



Approche pour un meilleur alignement des processus métiers. De la modélisation à l'implémentation

Jean-Stéphane Ulmer, Jean-Pierre Belaud, Jean-Marc Le Lann

► To cite this version:

Jean-Stéphane Ulmer, Jean-Pierre Belaud, Jean-Marc Le Lann. Approche pour un meilleur alignement des processus métiers. De la modélisation à l'implémentation. Revue des Sciences et Technologies de l'Information - Série ISI: Ingénierie des Systèmes d'Information, 2012, 17 (4), pp.17-47. 10.3166/ISI.17.4.17-47 . hal-03468430

HAL Id: hal-03468430

<https://hal.science/hal-03468430>

Submitted on 7 Dec 2021

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.

Approche pour un meilleur alignement des processus métiers

De la modélisation à l'implémentation

Jean-Stéphane Ulmer, Jean-Pierre Belaud, Jean-Marc Le Lann

1. Université de Toulouse - Institut National Polytechnique ENSIACET

CNRS UMR 5503 / Laboratoire de Génie Chimique

4, allée Emile Monso – BP 44362

F-31432 Toulouse cedex 4

{jeanstephane.ulmer, jeanpierre.belaud, jeanmarc.lelann}@ensiacet.fr

RÉSUMÉ. Une entreprise doit être capable de décrire et de demeurer réactive face à un événement endogène ou exogène. Une telle flexibilité peut s'obtenir par la gestion des processus d'entreprise. Lors d'une telle démarche, différentes transformations interviennent sur les modèles de processus développés par l'analyste métier et l'expert en technologies de l'information (TI). Un non-alignement se crée entre ces modèles hétérogènes lors de leurs manipulations : il s'agit du « fossé métier-TI » tel que décrit dans la littérature. L'objectif de notre travail est de proposer un cadre méthodologique permettant un meilleur pilotage des processus métiers, afin de tendre vers un alignement systématique de leur modélisation à leur implémentation au sein du système cible. En permettant la restitution intégrale d'un modèle transformé au sens de l'ingénierie inverse, notre plateforme permet une synchronisation entre modèle d'analyse et modèle d'implémentation.

ABSTRACT. A company must be able to describe and remain reactive against endogenous or exogenous event. Such flexibility can be achieved through the Business Process Management. In this approach, different transformations intervene on process models developed by the business analyst or the IT expert. A non-alignment is created between these heterogeneous domains during model manipulations: this is the "business-IT gap" as described in the literature. The aim of our works is to propose a methodological framework enhancing business process control, in order to obtain a systematic alignment from modelling to its implementation within the target system. By allowing a full restitution of a transformed model, our platform enables synchronization between analysis and implementation models.

MOTS-CLÉS : ingénierie de systèmes d'information, ingénierie dirigée par les modèles, alignement de modèles, ingénierie des processus métiers, métamodèles, transformation.

KEYWORDS: information system engineering, model-driven engineering, model alignment, business process management, metamodels, transformation.

1. Introduction

Pour que les organisations demeurent compétitives dans un environnement socio-économique dynamique, elles doivent être en mesure de saisir de nouvelles opportunités, de réagir efficacement face à de nouvelles demandes. Dans cette optique, dès les années 1960, les systèmes d'information et les technologies d'information associées (SITI) furent mis en œuvre afin de supporter la création de valeur au sein d'une entreprise. Rapidement, ces SITI ont pris une place de plus en plus importante au sein des organisations. En effet, posséder un système d'information efficace et efficient supportant les stratégies métiers et les processus qui y sont rattachés est devenu un facteur-clé de succès (Luftman, Maclean, 2004). Les SITI contribuent à la création de valeur, et ce, d'une manière reconnue : l'alignement entre la stratégie d'entreprise et les choix de déploiement des SITI est devenu un élément important de préoccupation pour les organisations (Silvius, 2007).

Selon (Henderson, Venkatraman, 1993 ; Avila Cifuentes, 2009) cet alignement peut être un ajustement stratégique dont la finalité est d'accroître la performance organisationnelle de l'entreprise. L'alignement peut également être d'ordre opérationnel (figure 1). Egalement désigné comme l'intégration fonctionnelle de l'entreprise, il en améliore la flexibilité métier. Cette amélioration se traduit par une réponse plus rapide aux nouveaux besoins client.

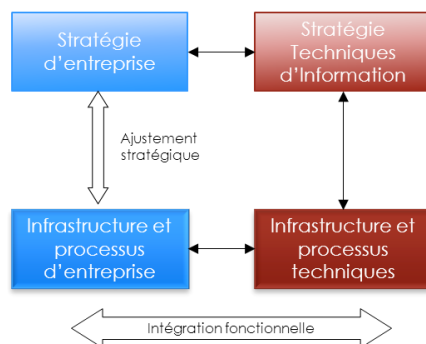


Figure 1. Modèle d'alignement stratégique selon (Henderson, Venkatraman, 1993)

Cette flexibilité se traduit par l'utilisation de processus d'entreprise supportés efficacement par les systèmes d'information (SI). Ils permettent notamment de mettre en œuvre la stratégie concurrentielle de l'entreprise (*Business strategy*). Cette stratégie résulte de l'ajustement entre la stratégie de l'entreprise et de ses processus métiers ainsi que de l'intégration fonctionnelle de ces derniers avec les processus SITI.

Si l'intérêt de réaliser un tel alignement est reconnu, sa mise en œuvre reste trop souvent limitée, en particulier pour l'intégration fonctionnelle. Il n'est pas rare que les acteurs de l'organisation ignorent les notions propres à cet alignement, de même qu'il existe une absence de communication entre domaine métier et domaine SITI.

(Etien, 2009). Une discontinuité franche entre le domaine métier et le domaine SITI se crée. Au lieu de permettre et d'accompagner le changement des processus, les SI peuvent brider ou ralentir la mise en place de telles évolutions.

2. Problématique et objectifs

Nos travaux sont motivés par la réalisation d'une intégration fonctionnelle réversible au sens de la rétro-ingénierie, que nous qualifierons d'alignement opérationnel, entre domaine métier et domaine SITI selon une approche orientée processus. Néanmoins plusieurs difficultés inhérentes à l'utilisation d'une telle approche subsistent. Tout d'abord, une définition appropriée de l'alignement opérationnel et de ses propriétés est requise. Ces différentes propriétés doivent être respectées par notre approche solution.

Nous examinons les différentes représentations que revêt un processus au cours de son cycle de vie. Considérons ainsi la manipulation de modèles de processus schématisée au sein de la figure 2.

Typiquement, après sa conception, un modèle « papier » abstrait et décrivant un processus et son objectif stratégique, est transformé en un modèle métier. Ce modèle, plus ou moins formel, est fréquemment utilisé dans un but de documentation ou de communication. Il subit à son tour une transformation afin d'être utilisé par un SI. Cependant des modifications et ajustements sont souvent nécessaires afin de rendre le modèle SITI obtenu pleinement exploitable. Ces modifications contribuent à l'obtention d'un modèle SITI raffiné.

Il faut garder à l'esprit que ces diverses transformations s'effectuent essentiellement manuellement et de manière *ad hoc*, ponctuelle. Dès lors, pouvons-nous considérer ces différents modèles comme étant cohérents entre eux, représentent-ils le même processus ?

De plus ces modèles sont également amenés à évoluer. Le modèle métier peut décrire de nouveaux buts à atteindre, le modèle SITI devra s'adapter à de nouvelles technologies. Ces cycles de vie asynchrones entre modèles d'abstraction et de domaines différents entraînent une discontinuité entre le processus souhaité et le processus implémenté par le SI. Dès lors, des efforts de synchronisation et de cohérence entre modèles s'avèrent nécessaires.

Afin de répondre à ces problématiques, le reste de l'article est organisé de la manière suivante. Tout d'abord nous présentons les principaux concepts rattachés aux notions de processus et d'ingénierie des processus dans la section 3. Nous cherchons ainsi à mettre en évidence l'hétérogénéité des modèles manipulés au sein d'une approche orientée processus. La section 4 se concentre sur la notion d'alignement opérationnel qui devient dès lors nécessaire. Nous proposons des propriétés le définissant et que notre approche doit prendre en compte. A ces propriétés s'ajoutent celles présentées section 5 dont la motivation est d'améliorer l'agilité d'entreprise. La section 6 caractérise l'approche et la section 7 présente la plateforme associée que nous avons développée. Cette plateforme est ensuite éprouvée à travers un scénario-

test défini dans la section 8. Enfin la section 9 clôture cet article en énonçant les conclusions et perspectives relatives à nos travaux de recherche.

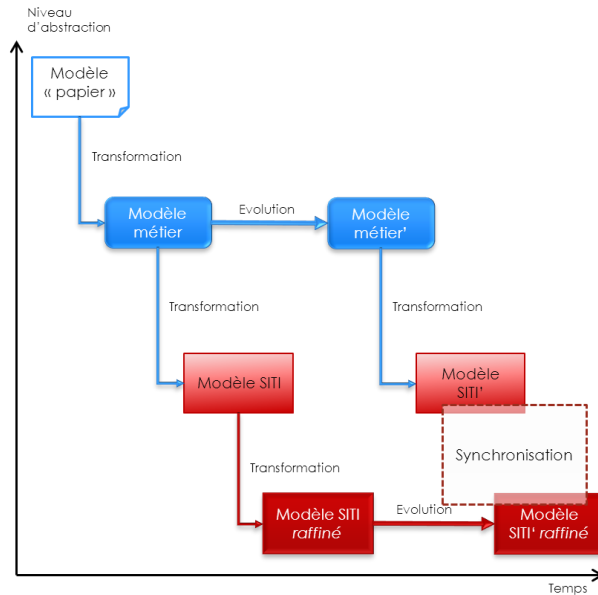


Figure 2. Processus de transformation depuis un modèle métier vers un modèle SITI

3. Cadre méthodologique

Cette section présente les différentes notions accompagnant le terme processus. Elle décrit également le cycle de vie d'un processus et les différentes étapes qui le jalonnent.

Un processus métier est une orchestration d'activités qui incluent une interaction entre différents acteurs sous la forme d'échange d'informations et réalisent des objectifs métiers. Le métier d'une entreprise représente un ensemble d'activités d'un champ d'expertise donné nécessitant des compétences et savoir-faire des acteurs de l'entreprise.

Les différentes étapes parcourant le cycle de vie d'un processus métier sont représentées sur la figure 3, inspirée de (Zur Muehlen, 2004). L'ingénierie des processus métiers ou *Business Process Management* (BPM) considère ce cycle de vie en regroupant les étapes selon trois grandes phases (Debauche, Mégard, 2004).

Un processus peut être considéré selon deux domaines bien distincts. Le domaine métier regroupe les étapes relatives à l'analyse, la conception (qu'il s'agisse de la modélisation ou de l'optimisation), l'évaluation et la simulation du processus. Le domaine SITI possède les étapes ayant un objectif d'automatisation du

processus : son implémentation, son exécution, et le prélèvement de données relatives à cette exécution.

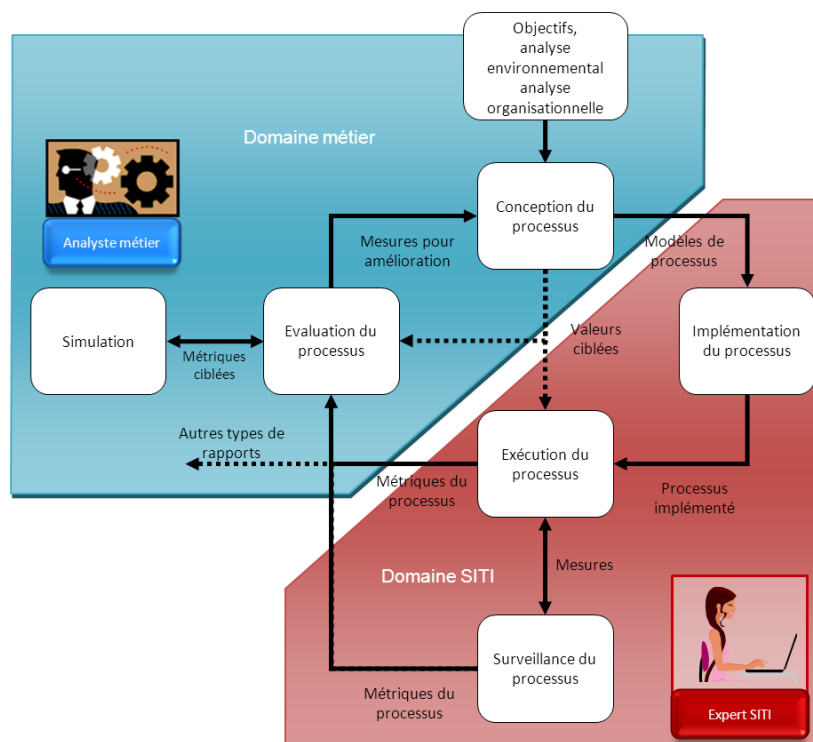


Figure 3. Cycle de vie du processus

Nous observons également que chaque domaine est régi par un acteur spécifique. L'analyste métier cherche de nouvelles façons d'améliorer l'efficacité « métier » de son entreprise, notamment en modifiant les processus (Various IIBA et Brennan, 2008). Une des caractéristiques importantes d'une démarche d'ingénierie des processus métier est la possibilité pour un analyste métier de définir les processus métiers sans aucune compétence technique. Ainsi un modèle source doit être réalisé indépendamment des considérations de l'environnement cible. Cette première phase regroupant les étapes d'analyse et de conception du processus est la phase d'analyse ou *Business Process Analysis* (BPA). Suite à la phase d'analyse, nous obtenons un modèle d'analyse, conceptuel, peu ou pas formel, reposant sur un langage graphique et créé dans un but de documentation.

A travers la phase d'implémentation, ou *Business Process Implementation* (BPI), l'expert SITI prend en charge ces besoins métiers et les transforme selon des considérations techniques afin de les rendre exploitables par un SI. Pour cela, il ajoute des informations techniques : le format des messages échangés, les protocoles

de transport utilisés, les transformations de données effectuées, etc. Une fois modifié, nous obtenons un modèle d'implémentation, technique et formel, adoptant un langage textuel. La finalité du modèle est d'être implémentée par un moteur d'exécution de processus.

Une phase de supervision des processus ou *Business Activity Monitoring* (BAM) conclut ce cycle. Elle fournit un ensemble d'outils assurant le pilotage de l'entreprise par un ensemble d'indicateurs d'analyse de performance, métriques calculées sur la base des mesures obtenues lors de la phase d'implémentation. Nous disposons ainsi des mesures nécessaires à l'évaluation et l'optimisation des processus. Dans le cadre de nos travaux, nous nous sommes plus particulièrement intéressés aux phases d'analyse et d'implémentation. La section suivante décrit plus en détail leur interaction.

4. Alignement opérationnel

L'alignement opérationnel intervient lors de l'interaction entre modèles hétérogènes et de domaines différents. Typiquement, le modèle conceptuel issu de la phase d'analyse est transformé en un modèle technique, le modèle d'implémentation. Or la représentation d'un processus selon des critères métiers peut être imprécise, ambiguë et sujette à interprétations d'un point de vue informatique. Dès lors les experts SITI peuvent considérer que le modèle d'implémentation obtenu présente des irrégularités et nécessite des ajustements afin d'être pleinement exploitable (figure 4).

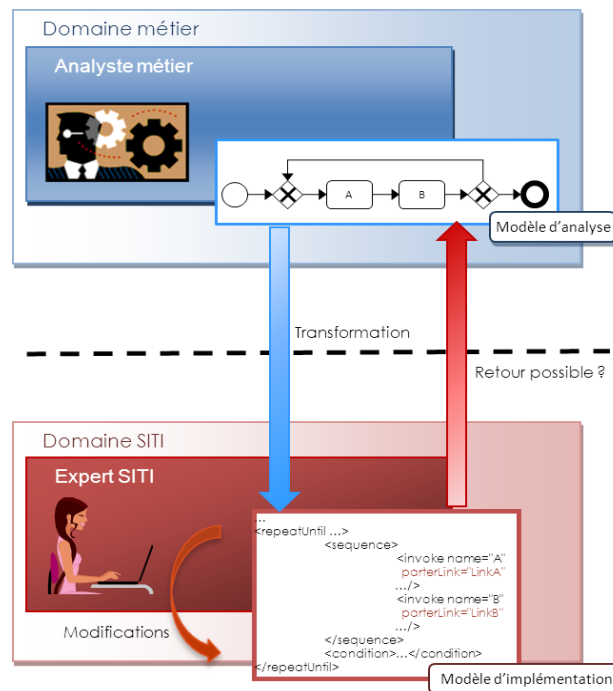


Figure 4. Du modèle d'analyse au modèle d'implémentation, et inversement

Un alignement opérationnel signifie qu'une transformation inverse soit réalisable, que le modèle obtenu soit utilisable par l'analyste métier. Une telle relation soulève plusieurs interrogations :

- les modèles sont-ils *synchronisés* ?
- s'agit-il de modèles sémantiquement équivalents ?
- au final, ces modèles sont-ils *cohérents* entre eux ?

Afin de demeurer le plus formel possible, nous proposons une définition de chacun de ces différents termes, ainsi que leurs répercussions sur l'approche solution proposée.

Nous considérons que deux modèles, i et j , sont dits synchronisés si et seulement si des changements significatifs effectués sur le modèle i peuvent être répercutés sur le modèle j (est considéré comme significatif tout changement modifiant la structure et/ou le comportement d'un modèle). La solution envisagée doit assurer la propagation des modifications significatives d'un modèle à un autre afin de maintenir ces modèles synchronisés. Dans ce contexte, nous considérons comme significatif tout changement modifiant la structure et/ou le comportement d'un modèle.

Nous définissons deux modèles i et j comme sémantiquement équivalents si pour chaque élément appartenant à E_i , un élément (ou un ensemble d'éléments) appartenant à E_j peut lui être associé. Chacun de ces éléments associés doit suivre la même orchestration. Notre approche doit être en mesure de déterminer les relations sémantiques existantes entre éléments de modèles différents.

Enfin, nous établissons qu'il existe une cohérence dite « intermodèle » entre deux modèles si nous observons une équivalence sémantique et s'ils sont synchronisés entre eux. Cette cohérence intermodèle est considérée comme une condition nécessaire et suffisante d'un alignement opérationnel. En respectant les différents points présentés ici, l'approche proposée permet d'obtenir un alignement opérationnel entre domaine métier et domaine SITI.

5. Principes de modélisation considérés

A l'aide des propriétés énoncées précédemment, l'approche proposée doit être en mesure de réduire l'écart métier-SITI. Pour cela notre approche s'appuie sur une plateforme permettant la transformation des modèles d'analyse vers les modèles d'implémentation et garantissant la cohérence intermodèle. Afin de faciliter et de formaliser ces transformations, nous définissons un couplage faible entre modèles ainsi que l'utilisation systématique de métamodèles.

La principale limite rencontrée au sein d'un cycle d'ingénierie des processus réside dans la « discontinuité » entre le domaine métier et le domaine SITI. Cette discontinuité intervient notamment lors du passage d'un modèle d'analyse vers un modèle d'implémentation. Majoritairement, la solution logicielle choisie pour supporter ces transformations est un BPMS (BPM Suite). Il convient de garder à l'esprit que les modèles sont amenés à être modifiés, les processus représentés et les

technologies de l'information utilisées pouvant évoluer. Des efforts de synchronisation, de mises à jour et de cohérence entre modèles s'avèrent nécessaires. L'entreprise et la nature de ses modèles étant en perpétuelle évolution, elles doivent pouvoir s'adapter à différents éditeurs ou à de nouvelles technologies. Cependant l'approche intégrée de modélisation basée sur un atelier logiciel unique comme le BPMS, se traduit par une dépendance forte à l'éditeur. L'entreprise ne possède plus la possibilité de modifier l'outil de modélisation voire la plateforme d'intégration, indépendamment l'un de l'autre. De ce fait l'entreprise n'est plus pleinement en mesure de modifier son outil de modélisation, son moteur d'exécution ou de s'adapter à de nouvelles technologies.

A l'inverse d'une approche fortement intégrée préconisant l'utilisation d'un éditeur logiciel unique, nous cherchons à obtenir un couplage faible entre les modèles d'analyse et les modèles d'implémentation. Ici la notion de couplage faible signifie que bien qu'il existe une interaction forte entre modèles, ceux-ci demeurent indépendants (l'analyste métier peut modifier le modèle d'analyse sans se soucier des répercussions sur le modèle technique, et inversement) et les environnements associés restent modifiables. Nous désirons également délimiter une stricte séparation entre ce que nous appellerons l'environnement de modélisation et l'environnement d'implémentation.

Qu'un modèle soit défini selon l'environnement d'analyse ou bien celui d'implémentation, il obéit à un langage d'expression et des règles structurales. Ce langage et ces règles peuvent être formalisés sous la forme d'un métamodèle. Ainsi, les environnements de modélisation et d'implémentation possèdent chacun un métamodèle. Sans métamodèles, les règles de transformation entre modèles d'analyse et d'implémentation sont peu flexibles car définies au cas par cas. Conformément aux principes de l'ingénierie dirigée par les modèles (IDM) (Combemale, 2008), nous utilisons des métamodèles afin de systématiser et de formaliser ces transformations. Les différentes relations entre ces métamodèles et modèles sont représentées figure 5. L'émergence d'un élément central, support de la transformation, est détaillée plus en détail dans la section suivante : l'environnement pivot.

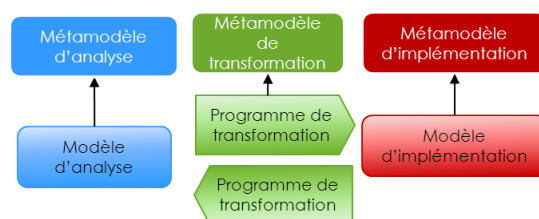


Figure 5. Métamodèles et modèles

6. Approche orientée pivot

Respectant « l'approche pivot » exploitée dans l'interopérabilité système (Meinadier, 2002), le rôle du pivot est d'assurer une équivalence sémantique entre un modèle conceptuel et le modèle d'implémentation lui correspondant.

A travers nos différents travaux (Ulmer *et al.*, 2009 ; 2010a), nous proposons une méthode centrée pivot afin d'améliorer l'alignement des différents modèles issus des phases d'analyse (BPA) et d'implémentation (BPI). L'alignement parfait tel que défini par (Etien, 2006) peut être obtenu par l'utilisation d'un pivot. Le concept de pivot a déjà été utilisé selon divers niveaux d'abstraction.

Pour l'implémentation des transformations de ses modèles, le projet FAROS¹ (Blay-Fornarino *et al.*, 2008) utilise cette notion de pivot. Il s'agit ici de transposer des éléments métiers, des contrats, au sein de plateformes techniques. L'utilisation d'un modèle et d'un métamodèle pivot permet de formaliser ces transformations depuis la maîtrise d'ouvrage vers la maîtrise d'œuvre et de les rendre reproductibles.

Nous pouvons également citer les travaux de (Thevenet, 2010) et la méthode INSTAL (« INstal Strategic ALignment »). Cette méthode s'intéresse aux intentions d'alignement partagées par les deux niveaux à aligner (stratégique et opérationnel) et qui représentent ici l'alignement stratégique.

A un niveau d'abstraction inférieur, nous pouvons considérer, par exemple, les systèmes de gestion de bases de données (SGBD) notamment pour les bases de données relationnelles et la méthode MERISE (Baptiste, 2009). L'utilisation de modèles différents dans les bases de données composantes pose des problèmes d'hétérogénéité syntaxique. Une solution possible est de traduire tous ces schémas en un modèle commun, le modèle pivot, nommé le modèle logique de données.

Le concept de pivot n'est pas non plus inconnu au sein du domaine métier du *Process Systems Engineering* (PSE). Le rapide développement d'outils d'ingénierie des procédés assistée par ordinateur a entraîné le développement du standard CAPE-OPEN (Belaud, 2002). CAPE-OPEN a pour rôle d'accroître l'interopérabilité entre outils logiciels PSE. Tel un pivot de bas niveau, CAPE-OPEN permet, *via* son middleware, à des logiciels hétérogènes d'éditeurs différents d'interopérer, indépendamment de leurs langages de programmation et de leur système opérant.

A travers l'approche proposée dans cet article, nous considérons que le système pivot doit être en mesure de faciliter la transformation entre des modèles de perspectives et d'abstractions différentes. Pour cela, il doit s'appuyer sur leurs métamodèles respectifs. Il doit pouvoir ajouter les informations nécessaires, préserver l'intégrité de ces informations et leur cohérence. L'utilisation d'un environnement pivot doit atténuer la dépendance existant entre l'environnement de modélisation et l'environnement d'implémentation. Cette indépendance assure un couplage faible entre modèles issus de ces environnements. Ce format intermédiaire

1. Projet ANR RNTL FAROS (composition de contrats pour la Fiabilité d'Architectures Orientées Services) : <http://www.lifl.fr/faros>

devient alors nécessaire pour stocker et échanger les informations des modèles entre les environnements de modélisation et d'implémentation.

Dans les sections suivantes, nous établissons la couverture de notre approche et les acteurs nécessaires à sa réalisation.

6.1. Vues

Nous cherchons à représenter la plupart des modèles de processus. Pour cela, nous devons considérer les différentes vues d'une entreprise. En effet un processus est représenté selon de nombreuses vues afin de rester compréhensible. Chacune de ces vues permet de décrire un aspect important du modèle de processus (Touzi, 2007 ; ISO, 2005). La vue fonctionnelle décrit les processus et leurs structures. La vue informationnelle décrit les données et les documents qu'un processus utilise ou produit. La vue organisationnelle définit les acteurs et les rôles qui sont responsables de l'exécution d'un processus donné. La vue des ressources définit les outils et les systèmes qui supportent l'exécution du processus. Enfin, la vue comportementale décrit l'ordre dans lequel les processus sont exécutés. La figure 6 met en évidence les liens existants avec ces vues.

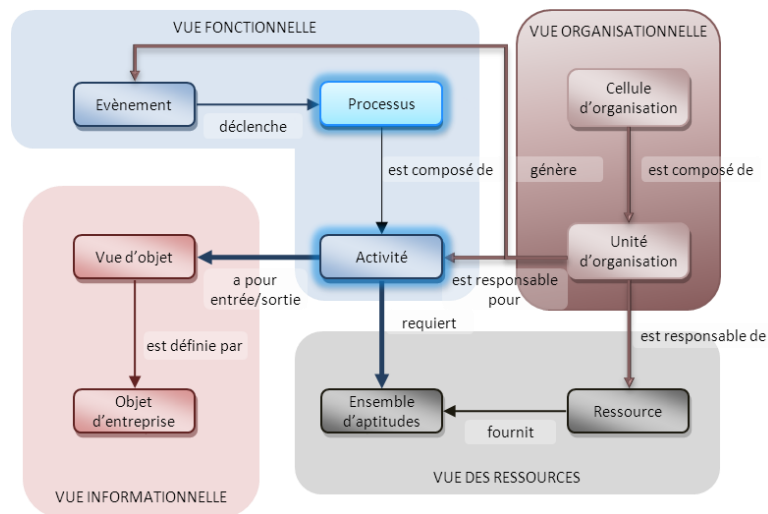


Figure 6. Vues de modélisation d'une entreprise

Nos modèles reposeront essentiellement sur une vue fonctionnelle. La vue fonctionnelle est d'ailleurs au cœur des autres vues de l'entreprise et est connectée à toutes ces autres vues par l'intermédiaire de l'élément Activité. Ceci permet à des concepts inhérents aux autres vues d'intervenir de manière ponctuelle, en fonction des standards et langages de modélisations utilisés au sein de notre approche.

Considérons par exemple BPMN², un langage standard de modélisation de processus d'entreprise, il est centré sur la vue fonctionnelle, mais il permet de représenter les documents émis ou reçus (vue informationnelle) et de définir des actions intervenant au sein des réseaux de circulation des produits/informations (vue des ressources).

6.2. Acteurs et données

Nous avons précédemment décrit que deux acteurs sont considérés lors d'une approche orientée processus classique : l'analyste métier et l'expert SITI. Lors d'une transformation de modèles depuis un environnement vers un autre, trois types de données sont échangées entre ces acteurs :

- (a) des données utilisables d'un modèle d'analyse lors de sa transformation vers un modèle d'implémentation. Il s'agit typiquement de données génériques et qui peuvent être réutilisées en modifiant l'environnement d'analyse ou encore celui d'implémentation ;
- (b) des données non prises en compte, car jugées inutiles, par exemple des données d'ordre graphique, des annotations sur le modèle ;
- (c) des données apportées de manière interne, indépendamment de l'autre acteur ici l'analyste métier. Il s'agit par exemple de données permettant de rendre le modèle SITI exploitable.

Ces différents acteurs et les données échangées sont représentés sur la figure 7.

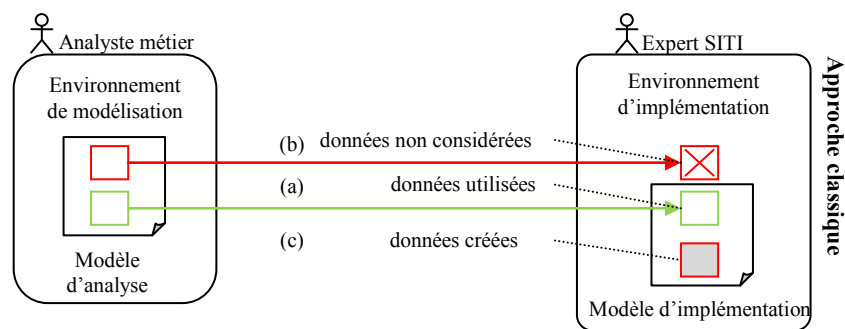


Figure 7. Différents acteurs, différents types de données

A travers notre approche, nous considérons qu'un troisième acteur se révèle nécessaire, l'*architecte des processus métiers*. En conciliant les deux niveaux d'abstractions propres à l'analyste métier et à l'expert SITI, l'acteur « pivot » permet de « fluidifier » les échanges et d'obtenir une meilleure cohérence entre

2. <http://www.omg.org/spec/BPMN/2.0/>

modèles. Il joue le rôle d'interface. Il possède les connaissances nécessaires métiers et techniques pour définir un « style architectural ». Il doit ainsi assurer la conservation de l'intégrité des informations d'un modèle d'analyse lors de la génération du modèle d'implémentation (données de type (b)). Il doit également permettre l'apport d'informations supplémentaires afin de diminuer l'apport en interne et d'avoir un modèle d'implémentation complet et en conformité avec le modèle d'analyse (figure 8).

L'opération inverse (depuis un modèle d'implémentation vers un modèle d'analyse) se réalise de la même manière et nécessite une même prise en charge par l'architecte de processus.

Nous avons vu le champ d'application de l'approche, ses acteurs et le type de données manipulées. Dans la suite de cette section, nous exposons les efforts de conception qui ont accompagné la définition de notre approche.

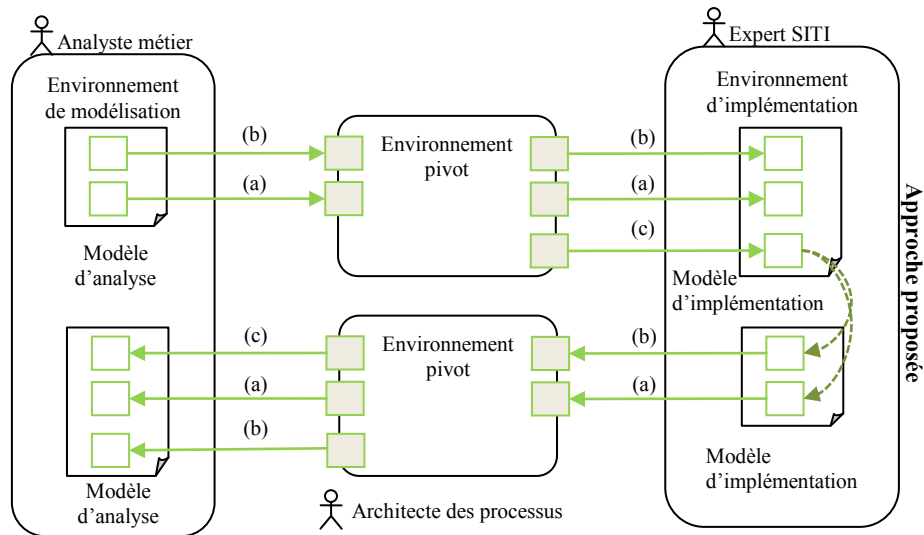


Figure 8. Trois environnements, trois acteurs

6.3. Conception

Notre approche se veut holistique. Il faut identifier tous les aspects décrivant un système afin de mieux le définir, ce qui dans notre cas revient à définir les activités à exécuter, les documents nécessaires à leur exécution, les agents impliqués et les ressources utilisées. Cependant, bien que nous désirions adopter une approche holistique et générique, cette globalité et cette généricité sont dépendantes d'un champ d'expertise particulier. Ainsi les métamodèles et modèles utilisés à travers notre approche obéissent à des règles et utilisent un vocabulaire propre à un champ d'expertise donné (banque, production, chimie...).

De manière à obtenir une approche formelle et rigoureuse, nous avons été amenés à définir formellement les métamodèles, qu'il s'agisse du métamodèle pivot, du métamodèle d'analyse ou encore du métamodèle d'implémentation. Lors d'une approche orientée processus, les métamodèles d'analyse et d'implémentation sont rarement explicites ou fournis par l'éditeur. Nous considérons qu'un métamodèle est défini (μ) à l'aide de deux éléments : un langage de représentation (MM_Rep) et un ensemble de règle (RefMet). Un modèle respectant RefMet et conforme à MM_Rep serait dès lors conforme (χ) au métamodèle défini (Ulmer *et al.*, 2010c).

Ensuite nous avons cherché à déterminer explicitement la notion de pivot et les relations d'équivalence pouvant exister entre les modèles et leurs métamodèles. Une relation bijective entre modèle permet de justifier mathématiquement la cohérence intermodèle. Cependant une telle relation serait difficile voire impossible à réaliser si les cardinalités entre métamodèles étaient différentes (ce qui est généralement le cas entre métamodèles de domaines différents). Une approche possible est d'utiliser un des modèles et de le transformer de manière à ce qu'il puisse contenir l'ensemble des éléments, concepts et informations des modèles. La transformation bijective devient une transformation surjective, ne modifiant ni les modèles d'entrée, ni les modèles de sortie. Notre environnement pivot intervient au niveau de cette nouvelle équivalence, les transformations entre modèles définissant les fonctions de conformité constructive de notre approche (Favre *et al.*, 2006), le modèle résultant étant le modèle pivot. Les fonctions de conformité constructive assurent la consistance des modèles générés lors de ces transformations, depuis l'environnement de modélisation vers celui d'implémentation et inversement. Cet ensemble de fonctions permet d'obtenir un lien entre la phase d'analyse et la phase d'implémentation et leurs modèles respectifs, qui sont alors considérés comme équivalents. La figure 9 détaille ces différentes relations, avec MM_i le métamodèle i et m_i le modèle i qui lui est conforme.

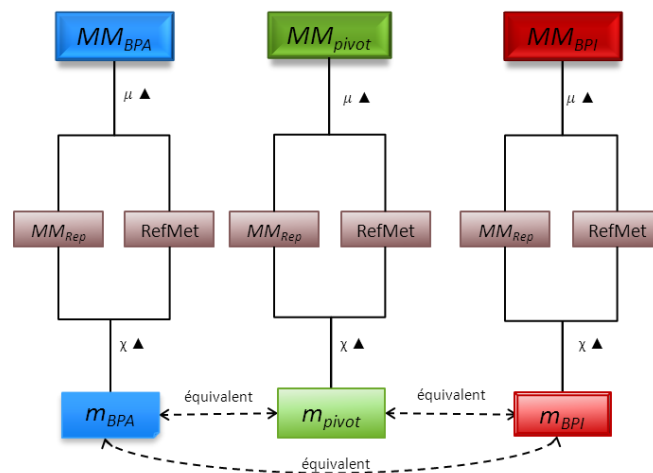


Figure 9. Modèles, métamodèles, métamodèle pivot

L'ensemble de cette section relative à la conception de notre approche est explicité plus en détail dans (Ulmer *et al.*, 2010b). Les différents points de conception abordés dans cet article sont illustrés au sein du projet de développement logiciel décrit dans la section suivante.

7. Plateforme SCALP

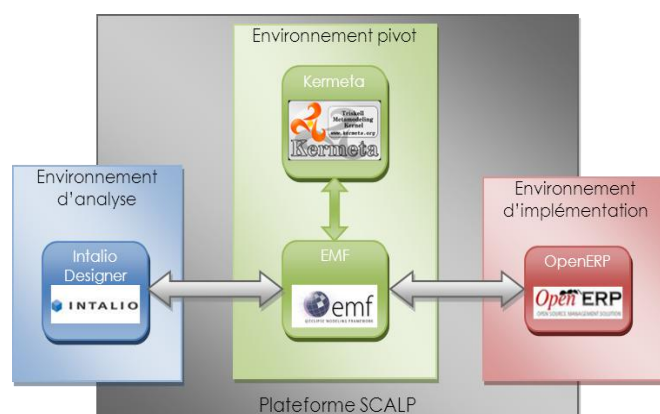


Figure 10. Plateforme SCALP

La plateforme logicielle développée propose une Solution pour la Cohérence et ALignement des Processus (SCALP). Cette plateforme repose sur les trois environnements pris en compte dans notre approche. L'environnement de modélisation prend en charge le modèle conceptuel de processus dessiné par un analyste métier. La fonctionnalité découlant de cet environnement est la modélisation du processus. A cet effet, nous utilisons l'outil de modélisation Intalio Designer 6.0.3³. L'environnement d'implémentation comprend la plateforme cible d'exécution. Au sein de cet environnement l'expert SITI modifie le modèle d'entrée obtenu précédemment, de manière à le rendre exécutable. Le moteur d'exécution de processus utilisé est un ERP, OpenERP 5.0.14⁴. Enfin l'environnement pivot, contient la plateforme pivot. Cette plateforme comporte deux fonctionnalités bien distinctes : la transformation d'un modèle provenant d'un environnement (le modèle d'entrée) vers un autre (le modèle de sortie) et la préservation de l'intégrité des informations contenues dans le modèle d'entrée. Pour instancier cette plateforme pivot, nous utilisons le framework EMF 1.4.0 (*Eclipse Modelling Framework*) de l'IDE (*Integrated Development Environment*) Eclipse 3.4.2⁵. La métamodélisation

3. <http://www.intalio.com/bpms/designer>

4. <http://www.openerp.com/>

5. <http://www.eclipse.org/>

est réalisée à l'aide d'Ecore 0.7.0 et est appuyée par le langage Kermeta⁶ 1.3.0 (« Kernel Metamodeling »). La plateforme SCALP et les environnements considérés sont représentés figure 10. Les différents outils et technologies impliqués sont répertoriés au sein du tableau 1. Si le prototype obtenu repose sur des outils spécifiques, l'approche mise en œuvre reste indépendante des outils technologiques utilisés. Un diagramme des cas d'utilisation est présent en annexe 1 (figure 21).

Abordons plus en détail l'architecture de cette plateforme logicielle. Tout d'abord, les différents métamodèles sont définis : le langage de représentation (MM_Rep) est décrit à l'aide d'Ecore et l'ensemble des règles (RefMet) grâce à Kermeta. Ulmer (2011) décrit en détail cette conception. Le contenu du RefMet est très dissemblable selon qu'il exprime les règles relatives à l'analyse, l'implémentation ou encore au pivot. Le RefMet du métamodèle d'analyse décrit les règles métiers portant sur le comportement. Le RefMet du métamodèle d'implémentation désigne l'ensemble des règles et contraintes que nous avons rencontrées lors de la réalisation de l'environnement d'implémentation. Le RefMet pivot assimile les deux RefMet précédemment cités afin de fournir des règles de contrôle et des valeurs par défaut des différents éléments du métamodèle pivot lors de leur instanciation.

Tableau 1. Technologies et applications impliquées

Niveau	Outil de modélisation	Environnement d'analyse	Environnement pivot	Environnement d'implémentation	Progiciel de gestion
Applications, applicatifs	Intalio Designer 6.0.2	Ecore 0.7.0, EMF 1.4.0, XML JDom	Ecore 0.7.0, EMF 1.4.0, XML JDom, Kermeta 1.3.0	Ecore 0.7.0, EMF 1.4.0, XML JDom	OpenERP 5.0.14, PGAdmin III
Technologies, langages	BPMN, XML	XML, XMI, Java	XML, XMI, Java	XML, XMI, Java	Python, PostgreSQL, XML
Format des fichiers	.bpmn, .bpmn_diagram	.xmi, .ecore, .java	.xmi, .ecore, .java, .kmt	.xmi, .ecore, .java	.xml, .py

La figure 11 illustre les opérations nécessaires au passage d'un modèle conceptuel issu de l'outil de modélisation Intalio Designer vers un modèle technique et son module OpenERP correspondant. Préalablement à ces transformations, les mappings (m1, m2, m3, m4) entre métamodèles et métamodèle Pivot doivent être réalisés.

Les différents mappings sont réalisés à l'aide de Kermeta. Kermeta peut être utilisé comme un tisseur d'aspect adapté aux métamodèles Ecore, capable de les manipuler sans les modifier. Pour cela, nous utilisons Kermeta avec le pattern de conception Visiteur (Gamma *et al.*, 1999).

6. <http://www.kermeta.org/>

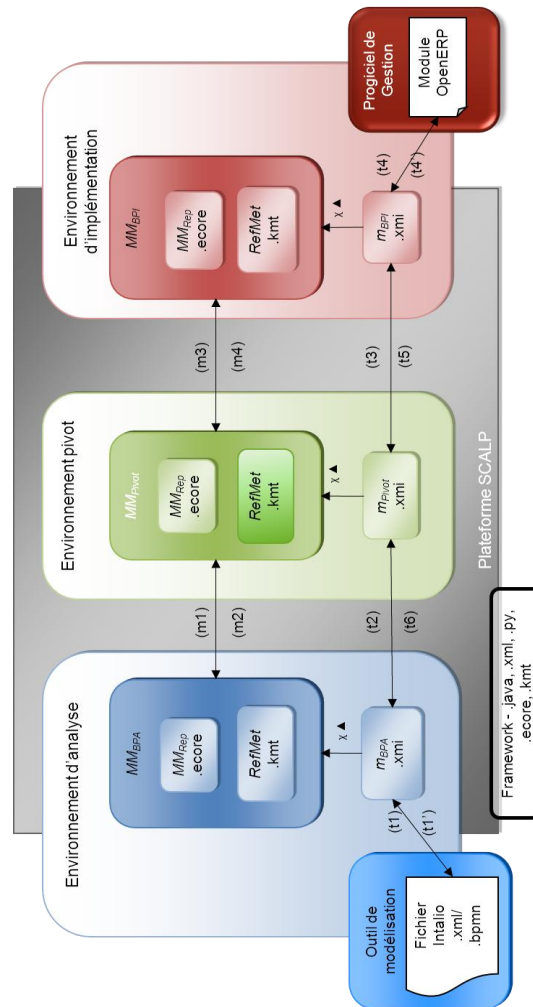


Figure 11. Architecture de SCALP

Ensuite les fonctions de conformité constructive sont réalisées : les transformations $t2$ et $t6$ (modèles d'analyse/pivot) et les transformations $t3$ et $t5$ (modèles d'implémentation/pivot). Les modèles générés sont par construction conformes à leurs métamodèles. Les transformations $t1$ et $t4'$ désignent les opérations nécessaires pour importer les modèles ou fichiers depuis les applications Intalio Designer et OpenERP vers la plateforme SCALP. A l'inverse, les transformations $t1'$ et $t4$ permettent de transformer les modèles issus de la plateforme SCALP en modèles utilisables par lesdites applications. Une démonstration de notre plateforme logicielle avec un cas d'étude est brièvement exposée dans la section suivante.

8. Mise en œuvre

Le processus métier que nous manipulons afin d'étayer nos propos est un processus de réintégration de produits ayant subi un contrôle qualité. Ce processus a été récemment mis en place par un grand groupe de cosmétique pour ses unités de production de parfum. Le but de ce processus est de réinjecter dans le flux de production le lot prélevé et contrôlé par le département Qualité, de manière à réduire les pertes financières. Il est utilisé comme un exemple de référence pour valider notre plateforme et démontrer notre approche. Pour cet article, nous utilisons ce modèle de processus à travers un scénario à deux phases. Durant la première phase, nous effectuons une transformation depuis le diagramme Intalio de processus vers un module OpenERP. Puis à travers la deuxième phase nous réalisons la transformation inverse, depuis le module OpenERP vers le diagramme Intalio. Au préalable, nous effectuons des modifications fonctionnelles et structurelles au module OpenERP créé (figure 12). Ces modifications sont décrites section 8.2.

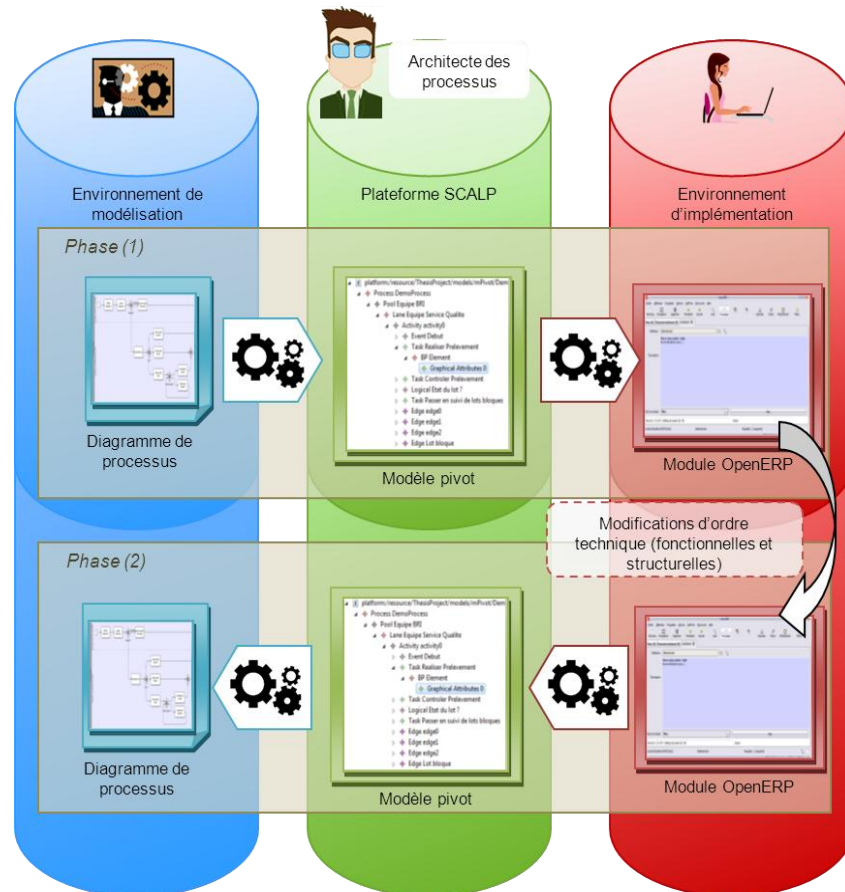


Figure 12. Scénario de démonstration

Ce scénario implique l'ensemble des acteurs décrits dans la section 3.1. Il permet de mettre en évidence les différents mécanismes et interactions intervenant lors d'une transformation « simple » depuis un modèle d'analyse vers un modèle d'implémentation. Le scénario met également en exergue l'utilité de notre plateforme et notamment les efforts de synchronisation, d'équivalence sémantique et donc de cohérence entre modèles lors de la transformation inverse.

8.1. Première phase : du diagramme de processus Intalio Designer au module OpenERP

Nous considérons que les métamodèles d'analyse, d'implémentation et pivot, soit respectivement MM_{BPA} , MM_{BPI} et MM_{Pivot} , existent et sont bien définis. La première phase consiste à transformer un diagramme de processus respectant une charte graphique en un module exécutable sous un ERP.

8.1.1. Du diagramme de processus Intalio vers un modèle d'analyse m_{BPA}

Nous importons les fichiers XML obtenus à partir d'Intalio Designer au sein de notre plateforme SCALP : le premier fichier décrit les informations graphiques, le second considère la logique exprimée par le processus. La transformation t_1 est un programme java utilisant le parseur XML JDOM⁷. Cette transformation crée le modèle m_{BPA} à partir des deux fichiers cités précédemment (figure 13).

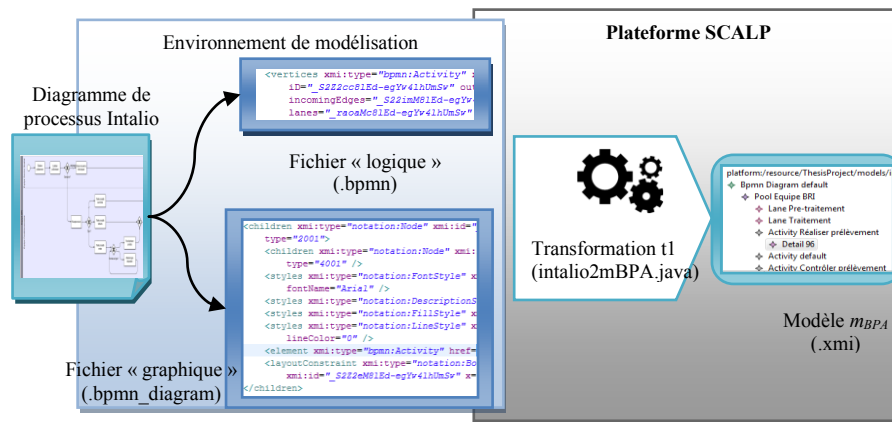


Figure 13. Transformation t_1 : du diagramme de processus au m_{BPA}

Le modèle obtenu est au format .xmi et est conforme au métamodèle BPA, le MM_{BPA} exprimé sous format ecore. Chaque modèle m_i rencontré durant les différentes

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

phases du scénario de validation doit être conforme à son métamodèle MM_i associé afin d'être manipulable par Kermeta. Cette conformité est vérifiée à l'aide du menu contextuel proposé par Ecore (clic-droit → « Validate »). A partir de ce modèle m_{BPA} , nous effectuons la transformation t2 afin d'obtenir notre modèle m_{Pivot} .

8.1.2. Du modèle d'analyse vers le modèle d'implémentation

La transformation t2 permettant d'obtenir un modèle pivot m_{Pivot} à partir de m_{BPA} est un programme Kermeta. La transformation t2 est effectuée en reprenant le paradigme du pattern visiteur. Un programme Visiteur détermine les éléments (et leurs attributs) appartenant à MM_{BPA} . Puis un programme Transformation les associe aux éléments et attributs constituant le MM_{Pivot} .

Avant la réalisation de la transformation t2, les mappings m1 ($MM_{BPA} \rightarrow MM_{Pivot}$) et m2 ($MM_{Pivot} \rightarrow MM_{BPA}$) doivent être effectués. Le tableau 2 reprend ces mappings.

Tableau 2. Mappings m1 et m2 : correspondance sémantique entre MM_{BPA} et MM_{Pivot}

Elément MM_{BPA} (A)	Correspondance sémantique	Elément MM_{Pivot} (B)
BpmnDiagram	$\overset{s}{A = B}$	Process
Pool	$\overset{s}{A = B}$	Pool
Lane	$\overset{s}{A = B}$	Lane
Activity	$\overset{s}{A = B}$	Activity
SubProcess	$\overset{s}{A = B}$	SubProcess
SequenceEdge	$\overset{s}{A = B}$	Edge
Detail	$\overset{s}{A \subset B}$	BPElement

A titre d'illustration d'utilisation du modèle pivot, considérons le mapping m1 de l'élément Activity. Le tableau 3 définit les correspondances existantes entre les différents attributs selon qu'il s'agisse d'une activité définie par le MM_{BPA} ou par le MM_{Pivot} . Nous pouvons constater que de manière générale, la définition des éléments du MM_{Pivot} englobe celles des éléments du MM_{BPA} . Il en va logiquement de même entre MM_{Pivot} et MM_{BPI} . Les informations concernant l'aspect graphique de l'élément Activity sont contenues dans l'élément BPElement du métamodèle pivot. Cet élément permet de prendre en considération des informations n'influençant pas le modèle BPA et/ou le modèle BPI mais jugées nécessaires dans un but de rétro-ingénierie des processus (figure 14).

Tableau 3. Mapping $m1$ de l'élément Activity entre MM_{BPA} et MM_{Pivot}

Elément MM_{BPA} (A)	Attribut		Elément MM_{Pivot} (B)	Attribut	Attribut équivalent
Activity		$A \overset{s}{\subset} B$	Activity		
	name	$A \overset{s}{=} B$		name	
	iD	$A \overset{s}{\subset} B$		BPElement	BPElement:GraphicalAttributes : businessData
	activityType	$A \overset{s}{\subset} B$		Node	Node:Logical:LogicalType
	x	$A \overset{s}{\subset} B$		BPElement	BPElement:GraphicalAttributes : CoordinatesType :xCordinate
	y	$A \overset{s}{\subset} B$		BPElement	BPElement:GraphicalAttributes : CoordinatesType :yCoordinate
	height	$A \overset{s}{\subset} B$		BPElement	BPElement:GraphicalAttributes : height

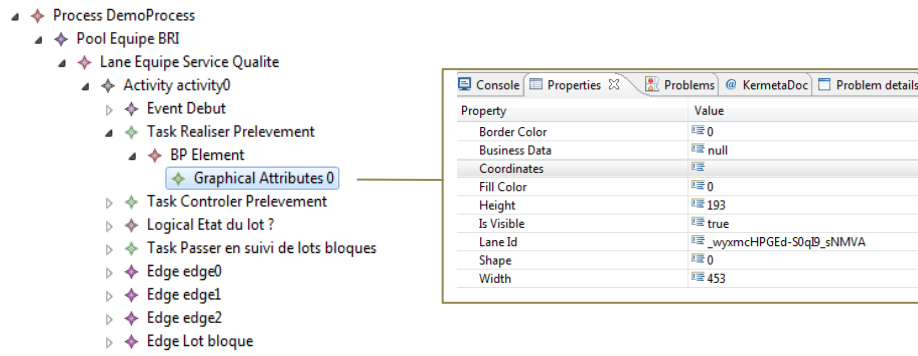


Figure 14. Exemple d'utilisation de l'élément BPElement

La transformation $t3$ possède les mêmes mécanismes que la transformation $t2$. Elle nécessite un modèle XMI en entrée, le m_{pivot} et fournit un modèle XMI en sortie, le m_{BPI} et nécessite également au préalable la définition des mappings $m3$ et $m4$. Ces équivalences sémantiques entre MM_{Pivot} et MM_{BPI} sont définies tableau 4.

Tableau 4. Correspondance sémantique entre MM_{Pivot} et MM_{BPI}

Elément MM_{Pivot} (A)	Correspondance sémantique	Elément MM_{BPI} (B)
Process	$\overset{s}{B \subset A}$	Data
Pool	$\overset{s}{B \subset A}$	Workflow
Lane	$\overset{s}{A \cap B}$	*
Activity	$\overset{s}{B \subset A}$	Activity
SubProcess	$\overset{s}{B \subset A}$	Activity
Edge	$\overset{s}{B \subset A}$	Transition

* : L'élément Lane n'existe pas en tant que tel dans un modèle mBPI. Cependant, son attribut name définit l'attribut role_id de l'élément Transition. Concrètement, lors de l'exécution du module, seule une catégorie d'utilisateurs reconnue ou disposant des mêmes droits que la catégorie role_id pourra valider une transition possédant l'attribut role_id.

Suite à la transformation t3, nous obtenons donc un modèle au format .xmi, le mBPI. Ce format, manipulable par Kermeta, permet la vérification de sa conformité avec le métamodèle BPI, MM_{BPI} , exprimé sous format ecore.

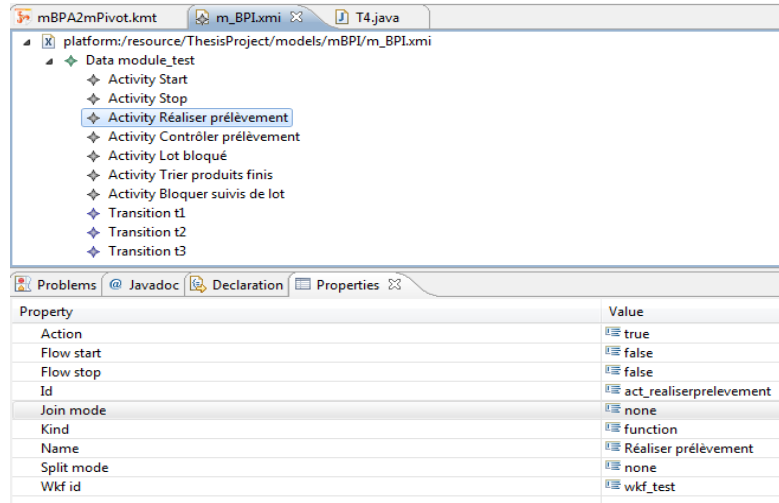


Figure 15. Extrait du m_{BPI} obtenu

La figure 15 montre un extrait du m_{BPI} obtenu. En effectuant une comparaison avec la figure 14, nous pouvons constater que les informations d'ordre graphique ne sont plus renseignées. De nouvelles propriétés nécessaires au fonctionnement d'un

module OpenERP viennent enrichir ce modèle de processus. Ces nouvelles données, issues de la transformation t3 possèdent des valeurs par défaut et permettent d'obtenir un mBPI conforme à son métamodèle M_{BPI} et de faciliter la génération d'un module OpenERP fonctionnel.

8.1.3. Du modèle d'implémentation vers le module OpenERP

A partir du mBPI obtenu, nous constituons le module OpenERP, il s'agit de la transformation t4 représentée figure 16.

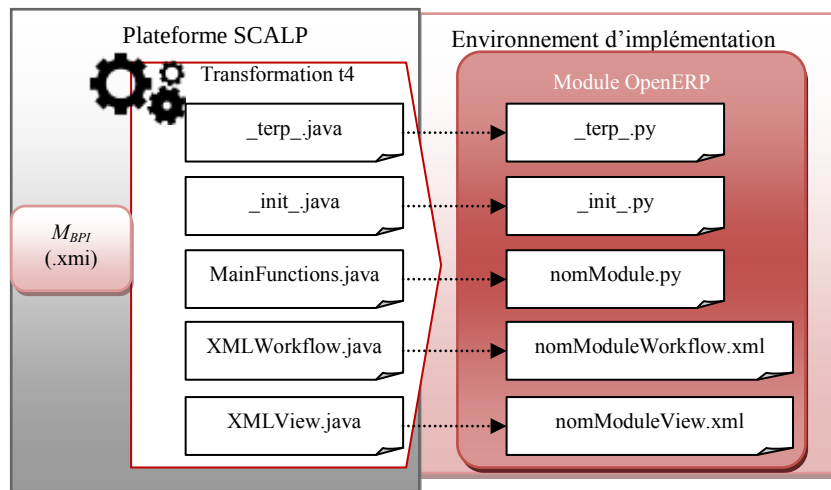


Figure 16. Transformation t4 : du m_{BPI} au module OpenERP

Pour cela, cinq programmes java sont définis. Chacun de ces programmes fournit en sortie un fichier constituant le module OpenERP. Il se traduit ainsi par un dossier contenant cinq types de fichiers qui sont sous format python (.py) et xml. Le fichier `__terp__.py` est une description du module indiquant le nom du module, sa version, ses dépendances vis-à-vis d'autres modules, etc. Le fichier `__init__.py` est un fichier de démarrage indiquant les imports à effectuer lors du lancement du module, notamment les *wizards* associés au module. Un fichier `nomModule.py` décrit les classes spécifiques au module. Ces classes décrivent les formulaires associés au module. Elles définissent également la structure de l'interface. Enfin la dernière catégorie de fichiers regroupe les fichiers en .xml, dans lesquels nous fournissons une description du séquençement d'activités du module (`nomModule_workflow.xml`), ou encore son interface (`nomModule_view.xml`).

Ces différents types de fichier sont représentés et détaillés dans (Ulmer *et al.*, 2010a).

8.2. Deuxième phase : du module ERP au diagramme de processus

Le module OpenERP obtenu à l'étape précédente n'est pas intégralement utilisable en l'état. Ainsi les fichiers `nomModule.py` et `nomModuleView.xml` demandent un apport d'information afin d'être utilisables. A cet effet, ces fichiers sont amenés à être modifiés par l'expert SITI : des fonctionnalités liées à l'interface OpenERP ou à l'affichage du module doivent être implémentées. En effet, à ce stade de développement, la génération de l'interface du module OpenERP reste très générique. L'ergonomie est à retravailler au cas par cas, de même que le choix des informations à afficher.

Conformément au scénario décrit en début de section 8, nous effectuons des modifications d'ordre technique sur le module OpenERP. Le fichier `nomModule.py` décrit la vue fonctionnelle de notre processus, des fonctions logiques exprimées en langage python sont ajoutées. Ces fonctions représentent les actions réalisées par les différentes activités du processus de réintégration. Par exemple, nous désirons ajouter une fonctionnalité évaluant le temps effectif passé sur un lot ciblé. Nous effectuons également des modifications de type structurel, nous simulons l'ajout d'une activité permettant de mettre à jour la liste des lots de produits semi-finis utilisés dans le processus de réintégration, il s'agit de l'activité « Mettre la liste à jour ». Cet ajout se traduit par une modification du control-flow. En effet l'ajout de l'activité provoque des changements au niveau du séquençement : il s'agit d'une modification significative. Les fichiers `nomModule.py` et `nomModule_workflow.xml` sont modifiés en conséquence à l'aide de l'éditeur d'OpenERP.

8.2.1. Du module OpenERP vers le modèle d'implémentation

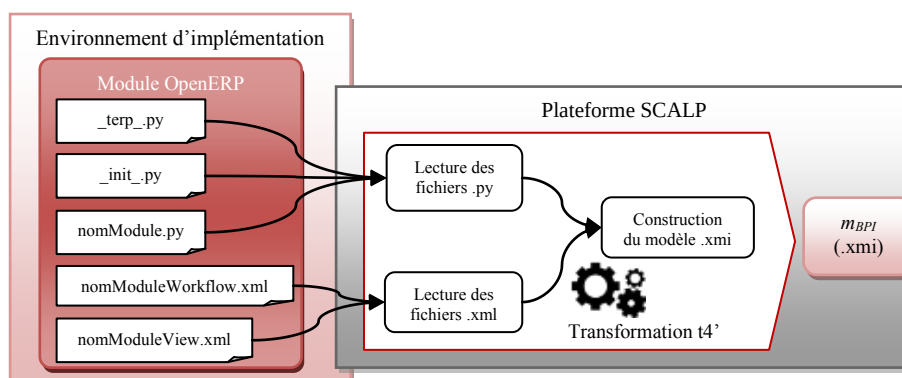


Figure 17. Transformation $t4'$: du module OpenERP au m_{BPI}

La transformation $t4'$ est un programme java se décomposant en trois étapes bien distinctes (figure 19). Une première étape consiste à parcourir les différents fichiers

python et à relever les informations nécessaires au m_{BPI} . De même la deuxième étape utilise le parseur XML JDOM sur les fichiers XML du module OpenERP. Puis la transformation génère la création d'un modèle m_{BPI} , également réalisé à l'aide du parseur XML JDOM. Le modèle obtenu est au format .xmi et est conforme au métamodèle BPI, le MM_{BPI} . La figure 17 retrace le déroulement de cette transformation.

8.2.2. Du modèle d'implémentation au diagramme de processus

Les transformations t5 et t6 permettent respectivement d'obtenir m_{Pivot} à partir de m_{BPI} et m_{BPA} à partir dudit m_{Pivot} . Ces transformations se réalisent de la même manière que les transformations t2 et t3 :

- à l'aide du langage Kermeta ;
- et en utilisant le paradigme du pattern visiteur.

Cependant, deux différences notables sont à signaler. Tout d'abord, lors de la transformation t5, nous apportons des informations supplémentaires concernant le modèle de processus. Ces informations, issues de la plateforme d'implémentation sont notifiées à travers le sous-élément *ImplAttributes* de l'élément *BPElement*. Dès lors, en récupérant les données contenues dans le sous-élément *GraphicalAttributes* du m_{Pivot} précédent, nous générons un nouveau modèle pivot contenant à la fois les données spécifiques au m_{BPA} et au m_{BPI} . Afin d'éviter toutes ambiguïtés, le modèle pivot généré est appelé $m_{Pivot}^{\phi 2}$ (figure 18).

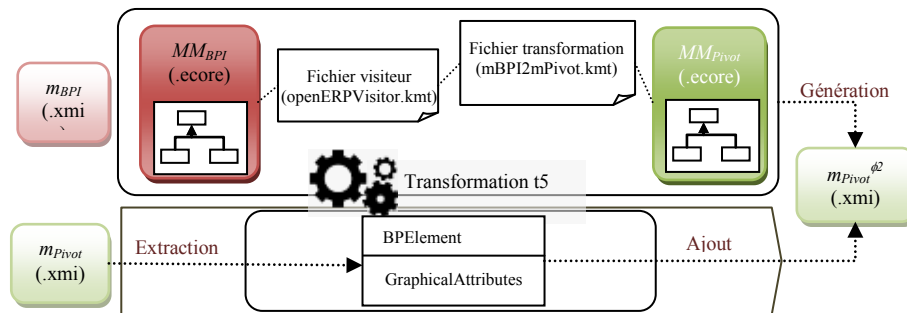


Figure 18. Transformation t5 : du m_{BPI} au $m_{Pivot}^{\phi 2}$

Lors de la transformation t6, nous utilisons les données contenues dans *BPElement* afin de générer un modèle d'analyse possédant les mêmes caractéristiques graphiques que le modèle d'analyse précédent. Pour éviter toutes ambiguïtés, le modèle obtenu est nommé $m_{BPA}^{\phi 2}$.

Enfin, la transformation t1' permet de transformer le modèle d'analyse obtenu durant cette deuxième phase en un diagramme de processus. Pour cela, nous utilisons un programme java utilisant le parseur XML JDOM. Ce programme génère

les deux fichiers nécessaires à l'outil Intalio Designer pour représenter un processus : un fichier « logique » et un fichier « graphique » (figure 19).

REMARQUE. — Si lors de cette deuxième phase, les modifications apportées ne concernaient que les fonctionnalités des activités du processus (autrement dit leur exécution sous OpenERP) et l'apparence du module OpenERP, le diagramme de processus obtenu est identique à celui utilisé lors de la première phase.

Dans cet exemple les modifications portant sur l'orchestration des activités effectuées par l'expert SITI sont correctement répercutées sur le diagramme Intalio. L'ajout de l'activité « Mettre la liste à jour », activité suivant l'activité « Traiter les produits semi-finis » et précédant l'activité « Traiter en interne » est bien représenté comme le montre la figure 20. Lors de la génération du modèle pivot, les attributs graphiques de cet élément ont une valeur par défaut définie par l'architecte des processus (« 0 », « null » ou « empty » selon les cas), d'où sa présence à l'origine du diagramme (avec ses coordonnées $x = 0$ et $y = 0$).

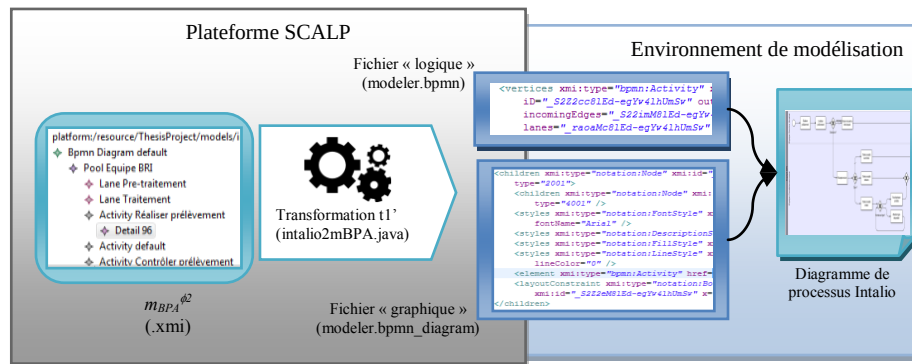


Figure 19. Transformation $t1'$: du $mBPA^{\phi 2}$ au diagramme Intalio

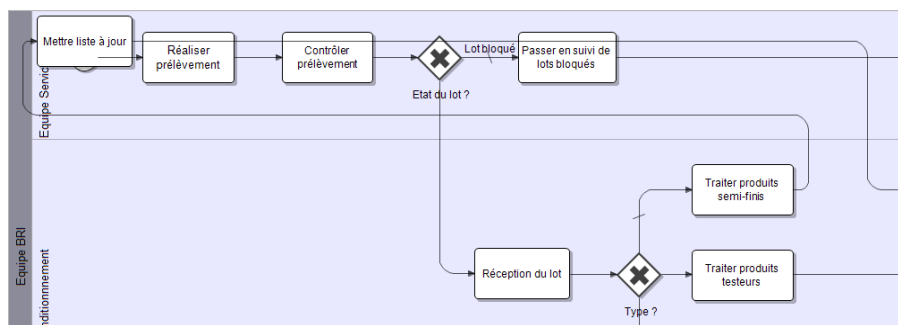


Figure 20. Extrait du diagramme Intalio obtenu

8.3. En résumé

Lors de la première phase, nous avons réalisé la transformation d'un modèle conceptuel, un diagramme de processus, en un modèle technique, un module ERP. A chaque étape de cette transformation, la conformité entre le modèle xmi obtenu et son métamodèle est validée à l'aide d'Ecore. Durant cette phase, nous avons pu montrer comment nous préservions les données non utilisables par le modèle technique, à l'aide du modèle pivot. Ces données sont restituées lors de la seconde phase, nous permettant d'obtenir un modèle conceptuel complet.

A travers les différentes transformations, nous générons un modèle de sortie prenant directement en compte les modifications subies par le modèle d'entrée (par exemple la création d'une activité) ou indirectement (modifications des instructions python associées à une activité). Les modifications étant propagées lors de ces transformations et l'intégrité des informations étant assurée, notre approche permet une synchronisation et une équivalence sémantique des modèles. Nous obtenons alors une cohérence intermodèle et donc un alignement opérationnel.

Les contributions apportées ainsi que nos travaux futurs sont décrits dans la section suivante.

9. Conclusions et perspectives

Nos principales contributions furent de définir l'alignement opérationnel et sa réalisation au sein d'une entreprise. Pour cela nous préconisons l'utilisation d'une approche orientée pivot permettant une cohérence intermodèle entre modèles hétérogènes et un couplage faible entre environnements d'analyse et d'implémentation. L'obtention systématique d'une synchronisation, d'une équivalence sémantique et finalement d'une cohérence intermodèle permettent de réduire l'écart existant entre le domaine métier et le domaine SITI.

Afin de supporter cette approche, une solution logicielle a été développée, la plateforme SCALP dont une démonstration a été réalisée à l'aide d'un processus issu de l'industrie cosmétique. Son utilisation améliore l'alignement opérationnel entre d'une part, les infrastructures et les processus d'entreprise et, d'autre part, les infrastructures et les processus techniques. Et en permettant un couplage faible entre modèles ainsi qu'une formalisation des transformations à l'aide de métamodèles, notre approche et la plateforme qui lui est associée améliorent l'agilité d'entreprise.

Diverses perspectives sont envisagées suite à nos travaux. Il s'agira tout d'abord d'éprouver notre plateforme en utilisant des processus plus spécifiques et/ou détaillés issus de PME/PMI. Cela nous permettra d'évaluer si l'expressivité du métamodèle pivot est suffisante.

Nous chercherons également à étayer l'aspect « couplage faible » de notre plateforme. Pour cela, son utilisation avec de nouveaux outils est envisagée (avec par exemple Bizagi Process Modeler en amont et des outils de suites bureautiques type Microsoft Excel en aval).

L'approche que nous proposons a été développée dans une optique de génération de code. Prendre en considération une optique d'orchestration de services préexistants pourrait la compléter. Pour cela, les mécanismes manipulant la partie pivot de notre approche devront être modifiés et fournir des méthodes de décomposition ou détection des éléments du processus en services déclarés préalablement. La plateforme cible serait des SI à base de composants ou packages standard de services.

A l'aide de notre approche, nous avons cherché à définir les liens existant entre les SI (infrastructures et processus techniques) et les unités métiers (infrastructures et processus d'entreprise), et ainsi apporter notre contribution à la littérature. Nous avons réduit nos travaux au domaine interne en nous préoccupant essentiellement de ces infrastructures SITI et des processus d'entreprise. Des travaux futurs pourront être menés sur la relation entre domaine interne et les stratégies employées par l'entreprise. L'étude de cet ajustement stratégique, menée notamment par (Thevenet *et al.*, 2009) et (Avila Cifuentes, 2009) permettrait d'élargir les possibilités de notre approche. Nous serions en mesure de conceptualiser l'ensemble des séquences d'alignement telles que définies par le *Strategic Alignment Model* (SAM).

Une autre perspective possible à prendre en compte est la possibilité d'inclure un environnement dédié à la supervision des processus métiers, le BAM. Son rôle serait de fournir les outils nécessaires permettant d'obtenir un retour d'informations sur l'utilisation et l'exécution des processus issus de l'environnement SITI. Les informations devront ensuite être transmises à l'environnement métier sous forme d'indicateurs de performance significatifs.

Bibliographie

- Avila Cifuentes O.J. (2009). *Contribution à l'alignement complet des systèmes d'information techniques*. Thèse de doctorat, Université de Strasbourg.
- Baptiste J.-L. (2009). *Merise - Guide Pratique - Modélisation des données et des traitements, langage SQL*. Eyrolles.
- Belaud J.-P. (2002). *Architectures et technologies des systèmes logiciels ouverts - Cape-Open, un standard pour l'interopérabilité et l'intégration des composants logiciels de l'ingénierie des procédés*. Thèse de doctorat, Institut National Polytechnique (INP) de Toulouse.
- Blay-Fornarino M., Dao M., Lahire P., Rivierre N. (2008). *Transformations depuis les modèles métier*. Research Report RNTL FAROS, number F-2.4, p. 1-42.
- Combemale B. (2008). *Approche de métamodélisation pour la simulation et la vérification de modèle. Application à l'ingénierie des procédés*. Thèse de doctorat, Institut de Recherche en Informatique de Toulouse.
- Debauche B., Mégard P. (2004). *BPM Business Process Management : Pilotage métier de l'entreprise*. Hermes Science Publications.
- Etien A (2009). Business process/information system co-evolution. *Ingénierie des systèmes d'information*. Hermes-Lavoisier, vol. 14, n° 6, p. 41-57.

- Etien A. (2006). *Ingénierie de l'alignement: Concepts, Modèles et Processus - La méthode ACEM pour l'alignement d'un système d'information aux processus d'entreprise*. Thèse de doctorat, Université Paris I - Panthéon - Sorbonne.
- Favre J.-M., Estublier J., Blay-Fornarino M. (2006). *L'ingénierie dirigée par les modèles - Au-delà du MDA*. Hermès-Lavoisier.
- Gamma et al. (1999). *Design patterns. Catalogue des modèles de conception réutilisables*. Vuibert informatique.
- Henderson J.C., Venkatraman N. (1993). Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal*, p. 3-16.
- International Standards Organization (ISO) (2005). *Enterprise integration - Framework for enterprise modeling, ISO/FDIS 19439*.
- Luftman J., Maclean E.R. (2004). Key issues for IT executives. *MIS Quaterly Executive*, University of Minnesota.
- Meinadier J.-P. (2002). *Le métier d'intégration de systèmes*. Hermès Science Publications.
- Silvius D.A.J.G. (2007). Business & IT Alignment in theory and practice. *40th Hawaii International Conference on System Sciences*, Hawaii.
- Thevenet L.-H. (2010). *Proposition d'une modélisation conceptuelle d'alignement stratégique : La méthode INSTAL*. Thèse de doctorat, Université Panthéon-Sorbonne.
- Thevenet L.-H., Rolland C., Salinesi C. (2009). Alignement de la stratégie et du système d'information. Présentation de la méthode INSTAL. *Ingénierie des systèmes d'information*, vol. 14, n° 6, Hermès-Lavoisier, p. 19-39.
- Touzi J. (2007). *Aide à la conception de Système d'Information Collaboratif, support de l'interopérabilité des entreprises*. Thèse de doctorat, Ecole des Mines d'Albi Carmaux.
- Ulmer J.-S. (2011). *Approche générique pour la modélisation et l'implémentation des processus*. Thèse de doctorat, ENSIACET - Laboratoire de Génie Chimique.
- Ulmer J.-S., Belaud J.-P., Le Lann J.-M. (2010a). Développement d'un framework pour une modélisation multi-perspectives des processus. *XXVIII^e Congrès INFORSID - Atelier Ingénierie d'Entreprise et des Systèmes d'Information (IESI'10)*, Marseille, France.
- Ulmer J.-S., Belaud J.-P., Le Lann J.-M. (2010b). Proposition d'une approche générique pour la formalisation et l'implémentation des processus. *Ingénierie des Systèmes d'Information*, vol. 15, n° 4, Hermès-Lavoisier, p. 63-87.
- Ulmer J.-S., Belaud J.-P., Le Lann J.-M. (2010c). Towards a pivotal-based approach for Business process alignment. *8th International Conference of Modeling and Simulation - MOSIM'10*, Hammamet - Tunisia.
- Various IIBA, Brennan K. (2008). *A Guide to the Business Analysis Body of Knowledge (BABOK)*. Canada: International Institute of Business Analysis.
- Zur Muehlen M. (2004). *Workflow-based Process Controlling Foundation, Design, and Application of Workflow-driven Process Information Systems*. Berlin: Logos Verlag.

Annexes

Annexe A. Diagramme des cas d'utilisation de la plateforme SCALP

Le système considéré (« System » sur la figure suivante) est la plateforme SCALP.

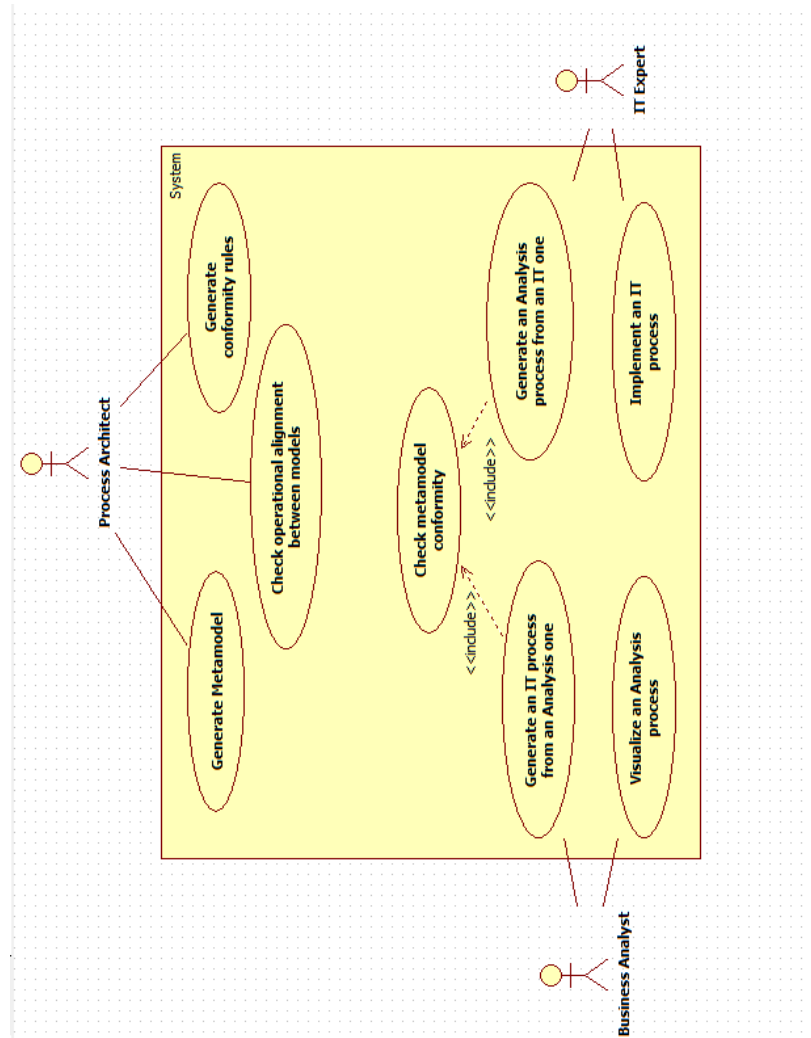


Figure 21. Diagramme des cas d'utilisation de la plateforme SCALP

Annexe B : Présentation du processus étudié

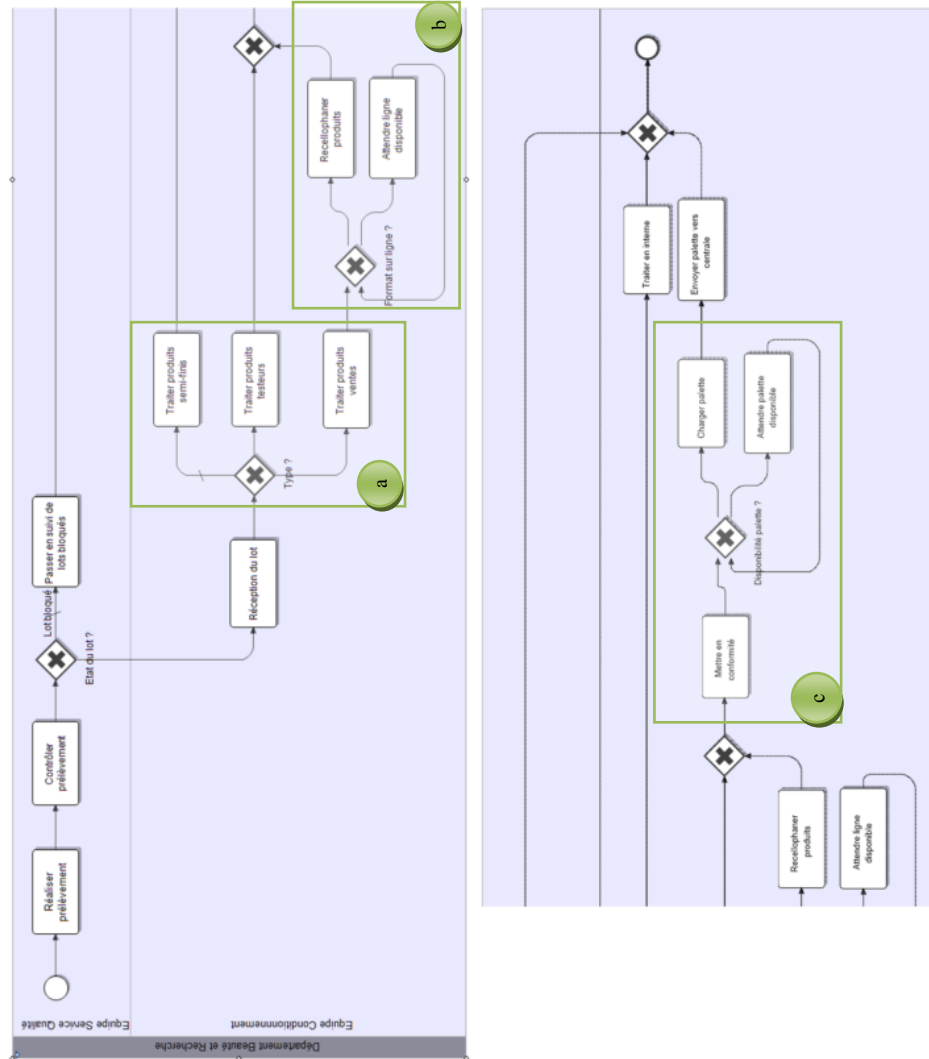


Figure 22. Processus de réintégration de parfum

Le processus considéré est un processus de réintégration de produits ayant subi un contrôle qualité. Ce processus a été récemment mis en place par un grand groupe de cosmétique pour ses unités de production de parfum. Nous l'utilisons ici comme un exemple de référence pour valider notre plateforme et démontrer notre approche.

Durant le conditionnement des produits, un pourcentage de la production est prélevé afin d'être contrôlé par le département Qualité. Le lot prélevé doit ensuite être réinjecté dans le flux de production afin d'éviter des pertes financières importantes. Cependant, les produits testés constituant ce lot sont ouverts durant ce contrôle. La réintégration concerne un flux de 300 000 produits par an, selon 94 formats de produits différents. Ceci implique de reconstruire l'emballage du produit avec un film cellophane sur une ligne de fabrication adéquate et de l'encrypter contre le détournement de produits. Un processus de réintégration de parfums est alors défini afin de rendre la majorité du lot réutilisable. Il est mis en place au sein de l'organisation

Le processus commence lors du prélèvement d'un lot de produits sur une ligne de production. Le lot est contrôlé par le laboratoire de Qualité. Les produits du lot prélevé à un instant t peuvent être de trois types différents (figure 22 - a). (1) Les produits destinés à la vente nécessitent une étape de cellophanage avant d'être envoyés à une centrale de distribution. (2) Les produits testeurs ne nécessitent pas de cellophanage avant leur envoi en centrale de distribution. (3) Les produits semi-finis ne nécessitent pas non plus une étape de cellophanage mais requièrent un traitement interne pour une éventuelle utilisation future. Il faut également considérer que l'étape de cellophanage ne s'effectue que sur une ligne de production vide afin d'éviter des mélanges de numéro de lot (figure 22 - b). Avant leur envoi en centrale, les lots constitués de produits de type 1 ou 2 suivent une étape de remise en conformité puis une étape de constitution de palette (figure 22 - c).