
Interopérabilité des systèmes d'information : approches dirigées par les modèles

Annie Authosserre-Cavarero, Frederic 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 Goer, Johan Montagnat, Anne-Marie Pinna-Dery

Action Spécifique : "Interopérabilité des SI et ingénierie des modèles"
annie.authosserre@unice.fr, 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 : quels défis, quelles solutions ? » soutenue par INFORSID. Nous proposons une synthèse d'un ensemble d'approches basées sur l'IDM et l'ingénierie des connaissances répondant à des problèmes du monde industriel 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 with no entropy creation. 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, Evolution

KEYWORDS: Interoperability, Model-Driven Engineering, Knowledge Engineering, Evolution

1. Introduction

Les systèmes d'information (SI) sont une 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. 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 à la fois 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 [FAV 06]. 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 des 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 terme d'interopérabilité au sein des SI (*cf.* SEC. 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)¹ fourni par la Direction générale de la modernisation de l'état (DGME) distingue six niveaux d'interopérabilité. Les niveaux politique, juridique et organisationnel, à l'instar de ce référentiel, ne sont pas étudiés dans cet article. Les trois autres niveaux d'interopérabilité : *sémantique* : « savoir se comprendre », *syntactique* : « savoir communiquer », et *technique* : « pouvoir communiquer » sont transversaux à nos axes. En effet, nous avons un point de vue « support » à l'interopérabilité et en fonction de l'étape traitée, nous utilisons différentes techniques qui couvrent parfois plusieurs niveaux d'interopérabilité. La figure 1 donne une vision globale de l'article.

Nous avons alors dégagé deux axes de réflexion : (1) la préparation à l'interopérabilité par extraction de modèles et respect de normes, (2) la mise en œuvre par les modèles de l'interopérabilité. Le premier axe nous ramène (i) au problème de la compréhension du SI et des techniques de modernisation que nous abordons en SEC. 3 et (ii) à la gestion de la conformité du SI relativement à des référentiels normatifs (*cf.* SEC. 4). Nous avons décliné notre étude du second axe selon les

1. http://references.modernisation.gouv.fr/sites/default/files/RGI_Version1%200.pdf

points de vue d'interopérabilité suivants : les domaines (*cf.* SEC. 5), les données (*cf.* SEC. 6), les processus (*cf.* SEC. 7), la gestion des règles de sécurité (*cf.* SEC. 8) et le dernier point de vue, essentiel, capture l'interopérabilité au niveau même des usages (*cf.* SEC. 9).

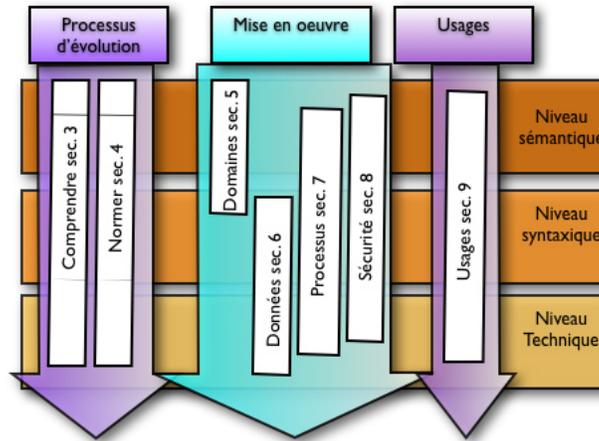


Figure 1. Axes d'étude et niveaux d'interopérabilité relativement aux travaux liés à l'IDM

2. Problématique de l'interopérabilité des SI

« On ne choisit pas son SI » résume une part des difficultés auxquelles les DSI doivent faire face aujourd'hui [BLA 11]. La plupart des SI ont les mêmes symptômes que les systèmes patrimoniaux [DEM 02]. Seule leur place centrale empêche leur mise en retraite illustrant par là même les lois de Lehman et Belady [LEH 85]. 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 stratégique au sein des entreprises. Elle doit permettre de prendre en compte :

la rapidité d'évolution des SI ; l'évolution technologique, réglementaire et structurale 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 [FIG 08]. 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. En effet, les périmètres applicatifs sont en règle générale fermés. Avoir une vue sur le SI ou sur des éléments du SI implique de définir une autre politique d'accès aux données basée non plus sur des domaines applicatifs mais sur des usages nécessitant des vues 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. 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. Pour cela nous nous sommes appuyés sur trois cas d'étude portés par des industriels : un SI construit graduellement au cours des vingt dernières années sous forme d'applications indépendantes souvent propriétaires (présentation autour d'une partie du SI des hôpitaux de Marseille), un SI centralisé sur les données construit sur un socle commun avec des problèmes d'évolution des règles métiers (CERTIAM) et un système construit autour d'un socle commun mais avec des applications satellites et la mise en place de mécanismes d'interopérabilité aux différents niveaux des couches logicielles en fonction des besoins (ProBTP). Nous déclinons à présent les différents travaux liés à l'IDM et à l'ingénierie des connaissances, relativement à notre problématique d'interopérabilité, et utilisons ces exemples comme supports. Nous l'abordons selon les deux axes suivants : (1) préparation à l'interopérabilité des SI, (2) mise en œuvre de l'interopérabilité.

3. Comprendre le SI par les modèles

La problématique de l'interopérabilité de deux SI A et B peut être repensée en terme d'interopérabilité du modèle de A avec celui de B , que nous notons $m(A)$ et $m(B)$. 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 : $m_{\text{domaine}}(SI) \oplus m_{\text{donnée}}(SI) \oplus m_{\text{processus}}(SI) \oplus m_{\text{sécurité}}(SI) \oplus m_{\text{usage}}(SI)$. Ceci nous amène naturellement à décomposer l'interopérabilité de A et B pour chaque point de vue : $m_{\text{domaine}}(A)$ et $m_{\text{domaine}}(B)$, $m_{\text{donnée}}(A)$ et $m_{\text{donnée}}(B)$, 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 systèmes², sous-tendant l'idée qu'il est possible d'appréhender le SI et ses évolutions comme un objet mathématique classique.

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'interopérabilité. 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 des années 80 [CHI 90], 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.

Du point de vue technique, la phase de rétro-ingénierie implique l'utilisation d'analyseurs sur les différents artefacts qui constituent le SI pour produire des modèles bruts, des métriques pour les caractériser, et diverses transformations pour en dériver des modèles à plus forte valeur ajoutée. Par exemple, des visualisations sont souvent adéquates pour obtenir des cartes de ces systèmes [LAN 03, DUC 06], permettant d'accéder à un degré intéressant de compréhension de l'existant. La conformité des modèles obtenus vis-à-vis de leurs méta-modèles respectifs est aussi une question centrale, où deux écoles de pensée co-existent : celle d'un portefeuille de métamodèles standardisés – ou unifiés – comme c'est le cas avec l'initiative ADM de l'OMG [GRO 11a, GRO 11b], ou au contraire celle d'un portefeuille de méta-modèles créés sur mesure comme le propose FAMIX dans la plate-forme Moose [DUC 11] par exemple.

Une réelle compréhension du SI exige une modélisation fidèle à l'existant, nécessitant d'appréhender simultanément les différents points de vue (domaines,

2. <http://people.irisa.fr/Benoit.Combemale/research/gemoc/as2011/>

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 méta-modèles complets/multi-points de vues, 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. L'objectif de ces normes est de s'assurer que toutes les exigences sont bien définies pour le système considéré, que le système est conforme à celles-ci et qu'un utilisateur juge le système final satisfaisant. 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 04] (management environnemental) et la norme ISO9001 [ISO 08] (management de la qualité). Les autres normes considérées sont des normes liées au domaine d'application (nucléaire, transport aérien, ferroviaire ou automobile, géographique [Eur 07] 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 référentiels nationaux ou internationaux est sous-jacente pour chacun d'eux. Que cela concerne l'adéquation aux droits français ou européen, à la qualité logicielle et des systèmes, 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.

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 [LED 12]. 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 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 CESAR³ 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. La composition de SSI amène là aussi de nouveaux challenges dans la

3. <http://www.cesarproject.eu>

prise en compte des études déjà réalisées pour des SI existants afin d'aider à la certification des SSI qui interopèrent. Avoir des processus dédiés aux activités de mise en conformité ainsi que des outils les implémentant faciliterait le travail et offrirait un support à ces activités qui concernent processus, produits et services.

Nous abordons à présent le deuxième axe de notre étude qui est celui des mises en oeuvre 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 terme de variabilité en ingénierie des lignes de produits logicielles et en terme 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. 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 [EUZ 07]. 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 subsumption. 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. 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. *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 (SPL) vise à produire des familles de variantes de programmes pour un domaine donné [POH 05]. 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. Une des approches les plus répandues est d’utiliser le formalisme des *Feature Model* (FM) qui permet de définir de manière compacte toutes ces caractéristiques et leurs combinaisons valides ; il s’agit pour simplifier d’un graphe AND-OR avec des contraintes propositionnelles [SCH 07, CZA 07, THÜ 09]. Les FM sont utilisés pour décrire la variabilité lors de la conception de logiciels mais également dans les autres phases de leur développement. Ainsi on utilise des FMs pour décrire les différentes parties du système, que ce soit la description du matériel [KAN 98], la structure organisationnelle [REI 07], les détails du commerce ou de la mise en œuvre [MET 07]. Des travaux abordent aujourd’hui l’extraction automatisée de FM à partir de grands systèmes logiciels [ACH 11c, SHE 11]. Ces FMs 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 FMs [ACH 11a]. Les FMs qui résultent de la capture des domaines et des besoins des utilisateurs peuvent alors être réassemblés pour construire des « produits » logiciels cohérents [ACH 11b]. 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.

5.3. *Gestion des exigences*

L’interopérabilité de 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 [DUB 12]). Ces travaux permettent de faciliter le 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 [MOS 11]. Prenons par exemple l’outil DODT[FAR 11] qui offre la possibilité de vérifier la cohérence entre une base d’exigences dans l’outil de gestion d’exigences DOORS[Doo] et une description du modèle de domaine à l’aide d’une ontologie. Des connexions existent

aussi entre des modèles SysML[OMG 10] 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.

Même lorsque les domaines sont alignés, la mise en oeuvre au niveau des données et des processus métier 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 dans une optique de *mashup*. Le développement des données ouvertes offre une opportunité pour reconsidérer l'échange de données entre SI et développer l'interopérabilité de processus relevant d'espaces de représentation différents.

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 [MIC 10]. 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 du 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 SEC. 8).

6.2. *Éléments 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 » [FOW 03] 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 [ORE 08]). 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 objet/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 [QUA 09]. 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 [HIL 08] [MEL 03]. Plus précisément, à partir d'un modèle métier enrichi, l'application di-

recte des principes de traduction entre UML et OWL propres à ODM⁴ permet la génération automatique de l'ontologie image. Celle-ci peut constituer un entrepôt interrogeable en SPARQL donnant un accès public aux données métier. 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 objet (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 [BIZ 06] [KON 08] et fait aujourd'hui l'objet d'une proposition du W3C (Working Draft R2RML : RDB to RDF Mapping language⁵).

Les apports de ces approches sont patents en termes de réutilisabilité et de maintenance. En donnant au modèle de classes métier 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 métier et des activités en charge d'assurer la traçabilité 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 métier

Une des composantes essentielles d'un système d'information est l'ensemble des processus métier qu'il supporte. Lorsque des SI doivent interopérer, on constate à la fois la construction de nouveaux processus métier (*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 orien-

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

5. <http://www.w3.org/TR/r2rml/>

tées services promeuvent un couplage lâche, la distribution des services, la dynamique et l'agilité [OAS 06]. 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 [FIG 08]. Dans le contexte SOA, les relations entre services sont portées par des orchestrations, qui correspondent à une mise en œuvre des processus métier. 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 [CHA 04] ou plus généralement aux différents niveaux de la définition des processus (analyse sémantique, moteur et orchestrations BPEL) [COU 05].

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 métier est aujourd'hui en pleine effervescence, en particulier avec le standard BPMN [OMG 11], 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 métier, 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 modèles. C'est ainsi que des travaux relatifs à l'introduction de fragments d'orchestration [MOS 10] ou de composition de scénarios sont proposés [KLE 07]. Nous noterons que ces compositions impactent également les points de vue domaine et données [KIE 09, CLA 11]. Dans ce contexte, différents travaux abordent alors la question du respect des propriétés initiales des processus [VER 07] (*ex. appel à un système d'anonymisation avant tout échange de données*), d'identification des services « redondants » [NEM 07] (*ex. appel à un unique service de facturation en regroupant un ensemble de requêtes d'authentification*), ... 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 [HAC 09].

Si la composition des processus métier 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 chiffrement 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 inaccessible, 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 X509⁶ 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 XrML⁷ (eXtensible Right Markup Language) utilisé comme base du langage d'expression des droits du standard MPEG-21⁸, ODRL⁹ (Open Digital Right Language)

6. X509 certificates : <http://www.ietf.org/rfc/rfc2459.txt>

7. <http://www.xrml.org/>

8. <http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm>

9. <http://www.w3.org/TR/odrl/>

implémenté par l'Open Mobile Alliance (OMA) et XACML¹⁰ (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 System¹¹. Plus récemment, dans [HOL 09], 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 [BUF 10]. 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 [COY 04]. Citons notamment les travaux sur l'architecture Fedora de gestion des ressources numériques [LAG 06] et ceux sur la bibliothèque numérique sémantique JeromeDL [KRU 08].

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 effectués 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 métier transparaissent au niveau des IHMs qui exposent par

10. <http://www.oasis-open.org/committees/xacml/>

11. <http://www.w3.org/2001/04/20-ACLs>

exemple des labels identiques pour des données différentes ou des labels différents en cas de données identiques.

Les solutions dans le monde industriel 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. À 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. L'un de nos exemples industriels a défini tout un processus 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 de faire évoluer leur système. Les modèles conceptuels sous-jacents à la conception d'IHM [SOT 09] 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 [PAN 95, FRE 10] a pour but de fournir des explications à base de modèles sur une interface lors de son utilisation. L'hétérogénéité des IHM est alors conservée, mais la compréhension du système et son utilisabilité sont améliorées. Par exemple, il est possible d'expliquer pourquoi un menu a été grisé en utilisant un modèle *Question Options Criteria* [MAC 91].

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 composition 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 [GAB 11, CED 11, BRE 11]. 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.

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 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

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, à la fois, promouvoir une montée en abstraction « productive » et 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 peut nécessiter une adaptation, souvent due à une prise en compte plus importante des utilisateurs finaux.

La prise en compte 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 au travers d'un mouvement dit *Aspect Oriented Modeling* contribue aujourd'hui largement à ce questionnement, en particulier au travers d'articles 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 interactions qui existent entre les différents modèles qui permettent de capturer l'essence des SI. Enfin, les différents travaux présentés devraient supporter l'aide au pilotage du SI, ce qui présente une voie d'applications particulièrement intéressantes car à très grande valeur ajoutée.

11. Bibliographie

- [ACH 11a] ACHER M., COLLET P., LAHIRE P., FRANCE R., « A Domain-Specific Language for Managing Feature Models », *SAC'11*, PL Track, ACM, 2011.
- [ACH 11b] ACHER M., COLLET P., LAHIRE P., GAIGNARD A., FRANCE R., MONTAGNAT J., « Composing Multiple Variability Artifacts to Assemble Coherent Workflows », *Software Quality Journal (Special issue on Quality Engineering for SPLs)*, , 2011.
- [ACH 11c] ACHER M., CLEVE A., COLLET P., MERLE P., DUCHIEN L., LAHIRE P., « Reverse Engineering Architectural Feature Models », 2011, ECSA'11 conference
- [BIZ 06] BIZER C., CYGANIAK R., « D2R Server Publishing Relational Databases on the Semantic Web (poster) », *5th Int. Semantic Web Conference*, Athens, 2006.

- [BLA 11] BLANCHARD T., « "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, Dec. 2011.
- [BRE 11] BREL C., RENEVIER-GONIN P., RIVEILL M., CEDEX S. A., « OntoCompo : A Tool to Enhance Application Composition », *Ifip International Federation For Information Processing*, , 2011, p. 588–591.
- [BUF 10] BUFFA M., FARON-ZUCKER C., KOLOMOYSKAYA A., « Gestion sémantique des droits d'accès au contenu : l'ontologie AMO », YAHIA S. B., PETIT J.-M., Eds., *EGC*, vol. RNTI-E-19 de *Revue des Nouvelles Technologies de l'Information*, Cépaduès-Éditions, 2010, p. 471-482.
- [CED 11] JOFFROY C., CAMEL B., PINNA AM, RIVEILL M., « When the Functional Composition Drives the User Interfaces Composition : Process and Formalization », *Interfaces*, 2011, p. 207–216.
- [CHA 04] CHARFI A., MEZINI M., « Aspect-Oriented Web Service Composition with AO4BPPEL », *European Conference on Web Services*, 2004, p. 168–182.
- [CHI 90] CHIKOFFSKY E. J., CROSS II J. H., « Reverse Engineering and Design Recovery : A Taxonomy », *IEEE Softw.*, vol. 7, 1990, p. 13–17, IEEE Computer Society Press.
- [CLA 11] CLAVREUL M., MOSSER S., BLAY-FORNARINO M., FRANCE R., « Service-oriented Architecture Modeling : Bridging the Gap Between Structure and Behavior », *14th Inter. Conference on Model Driven Engineering Languages and Systems(MODELS'11)*, , Wellington, New Zealand, octobre 2011, ACM/IEEE, p. 1-10.
- [COU 05] COURBIS C., FINKELSTEIN A., « Towards Aspect Weaving Applications », *ICSE '05 : Proceedings of the 27th inter. conference on Software engineering*, New York, NY, USA, 2005, ACM, p. 69–77.
- [COY 04] COYLE K., « Rights Management and Digital Library Requirements », *Ariadne*, vol. 40, 2004.
- [CZA 07] CZARNECKI K., WASOWSKI A., « Feature Diagrams and Logics : There and Back Again », *SPLC'07*, 2007, p. 23-34.
- [DEM 02] DEMEYER S., DUCASSE S., NIERSTRASZ O., *Object-Oriented Reengineering Patterns*, Morgan Kaufmann, 2002.
- [Doo] « Telelogic DOORS », <http://www2.telelogic.com/products/doorsers/doors/index.cfm>.
- [DUB 12] DUBOIS H., IBANEZ V., LOPEZ C., MACHROUH J., MELEDO N., SILVA A., « The product-line engineering approach in a model-driven process », *Proceedings of the 6th European Congress for Embedded Real Time Software and Systems Conference, ERTSC*, Toulouse, France, February 2012.
- [DUC 06] DUCASSE S., GIRBA T., KUHN A., « Distribution Map », *Proceedings of 22nd IEEE International Conference on Software Maintenance (ICSM '06)*, Los Alamitos CA, 2006, IEEE Computer Society, p. 203–212.
- [DUC 11] DUCASSE S., ANQUETIL N., BHATTI U., CAVALCANTE HORA A., LAVAL J., GIRBA T., « MSE and FAMIX 3.0 : an Interexchange Format and Source Code Model Family », rapport, 2011, RMod – INRIA Lille-Nord Europe.
- [Eur 07] EUROPEAN COMMISSION, « 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, 2007, p. 1–14, Official Journal of the European Union.
- [EUZ 07] EUZENAT J., SHVAIKO P., *Ontology matching*, Springer, 2007.
- [FAR 11] FARFELEDER S., MOSER T., KRALL A., STALHANE T., ZOJER H., PANIS C., « DODT : Increasing requirements formalism using domain ontologies for improved embedded systems development », *Design and Diagnostics of Electronic Circuits Systems (DDECS), 2011 IEEE 14th Inter. Symposium on*, avril 2011, p. 271 -274.
- [FAV 06] FAVRE J.-M., ESTABLIER J., BLAY-FORNARINO M., Eds., *L'ingénierie dirigée par les modèles : au-delà du MDA*, Hermes-Lavoisier, Cachan, France, 2006.
- [FIG 08] FIGER J.-P., « Principes d'urbanisation pour un système d'information, Référence H6000 », *Traité Technologies Logicielles et architecture des systèmes*, , page 24, Techniques de l'Ingénieurs, août 2008.
- [FOW 03] FOWLER M., *Patterns of Enterprise Application Architecture*, Addison-Wesley Professional, 2003.
- [FRE 10] FREY A. G., CALVARY G., DUPUY-CHESSA S., « Xplain : an editor for building self-explanatory user interfaces by model-driven engineering », *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing System, EICS 2010*, ACM, 2010, p. 41-46.
- [GAB 11] GABILLON Y., CALVARY G., FIORINO H., « Composition d'Interfaces Homme-Machine en contexte : approche par planification automatique », *Technique et science informatiques*, vol. 30, 2011, page 25.
- [GRO 11a] GROUP) O. M., « Knowledge Discovery Metamodel - Version 1.3 », <http://www.omg.org/spec/KDM/1.3>, 2011.
- [GRO 11b] GROUP O. M., « Syntax Tree Metamodel - Version 1.0 », <http://www.omg.org/spec/ASTM/1.0>, 2011.
- [HAC 09] HACID M.-S., LÉCUÉ F., LÉGER A., REY C., TOUMANI F., « Les web services sémantiques, automate et intégration. I. Introduction, éléments et scénarios, découverte de services web », *TSI*, vol. 28, n° 2, 2009, p. 229–262.
- [HIL 08] HILLAIRET G., BERTRAND F., LAFAYE J.-Y., « MDE for Publishing Data on the Semantic Web », *TWOMDE International Workshop on Transformation and Weaving Ontologies in MDE at MODELS'08*, Toulouse, 2008, p. 32–46.
- [HOL 09] HOLLENBACH J., PRESBREY J., BERNERS-LEE T., « Using RDF Metadata to Enable Access Control on Social Semantic Web », *International Semantic Web Conference, ISWC 2009*, LNCS, Springer, 2009.
- [ISO 04] ISO, « NF EN ISO 14001. Systèmes de management environnemental. Exigences et lignes directrices pour son utilisation », December 2004.
- [ISO 08] ISO, « NF EN ISO 9001. Systèmes de management de la qualité », Nov. 2008.
- [KAN 98] KANG K., KIM S., LEE J., KIM K., SHIN E., HUH M., « FORM : A feature-oriented reuse method with domain-specific reference architectures », *Annals of Software Engineering*, vol. 5, n° 1, 1998, p. 143–168.
- [KIE 09] KIENZLE J., AL ABED W., JACQUES K., « Aspect-oriented multi-view modeling », *Proceedings of the 8th ACM international conference on Aspect-oriented software development AOSD 09*, ACM Press, 2009, page 87.
- [KLE 07] KLEIN J., FLEUREY F., JÈZÈQUEL J. M., « Weaving Multiple Aspects in Sequence Diagrams », *Transactions on Aspect-Oriented Software Development*

- (*TAOSD*), vol. LNCS 4620, 2007, p. 167–199.
- [KON 08] KONSTANTINOUCHE N., SPANOS D.-E., MITROU N., « Ontology and Database Mapping : A Survey of Current Implementations and Future Directions », *Journal of Web Engineering*, vol. 7, n° 1, 2008, p. 1–24, Rinton Press.
- [KRU 08] KRUK S. R., CYGAN M., GZELLA A., « JeromeDL - Semantic and Social Technologies for Improving User Experience in Digital Libraries », *World Wide Web Conference, WWW 2008*, ACM, 2008.
- [LAG 06] LAGOZE C., PAYETTE S., SHIN E., WILPER C., « Fedora : an Architecture for Complex Objects and their Relationships », *Int. J. on Digital Libraries*, vol. 6, n° 2, 2006, p. 124–138.
- [LAN 03] LANZA M., DUCASSE S., « Polymetric Views—A Lightweight Visual Approach to Reverse Engineering », *Transactions on Software Engineering (TSE)*, vol. 29, n° 9, 2003, p. 782–795, IEEE Computer Society.
- [LED 12] LEDINOT E., GASSINO J., BLANQUART J.-P., BOULANGER J.-L., QUÉRÉ P., RICQUE B., « A cross-domain comparison of software development assurance », *Proceedings of the 6th European Congress for Embedded Real Time Software and Systems Conference, ERTSC*, Toulouse, France, February 2012.
- [LEH 85] LEHMAN M., BELADY L., *Program Evolution : Processes of Software Change*, London Academic Press, London, 1985.
- [MAC 91] MACLEAN A., YOUNG R. M., BELLOTTI V. M. E., MORAN T. P., « Questions, options, and criteria : Elements of design space analysis », *Human-Computer Interaction*, vol. 6, n° 3, 1991, p. 201–250, Taylor & Francis.
- [MEL 03] MELNIK S., RAHM E., BERNSTEIN P. A., « Rondo : A Programming Platform for Generic Model Management », HALEVY A. Y., IVES Z. G., DOAN A., Eds., *SIGMOD Conference*, ACM, 2003, p. 193–204.
- [MET 07] METZGER A., POHL K., HEYMANS P., SCHOBGENS P.-Y., SAVAL G., « Disambiguating the Documentation of Variability in Software Product Lines : A Separation of Concerns, Formalization and Automated Analysis », *RE'07*, 2007, p. 243–253.
- [MIC 10] MICHEL F., GAIGNARD A., AHMAD F., BARILLOT C., BATRANCOURT B., DOJAT M., GIBAUD B., GIRARD P., GODARD D., KASSEL G., LINGRAND D., MALANDAIN G., MONTAGNAT J., PÉLÉGRINI-ISSAC M., PENNEC X., ROJAS BALDERRAMA J., WALI B., « Grid-wide neuroimaging data federation in the context of the NeuroLOG project », *HealthGrid'10*, Paris, France, juin 2010, IOS Press, p. 112–123.
- [MOS 10] MOSSER S., BLAY-FORNARINO M., FRANCE R., « Workflow Design using Fragment Composition (Crisis Management System Design through ADORE) », *Transactions on Aspect-Oriented Software Development (TAOSD) Special issue on Aspect Oriented Modeling*, 2010, p. 1–34.
- [MOS 11] MOSSER S., MUSSBACHER G., BLAY-FORNARINO M., AMYOT D., « From Aspect-oriented Requirements Models to Aspect-oriented Business Process Design Models », *10th international conference on Aspect Oriented Software Development(AOSD'11)*, Porto de Galinhas, mars 2011, ACM.
- [NEM 07] NEMO C., BLAY-FORNARINO M., KNEISEL G., RIVEILL M., « SEMANTIC ORCHESTRATIONS MERGING - Towards Composition of Overlapping Orchestrations », FILIPE J., Ed., *9th International Conference on Enterprise Information Systems (ICEIS'2007)*, Funchal, Madeira, 2007.

- [OAS 06] OASIS, « Reference Model for Service Oriented Architecture 1.0 », rapport n° wd-soa-rm-cd1, 2006, OASIS.
- [OMG 10] OMG, « OMG Systems Modeling Language (OMG SysML 2.1) », June 2010.
- [OMG 11] OMG, « Business process management notation (BPMN 2.0) », 2011.
- [ORE 08] OREN E., HEITMANN B., DECKER S., « ActiveRDF : Embedding Semantic Web data into object-oriented languages », *J. Web Sem.*, vol. 6, n° 3, 2008, p. 191–202.
- [PAN 95] PANGOLI S., PATERNÓ F., « Automatic generation of task-oriented help », *Proceedings of the 8th annual ACM symposium on User interface and software technology UIST 95*, , 1995, p. 181–187, ACM Press.
- [POH 05] POHL K., BÖCKLE G., VAN DER LINDEN F. J., *Software Product Line Engineering : Foundations, Principles and Techniques*, Springer-Verlag, 2005.
- [QUA 09] QUASTHOFF M., MEINEL C., « Design Pattern for Object Triple Mapping », *IEEE SCC*, IEEE Computer Society, 2009, p. 443–450.
- [REI 07] REISER M.-O., WEBER M., « Multi-level feature trees : A pragmatic approach to managing highly complex product families », *Requir. Eng.*, vol. 12, n° 2, 2007, p. 57–75, Springer-Verlag New York, Inc.
- [SCH 07] SCHOBEBENS P.-Y., HEYMANS P., TRIGAUX J.-C., BONTEMPS Y., « Generic semantics of feature diagrams », *Comput. Netw.*, vol. 51, n° 2, 2007, p. 456–479, Elsevier North-Holland, Inc.
- [SHE 11] SHE S., LOTUFO R., BERGER T., WASOWSKI A., CZARNECKI K., « Reverse Engineering Feature Models », *ICSE'11*, 2011.
- [SOT 09] SOTTET J. S., CALVARY G., FAVRE J. M., COUTAZ J., « Megamodeling and metamodel-driven engineering for plastic user interfaces : mega-ui », *HumanCentered Software Engineering*, vol. 2, 2009, p. 173–200, Springer.
- [THÜ 09] THÜM T., BATORY D., KÄSTNER C., « Reasoning about Edits to Feature Models », *ICSE'09*, IEEE, 2009, p. 254–264.
- [VER 07] VERJUS H., POURRAZ F., « A Formal Framework For Building, Checking And Evolving Service Oriented Architectures », *Fifth European Conference on Web Services ECOWS07*, , 2007, p. 245–254.