HAL
open science

Ingénierie dirigée par les modèles : quels supports à l'interopérabilité des systèmes d'information ?

Annie Authosserre-Cavarero, Frédéric Bertrand, Mireille Fornarino, Philippe Collet, Hubert Dubois, Stéphane Ducasse, Sophie Dupuy-Chessa, Catherine Faron Zucker, Cyril Faucher, Jean-Yves Lafaye, et al.

To cite this version:

Annie Authosserre-Cavarero, Frédéric Bertrand, Mireille Fornarino, Philippe Collet, Hubert Dubois, et al.. Ingénierie dirigée par les modèles : quels supports à l'interopérabilité des systèmes d'information ?. Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information, 2013. hal-00813675v2

HAL Id: hal-00813675

https://inria.hal.science/hal-00813675v2

Submitted on 24 Apr 2013

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.

Ingénierie dirigée par les modèles : quels supports à l'interopérabilité des systèmes d'information ?

Annie Authosserre-Cavarero, Frédéric Bertrand, Mireille Blay-Fornarino, Philippe Collet, Hubert Dubois, Stéphane Ducasse, Sophie Dupuy-Chessa, Catherine Faron-Zucker, Cyril Faucher, Jean-Yves Lafaye, Philippe Lahire, Olivier Le Goaer, Johan Montagnat, Anne-Marie Pinna-Dery

Action Spécifique : « Interopérabilité des SI et ingénierie des modèles »
blay@unice.fr

RÉSUMÉ. *Les systèmes d'information sont de plus en plus souvent construits à partir de l'agrégation de systèmes informatiques qu'il convient de maintenir et faire évoluer avec agilité et sans entropie non contrôlée. Ceci n'est pas sans poser des problèmes d'interopérabilité ! L'ingénierie dirigée par les modèles (IDM) a entre autres objectifs d'apporter des solutions aux difficultés d'interopérabilité entre les systèmes. Cet article est le résumé des réflexions menées au sein de l'action spécifique « Interopérabilité des systèmes d'information et ingénierie des modèles » soutenue par* INFORSID *: quels défis, quelles solutions ? Nous proposons une synthèse d'un ensemble d'approches basées sur l'IDM répondant à différents problèmes posés par l'interopérabilité. De nombreuses questions et limites ont été soulevées lors de nos rencontres qui sont ici également rapportées dans cet article.*

ABSTRACT. *Information systems are more and more often based on aggregation of other systems that must be maintained and evolved in an agile way and without uncontrolled entropy. This is not without interoperability problems! Among others, the aim of Model-Driven Engineering (MDE) is to provide solutions for interoperability issues between systems. This paper summarizes thoughts that have come up from the* specific action *"Interoperability of information systems and model-driven engineering: What challenges ? What solutions ?" supported by* INFORSID. *We propose a summary of approaches that are based on MDE and knowledge engineering and that tackle interoperability issues in the industry. Open questions and limitations that raised during the meetings are also reported.*

MOTS-CLÉS : *interopérabilité, ingénierie dirigée par les modèles, ingénierie des connaissances, évolution des modèles et des SI.*

KEYWORDS: *interoperability, model-driven engineering, knowledge engineering, models and IS evolution.*

DOI:10.3166/ISI.18.2.13-44 © 2013 Lavoisier

1. Introduction

Les systèmes d'information (SI) sont la brique essentielle à l'organisation de nos entreprises et plus généralement de notre société. Ils sont aujourd'hui construits à partir de l'agrégation de systèmes informatiques (SSI) qu'il convient de maintenir et faire évoluer avec agilité et sans entropie [1] non contrôlée. Les objectifs sont alors entre autres d'améliorer la qualité des services offerts tout en préservant l'autonomie des acteurs, l'ouverture des SI, une gestion cohérente des informations, des temps de production réduits et une meilleure maîtrise des coûts de maintenance. Cependant, face à la complexité des SI, la mise en place de l'interopérabilité entre SI est difficile aux niveaux conceptuels et techniques.

Depuis novembre 2000, sur l'initiative de l'OMG, l'ingénierie dirigée par les modèles a proposé différentes approches et technologies pour le développement et la maintenance des systèmes à prépondérance logicielle. Dans ce contexte, un des objectifs visés était de pallier les difficultés d'interopérabilité entre les systèmes en s'appuyant sur les modèles au lieu des intergiciels (Favre *et al.*, 2006). Depuis, les architectures à base de services sont venues compléter le tableau en mettant en avant des standards de développement et de communication visant à favoriser les intégrations de systèmes et une meilleure « agilité » par l'adaptation des systèmes.

L'interopérabilité des systèmes d'information existants reste cependant un verrou par leur complexité intrinsèque et extrinsèque due aux interactions avec les autres systèmes, par les technologies utilisées et l'évolution constante de ces systèmes alors même qu'ils sont les pièces maîtresses des entreprises. C'est sur la base de ces réflexions que les 11 et 12 décembre 2011, nous nous sommes réunis pour écouter les problèmes posés par les industriels en termes d'interopérabilité au sein des SI (cf. section 2) et porter un regard commun sur les travaux de recherche visant à les résoudre en utilisant des approches dirigées par les modèles. Le référentiel général d'interopérabilité (RGI) [2] fourni par la Direction générale de la modernisation de l'état (DGME) distingue six niveaux d'interopérabilité. Les niveaux politique, juridique et organisationel, à l'instar de ce référentiel, ne sont pas de notre ressort et ne sont pas étudiés dans cet article. Les trois autres niveaux d'interopérabilité : *sémantique* : « savoir se comprendre », *syntaxique* : « savoir communiquer », et *technique* : « pouvoir communiquer » sont transversaux à nos axes d'étude qui couvrent parfois plusieurs niveaux d'interopérabilité.

Après une étude plus précise de la problématique de l'interopérabilité des SI (cf. section 2), nous avons travaillé selon deux axes de réflexion : (1) la préparation à l'interopérabilité par extraction de modèles et respect de normes,

1. Ce terme « désigne le désordre qui règne dans une organisation lorsque le SI est suffisamment complexe... » (Gillet, Gillet, 2011)
2. http://references.modernisation.gouv.fr/sites/default/files/RGI_Version1%200.pdf

(2) la mise en œuvre par les modèles de l'interopérabilité. La figure 1 donne une vision globale du contenu et de la structuration de l'article. On peut ainsi constater que notre approche est transversale relativement aux différents niveaux identifiés.

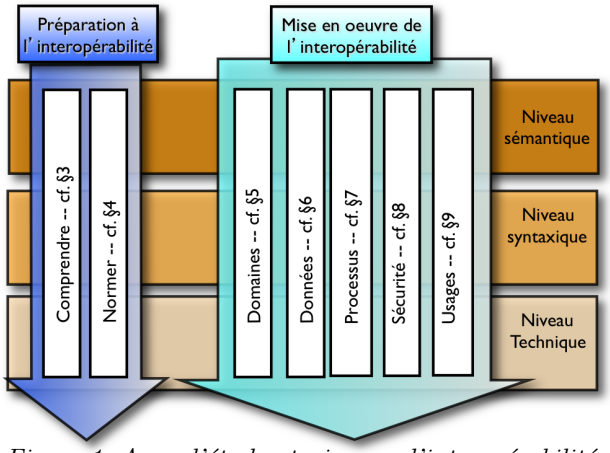

Figure 1. Axes d'étude et niveaux d'interopérabilité relativement aux travaux liés à l'IDM

Selon l'angle de l'ingénierie des modèles et la décomposition en métamodèle, modèle et système, la figure 2 illustre les différents champs étudiés dans cet article. Ainsi en section 3, nous explicitons le besoin de modèle pour appréhender l'interopérabilité et présentons différents travaux supportant cette montée en abstraction. Dans la figure 2 cette étape est schématisée par la relation μ entre le système d'information et la modélisation conformément à la notation introduite par (Favre *et al.*, 2006). De manière générale, la construction et la maintenance d'un SI reposent sur le respect de normes établies tant sur le processus de construction du SI que sur leur bonne formation et leurs qualités intrinsèques ; la conformité des SI par rapport à ces normes est abordée en section 4. Puis nous nous intéressons plus spécifiquement à la mise en œuvre de l'interopérabilité. Ainsi la section 5 aborde l'interopérabilité des domaines par les alignements de vocabulaire dans le contexte de la représentation des connaissances (modélisation des ontologies) (cf. 5.1) et par composition de modèles dans le contexte de l'ingénierie dirigée par les modèles (cf. 5.2). Puis nous nous intéressons à la grande variabilité latente à certains SI et aux techniques mises en place aujourd'hui pour la maîtriser et supporter à la fois l'abstraction automatique des systèmes et leur production dans le contexte des usines logicielles (cf. 5.3). Nous concluons cette analyse de l'interopérabilité des domaines par la gestion des exigences (cf. 5.4). Sur cette base, nous abordons alors l'interopérabilité applicative c'est-à-dire l'interopérabilité des données (cf. 6), des processus (cf. 7), des règles de sécurité (cf. 8) et des usages (cf. 9). Les travaux présentés dans ces sections prennent leurs racines dans les approches

citées précédemment et les enrichissent pour prendre en compte la spécificité des domaines étudiés.

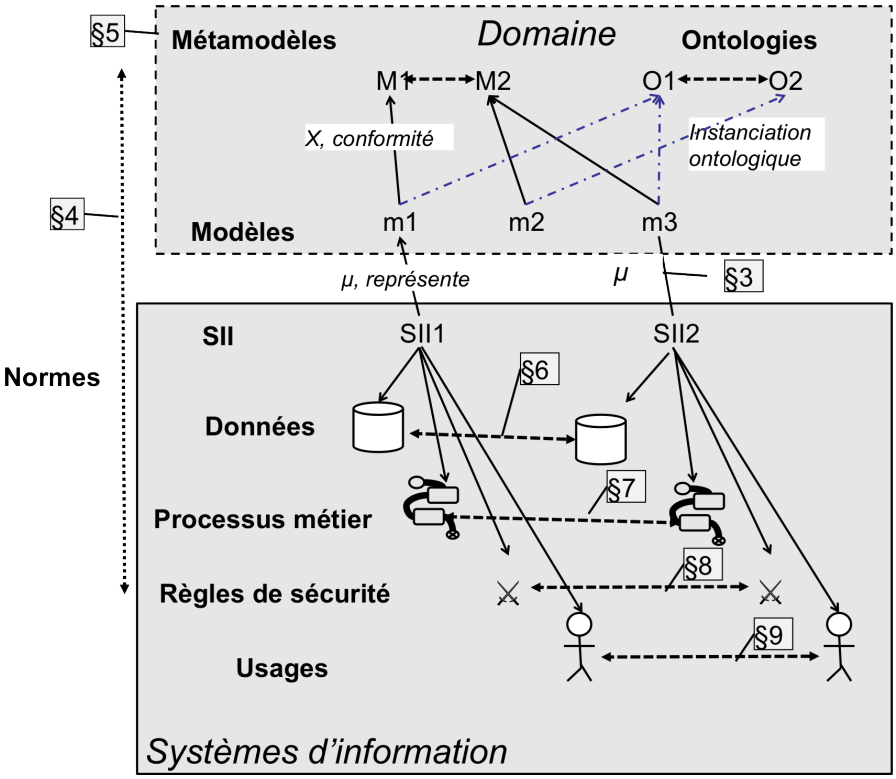

Figure 2. Interopérabilité entre SI *relativement à l'ingénierie dirigée par les modèles et à la représentation des connaissances étudiée dans l'article*

Cet article étend l'article présenté à INFORSID selon deux axes : d'une part, nous avons explicité et illustré certains travaux, d'autre part, nous avons enrichi cette étude par des travaux récents ou issus d'approches que nous n'avions pas présentés dans la version courte de cet article.

2. Problématique de l'interopérabilité des SI

« On ne choisit pas son SI » résume une part des difficultés auxquelles les DSI doivent aujourd'hui faire face (Blanchard, 2011). La plupart des SI ont les mêmes symptômes que les systèmes patrimoniaux (Demeyer *et al.*, 2002). Seule leur place centrale empêche leur mise en retraite illustrant par là même les lois de (Lehman, Belady, 1985). Toute la complexité est alors dans l'« intégration » de ces systèmes entre eux ou avec de nouvelles applications. Or, il s'agit souvent de systèmes de plusieurs décennies et de taille conséquente. Une grande part

du savoir interne du système n'est plus connue des équipes de maintenance. De par leur âge, les méthodes de développement et les langages utilisés sont souvent obsolètes et/ou propriétaires. Les modèles de calcul sont difficiles à identifier et les règles métiers sont codées en dur dans le système. Les interfaces homme-machine (IHM) consistent parfois en des écrans qui n'hésitent pas à violer les principes des architectures en couches : il n'est pas rare de construire les requêtes directement dans le code des interfaces. De manière générale, les préoccupations qui sous-tendent la mise en œuvre du système sont difficiles à séparer. Tous ces éléments complexifient la compréhension « rétrospective » des SI et l'absence de tests rend tout changement une aventure périlleuse. Ces caractéristiques conduisent souvent à un immobilisme et un pilotage à vue du SI.

La mise en place de l'interopérabilité entre ceux qui deviennent des sous-systèmes d'information exacerbe ces difficultés. Pourtant, l'interopérabilité des SI est vitale au sein des entreprises. Elle doit permettre de prendre en compte : la rapidité d'évolution des SI ; l'évolution technologique, celle des règles de gestion, celle de distribution des compétences sur les services de l'entreprise ; le partage de connaissances élaborées par des acteurs ayant des cultures différentes et des objectifs spécifiques ; la multiplication de parties de systèmes sur des réseaux interconnectés. La construction des SI et la gestion de leur évolution reposent alors sur la composition des SI, souvent autour de référentiels de données transverses, accessibles et utilisés par l'ensemble des traitements informatiques (Figer, 2008). On constate ainsi que le terme *interopérabilité des* SI regroupe plusieurs points de vue dont : interopérabilité des domaines structurels (nommage, structure, description), interopérabilité des données (partage, cohérence) ; interopérabilité des processus (processus opération, processus de support, règles de gestion) et des contrôles d'accès ; interopérabilité des usages (IHM universelle et présentation adaptable et progressive/public hétérogène).

Différentes stratégies sont utilisées pour permettre l'interopérabilité : rétro-ingénierie (bien souvent utilisée pour reconstituer un modèle de données à partir des structures physiques des fichiers ou des tables), re-ingénierie (réorganisation d'un système d'information existant afin de le rendre plus efficient), ingénierie avant (cibler de nouvelles plateformes technologiques, intégrer des notions nouvelles), création de surcouches avec des IHM qui masquent les différents applicatifs et utilisent la composition d'interfaces existantes ; mise en place d'auto-explications, etc. Ces approches doivent de plus prendre en compte la gestion des droits d'accès aux données basée non seulement sur les domaines applicatifs mais également sur les usages nécessitant des visions transverses sur les données. Il faut passer d'une stratégie d'exploitation verticale des données à une exploitation horizontale inter-applicatifs.

L'interopérabilité des SI est donc à la fois essentielle et difficile à mettre en place. L'un des premiers articles sur l'ingénierie des modèles commence par *"It's about integration. It's about interoperability (Soley, OMG Staff Strategy Group, 2000)"*. La différence mise en avant relativement aux approches de modélisation

antérieures est l'objectif de productivité (Combemale, 2008). Ainsi, l'ingénierie des modèles se veut productive en automatisant la mise en cohérence de différentes vues sur les modèles, la prise en charge des outils relatifs à la validation des modèles, la transformation de modèles et la génération de code (Favre *et al.*, 2006).

Comment les travaux relatifs à l'ingénierie des modèles mais aussi à la rétro-ingénierie et à l'ingénierie des connaissances supportent ou pourraient supporter une part de l'interopérabilité des SI ? Telle est la question à laquelle notre action spécifique a tenté de répondre. Nous déclinons à présent les différents travaux liés à l'IDM et à l'ingénierie des connaissances, relativement à notre problématique d'interopérabilité.

3. Comprendre le SI par les modèles

La problématique de l'interopérabilité de deux SI A et B peut être repensée en termes d'interopérabilité du modèle de A (noté $m(A)$) avec celui de B (noté $m(B)$). Or, ce que l'on appelle le modèle d'un SI est en réalité l'agrégation de plusieurs modèles, chacun correspondant à un point de vue particulier sur le système. L'expérience des journées de cette action spécifique nous a permis de dégager 5 points de vue complémentaires (mais non exhaustifs) sur le SI qui doivent être pris en compte pour son interopérabilité avec un SI tiers : $m_{domaine}(SI) \oplus m_{donnée}(SI) \oplus m_{processus}(SI) \oplus m_{sécurité}(SI) \oplus m_{usage}(SI)$. Ceci nous amène naturellement à décomposer l'interopérabilité de A et B pour chaque point de vue : $m_{domaine}(A)$ et $m_{domaine}(B)$, $m_{donnée}(A)$ et $m_{donnée}(B)$, etc. Cette vision suggère une sorte d'« algèbre » pour l'interopérabilité, dans l'esprit de celle amorcée par les membres de l'AS du GDR GPL sur l'hétérogénéité des systèmes [3], sous-tendant l'idée qu'il est possible d'appréhender le SI et ses évolutions comme un objet mathématique classique.

3.1. Techniques de rétro-ingénierie

L'hypothèse de départ de cette approche est donc que l'on dispose d'un modèle du SI. Malheureusement, ceci est très rare en pratique, sauf naturellement à considérer que ce dernier a été construit selon une pure démarche IDM, et que de surcroît les modèles sont disponibles. Puisque le « cas nominal » est l'absence de modèle du SI existant, sa récupération devient un facteur-clé de réussite pour l'intéropérabilité. De manière générale, la rétro-ingénierie (*reverse-engineering* en anglais) est l'ensemble des techniques pour comprendre et déterminer le fonctionnement d'un objet. Appliquée au logiciel, elle consiste à extraire des entrailles d'un système logiciel des vues de plus haut niveau : ce sont les modèles. On trouve une littérature abondante sur le sujet depuis la fin

3. http://people.irisa.fr/Benoit.Combemale/research/gemoc/as2011/

des années 1980 (Chikofsky, Cross II, 1990), selon le type du système en présence, le niveau d'abstraction à atteindre ou encore le nombre et la nature des éléments à prendre en compte. Par exemple, les treillis de concepts permettent (entre autres) de factoriser des concepts et s'appliquent bien aux diagrammes de classes UML (Amar *et al.*, 2012).

Du point de vue technique, la phase de rétro-ingénierie peut inclure des analyses statiques et dynamiques du système existant. L'analyse statique repose sur l'utilisation de parseurs sur les différents artefacts qui constituent le SI tandis que l'analyse dynamique repose sur des traçeurs d'exécution. Ces analyses produisent en sortie des modèles bruts, d'abord caractérisés par des métriques, puis dérivés en modèles à plus forte valeur ajoutée. Par exemple, des visualisations sont souvent adéquates pour obtenir des cartes de ces systèmes (Lanza, Ducasse, 2003 ; Ducasse *et al.*, 2006), permettant d'accéder à un degré intéressant de compréhension de l'existant. Cette montée en abstraction est primordiale pour que le processus de « refonte » ou de « composition » des systèmes puisse opérer, car basé sur une véritable compréhension des SI en présence. Par exemple, dans le cas difficile des hôpitaux de Marseille, une approche par rétro-ingénierie des SI existants, en particulier des processus métiers et de leurs accès aux données, semble un prérequis indispensable à la mise en place de l'interopérabilité.

3.2. Référentiels de rétro-ingénierie

La maîtrise de la conformité des SI relativement à des référentiels normatifs est aujourd'hui exigée tout au long du cycle de vie du SI et impacte les choix et les approches en matière d'interopérabilité. Ceci implique que la rétro-conception de deux SI A et B selon deux référentiels distincts limite la possibilité de les faire intéropérer car il est plus difficile de les comparer entre eux. C'est là que les métamodèles entrent en jeu, puisqu'ils définissent les langages qui seront utilisés pour capturer les différents aspects des systèmes logiciels étudiés.

Parmi les référentiels les plus connus, nous pouvons citer, dans l'ordre chronologique :

– 1996 - Les métamodèles de FAMIX (Ducasse *et al.*, 2011), intégrés dans la plateforme Moose (Nierstrasz *et al.*, 2005), permettent à la fois l'analyse statique et dynamique de code et autres artefacts tels que des informations de bugtracker, des logs d'activités. Dans ce sens, FAMIX est une famille de métamodèles orientés pour l'analyse de SI ;

– 2004 - Initié par le célèbre *Dagstuhl Seminar* sur l'intéropérabilité des outils de retro-ingénierie, le *Dagstuhl Middle Metamodel* (DMM (Lethbridge *et al.*, 2004)) a vu le jour, et supporte l'analyse statique de code source objet ;

– 2005 - Même s'il cible plutôt l'ingénierie avant, l'*Unified Modeling Language* (UML 2.x) de l'OMG, à travers ses 13 sous-langages (*i.e.*, ses 13 diagrammes), reste le langage de modélisation le plus largement utilisé pour re-

présenter un système logiciel existant (orienté objet ou non). Cependant il est mal adapté aux tâches de ré-ingénierie (Demeyer *et al.*, 1999) ;

– 2005 - L'initiative ADM de l'OMG a débouché sur la proposition *Knowledge Discovery Metamodel* (KDM (OMG, 2011)), constituée de 9 sous-langages pour cartographier le portefeuille logiciel des entreprises. Elle inclut également l'*Abstract Syntax Tree Metamodel* (ASTM (OMG, 2011)) pour traiter spécifiquement l'extraction d'information syntaxique pour les langages de programmation de 4e génération.

Une réelle compréhension du SI exige une modélisation fidèle à l'existant, nécessitant d'appréhender simultanément les différents points de vue (domaines, données, etc.). Des pistes de recherche concernent la prise en compte des relations « intra- » et « inter- » des modèles extraits. Ceci inclut l'élaboration de métriques adaptées qui allient ces différents aspects, la définition de métamodèles complets/multipoints de vue, des techniques d'analyse d'impacts, ou encore de nouveaux modes de visualisation permettant de cartographier un système dans son ensemble.

4. Conformité à des référentiels normatifs

Les systèmes d'information sont soumis depuis de nombreuses années à la difficulté d'assurer leur conformité face à des référentiels normatifs divers tant au niveau des processus de développement que des modèles produits. Quels que soient la norme considérée et son objectif, leur but est de s'assurer que toutes les exigences explicitées sont bien remplies pour le système considéré, que le système est conforme à celles-ci et qu'un utilisateur a les moyens de juger le système final conforme aux besoins et contraintes exprimés. Le travail de mise en conformité entre un modèle et une norme dépasse la seule considération de la traçabilité des exigences : on doit en effet considérer les processus utilisés pour définir les systèmes et pour les vérifier/valider, on doit considérer la prise en compte de propriétés spécifiques de ces modèles, on doit produire différents documents supports à la certification. Lorsque l'on considère l'interopérabilité des SI, tout doit être également remis en question : deux SI conformes à un référentiel donné peuvent engendrer un SI global qui perd alors toutes les bonnes propriétés par le seul fait de leur interopérabilité.

Lors d'un atelier de travail organisé par l'ISO fin 2010, on apprend que les standards les plus couramment considérés (près de 40 % des études de conformité réalisées) sont la norme ISO14001 (ISO, 2004b) (management environnemental) et la norme ISO9001 (ISO, 2008) (management de la qualité). Les autres normes considérées sont des normes liées au domaine d'application (nucléaire, transport aérien, ferroviaire ou automobile, géographique (European Commission, 2007) etc.) ou des normes spécifiques comme celles pour la sûreté de fonctionnement des systèmes notamment. Dans les exemples industriels présentés lors de ces journées sur l'interopérabilité, la conformité à des réfentiels nationaux ou internationaux est sous-jacente pour chacun d'eux. La qualité du

produit est fortement considérée dans ces normes (avec l'ISO9001 notamment) : on s'intéresse dans ce cas à la certification d'un processus de développement pour obtenir le produit final mais aussi à la qualité sur le résultat obtenu. Dans ce dernier cas, on s'intéresse à la certification du produit final par rapport à des propriétés attendues, conformément à des normes comme celles pour la sûreté de fonctionnement avec la norme EN61508 (CENELEC, 2001) (ou une de ses nombreuses dérivées selon le domaine d'application considéré).

Dans les approches à base de modèles, de nombreux travaux ont été réalisés pour faciliter la prise en compte des normes systèmes dans les processus de développement (Ledinot *et al.*, 2012). Ces approches, permettant d'assurer tout autant la conformité des processus de développement mis en place que celles des produits développés, sont basées sur l'utilisation de modèles informels, semi-formels (comme UML/SysML) ou formels et elles ont été développées pour faciliter la certification des systèmes logiciels et le suivi des exigences tout au long du développement.

La gestion de la conformité est d'autant plus difficile à s'inscrire dans le cadre de l'interopérabilité que celle-ci peut être abordée à tous les niveaux du SI et, ainsi, l'intégration des résultats obtenus dans chaque sous-système et aux différents niveaux est un défi qui reste à relever. En effet, malgré les récentes avancées dans le domaine (on peut se référer au projet européen CESAR [4] qui propose comme résultat une plateforme basée sur les modèles pour faciliter l'étude de sûreté et de sécurité des systèmes informatiques), les études de conformité des systèmes restent difficiles à mener de bout en bout par manque de guidage méthodologique lors de ces études. Ainsi, il faudra suivre attentivement les résultats du projet européen OPENCOSS [5] dont l'objectif est de fournir une plateforme ouverte d'aide à la certification de systèmes complexes dédiée à la gestion de la sûreté. Malgré le côté spécifique de cet objectif de certification, les résultats obtenus permettront d'avancer sur ce sujet.

Dans le cadre de la certification d'un système à la fois dans son processus de développement (on s'intéresse ici au « comment » un système est modélisé) que dans les propriétés attendues du système résultant (on s'intéresse au « quoi » dans ce cas), une première étape a été réalisée dans les travaux de M. Adedjouma *et al.* (Adedjouma, Dubois, Terrier, Kitouni, 2012 ; Adedjouma, Dubois, Terrier, 2012) dans lesquels les auteurs se sont employés à étudier comment optimiser la réutilisation d'artefacts d'une certification donnée dans le cadre d'un autre processus de certification. Un artefact est ici un élément d'entrée à la certification (un modèle, une exigence, un document, etc.), un élément en sortie (une justification, une preuve, un statut, etc.) ou une activité du processus de certification (une action par exemple). Ces études se sont faites dans le domaine automobile avec tout d'abord la certification du processus de déve-

4. http://www.cesarproject.eu
5. http://www.opencoss-project.eu

loppement par rapport à une dérivée automobile de l'ISO15504 (ISO, 2004a) (aussi appelée SPICE (*Software Process Improvement and Capability dEtermination*)) et du comment on peut ensuite faciliter la certification en sûreté (prise en compte de l'ISO26262 (ISO, 2011) dans ce cas) de ce même système. Ces travaux montrent comment un processus basé sur l'IDM facilite la réutilisation d'artefacts de certification pour faciliter la certification vis-à-vis d'une autre norme.

Ces travaux montrent l'intérêt d'une telle démarche mais ne font qu'esquisser comment l'IDM peut répondre à ces attentes. Ils illustrent bien comment des processus dédiés aux activités de mise en conformité ainsi que des outils les implémentant faciliteraient le travail et offriraient un support à ces activités qui concernent les processus, produits et services. Les uniques travaux sur le sujet se font à travers diverses implantations du standard BPMN notamment (tel que EPF sous Eclipse) et ces offres ne sont que peu réellement utilisables dans un contexte de certification opérationnel.

Une collaboration étroite entre la modélisation des processus et la modélisation du système est un objectif primordial pour faciliter une plus large diffusion de ces activités et l'IDM offre certaines briques importantes pour améliorer chacune de ces activités ; les techniques permettant de les relier de façon très fine doivent être étudiées et outillées.

La composition de SSI amène là aussi de nouveaux challenges dans la prise en compte des études déjà réalisées pour des SI existants afin d'aider à la certification des SSI qui interopèrent. Quel que soit le type de certification (produit final ou processus de production) qui nous intéresse ici, la composition et l'interopérabilité de ces systèmes engendrent une nécessaire prise en compte de la gestion de leur certification en limitant au maximum leurs délais et leurs coûts. En effet, outre la certification d'un système en lui-même qui doit être facilitée et outillée, le fait de composer de tels systèmes et de les faire interopérer pose des questions sur la réutilisation potentielle de la certification des systèmes composites afin d'éviter un surcoût en certification du système global ; la composition pose aussi des problèmes quant à la perte potentielle de certification avec l'ajout d'un nouveau composant au système. L'IDM peut apporter certaines réponses à cette problématique.

Nous abordons à présent le deuxième axe de notre étude qui est celui des mises en œuvre en commençant par l'interopérabilité entre les domaines.

5. Interopérabilité des domaines

L'interopérabilité des domaines des SI fait référence au problème bien connu en ingénierie des connaissances de l'alignement des vocabulaires. Elle est également abordée en termes de variabilité en ingénierie des lignes de produits logicielles et en termes de gestion des exigences en ingénierie des modèles.

5.1. Alignement de vocabulaires

Une ontologie permet de modéliser un domaine en définissant ses concepts et les règles qui les régissent sous la forme d'axiomes. Ces dernières années ont vu la publication sur le web de nombreuses ontologies devenues des références pour un domaine particulier, une communauté particulière. Leur réutilisation garantit l'interopérabilité de différentes sources de données ; c'est le principe du web de données (*Web of Linked Data* en anglais). Le développement d'un SI nécessite de combiner différentes terminologies adoptées par différentes classes d'utilisateurs du SI : le sens des termes varie selon les organisations, les métiers, les acteurs et les contextes, ce qui entraîne que le modèle sous-jacent soit souvent différent. La modélisation du domaine sous la forme d'une ontologie permet de donner une place à ces différentes terminologies au sein d'un même modèle, en associant à chaque concept les différents termes qui le désignent. Le langage RDFS permet de représenter ces relations à l'aide des propriétés `label` et `comment` ; le langage SKOS permet également de modéliser plus finement un thésaurus avant de l'associer à une ontologie.

Au-delà de cet alignement des terminologies utilisées sur un même domaine, le développement d'un SI nécessite également souvent d'intégrer différents sous-systèmes et donc de combiner des données ou des connaissances hétérogènes, issues de différents domaines, référant à différentes ontologies. Le problème qui se pose alors est de découvrir, représenter et exploiter les correspondances qui existent entre concepts issus de différentes ontologies. C'est le problème de l'alignement d'ontologies (Euzenat, Shvaiko, 2007). Dans le cas le plus simple, deux concepts issus de deux ontologies sont considérés comme équivalents (le langage OWL permet de représenter cette correspondance avec les propriétés `equivalentClass` et `equivalentProperty`), ou l'un subsumant l'autre (le langage RDFS permet de représenter cette correspondance avec les propriétés `subClassOf` et `subPropertyOf`). Etant donnée la définition en intension des concepts, des méthodes de classification classiques en logique de description permettent d'établir des correspondances d'équivalence ou de subsomption. Des liens sémantiques non logiques peuvent également être établis, basés sur des mesures de similarités entre les définitions de concepts. D'autres approches utilisent des techniques de fouille de données ou de textes pour établir des relations sémantiques entre concepts basées sur leur définition en extension. Parmi celles-ci, plusieurs travaux reposent sur l'analyse formelle de concepts (FCA) (Stumme, Maedche, 2001) (Formica, 2006) (Tatsiopoulos, Boutsinas, 2009). Le nombre de méthodes d'alignement progresse constamment et il est difficile de les comparer car elles dépendent beaucoup des données et domaines à aligner.

5.2. Composition de modèles

Dans le contexte de l'IDM, de nombreux travaux se sont intéressés aux approches par points de vue pour introduire des éléments spécifiques aux plates-

formes (Thomas *et al.*, 2007), abordant ainsi l'un des aspects de l'interopérabilité des domaines. Mais cette décomposition du modèle en différents points de vue dirigés par les projections vers les plates-formes ne suffit pas à répondre à la complexité du développement logiciel qui adresse simultanément différents domaines de préoccupations (France, Rumpe, 2007). Aussi la modélisation orientée aspects a-t-elle émergée pour permettre une gestion des différentes préoccupations au niveau de la modélisation. L'objectif est de donner les moyens aux experts de différents domaines de travailler ensemble à la construction d'un même modèle par la définition de « correspondances » (Bezivin *et al.*, 2006). La correspondance peut être automatiquement calculée en recherchant des éléments dont les signatures sont équivalentes, où une signature est un ensemble de propriétés syntaxiques associées à un type élément (Reddy *et al.*, 2005). Sur la base de ces modèles de correspondance, des algorithmes de composition permettent d'obtenir soit les modèles communs et de préserver les vue (Anwar *et al.*, 2010), soit de construire un modèle global (France *et al.*, 2007). La modélisation par point de vue consiste alors à adapter un modèle en l'enrichissant des concepts issus des différents points de vue tout en maintenant valides les contraintes propres à chaque domaine (Blanc *et al.*, 2008 ; Anwar *et al.*, 2010).

Les travaux cités précédemment sont centrés sur la composition des structures et non pas sur celle des comportements. Or, l'interopérabilité des domaines impose également une prise en compte de l'étude et de la composition des comportements. Nous abordons cet aspect de l'interopérabilité à la section 7. Notons que la thèse de (Wouters, 2013) aborde le mariage entre les techniques d'alignements des ontologies et les modèles pour supporter la modélisation multi-expert de différents domaines, en prenant en compte les comportements induits par les modèles.

5.3. Capture de la variabilité

Compte tenu des points communs et des différences (c'est-à-dire de la variabilité) qui existent entre les logiciels d'un même domaine ou les logiciels qui répondent à un même besoin mais qui ne sont pas développés dans le même contexte, l'ingénierie des lignes de produits logicielles (LPL) vise à produire des familles de variantes de programmes pour un domaine donné (Pohl *et al.*, 2005). Elle s'appuie notamment sur une capture de la variabilité (points de variation et variantes), elle-même reconnue comme nécessaire depuis longtemps (Jacobson *et al.*, 1997). Une manière de capturer cette variabilité est d'exprimer la ligne de produits sous la forme d'un ensemble de caractéristiques qui décrivent ce domaine avec un point de vue qui dépend des besoins de l'utilisateur. Plusieurs approches permettent ensuite de modéliser la variabilité, soit par modèles de décision – les décisions forment un ensemble adéquat de choix distinguant les membres de la ligne (Rabiser *et al.*, 2007 ; Schmid *et al.*, 2011) –, soit par *Feature Model* (FM) – les caractéristiques et leurs combinaisons valides sont organisées de façon compacte –, ou soit enfin par modèles pouvant

se ramemer aux FM (Pohl *et al.*, 2005). Le formalisme des FM est actuellement le plus répandu ; il s'agit pour simplifier d'un graphe AND-OR avec des contraintes propositionnelles (Schobbens *et al.*, 2007 ; Czarnecki, Wasowski, 2007 ; Thüm *et al.*, 2009). Les FM sont utilisés pour décrire la variabilité lors de l'analyse et conception de logiciels, mais également dans les autres phases de développement. Ainsi on utilise des FM pour décrire les différentes parties du système, que ce soit la description du matériel (Kang *et al.*, 1998), la structure organisationnelle (Reiser, Weber, 2007), les détails du métier ou de la mise en œuvre (Metzger *et al.*, 2007). Des travaux abordent aujourd'hui l'extraction automatisée de FM à partir de grands systèmes logiciels (Acher *et al.*, 2011 ; She *et al.*, 2011 ; Acher, Cleve *et al.*, 2012). Ces FM sont souvent de grande taille et comme les techniques de l'état de l'art ne passent pas à l'échelle, il est donc nécessaire de proposer des opérateurs de décomposition et de composition des FM (Acher, Collet, Lahire, France, 2012). Les FM qui résultent de la capture des domaines et des besoins des utilisateurs peuvent alors être réassemblés pour construire des « produits » logiciels cohérents (Acher, Collet, Gaignard *et al.*, 2012) (cf. figure 3).

Dans ce contexte d'évolutions fréquentes à la fois des exigences et des produits, des travaux ont été amorcés pour la gestion agile des lignes de produits. Ces travaux sont basés en particulier sur une décomposition en domaines liés par des contraintes externes exprimées sous forme de règles (Urli *et al.*, 2012). Ces travaux complètent ceux visant à gérer la co-évolution des modèles de variabilité et des différents autres modèles (Neves *et al.*, 2011 ; Seidl *et al.*, 2012). Ainsi l'ingénierie des lignes de produits logicielles propose de nouvelles approches et techniques pour appréhender au moins une partie de l'interopérabilité des domaines.

Dans la figure 3, deux FM sont visualisés à gauche, ils sont extraits de systèmes pré-existants dans le cadre du projet YourCast http://yourcast.fr. Le premier présente une source d'information diffusant des calendriers qu'il est possible de trier par date. Le second diffuse des tweets. Ceux-ci peuvent être triés ou filtrés. L'analyse d'un ensemble de tels systèmes a conduit à la modélisation de différents FM, fusionnés à droite pour modéliser la variabilité des sources d'informations disponibles.

5.4. Gestion des exigences

L'interopérabilité des systèmes initialement indépendants répondant chacun à leurs propres exigences suppose de prendre en considération l'adéquation des exigences de chacun et d'assurer les exigences portant sur le SI résultant. Dans le cadre des travaux sur l'IDM, des passerelles ont été mises en place pour lier les outils dédiés à la gestion des ontologies, les outils de gestion d'exigences et les outils de modélisation (dans le cadre de la gestion de la variabilité par exemple avec (Dubois *et al.*, 2012)). Ces travaux permettent de faciliter le

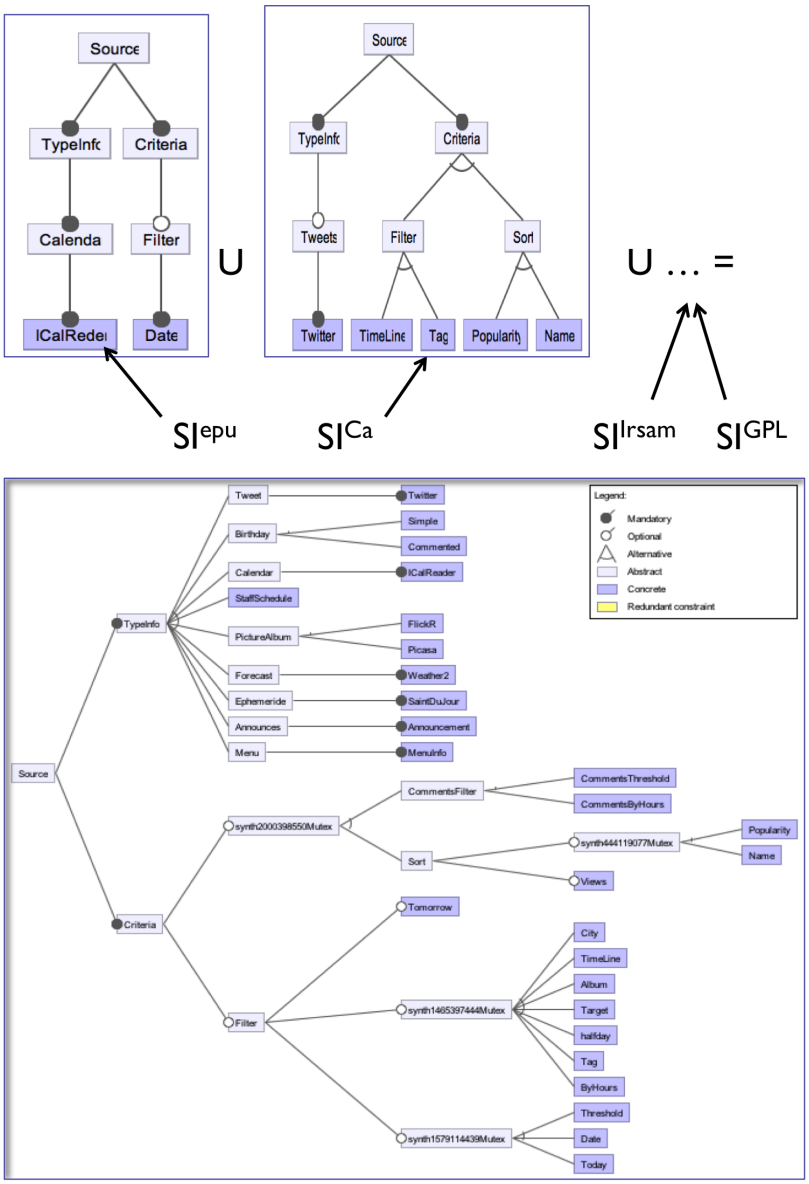

Figure 3. Capture de la variabilité par « feature models »

travail de gestion de la conformité des exigences avec la capture du domaine, d'avoir également un suivi entre les exigences et les modèles réalisés pour décrire le système (Mosser *et al.*, 2011). Prenons par exemple l'outil DODT (Farfeleder *et al.*, 2011) qui offre la possibilité de vérifer la cohérence entre une base d'exigences dans l'outil de gestion d'exigences DOORS (Doors, 2008) et une description du modèle de domaine à l'aide d'une ontologie. Des connexions existent aussi entre des modèles SysML (OMG, 2010) et des ontologies décrites en OWL. Ces travaux doivent ensuite être améliorés afin de gérer l'interopérabilité des domaines, des exigences et des modèles associés. Dans le cadre de l'IDM, diverses techniques basées sur la fusion de modèles peuvent aussi être utilisées pour considérer l'interopérabilité des bases d'exigences. Par contre, au-delà de la fusion des modèles, aucune technique dédiée étudiant la gestion de la traçabilité ou de la cohérence de ces modèles n'a été spécifiquement abordée et ce domaine de recherche reste encore complètement en friche.

Même lorsque les domaines sont alignés, la mise en œuvre au niveau des données et des processus métiers peut différer énormément d'un système à l'autre engendrant des difficultés lorsqu'il s'agit de faire travailler ces systèmes entre eux.

6. Interopérabilité des données

La communauté des SI est confrontée à l'intégration d'informations hétérogènes, tant dans les contenus et les formats que dans la diversité des points de vue utilisateurs. Il s'agit de partager ces informations -données et métadonnées- afin d'assurer la coopération d'applications ou de services, ou bien de favoriser la publication d'informations consolidées qui facilitent la prise de décisions. En pratique, on peut distinguer plusieurs niveaux d'intégration de SI au niveau des données, du plus fort au plus faible couplage : (i) l'utilisation de données ouvertes ou d'entrepôts de données partagés entre les SI, (ii) l'utilisation d'un modèle d'échange, et (iii) la construction de ponts entre des espaces de représentation différents, tels que les entrepôts intermédiaires construits par méthode ETC (*Extraction-Transformation-Chargement*) utilisée en informatique décisionnelle (Atigui *et al.*, 2010). Le premier niveau résout le problème d'interopérabilité des données dès la conception des SI. Lorsque cela n'est pas possible, le développement de référentiels sémantiques communs et d'interfaces de médiation est nécessaire.

6.1. Principes généraux

La modélisation des connaissances de domaine d'un SI sous la forme d'une ontologie ou de plusieurs ontologies alignées ne suffit pas à l'interopérabilité des données. L'adoption d'un modèle d'échange ou de médiation voire d'un modèle unique de représentation des données facilite l'intégration de sous-systèmes dans la conception d'un SI (Michel *et al.*, 2010). Tout l'enjeu est de propager

et partager des données d'un sous-système à l'autre pour éviter leur duplication qui rend les mises à jour difficiles. La gestion des données dans le contexte de l'interopérabilité des SSI doit alors répondre à plusieurs besoins :

– *référencement global des données*, qui doit permettre d'identifier de manière unique une donnée appréhendée différemment dans les sous-systèmes (*ex. l'identification des personnes par un identifiant unique est un problème connu lorsque le numéro de sécurité sociale ne peut pas être utilisé*) ;

– gestion des *modifications de données* (valeur, création, destruction) qui doivent être propagées dans différents SSI tout en maîtrisant les délais de propagation entre SSI. En effet, certains applicatifs imposent que les mises à jour ne se fassent qu'à des périodes définies (*ex. la propagation du changement d'adresse d'un locataire dans un* SSI *doit tenir compte du fait qu'il faut conserver l'adresse au 1er janvier de l'année pour le paiement de la taxe d'habitation dans un autre* SSI). Ainsi la gestion des modifications impose une prise en compte et une modélisation des notions de synchronisation, d'historisation et de temps de validité de la valeur ;

– gestion des *plages de valeurs* qui nécessite la mise en place de tables de correspondance bien souvent gérées manuellement et qui font la plupart du temps partie des passerelles inter-applicatives. Ces tables de correspondance sont difficiles à utiliser dès qu'il n'existe pas une bijection entre les valeurs des SSI mis en jeu ;

– *compréhension sémantique* de la donnée. En effet une donnée n'existe et n'a de sens que par rapport aux autres données avec lesquelles elle interagit à l'intérieur d'un SSI. La transposition dans un autre SSI ne peut se faire qu'en tenant compte du nouveau contexte d'interprétation ; Ce contexte d'interprétation se positionne à deux niveaux de sémantique : par rapport au domaine applicatif et par rapport aux interacteurs du SSI ;

– gestion des *droits d'accès à la donnée*. La construction de surcouches inter-applications doit permettre de filtrer les accès aux données suivant les modalités spécifiques de chaque domaine. La politique de définition des droits d'accès devient une politique globale et nécessite la définition de rôles et de profils nouveaux qui sont davantage calqués sur les processus métiers que sur les domaines applicatifs (voir section 8).

6.2. Eléments de mise en œuvre

Dans une optique idéale d'interopérabilité, les données brutes doivent être modélisées avec l'ensemble de leurs contraintes d'intégrité. Le paradigme objet (UML/OCL) est largement utilisé à cet effet. Il présente, sur des niveaux différents, des classes et leurs instances. Les modèles de classes sont à la base du développement d'applications, tandis que le modèle relationnel est privilégié à des fins de gestion de la persistance. La mise en correspondance objet/relationnel (ORM) a été extensivement étudiée ; ses difficultés sont connues et traitées au

mieux tandis que des solutions de traduction plus ou moins automatiques exploitent les patrons « Active Record » (Fowler, 2003) ou « Data Mapper », qu'elles implémentent (Hibernate, ActiveJDBC, ADO.net, RubyActiveRecord, ActiveJS).

La correspondance objet/ontologie (OOM) ou objet/RDF soulève des questions homologues (cf. ActiveRDF (Oren *et al.*, 2008)). La flexibilité des schémas RDF est un atout pour l'échange de données ouvertes, et l'exigence de reconstruction exacte des données dans le cycle object/relationnel/objet qui prévaut dans l'ORM n'est pas de mise dans la correspondance OOM. En effet, les cas d'utilisation « importation » et « exportation » y sont dissociés.

Une première approche pour mettre en correspondance par l'OOM les éléments de paradigmes différents procède par annotation du modèle de classes. On indique quels éléments sont l'objet de transformations appariant et traduisant les concepts, pour finalement permettre l'échange de données (Quasthoff, Meinel, 2009). Cette spécification statique impose de coder et d'embarquer dans les programmes de transformation toute la complexité liée à la mise en correspondance, qui ne peut être gérée qu'à l'exécution, les valeurs des instances et non pas seulement les éléments de modélisation étant en cause.

Une approche alternative fractionne le processus en séparant les aspects et les préoccupations ; elle explicite les métamodèles intermédiaires. Les transformations de niveau supérieur sont modélisées donnant ainsi un statut de premier rang aux règles de correspondance, et automatisant la production de code (Hillairet *et al.*, 2008) (Melnik *et al.*, 2003). Plus précisément, à partir d'un modèle métier enrichi, l'application directe des principes de traduction entre UML et OWL propres à ODM [6] permet la génération automatique de l'ontologie image. Celle-ci peut constituer un entrepôt interrogeable en SPARQL donnant un accès public aux données métiers. Enfin, on utilise les facilités d'alignement intrinsèques aux ontologies pour associer les ontologies cibles avec l'ontologie image du modèle métier enrichi. Ces pratiques mettent en jeu un grand nombre de métamodèles : métier, annotations, métier enrichi, ontologie image, ontologies cibles, requêtes (SQL, OQL, SPARQL). Cette complexité est maîtrisée grâce à l'IDM qui permet de spécifier des applications génériques pour mettre en correspondance et gérer les modèles concernés. Des modèles de transformation de niveau supérieur permettent de traduire des requêtes sur les données objets (ex. OQL) en requêtes sur les bases de connaissance (ex. SPARQL) ou *vice versa*.

A contrario, une application objet/web métier peut instancier à l'exécution ses paramètres en interrogeant le web des données *via* une requête SPARQL automatiquement construite par l'exploitation des correspondances spécifiées dans le modèle d'OOM. La mise en correspondance directe de graphes RDF et de bases relationnelles existe (Bizer, Cyganiak, 2006 ; Konstantinou *et al.*,

6. http://www.omg.org/spec/ODM/1.0/ - 2009, Chapter 16 : Mapping UML to OWL

2008) et fait aujourd'hui l'objet d'une proposition du W3C (*Working Draft R2RML : RDB to RDF Mapping language*[7]).

Les apports de ces approches sont patents en termes de réutilisabilité et de maintenance. En donnant au modèle de classes métiers un rôle de pivot, l'interopérabilité s'exerce entre outils de calcul, de visualisation, de persistance ou de raisonnement.

L'interopérabilité des données va de pair avec celle des processus métiers et des activités en charge d'assurer la traçablilité des mises à jour, l'identification des acteurs et les aspects temporels liés à la synchronisation inter-domaines. La composition des processus est ainsi un des points durs de l'interopérabilité des SI.

7. Interopérabilité des processus métiers

Une des composantes essentielles d'un système d'information est l'ensemble des processus métiers qu'il supporte. Lorsque des SI doivent interopérer, on constate à la fois la construction de nouveaux processus métiers (*ex. l'ajout d'un sous-système « clef en main » de gestion de vacances implique de donner la possibilité à un agent d'enchaîner la réservation de vacances après avoir créé un nouveau sociétaire*), la modification des processus (*ex. ne pas demander plusieurs fois à un hospitalier de s'identifier lorsqu'il étudie un dossier médical en fonction des sous-systèmes contribuant au dossier*), et la disparition de processus (*ex. interdire de saisir des informations sur un nouveau sociétaire dans le système de gestion des vacances, puisqu'il doit à présent puiser ses données dans le système de gestion des sociétaires*).

Les services web sont aujourd'hui employés pour créer de larges applications distribuées et évolutives à partir de composants existants. Les architectures orientées services promeuvent un couplage lâche, la distribution des services, la dynamicité et l'agilité (OASIS, 2006). L'appréhension de l'architecture du SI sous la forme de services fonctionnels silos (ils fournissent à l'ensemble du SI les services liés à leurs données) et services fonctionnels pilotes (ils fournissent la réalisation d'un ensemble de traitements) vise à favoriser l'interopérabilité des systèmes sous réserve du respect d'un corpus de principes de conception (Figer, 2008). Dans le contexte SOA, les relations entre services sont portées par des orchestrations, qui correspondent à une mise en œuvre des processus métiers. L'évolution même des orchestrations est alors appréhendée de manière « technique » *via* les travaux sur les aspects, qui supportent la séparation des préoccupations par exemple par modification des moteurs d'exécution BPEL (Charfi, Mezini, 2004) ou plus généralement aux différents niveaux de la définition des processus (analyse sémantique, moteur et orchestrations BPEL) (Courbis, Finkelstein, 2005).

7. http://www.w3.org/TR/r2rml/

Des travaux récents se concentrent sur la conception des systèmes évolutifs et de qualité, conscients de la nécessité de les aborder à un niveau d'abstraction permettant de raisonner et de contrôler ces systèmes. Ainsi la modélisation des processus métiers est aujourd'hui en pleine effervescence, en particulier avec le standard BPMN (OMG, 2011), dont la nouvelle version introduit le concept de chorégraphies de services. La chorégraphie trace les échanges des messages entre les différents participants d'un point de vue global. Il s'agit d'un contrat métier où les participants se sont mis d'accord sur l'ordre et la manière d'interagir (*ex. dans le cadre d'une intervention chirurgicale, la chorégraphie n'étudiera pas ce que doit faire chaque praticien avec le dossier du patient, mais se focalisera sur la manière dont les informations vont être échangées entre les praticiens*). Cette vision globale complète la vision locale de chaque participant qui définit son propre processus interne en une orchestration. Elle s'approche du concept de processus métiers, visant ainsi à capturer un niveau d'abstraction « accessible » à l'homme du métier.

La composition des systèmes est alors appréhendée comme une composition de ces différents modèles de processus métiers. C'est ainsi que des travaux relatifs à l'introduction de fragments d'orchestration (Mosser, Blay-Fornarino, 2012) ou de composition de scenarios sont proposés (Klein *et al.*, 2007). Nous noterons que ces compositions impactent également les points de vue domaine et données (Kienzle *et al.*, 2009 ; Clavreul *et al.*, 2011). Dans ce contexte, différents travaux abordent alors la question du respect des propriétés initiales des processus (Verjus, Pourraz, 2007) (*ex. appel à un systeme d'anonymisation avant tout échange de données*), d'identification des services « redondants » (Nemo *et al.*, 2007) (*ex. appel à un unique service de facturation en regroupant un ensemble de requêtes d'authentification*), et la nécessité d'une absence d'ordre dans les compositions (Mosser, Blay-Fornarino, Duchien, 2012). Les travaux sur la caractérisation des services par l'ajout de propriétés sémantiques viennent alors enrichir cette panoplie d'outils en autorisant des évolutions adaptatives au contexte des systèmes (Hacid *et al.*, 2009). Enfin notons que les avancées sont telles aujourd'hui dans le domaine de la composition des processus que se posent aujourd'hui des problèmes de passage à l'échelle et de prise en compte dans ce contexte de la variabilité nécessitant la mise en place d'une modélisation par « features » comme vue en section 5.3 (Mosser, Parra *et al.*, 2012 ; Acher, Collet, Gaignard *et al.*, 2012).

Si la composition des processus métiers est aujourd'hui un sujet d'étude en pleine effervescence, ses conséquences sur l'interopérabilité des données et l'analyse en termes de cohérence et de qualité des données reste à notre connaissance un sujet ouvert. De manière transversale à ces différents aspects de l'interopérabilité des SI, la gestion des règles de sécurité entre les SI est un point difficile.

8. Interopérabilité des règles de sécurité

Tout système d'information non complètement fermé vers l'extérieur nécessite un système de contrôle qui vérifie l'identité des personnes manipulant l'information et limite l'utilisation qui peut être faite de cette information selon les fonctions et les privilèges attachés à ces personnes. Définis indépendamment, les SSI se protègent à la fois par une sécurisation de l'infrastructure technique et par la mise en place de systèmes d'authentification et d'autorisation. Le chiffrage des transactions permet de garantir l'intégrité et la confidentialité des informations échangées entre SSI. Le contrôle d'identité est en soi un problème non trivial lors de la mise en commun de SSI utilisant des moyens d'identification différents. Le contrôle d'accès à l'information est d'autant plus complexe que l'expression et la vérification de règles de contrôle d'accès s'appuient potentiellement sur des langages différents.

Le contrôle d'identité est basé sur la possession par les utilisateurs d'un secret incessible, tel qu'un mot de passe, un artefact physique (empreintes, cartes électroniques...), ou un certificat d'identité utilisant des moyens cryptographiques pour rendre sa falsification très difficile (les certificats répondant à la norme X509 [8] sont majoritairement utilisés). Certains secrets, tels les certificats cryptographiques signés par des autorités de certification à reconnaissance large, sont parfois utilisables dans plusieurs SSI simultanément. Cependant, lorsqu'un utilisateur est reconnu à travers plusieurs identités dans différents SSI, un mécanisme d'alignement des identités est nécessaire. Une caractéristique importante pour l'utilisateur est que ce mécanisme permette une identification unique (*single sign-on*), c'est-à-dire que son identité soit automatiquement présentée dans le format adéquat à chaque opération ordonnée par l'utilisateur, sans que celui-ci ne doive représenter son secret. Dans certains cas, cela nécessite l'enregistrement du secret au niveau du mécanisme d'alignement des identités, ce qui le rend très sensible du point de vue de la sécurité.

Une fois l'identité d'un utilisateur établie, il est nécessaire de vérifier les actions auxquelles il est autorisé. La plupart des mécanismes de contrôle d'accès implémentés dans les systèmes de gestion de contenus sont basés sur des langages XML permettant de décrire des politiques de contrôle d'accès et de gestion des droits numériques. Ces systèmes exploitent des métadonnées formatées associées aux ressources dont il s'agit de contrôler l'accès. Parmi ceux-ci, les plus connus sont XrML [9] (*eXtensible Right Markup Language*) utilisé comme base du langage d'expression des droits du standard MPEG-21 [10], ODRL [11](*Open Digital Right Language*) implémenté par l'Open Mobile Al-

8. X509 certificates : http://www.ietf.org/rfc/rfc2459.txt
9. http://www.xrml.org/
10. http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
11. http://www.w3.org/TR/odrl/

liance (OMA) et XACML [12] (*Extensible Access Control Markup Language*) développé par OASIS. Le modèle ODRL repose sur les notions d'*Asset*, *Party*, *Permission*, *Constraint*, *Requirement*, *Condition*, *Rights holder*, *Context*, *Offer*, *Agreement*, et *Revoking rights*. Le modèle XACML permet de représenter des stratégies de contrôle d'accès sous forme de règles et repose sur les notions de *Rule*, *Policy* et *Policy Set* ; une *Rule* comprend des *Conditions* et *Effects*, et une *Policy* des *Rules* et *Obligations*.

Avec l'émergence du web 2.0 et du web sémantique, de nouveaux systèmes de gestion de contenu ont vu le jour qui reposent sur ces nouvelles approches du web pour gérer l'accès au contenu. Citons le système de contrôle d'accès basé sur RDF qu'utilise le W3C depuis 2001 pour contrôler l'accès aux fichiers de ses serveurs : W3C ACL System [13]. Plus récemment, dans (Hollenbach *et al.*, 2009), une évolution de ce système a été proposée pour contrôler les accès de manière décentralisée, à grande échelle. Dans le prolongement de ces premiers travaux, différents modèles de contrôle des droits d'accès ont été proposés, parmi lesquels un modèle pour un wiki sémantique qui repose sur une ontologie et des règles d'inférence, assurant ainsi son adaptabilité à différentes politiques d'autorisation (Buffa *et al.*, 2010). On retrouve dans les différents modèles sémantiques qui existent les notions déjà présentes dans les langages XML plus anciens. En reposant sur des connaissances intégrées de façon homogène aux bases de connaissances du système, ils permettent en outre une plus grande flexibilité et expressivité des droits personnalisés, prenant en compte le profil des utilisateurs, leur géo-localisation, un nombre d'accès, des contraintes de temps, etc.

Les besoins en termes de gestion des droits d'accès s'apparentent également à ceux des bibliothèques numériques dont une synthèse est proposée dans (Coyle, 2004). Citons notamment les travaux sur l'architecture Fedora de gestion des ressources numériques (Lagoze *et al.*, 2006) et ceux sur la bibliothèque numérique sémantique JeromeDL (Kruk *et al.*, 2008).

Dans le prolongement des problèmes d'interopérabilité, la composition de SI peut amener à modifier les usages, ce qui correspond probablement à l'un des points les moins étudiés dans ce cadre bien que se révélant l'un des plus critiques.

9. Interopérabilité des usages

L'interopérabilité des SI met en exergue de difficiles problèmes d'homogénéité aussi bien au niveau de l'interaction homme-machine (IHM) que du contenu fourni aux utilisateurs. En effet, chaque SI intégré disposait préalablement de sa propre IHM. Les choix de conception, d'interaction et de design ef-

12. http://www.oasis-open.org/committees/xacml/
13. http://www.w3.org/2001/04/20-ACLs

fectués dans un contexte d'usage spécifique ont conduit à des IHM hétérogènes. Ainsi les problèmes d'alignement de concepts/contenus rencontrés au niveau de l'interopérabilité des données et des processus métiers transparaissent au niveau des IHM qui exposent par exemple des labels identiques pour des données différentes ou des labels différents en cas de données identiques.

Les solutions reposent actuellement soit sur une refonte complète des IHM, soit sur une juxtaposition des IHM de chaque SI. Dans le cas d'une refonte, les choix de conception sont totalement revus et repensés. Une nouvelle IHM est conçue, nécessitant un investissement lourd équivalent à la création de l'interface d'un nouveau système. A l'inverse l'intégration qui consiste à récupérer autant que possible les IHM pré-existantes en les « adaptant » n'est pas sans difficulté. Dans cette approche, le choix même des applications à intégrer peut alors être guidé par la capacité de celles-ci à voir leurs IHM être adaptées. Dans le contexte d'un exemple industriel, tout un processus a été défini afin de sélectionner parmi plusieurs SI le meilleur candidat à l'intégration. En matière d'IHM, ils étudient la possibilité d'effectuer des changements comme supprimer des menus ou les griser, masquer des parties de l'interface lorsqu'elles sont définies en HTML, etc. Ces changements doivent cependant supporter des évolutions potentielles des SI intégrés ce qui explique qu'ils ne touchent jamais aux styles (CSS) des IHM fournies avec les SI intégrés, ni ne retirent de composants, actions lourdes qui devraient être répétées à chaque nouvelle version du SI intégré. Les IHM du SI sont alors une juxtaposition améliorée mais dont le style n'est pas homogène. Il existe souvent une forte discontinuité dans l'interaction lors du passage d'une entité à une autre. De telles discontinuités dégradent l'utilisabilité des logiciels et du SI dans son ensemble. De telles solutions ne sont donc pas convaincantes.

Pour pallier ces problèmes, l'utilisation de modèles offre des pistes pour raisonner de manière plus abstraite et permettre aux utilisateurs de mieux comprendre et faire évoluer leur système. Les modèles conceptuels sous-jacents à la conception d'IHM (Paterno, Santoro, 2002 ; Sottet *et al.*, 2009 ; Blumendorf *et al.*, 2010) sont tout ou partie embarqués à l'exécution. Ils peuvent alors être utilisés pour commenter l'interaction homme-machine. Dans ce cadre, l'auto-explication des IHM (Pangoli, Paternó, 1995 ; Frey *et al.*, 2010 ; 2012) a pour but de fournir des explications à base de modèles sur une interface lors de son utilisation. L'hétérogénéité des interfaces homme-machine est alors conservée, mais la compréhension du système et son utilisabilité sont améliorées. Par exemple, dans (Frey *et al.*, 2011), le modèle *Question Options Criteria* (MacLean *et al.*, 1991), QOC, documente les raisons justifiant un choix de conception : les Questions identifient les questions-clés de la conception ; les Options fournissent les réponses possibles aux Questions et les Critères permettent l'évaluation et la comparaison des Options. Les critères de QOC peuvent être vus comme des critères de qualité. Les concepteurs d'IHM doivent définir leurs IHM en utilisant les modèles de définition de l'IHM. Quand un choix de conception s'offre à eux, ils doivent l'exprimer sous la forme d'une question, d'options

et de critères. Les options sont les alternatives qui peuvent être proposées à l'utilisateur et les critères fournissent les justifications des différents choix en termes de qualité. Ainsi la figure 4 montre comment un concepteur peut avoir le choix entre deux interacteurs différents pour la saisie d'une date et peut justifier chacune des options par des modèles de qualité, représentant ici des critères ergonomiques (Frey *et al.*, 2011). Il reste à embarquer ces modèles à l'exécution afin de pouvoir les utiliser pour répondre aux questions des utilisateurs. Par exemple, si l'utilisateur se demande pourquoi il doit utiliser un calendrier pour saisir une date, le système pourra lui répondre que ce choix favorise les critères ergonomiques d'adéquation à la tâche, de conformité aux attentes des utilisateurs, de contrôlabilité et de tolérance aux fautes.

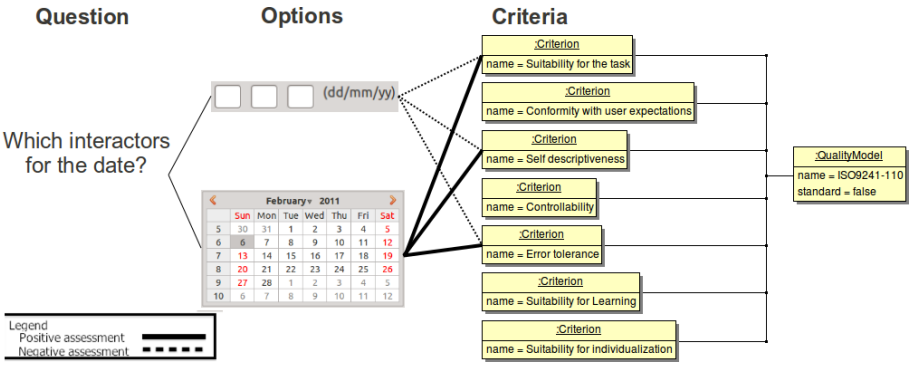

Figure 4. Exemple d'options de conception issu de (Frey et al.*, 2010)*

La composition des IHM en elle-même vise soit à assembler des IHM existantes, soit à les redéfinir complètement en les uniformisant (*ex. Doodle et GoogleCalendar peuvent être composés pour proposer aux participants la visualisation de leur agenda personnel avec les dates et heures proposées par le sondage*). Les modèles ciblent la possibilité d'obtenir de telles compositions de manière semi-automatique en prenant en compte toute la complexité à la fois des composants graphiques et de leurs interactions avec la partie fonctionnelle (Gabillon *et al.*, 2011 ; Joffroy *et al.*, 2011 ; Brel *et al.*, 2011). Ils peuvent aussi, lors de l'évolution des SSI, assurer la cohérence globale du SI, en maintenant les interactions entre la partie métier et la partie usage. La figure 5 illustre une telle approche où des modèles sont issus des applications existantes pour pouvoir composer les interfaces avant de générer une nouvelle application. On trouve des étapes 1) d'abstraction des applications pour obtenir des modèles du noyau fonction et de l'interface homme-machine représentant les applications (point 1), 2) d'abstraction de la composition fonctionnelle (point 2), 3) des étapes de la réalisation de la composition d'IHM basée sur les modèles (point 3) et de la concrétisation du résultat obtenu (point 4).

Dans la plupart des solutions actuelles, l'utilisateur, qu'il soit développeur dans la phase d'intégration ou utilisateur du SI final, est partie prenante pour

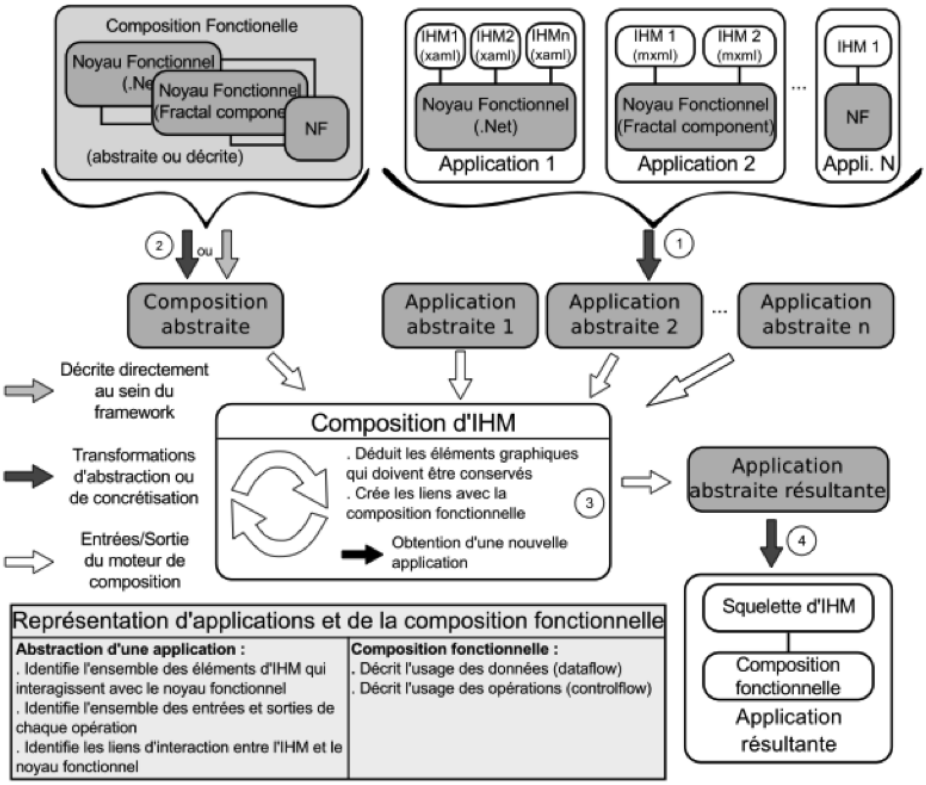

Figure 5. Démarche globale pour la réalisation de la composition d'applications proposée dans (Joffroy et al.*, 2011)*

le bon usage des SI intégrés. Il convient de lui offrir la possibilité de manipuler les modèles de son SI pour le faire évoluer. La modélisation par les utilisateurs finaux ou « *End User Modelling* » est assurément l'objectif pour assurer l'utilisabilité de systèmes.

10. Conclusion

L'interopérabilité des systèmes d'information est nécessaire à la maîtrise de l'évolution, constante, des systèmes informatiques agrégés. Les travaux issus de nos communautés couvrent une partie des problèmes liés à l'interopérabilité des systèmes. Ils se situent à tous les niveaux du cycle de vie et d'interopérabilité des SI.

Des avancées importantes ont ainsi été réalisées ces dernières années pour d'une part, promouvoir une montée en abstraction « productive » et, d'autre part, faciliter cette approche par une analyse des codes eux-mêmes appréhendés

comme des modèles. La très large palette de travaux introduits ici montre cette richesse en portant un regard commun sur ces approches dans un contexte *système d'information*. Nous ne prétendons pas à l'exhaustivité, mais à une vision croisée des travaux menés et des pistes à creuser. Ainsi ces travaux ne se substituent pas aux avancées obtenues dans le domaine des systèmes distribués, mais se positionnent soit en amont par l'abstraction des mises en œuvre, voire la production des codes, ou en aval par l'extraction des modèles nécessaires au raisonnement. De plus, les travaux présentés ont souvent été appliqués à des problèmes issus des systèmes industriels ; leur usage dans le cadre des SI nécessite vraisemblablement une adaptation, souvent due à une meilleure prise en compte des utilisateurs finaux.

La considération des différents points de vue sur le SI ne masque pas le besoin important d'une meilleure corrélation entre eux. La communauté de l'IDM, par un mouvement dit *Aspect Oriented Modeling*, contribue aujourd'hui largement à ce questionnement, en particulier à travers des articles présentés lors de la conférence AOSD. Ainsi, plusieurs d'entre nous envisagent d'approfondir cette première étude par une réflexion autour d'une algèbre qui permettrait de capturer formellement les liens entre les différents modèles qui constituent un SI et qui influent nécessairement sur son interopérabilité. Ainsi, on pourrait imaginer calculer l'interopérabilité entre deux SI comme on résout une équation.

La corrélation s'applique aussi du point de vue de l'humain. Outre les techniques de capture des interactions, il est aussi nécessaire d'aborder les différentes facettes sur l'interopérabilité en considérant les points de vue de l'utilisateur final et de son organisation (l'objectif étant de fournir un SI utile et utilisable), comme par exemple, avoir des processus métiers compréhensibles par les utilisateurs finaux et qui tiennent compte des règles de sécurité.

Enfin, les différents travaux présentés devraient faciliter l'aide au pilotage du SI, ce qui présente une voie d'applications particulièrement intéressante, car à très grande valeur ajoutée.

Remerciements

Nous remercions les relecteurs de la conférence INFORSID et de ce numéro spécial qui par leurs commentaires nous ont guidés dans la présentation de cette étude.

Bibliographie

Acher M., Cleve A., Collet P., Merle P. *et al.* (2011, septembre). Reverse Engineering Architectural Feature Models. In Springer (Ed.), *5th European Conference of Software Architecture (ECSA)*, vol. 6983, p. 220-235. Essen, Allemagne, Springer. http://hal.inria.fr/inria-00614984

Acher M., Cleve A., Perrouin G., Heymans P. *et al.* (2012, janvier). On Extracting Feature Models From Product Descriptions. In *Vamos'12*, p. 10. Leipzig, Germany, ACM. https://nyx.unice.fr/publis/acher-cleve-etal:2012.pdf

Acher M., Collet P., Gaignard A., Lahire P. *et al.* (2012, septembre). Composing Multiple Variability Artifacts to Assemble Coherent Workflows. *Software Quality Journal Special issue on Quality Engineering for Software Product Lines*, vol. 20, n° 3-4, p. 689–734. http://hal.archives-ouvertes.fr/hal-00733556/PDF

Acher M., Collet P., Lahire P., France R. (2012, mars). Separation of Concerns in Feature Modeling: Support and Applications. In *Aspect-oriented software development(aosd'12)*, p. 1-12. ACM. https://nyx.unice.fr/publis/acher-collet-etal:2012.pdf

Adedjouma M., Dubois H., Terrier F. (2012, juin). Merging the quality assessment of processes and products in automotive domain. In *Proceedings of the 13th international conference on Product-Focused software development and process improvement (PROFES 2012)*, p. 275-289. Madrid, Spain.

Adedjouma M., Dubois H., Terrier F., Kitouni T. (2012, mai). An experiment on merging quality assessment in automotive domain. In *Proceedings of the 12th international SPICE conference on process improvement and capability dEtermination in software, systems engineering and service management (SPICE 2012)*, p. 107-117. Palma de Mallorca, Spain.

Amar B., Osman Guédi A., Miralles A., Huchard M. *et al.* (2012, juin). Using Formal Concept Analysis to Extract a Greatest Common Model. In L. A. Maciaszek, A. Cuzzocrea, J. Cordeiro (Eds.), *ICEIS 2012: 14th International Conference on Enterprise Information Systems*, vol. 1, p. 27-37. Wroclaw, Pologne, SciTePress. http://hal-lirmm.ccsd.cnrs.fr/lirmm-00727009

Anwar A., Ebersold S., Coulette B., Nassar M., Kriouile A. (2010). A Rule-Driven Approach for composing Viewpoint-oriented Models. *Journal of Object Technology*, vol. 9, n° 2, p. 89–114.

Atigui F., Ravat F., Teste O., Zurfluh G. (2010). Démarche dirigée par les modèles pour la conception d'entrepôts de données multidimensionnelles. In *Journées Bases de Données Avancées (BDA), Toulouse, 18/10/2010-22/10/2010*, p. (support électronique). http://www.ups-tlse.fr, Université Paul Sabatier - Toulouse. ftp://ftp.irit.fr/IRIT/SIG/2010_ARTZ_BDA.pdf

Bezivin J., Bouzitouna S., Fabro M. D. D., Gervais M.-P. *et al.* (2006). A Canonical Scheme for Model Composition. In A. Rensink, J. Warmer (Eds.), *Ecmda-fa*, vol. 4066, p. 346–360. Springer.

Bizer C., Cyganiak R. (2006). D2R Server Publishing Relational Databases on the Semantic Web (poster). In *5th international semantic web conference.* Athens.

Blanc X., Mounier I., Mougenot A., Mens T. (2008). Detecting model inconsistency through operation-based model construction. In *Icse '08: Proceedings of the 30th international conference on software engineering*, p. 511–520. New York, NY, USA, ACM.

Blanchard T. (2011, Dec.). *"on ne choisit pas son si", assistance publique hôpitaux de marseille, journées de l'as interopérabilité des si.* https://glc.i3s.unice.fr/_media/public:seminars_manifestations:ppt_metro.pptx.

Blumendorf M., Lehmann G., Albayrak S. (2010). Bridging models and systems at runtime to build adaptive user interfaces. In *Proceedings of the 2nd acm sigchi*

symposium on engineering interactive computing systems, p. 9–18. New York, NY, USA, ACM. http://doi.acm.org/10.1145/1822018.1822022

Brel C., Renevier-gonin P., Riveill M. (2011). OntoCompo : A Tool to Enhance Application Composition. *Ifip International Federation For Information Processing*, p. 588–591.

Buffa M., Faron-Zucker C., Kolomoyskaya A. (2010). Gestion sémantique des droits d'accès au contenu : l'ontologie amo. In S. B. Yahia, J.-M. Petit (Eds.), *Egc*, vol. RNTI-E-19, p. 471-482. Cépaduès-Éditions.

CENELEC. (2001, December). *En61508. functional safety of electrical/electronic-/programmable electronic safety-related systems.*

Charfi A., Mezini M. (2004). Aspect-Oriented Web Service Composition with AO4BPEL. In *European conference on web services*, p. 168–182.

Chikofsky E. J., Cross II J. H. (1990, January). Reverse engineering and design recovery: A taxonomy. *IEEE Softw.*, vol. 7, p. 13–17. http://dx.doi.org/10.1109/52.43044

Clavreul M., Mosser S., Blay-Fornarino M., France R. (2011, octobre). Service-oriented Architecture Modeling: Bridging the Gap Between Structure and Behavior. In *Models'11*, p. 1-10. New Zealand.

Combemale B. (2008). Ingénierie Dirigée par les Modèles (IDM) État de l'art. *Management*, n° Idm, p. 1–19. http://hal.archives-ouvertes.fr/hal-00371565/

Courbis C., Finkelstein A. (2005). Towards Aspect Weaving Applications. In *Icse '05: Proceedings of the 27th international conference on software engineering*, p. 69–77. New York, NY, USA, ACM.

Coyle K. (2004). Rights Management and Digital Library Requirements. *Ariadne*, vol. 40. http://www.ariadne.ac.uk/

Czarnecki K., Wasowski A. (2007). Feature diagrams and logics: There and back again. In *Splc'07*, p. 23-34.

Demeyer S., Ducasse S., Nierstrasz O. (2002). *Object-oriented reengineering patterns.* Morgan Kaufmann. http://www.iam.unibe.ch/~scg/OORP

Demeyer S., Ducasse S., Tichelaar S. (1999, octobre). Why unified is not universal. UML shortcomings for coping with round-trip engineering. In B. Rumpe (Ed.), *Uml'99*, vol. 1723, p. 630–644. Kaiserslautern, Germany, Springer-Verlag. http://scg.unibe.ch/archive/papers/Deme99dUML99.pdf

Dubois H., Ibanez V., Lopez C., Machrouh J. *et al.* (2012, February). The product-line engineering approach in a model-driven process. In *Proceedings of the 6th european congress for embedded real time software and systems conference, ertsc.* Toulouse, France.

Ducasse S., Anquetil N., Bhatti U., Cavalcante Hora A. *et al.* (2011). *Mse and famix 3.0: an interexchange format and source code model family*. Rapport technique. RMod – INRIA Lille-Nord Europe. http://rmod.lille.inria.fr/archives/reports/Duca11c-Cutter-deliverable22-MSE-FAMIX30.pdf

Ducasse S., Gîrba T., Kuhn A. (2006). Distribution map. In *Proceedings of 22nd ieee international conference on software maintenance (icsm '06)*, p. 203–212. Los Alamitos CA, IEEE Computer Society. http://scg.unibe.ch/archive/papers/Duca06cDistributionMap.pdf

European Commission. (2007). Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). *Official Journal of the European Union*, vol. 50, n° L 108, p. 1–14. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:EN:PDF

Euzenat J., Shvaiko P. (2007). *Ontology matching*. Springer.

Farfeleder S., Moser T., Krall A., Stalhane T. *et al.* (2011, april). Dodt: Increasing requirements formalism using domain ontologies for improved embedded systems development. In *Design and diagnostics of electronic circuits systems (ddecs), ieee*, p. 271 -274.

Favre J.-M., Establier J., Blay-Fornarino M. (Eds.). (2006). *L'ingénierie dirigée par les modèles : au-delà du mda*. Cachan, France, Hermès-Lavoisier.

Figer J.-P. (2008, août). Principes d'urbanisation pour un système d'information, Référence H6000. In *Traité technologies logicielles et architecture des systèmes*, p. 24. Techniques de l'Ingénieurs.

Formica A. (2006). Ontology-based concept similarity in formal concept analysis. *Information Sciences*, vol. 176, n° 18, p. 2624-2641.

Fowler M. (2003). *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional.

France R., Fleurey F., Reddy R., Baudry B., Ghosh S. (2007). Providing Support for Model Composition in Metamodels. In *Edoc'07 (entreprise distributed object computing conference)*. Annapolis, MD, USA.

France R., Rumpe B. (2007). Model-driven Development of Complex Software: A Research Roadmap. In *Fose '07: 2007 future of software engineering*, p. 37–54. Washington, DC, USA, IEEE Computer Society.

Frey A. G., Calvary G., Dupuy-Chessa S. (2010). Xplain: an editor for building self-explanatory user interfaces by model-driven engineering. In *Proceedings of the 2nd acm sigchi symposium on engineering interactive computing system, eics 2010*, p. 41-46. ACM.

Frey A. G., Calvary G., Dupuy-Chessa S. (2012). Users need your models!: exploiting design models for explanations. In *Bcs-hci '12 proceedings of the 26th annual bcs interaction specialist group conference on people and computers, bcs-hci 2012, 12-14 september 2012, birmingham, uk*, p. 79-88.

Frey A. G., Ceret E., Dupuy-Chessa S., Calvary G. (2011). Quimera: a quality metamodel to improve design rationale. In *Proceedings of the 3rd acm sigchi symposium on engineering interactive computing system, eics 2011, pisa, italy, june 13-16, 2011*, p. 265-270. ACM.

Gabillon Y., Calvary G., Fiorino H. (2011). Composition d'interfaces homme-machine en contexte : approche par planification automatique. *Technique et science informatiques*, vol. 30, p. 25.

Gillet M., Gillet P. (2011). *Les systèmes d' information de a à z*. Dunod.

Hacid M.-S., Lécué F., Léger A., Rey C., Toumani F. (2009). Les web services sémantiques, automate et intégration. I. Introduction, éléments et scénarios, découverte de services web. *Techniques et sciences informatiques*, vol. 28, n° 2, p. 229–262. http://tsi.revuesonline.com/article.jsp?articleId=12997

Hillairet G., Bertrand F., Lafaye J.-Y. (2008). MDE for Publishing Data on the Semantic Web. In *Twomde international workshop on transformation and weaving ontologies in model driven engineering at models'08*, p. 32–46. Toulouse.

Hollenbach J., Presbrey J., Berners-Lee T. (2009). Using RDF Metadata to Enable Access Control on Social Semantic Web. In *International semantic web conference, ISWC 2009*. Springer.

ISO. (2004a). *ISO/IEC 15504. information technology - process assessment.*

ISO. (2004b, December). *NF EN ISO 14001. systèmes de management environnemental. exigences et lignes directrices pour son utilisation.*

ISO. (2008, November). *NF EN ISO 9001. systèmes de management de la qualité.*

ISO. (2011, November). *ISO 26262-1. road vehicles - Functional safety.*

Jacobson I., Griss M., Jonsson P. (1997). *Software reuse: Architecture, process and organization for business success*. Addison Wesley/ACM Press.

Joffroy C., Caramel B., Pinna-Dery A.-M., Riveill M. (2011). When the Functional Composition Drives the User Interfaces Composition : Process and Formalization. *Interfaces*, p. 207–216.

Kang K., Kim S., Lee J., Kim K. *et al.* (1998). Form: A feature-oriented reuse method with domain-specific reference architectures. *Annals of Software Engineering*, vol. 5, n° 1, p. 143–168.

Kienzle J., Al Abed W., Jacques K. (2009). Aspect-oriented multi-view modeling. In *Proceedings of aosd 09*, p. 87. ACM Press. http://portal.acm.org/citation.cfm ?doid=1509239.1509252

Klein J., Fleurey F., JÈzÈquel J. M. (2007). Weaving Multiple Aspects in Sequence Diagrams. *Transactions on Aspect-Oriented Software Development (TAOSD)*, vol. LNCS 4620, p. 167–199.

Konstantinou N., Spanos D.-E., Mitrou N. (2008). Ontology and Database Mapping: A Survey of Current Implementations and Future Directions. *Journal of Web Engineering*, vol. 7, n° 1, p. 1–24. http://www.citeulike.org/user/SwissJ74/article/ 3441433

Kruk S. R., Cygan M., Gzella A. (2008). JeromeDL - Semantic and Social Technologies for Improving User Experience in Digital Libraries. In *World wide web conference, WWW 2008*. ACM.

Lagoze C., Payette S., Shin E., Wilper C. (2006). Fedora: an Architecture for Complex Objects and their Relationships. *Int. J. on Digital Libraries*, vol. 6, n° 2, p. 124-138.

Lanza M., Ducasse S. (2003, septembre). Polymetric views—a lightweight visual approach to reverse engineering. *Transactions on Software Enginee-*

ring (TSE), vol. 29, n° 9, p. 782–795. http://scg.unibe.ch/archive/papers/Lanz03dTSEPolymetric.pdf

Ledinot E., Gassino J., Blanquart J.-P., Boulanger J.-L., Quéré P., Ricque B. (2012, February). A cross-domain comparison of software development assurance. In *Proceedings of the 6th european congress for embedded real time software and systems conference, ertsc.* Toulouse, France.

Lehman M., Belady L. (1985). *Program evolution: Processes of software change.* London, London Academic Press. ftp://ftp.umh.ac.be/pub/ftp_infofs/1985/ProgramEvolution.pdf

Lethbridge T., Tichelaar S., Plödereder E. (2004). The dagstuhl middle metamodel: A schema for reverse engineering. *Electronic Notes in Theoretical Computer Science*, vol. 94, p. 7-18.

MacLean A., Young R. M., Bellotti V. M. E., Moran T. P. (1991). Questions, options, and criteria: Elements of design space analysis. *Human-Computer Interaction*, vol. 6, n° 3, p. 201–250. http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:Questions,+Options,+and+Criteria:+Elements+of+Design+Space+Analysis\#0

Melnik S., Rahm E., Bernstein P. A. (2003). Rondo: A Programming Platform for Generic Model Management. In A. Y. Halevy, Z. G. Ives, A. Doan (Eds.), *Sigmod conference*, p. 193–204. ACM.

Metzger A., Pohl K., Heymans P., Schobbens P.-Y., Saval G. (2007). Disambiguating the documentation of variability in software product lines: A separation of concerns, formalization and automated analysis. In *Re'07*, p. 243–253.

Michel F., Gaignard A., Ahmad F., Barillot C. *et al.* (2010, juin). Grid-wide neuroimaging data federation in the context of the NeuroLOG project. In *Healthgrid'10*, p. 112–123. Paris, IOS Press. http://nyx.unice.fr/publis/michel-gaignard-etal:2010.pdf

Mosser S., Blay-Fornarino M. (2012). ADORE, a Logical Meta-model Supporting Business Process Evolution. *Science of Computer Programming (SCP)*, p. 1-35. https://nyx.unice.fr/publis/mosser-blay-fornarino:2012.pdf

Mosser S., Blay-Fornarino M., Duchien L. (2012, juillet). A Commutative Model Composition Operator to Support Software Adaptation. In *Ecmfa'12*, p. 4-19. Copenhagen, Denmark, Springer LNCS.

Mosser S., Mussbacher G., Blay-Fornarino M., Amyot D. (2011, mars). From Aspect-oriented Requirements Models to Aspect-oriented Business Process Design Models. In *10th international conference on aspect oriented software development(aosd'11), long paper: 1st round.* Porto de Galinhas.

Mosser S., Parra C., Duchien L., Blay-Fornarino M. (2012). Using Domain Features to Handle Feature Interactions. In *Vamos'12*, p. 101-110. Leipzig, Germany. https://nyx.unice.fr/publis/mosser-parra-etal:2012.pdf

Nemo C., Blay-Fornarino M., Kniesel G., Riveill M. (2007). SEMANTIC ORCHESTRATIONS MERGING - Towards Composition of Overlapping Orchestrations. In J. Filipe (Ed.), *Iceis'2007.* Funchal, Madeira.

Neves L., Teixeira L., Sena D., Alves V. *et al.* (2011). Investigating the safe evolution of software product lines. In E. Denney, U. P. Schultz (Eds.), *Gpce*, p. 33-42. ACM.

Nierstrasz O., Ducasse S., Gîrba T. (2005). The story of Moose: an agile reengineering environment. In *Proceedings of the european software engineering conference (esec/fse'05)*, p. 1–10. New York NY, ACM Press. http://scg.unibe.ch/archive/papers/Nier05cStoryOfMoose.pdf

OASIS. (2006). *Reference Model for Service Oriented Architecture 1.0*. Rapport technique n° wd-soa-rm-cd1. Auteur.

OMG. (2010, June). *OMG Systems Modeling Language (OMG SysML). Version 2.1.*

OMG. (2011). *Business process management notation (BPMN 2.0).* http://www.omg.org/spec/BPMN/2.0/

OMG. (2011). *Knowledge discovery metamodel - version 1.3.* http://www.omg.org/spec/KDM/1.3

OMG. (2011). *Syntax Tree Metamodel - Version 1.0.* http://www.omg.org/spec/ASTM/1.0

Oren E., Heitmann B., Decker S. (2008). ActiveRDF: Embedding Semantic Web data into object-oriented languages. *J. Web Sem.*, vol. 6, n° 3, p. 191–202.

Pangoli S., Paternó F. (1995). Automatic generation of task-oriented help. *Proceedings of the 8th annual ACM symposium on User interface and software technology UIST 95*, p. 181–187. http://portal.acm.org/citation.cfm?doid=215585.215971

Paterno F., Santoro C. (2002). One model, many interfaces. In C. Kolski, J. Vanderdonckt (Eds.), *Cadui'2002*, p. 143-154. Kluwer.

Pohl K., Böckle G., Linden F. J. van der. (2005). *Software product line engineering: Foundations, principles and techniques.* Springer-Verlag.

Quasthoff M., Meinel C. (2009). Design Pattern for Object Triple Mapping. In *Ieee scc*, p. 443–450. IEEE Computer Society.

Rabiser R., Grunbacher P., Dhungana D. (2007, sept.). Supporting product derivation by adapting and augmenting variability models. In *Software product line conference, 2007. splc 2007*, p. 141 -150.

Reddy R., France R., Ghosh S., Fleurey F., Baudry B. (2005). Model composition - a signature-based approach. In *Aspect oriented modeling (aom) workshop.* Montego Bay, Jamaica.

Reiser M.-O., Weber M. (2007). Multi-level feature trees: A pragmatic approach to managing highly complex product families. *Requir. Eng.*, vol. 12, n° 2, p. 57–75.

Schmid K., Rabiser R., Grünbacher P. (2011). A comparison of decision modeling approaches in product lines. In *Vamos '11*, p. 119–126. New York, NY, USA, ACM. http://doi.acm.org/10.1145/1944892.1944907

Schobbens P.-Y., Heymans P., Trigaux J.-C., Bontemps Y. (2007). Generic semantics of feature diagrams. *Comput. Netw.*, vol. 51, n° 2, p. 456–479.

Seidl C., Heidenreich F., Assmann U. (2012). Co-evolution of models and feature mapping in software product lines. In *Splc '12*, p. 76–85. New York, NY, USA, ACM. http://doi.acm.org/10.1145/2362536.2362550

She S., Lotufo R., Berger T., Wasowski A., Czarnecki K. (2011). Reverse engineering feature models. In *Icse'11.*

Soley R., OMG Staff Strategy Group the. (2000). *Model-driven architecture. white paper, draft 3.2. available at www.omg.org.*

Sottet J. S., Calvary G., Favre J. M., Coutaz J. (2009). Megamodeling and metamodel-driven engineering for plastic user interfaces: mega-ui. *HumanCentered Software Engineering*, vol. 2, p. 173–200. http://www.springerlink.com/index/k4349864gvv36k76.pdf

Stumme G., Maedche A. (2001). Fca-merge: Bottom-up merging of ontologies. In *Proceedings of the seventeenth international joint conference on artificial intelligence, ijcai 2001, seattle, washington, usa, august 4-10, 2001*, p. 225-234. Morgan Kaufmann.

Tatsiopoulos C., Boutsinas B. (2009). Ontology mapping based on association rule mining. In *Proceedings of the 11th international conference on enterprise information systems, iceis, milan, italy, may 6-10, 2009*, p. 33-40.

Telelogic. (2008). *Doors.* http://www2.telelogic.com/products/doorsers/doors/index.cfm

Thomas F., Delatour J., Gérard S., Brun M., Terrier F. (2007). Contribution à la modélisation explicite des plates-formes d'exécution pour l'IDM. *L'objet*, vol. 13, p. 9–32.

Thüm T., Batory D., Kästner C. (2009). Reasoning about edits to feature models. In *Icse'09*, p. 254–264. IEEE.

Urli S., Blay-Fornarino M., Collet P., Mosser S. (2012, septembre). Using Composite Feature Models to Support Agile Software Product Line Evolution. In *Models and evolution 2012(me'12), workshop.*

Verjus H., Pourraz F. (2007). A Formal Framework For Building, Checking And Evolving Service Oriented Architectures. *Fifth European Conference on Web Services ECOWS07*, p. 245–254. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4399753

Wouters L. (2013). *Multi-Domain Expert-User Modeling Infrastructure.* Thèse de doctorat non publiée, Pierre et Marie Curie.