interfaces for component-based systems. *Interacting with Computers*, Vol. 14, 175-194.
- Hariri, M.-A., Tabary, D., Kolski, C. (2006). Démarche en vue de la Génération d'Interfaces Mobiles et Plastiques. In *Actes de la troisième conférence UBIMOB2006 Mobilité et Ubiquité*, Defude, B., Lecolinet, E. (Eds.), CNAM, Paris, 5-8 septembre 2006, ACM Conference Proceedings Series, septembre, 127-130.

- Hartson, H.R., Hix, D. (1989). Towards empirically derived methodologies and tools for Human-Computer development. *International Journal of Man-Machine Studies*, Vol. 31, 477-494.
- Hassine I., Rieu D., Bounaas, F., Seghrouchni, O. (2002). Symphony : un modèle conceptuel de composants métier. In *Revue Ingénierie des systèmes d'information (ISI)* numéro spécial Connaissances métier en ingénierie des systèmes d'information, Cauvet, C. (ed.) Paris Hermès Science Lavoisier, Vol. 7, num. 4, 35-59.
- Jaulent P. (1994). *Génie Logiciel, les méthodes*. A. Colin, Paris.
- Lepreux, S. (2005). *Approche de Développement centré décideur et à l'aide de patron de Systèmes Interactifs d'Aide à la Décision, application à l'investissement dans le domaine ferroviaire*. Thèse de l'Université de Valenciennes et du Hainaut-Cambrésis, juin.
- Lepreux, S., Abed, M., Kolski, C. (2003). A Human-centred methodology applied to decision support system design and evaluation in a railway network context. *Cognition, Technology & Work*, Vol. 5, 248-271.
- Lepreux, S., Kolski, C., Abed, M. (2004). IHM et SIAD : vers une composition d'outils interactifs pour l'aide à la prise de décision. In *Proceedings of IHM 2004, International Conference Proceedings Series*, ACM Press, Namur, pp. 227-230
- Lepreux S. (2006). Supports pour la prise en compte des experts et utilisateurs dans le développement de Systèmes Interactifs d'Aide à la Décision. In *10ème conférence ERGO-IA, L'humain comme facteur de performance des systèmes complexes*, Brangier, E., Kolski, C., Ruault, J.R. (eds). ESTIA & ESTIA.INNOVATION, Biarritz, France, Octobre, 139-146.
- Lepreux, S., Vanderdonckt, J. (2006). Towards Supporting User Interface Design by Composition Rules. In *Proc. of the 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006*, Springer-Verlag, Berlin, 231-244.
- Lévine, P., Pomerol, J.-C. (1989). *Systèmes interactifs d'aide à la décision et systèmes experts*. Hermès, Paris.
- Lewandowski A. Tarby J-C., Bourguin, G. (2006). Les Modèles de Tâches pour la Contextualisation des Composants. In *10ème conférence ERGO-IA, L'humain comme facteur de performance des systèmes complexes*, Brangier, E., Kolski, C., Ruault, J.R. (eds). ESTIA & ESTIA.INNOVATION, Biarritz, France, Octobre, 147- 154.
- Lewandowski, A., Lepreux, S., et Bourguin, G. (2007a). Tasks models merging for high-level component composition. In *Proceedings of HCI-International 2007*, Beijing, Chine, juillet, LNCS 4550, Julie A. Jacko (ed), 1129-1138.
- Lewandowski A., Bourguin, G, Tarby J-C. (2007b). De l'Orienté Objet à l'Orienté Tâches, des modèles embarqués pour l'intégration et le traçage des composants. *Revue d'Interaction Homme-Machine*, Vol. 8(1), 1-32.
- Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., Lopez, V. (2005). USXML: a Language Supporting Multi-Path Development of User Interfaces. In *Proc. of 9th IFIP Working Conf. on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems EHCI-DSVIS'2004* (Hamburg, July 11-13, 2004), Lecture Notes in Computer Science, Vol. 3425, Springer-Verlag, Berlin, 200-220.

Oussalah, M, Rieu, D. (eds) (2004). Ingénierie des composants dans les systèmes d'information. *Revue des sciences et technologies de l'information (RSTI), série L'Objet*, Numéro Spécial Vol. 10 (1).

Palma-dos-Reis, A., Zahedi, F. M. (1999). Designing personalized intelligent financial decision support systems. *Decision Support Systems*, Vol 26, num. 1, 31-47.

Rasmussen, J. (1983). Skill, rules and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-13, Num. 3, 257-267.

Robert, J.-M. (2003). Que faut-il savoir sur les utilisateurs pour réaliser des interfaces de qualité ? In *Ingénierie cognitive, IHM et cognition*, Boy, G., (ed.), Hermès Lavoisier, Paris, 249–283.

Szyperski, C. (1998). *Component-Software-Beyond Object-Oriented Programming*. MA: Addison Wesley.

Tamzalit, D., Aniorté, P. (eds) (2005). Ingénierie des composants et systèmes d'information. *Revue des sciences et technologies de l'information (RSTI), série L'Objet*. Hermès Science, Paris, Vol 11, num. 4.

Vanderdonckt, J., Coyette A. (2007). Modèle, méthodes et outils de support au prototypage multi-fidélité des interfaces graphiques. *Revue d'Interaction Homme-Machine*, Vol 8 (1), 91-123.

van Welie, M., van der Veer G.C. (2003). Pattern Languages in Interaction Design: Structure and Organization. In *Proceedings of Interact '03 (1-5 September, Zürich, Switzerland)*, M. Rauterberg, M. Menozzi, J. Wesson (eds), IOS Press, Amsterdam, The Netherlands, 527-534.