

Alignement de modèles pour l'évolution de patrimoines applicatifs

Jonathan Pepin^{1,2}, Pascal André¹, Christian Attiogbé¹, and Erwan Breton²

¹ AeLoS Team

LINA CNRS UMR 6241 - University of Nantes

{firstname.lastname}@univ-nantes.fr

² Mia-Software - Nantes

ebreton@sodifrance.fr

Résumé La maintenance de systèmes d'informations implique de mettre en correspondance la vision stratégique du métier et la vision informatique, parfois via le prisme de l'urbanisation. La distance sémantique entre les points de vue rend difficile la mesure de l'impact de l'évolution du parc applicatif vis-à-vis des processus métiers. Nous proposons une approche pragmatique pour rapprocher les points de vue et aider à évaluer l'impact de restructurations sur l'évolution du parc applicatif. Une fois alignés les modèles des deux points de vue, des mesures estiment la qualité de l'alignement. L'approche présentée est mise en œuvre par des transformations de modèles et expérimentée sur des cas concrets.

Mots-clés : Systèmes d'information - Architecture d'entreprise - Rétro-ingénierie
- Alignement - Ingénierie des modèles - Mesure

1 Introduction

Le parc applicatif associé à un système d'information évolue tant du point de vue technique (distribution, interfaces, internet) que du point de vue métier (réorganisation, compétitivité, législation). Pour faire face à la concurrence, les entreprises raccourcissent le cycle de décision et exigent une réactivité forte du SI alors que le cycle de maintenance et de renouvellement du parc applicatif, souvent hétérogène, est plus long du fait de contraintes budgétaires ou organisationnelles.

Les travaux présentés ici contribuent à l'alignement entre les visions métier et technique (*business/IT*) pour maintenir le patrimoine applicatif en phase avec l'évolution des métiers. L'alignement *business/IT* n'est qu'une étape de la démarche d'architecture d'entreprise et d'urbanisation [1,2] mais elle est cruciale du point de vue de l'évolution du SI. Nous en avons tracé les contours dans [3] et exposé la méthode dans [4]. Nous en présentons ici l'application.

Cette communication est organisée comme suit. Dans la section 2 nous exposons notre approche opérationnelle de l'alignement des visions métier et technique. Cette approche est orientée modèles et le cœur du problème se réduit à relier la vue métier, la vue applicative et la vue fonctionnelle. Nous en proposons une solution non-intrusive par tissage. Notre approche est outillée et a été expérimentée sur plusieurs cas concrets relatés dans la section 3. L'évaluation de l'alignement est discutée en section 4. La section 5 résume l'état des travaux et trace des perspectives.

2 Un processus d'alignement de modèles

Dans une vision idéale, l'alignement est une ligne de traçabilité traversant les couches d'abstraction entre la vue stratégique du système d'information et son implantation. En pratique le but est surtout de rapprocher les différents modèles et de mettre en évidence des forces et faiblesses pour des scénarios d'évolution. Nous avons réduit le périmètre au problème de l'alignement entre la vision métier et la vision informatique parfois déformée par le spectre de la vision architecturale des urbanistes. Pour cela, nous faisons abstraction de la vision stratégique et la vision métier est instrumentée par des modèles de processus métiers et fonctionnels. A l'autre bout de la chaîne, le code déployé contient trop de détails. Le rapprochement des points de vue se fait en concrétisant la stratégie par des modèles de processus métier et en masquant les détails d'implantation (processus d'abstraction ou de rétro-ingénierie) jusqu'à un niveau acceptable pour le "langage commun". A chaque étape nous définissons des méta-modèles génériques dans un souci d'adaptation aux démarches d'architecture d'entreprise existantes. Le langage commun est alors défini comme un tissage entre trois méta-modèles afin d'identifier les concepts essentiels : processus (BPM), fonctionnel (Fun) et application (App).

Extraction du modèle applicatif par rétro-ingénierie du code source L'extraction d'un modèle applicatif à partir du code source est réalisé en trois étapes S1.1, S1.2 et S1.3.

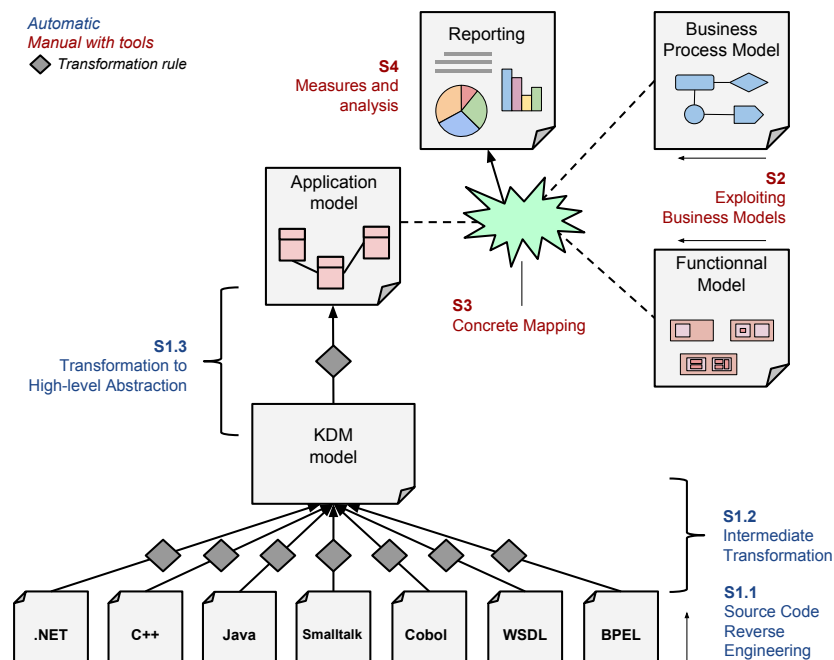


FIGURE 1. Les étapes de transformation

L'étape *S1.1* dépend du support technique de déploiement. Or les systèmes d'information sont composés de technologies hétérogènes évoluant rapidement dans le temps. Il est nécessaire de disposer d'un analyseur syntaxique différent pour chaque technologie. Les langages les plus communément rencontrés dans un patrimoine logiciel d'entreprise sont Java, C++, .Net, Smalltalk, Cobol... Nos expérimentations (section 3) sont faites avec un analyseur Java.

La seconde étape (*S1.2*) est une transformation vers un modèle intermédiaire pour s'abstraire de l'hétérogénéité. Il est défini par un méta-modèle unique capable de synthétiser les concepts des différentes technologies issues de la première étape *S1.1*. Le méta-modèle *Knowledge Discovery Metamodel (KDM)*, standard de l'OMG, a été choisi. KDM inclut de nombreuses couches pour stocker les différents aspects des langages de programmation communs [5]. Nous n'utilisons qu'une partie du méta-modèle KDM, notamment le paquetage *Code*.

L'étape *S1.3* abstrait plus fortement pour se séparer de la logique des programmes vers une logique applicative. Nous avons défini un méta-modèle (*App*), partiellement représenté en bas à gauche de la figure 2. Il définit la notion d'application constituée de composants, services, interfaces, fonctions et objets de données. La transformation détecte les aspects pertinents du modèle KDM. Bien que KDM soit un modèle pivot, des spécificités peuvent subsister. La rétro-ingénierie d'un code source complet ne se préoccupe pas de l'utilité des détails contenus dans le code source, s'il s'agit d'éléments métiers, utilitaires ou simplement techniques. Ainsi, cette étape de transformation va tamiser les différents éléments de l'application et ne capter que les concepts utiles pour peupler le modèle applicatif. Ce filtrage est un atout pour gérer de gros parcs applicatifs.

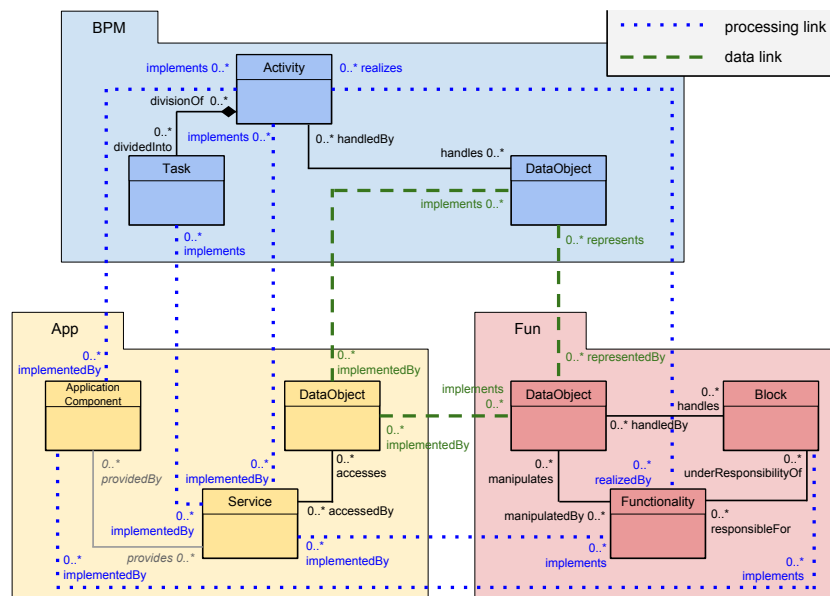


FIGURE 2. Trois méta-modèles d'architecture d'entreprise au cœur de l'alignement

Tissage entre les modèles métiers et applicatif Ce trio de méta-modèles (Fun, BPM et App) de la figure 2 est le support "langage" pour l'alignement du SI. Pour réaliser cet alignement, c'est-à-dire lier les fonctionnalités et activités de l'organisation à leur réalisation dans les applications exécutant ces opérations, des liens sont définis entre les concepts des différents méta-modèles. Ces liens inter-modèles sont créés selon deux axes de description : les traitements ou les données.

Les *Fonctionnalités* de Fun réalisent des *Activités* de BPM et sont implémentées par les *Services* dans App. En fonction du niveau de granularité de la modélisation BPM : si les *Tâches* sont détaillées elles sont implémentées par les *Services* ; si uniquement les *Activités* sont représentées, ces dernières sont implémentées par les *Services*. De même, de façon macroscopique, les *Activités* peuvent être implémentées directement par une *Composant applicatif*. Chaque méta-modèle possède une classe *Donnée*, un lien existe entre les classes *Donnée* équivalentes.

Dans la figure 2 seuls sont présents les concepts de chaque méta-modèle visibles et alignables entre eux. D'autres concepts figurent dans les méta-modèles complets. La mise en correspondance dépend du niveau de granularité choisi, de la portée de l'alignement (SI total ou partiel, une ou plusieurs applications), du type des éléments (structure ou comportement) et des objectifs de l'alignement.

3 Mise en œuvre du processus

Outillage Notre démarche repose sur l'ingénierie des modèles, nous avons fait le choix du cadriciel open source Eclipse EMF qui permet d'une part de modéliser ses méta-modèles dédiés, puis d'autre part de les instancier, manipuler et créer d'autres outils les référençant. Pour les étapes de rétro-ingénierie nous avons utilisé l'outil open source Eclipse *Modisco* qui inclut un mécanisme de découverte pour réaliser l'analyse d'un code source et la transformation de modèle EMF. *Modisco* inclut également des méta-modèles technologiques pour Java et KDM. Lors des différentes étapes de transformation nous avons créé nos propres règles écrites avec le logiciel *Mia-Transformation*³. L'outil est exécutable à partir de la plateforme Eclipse, lors de la transformation il permet de lire en entrée et créer en sortie des méta-modèle EMF. Le tissage (étape S3 de la figure 1) à été expérimenté avec plusieurs méthodes de composition existantes (*AMW*, *Virtual EMF*) que nous avons adapté : extension de méta-modèle, annotation, méta-modèle de tissage spécifique. Néanmoins aucune de ces méthodes n'a été satisfaisante : problèmes d'intrusion ou de résolution des liens de tissage ou encore problème de sémantique de liens génériques. La solution à été trouvée à l'aide du projet open source *EMF Facet* qui permet d'étendre un méta-modèle sans intrusion en créant de nouveaux attributs et relations typés. Certains mécanismes n'étant pas nativement couverts par l'API d'*EMF Facet* pour réaliser un tissage de modèles, nous avons contribué au projet. Grâce à *EMF Facet* nous avons donc pu implémenter les relations entre méta-modèles (cf figure 2 en créant un jeu de facettes. Pour aller plus loin nous avons étendu le requêteur OCL d'EMF pour qu'il soit compatible avec *EMF Facet*. Ainsi, il est désormais possible de naviguer d'un méta-modèle à un autre via les facettes. Cette navigation complète permet de réaliser l'évaluation de l'alignement (cf. section 4).

3. <http://www.mia-software.com/>

Cas d'études Nous avons eu l'opportunité de tester notre démarche sur trois cas d'études concrets, provenant de sociétés d'Assurance Mutuelle françaises que nous nommerons SAMM, SAMI et SAMUT. Chaque étude possède ses spécificités et couvre tout ou partie des modèles du processus de la section 2. L'expérimentation a pour but de vérifier la faisabilité de notre démarche : pertinence des modèles, automatisation des transformations et de leur enchaînement et la pertinence du tissage et de sa mise en œuvre.

Cas SAMM Le cas SAMM est composé *i*) d'un code source complet écrit en Java avec 33 400 classes, environ 3 400 000 lignes de codes ; et *ii*) d'un référentiel d'entreprise sous la forme d'un portail HTML exporté depuis le logiciel MEGA Enterprise Architecture⁴. Le référentiel d'entreprise contient 360 diagrammes de processus métier couvrant la totalité du SI. Le volume de l'application est conséquent et représente un véritable défi pour traiter ce volume d'informations. Ce cas d'étude nous a permis de tester notre approche sur un source code Java de taille importante, le chargement des modèles obtenus par rétro-ingénierie a été une épreuve pour nos outils. Le travail de transformation a été facilité par la présence d'une nomenclature des concepts qui se retrouve à différents niveaux et par là même montre de bonnes pratiques de codage, même si des exceptions subsistent. Néanmoins, nous regrettons ne pas avoir accès au source original du référentiel d'entreprise MEGA pour réaliser un tissage complet.

Cas SAMI Le cas SAMI est composé uniquement d'un référentiel MEGA dont le source est disponible. Nous avons extrait les différents concepts pour peupler à l'aide d'une transformation nos différents modèles (App, BPM, Fun) afin de vérifier la couverture et la compatibilité des concepts. Le cas présente 625 composants, 11894 services, 18 blocs fonctionnels, 131 fonctionnalités, 167 processus et 268 activités.

Cas SAMUT Le cas SAMUT présente un scénario particulier ; ce dernier n'avait aucune représentation métier existante. Les objets de données ont été extraits par rétro-ingénierie d'une ancienne base de données hiérarchique et les composants applicatifs de l'architecture existante sous forme de procédures stockées. Un architecte a ensuite modélisé et identifié des blocs fonctionnels et nous avons alors pu réaliser le tissage entre les blocs et les composants applicatifs. Le SI comporte 12 blocs, 1045 composants et 669 objets de données. Ce cas d'étude a permis de tester notre méthode de tissage, et isoler des composants qui étaient orphelins (rangés dans aucun bloc).

Ces trois cas d'études ont tous des caractéristiques particulières, les supports sources sont totalement hétérogènes et représentatifs de la disparité de maturité des SI. Ces différents aspects ont permis de prouver la souplesse de notre démarche qui à chaque étape peut s'adapter. Néanmoins, chaque cas d'étude avec un source d'une typologie nouvelle implique d'écrire ou adapter une transformation.

4. <http://www.mega.com/fr/solution/business-architecture>

4 Évaluation de l'alignement

L'évaluation de l'alignement souhaitée consiste à réaliser une analyse structurelle de l'existant en détectant les problèmes entre couches différentes à l'aide de la sémantique des liens de tissage. Cette évaluation se différencie d'une analyse d'écart entre une seule et même cartographie figée à deux instants différents du SI : *As-Is* et *To-Be*. Cette analyse d'écart se contente de réaliser une comparaison en isolant les différences et les contenus communs pour détecter ce qui a été modifié, ajouté ou supprimé. L'objectif est de fournir des outils d'assistance à l'évolution pour les urbanistes.

Au départ nous étions partis sur une mesure de la qualité de l'alignement avec un indicateur global permettant d'étalonner la cohérence globale du système d'information. L'analyse détaillée nous a convaincu que ce n'était ni la voie ni le besoin. Ce n'est pas la voie car trouver un référentiel de bonne qualité de l'alignement se révèle ardu et finalement trop complexe. Ce n'est pas le besoin car l'objectif des architectes n'est pas d'aboutir après plusieurs itérations à un "super" alignement final mais plutôt d'établir un constat à un moment donné avec des pistes d'améliorations puisque le système d'information évolue en permanence. Nous avons proposé une série de métriques de croisement de modèles, exprimées de manière concrète en OCL. Par exemple, la règle suivante permet d'obtenir la liste des *Activités* et *Tâches* de BPM non implantées dans App :

Listing 1.1. BPM not implemented by App

```
context BusinessProcessLayer :
bpm::Activity . allInstances () -> select ( services -> isEmpty () or
  applicationComponents -> isEmpty () ) -> collect ( oclAsType ( ecore ::
  EObject ) ) -> union ( bpm::Task . allInstances () -> select ( services ->
  isEmpty () ) -> collect ( oclAsType ( ecore :: EObject ) ) ) -> union ( bpm::
  DataObject . allInstances () -> select ( dataObjectsApp -> isEmpty () )
  -> collect ( oclAsType ( ecore :: EObject ) ) )
```

Nous avons trouvé assez peu de travaux sur la mesure de l'alignement. Les travaux de Simonin [6] se focalisent sur la couche fonctionnelle (urbanisation). Les autres traitent surtout de la couche métier (voire stratégique) avec le logiciel. Ils incluent la notion d'acteurs, qui n'a pas forcément de sens du point de vue logiciel. Aversano [7] traite de la couverture des processus métiers par le logiciel, celui-ci ne présente pas les modèles du logiciel. Rolland et Etien [8] alignent les processus métiers avec des modèles UML via des ontologies mais sans parc applicatif. Thévenet [9] s'intéresse à l'alignement stratégique et l'évolution, sans lien direct avec le code.

L'objectif est alors de mesurer l'alignement pour fournir aux décideurs des indicateurs guidant à la fois l'analyse de l'état actuel, la détection d'anomalies et la valorisation de scénario de projections futures. Nous travaillons actuellement sur une classification des indicateurs selon quatre axes : la couverture, la cohérence, la densité de l'alignement concret (entre les modèles métiers, applicatifs et fonctionnels) et la couverture du code final.

Nous travaillons aussi sur la modularité des modèles et de leur alignement via des algorithmes sur les matrices de dépendance (DSM - *Dependency Structure Matrix*) qui

présentent l'avantage de fournir une visualisation des dépendances. D'autres analyses seront également réalisées pour déterminer où sont disséminées les données dans le SI des objets métier ou pour identifier si le flot d'exécution est conforme à l'enchaînement des tâches d'un processus. L'affichage se ferait par agrandissement (*drill-down*) afin de se focaliser sur une partie précise en obtenant les détails, par exemple en cliquant sur une tâche, obtenir tous les services en relation.

5 Conclusion et perspectives

Nous avons proposé une approche pragmatique au problème d'alignement des points de vue métier et applicatif s'insérant dans la démarche d'urbanisation des architectures d'entreprise. Notre approche est instrumentée dans le cadre d'Eclipse EMF et a été expérimenté sur des cas réels d'entreprise, permettant d'éprouver la viabilité de notre approche. Certains points sont encore à améliorer dans l'automatisation du processus comme la découverte des composants logiciels ou la détection des correspondance du tissage. Le travail en cours sur les métriques doit permette la mise en place de tableaux de bord pour l'évaluation du tissage et les scénarios d'évolution.

Références

1. Ullah, A., Lai, R. : A systematic review of business and information technology alignment. *ACM Trans. Manage. Inf. Syst.* **4**(1) (April 2013) 4 :1–4 :30
2. Aversano, L., Grasso, C., Tortorella, M. : A literature review of Business/IT alignment strategies. In Cordeiro, J., Maciaszek, L.A., Filipe, J., eds. : *Enterprise Information Systems*. Number 141 in LNBIP. Springer (January 2013) 471–488
3. Pepin, J., André, P., Attiogbé, C., Breton, E. : Alignement de modèles métiers et applicatifs : une approche pragmatique par transformations de modèle. In : *Actes de CIEL 2014, Troisième Conférence en Ingénierie du Logiciel*, - (June 2014) 17–31
4. Pepin, J., André, P., Attiogbé, C., Breton, E. : A method for business-it alignment of legacy systems. In Hammoudi, S., Maciaszek, L.A., Teniente, E., eds. : *ICEIS 2015 - Proceedings of the 17th International Conference on Enterprise Information Systems*, Volume 3, Barcelona, Spain, 27-30 April, 2015, SciTePress (2015) 229–237
5. Normantas, K., Sosunovas, S., Vasilecas, O. : An overview of the knowledge discovery meta-model. In : *Proc. of the 13th International Conference on Computer Systems and Technologies*. *CompSysTech '12*, NY, USA, ACM (2012) 52–57
6. Simonin, J. : Conception de l'architecture d'un système dirigée par un modèle d'urbanisme fonctionnel. PhD thesis, Université de Rennes 1 (January 2009)
7. Aversano, L., Grasso, C., Tortorella, M. : Measuring the alignment between business processes and software systems : A case study. In : *Proceedings of the 2010 ACM Symposium on Applied Computing*. *SAC '10*, New York, NY, USA, ACM (2010) 2330–2336
8. Etien, A., Rolland, C. : Measuring the fitness relationship. *Requirements Engineering* **10**(3) (2005) 184–197
9. Thevenet, L.H. : Proposition d'une modélisation conceptuelle d'alignement stratégique : La méthode INSTAL. Theses, Université Panthéon-Sorbonne - Paris I (December 2009)