



Tâche 2.4 Développement du connecteur intelligent MDI

Livrable : 2.4. : Spécification des connecteurs intelligents

Auteurs:

Judicael Ribault – Yves Ducq

Objet du document

L'objectif du document est:

- Elaborer les spécifications des connecteurs
- Définir les modèles de transformation BSM/TIM/TSM
- Définir l'ontologie adaptée
- Présenter le connecteur intelligent MDI utilisé dans la SLM ToolBox (développée par le projet MSEE)

Historique du document

Versions du document	Date	Statut	Auteurs(s)
V 0.1	22/01/15	Initial document	JR
V 0.2	26/01/15		JR
V 0.3	28/01/15		JR

SOMMAIRE

1. Introduction.....	4
1.1 Rappels des objectifs du livrable	4
1.2 Plan du document	5
2. MDSEA.....	5
2.1 Business Service Model (BSM)	7
2.2 Technology Independent Model (TIM).....	8
2.3 Technology Specific Model (TSM).....	8
3. Langages de modélisation proposées.....	9
3.1 Extended Actigram Star (EA*).....	10
3.2 BPMN.....	13
4. Transformation de modèle	14
4.1 Approche méta-modèle.....	14
4.2 Transformer des modèles EA* en modèles BPMN.....	16
4.2.1 Diagramme de collaboration EA* vers BPMN – Mappage.....	16
4.2.2 Diagramme de collaboration EA* vers BPMN – Règles de transformation.....	17
4.2.3 Diagramme de processus de EA* vers BPMN – Mappage de modèle	19
4.2.4 Diagramme de processus de EA* vers BPMN – Règles de transformation	19
5. Conclusions.....	21
Références	21

LISTE DES FIGURES

Figure 1 The MDSEA architecture applied in a service network of two enterprises.....	7
Figure 2 Extended Actigram Star metamodel	12
Figure 3 Transformation architecture of EA* to BPMN.....	15

LISTE DES TABLEAUX

Table 1 - EA* vers BPMN - Mappage (diagramme de collaboration)	17
Table 2 - EA* vers BPMN - Mappage (diagramme de collaboration)	19

1. Introduction

1.1 Rappels des objectifs du livrable dans NOSCIFEL

La transformation de modèle est un concept qui a été largement utilisé dans le développement dirigé par les modèles (Model Driven Development (MDD)) et dans l'ingénierie dirigée par les modèles (Model Driven Engineering (MDE)). En effet, le rôle des modèles dans le processus de développement est passé d'un rôle contemplatif (i.e. utilisé à des fins de documentation) à un rôle productif grâce à la transformation de modèle de haut niveau vers des modèles d'exécutions. L'interopérabilité dirigée par les modèles (Model Driven Interoperability (MDI)) est un autre domaine qui envisage la transformation de modèle pour résoudre les problèmes d'interopérabilités entre plusieurs entreprises, pas seulement au niveau applicatif et logiciel mais également au niveau de la modélisation d'entreprise.

Dans le cadre du projet NOSCIFEL, nous ne devons pas uniquement résoudre le problème de modélisation du système transport mais également résoudre le problème de l'interopérabilité entre ces systèmes et notamment les plateformes de communication. Pour cela, nous avons utilisé la démarche proposée par le projet MSEE qui s'appuie sur les concepts de transformation de modèles. L'architecture MDSEA a été proposée pour résoudre le problème de l'interopérabilité par les modèles. Cette architecture permet à la fois la génération de logiciel par la transformation de modèles et également la génération de logiciel d'interopérabilité. Ces résultats découlent d'un long travail initialisé par INTEROP-VLab en 2007 qui s'appuyait sur les approches MDA (Model Driven Architecture) (OMG 2003 [1]) et sur l'approche MDI (Model Driven Interoperability) développé par le projet INTEROP-NoE [2].

1.2 Plan du document

Ce livrable sera décomposé comme suit:

- Une première section pour définir les objectifs du livrable et le plan.
- Une deuxième section présente la méthodologie MDSEA de transformation de modèle basée sur les concepts de MDA/MDI. Cette section présente les différents niveaux de modélisation de MDSEA (BSM, TIM, TSM),
- Une troisième section présente les langages de modélisation sélectionnés (EA* et BPMN2.0),
- Une quatrième section propose l'approche de transformation par méta-modèle (de EA* vers BPMN),

Enfin une conclusion terminera le contenu de ce livrable.

2. MDSEA

MDSEA (Model Driven Service Engineering Architecture) est inspiré de MDA/MDI. La méthodologie MDSEA s'inscrit dans le cadre du projet MSEE [3] dont le premier Grand Challenge est de faire évoluer SSME (Service Science, Management and Engineering) autour des systèmes de fabrication et des usines du futur. MDSEA propose une méthodologie basée sur des langages de modélisation à différents niveaux d'abstraction pour gérer des modèles de service et la conception et l'implémentation de système de service. La relation entre les niveaux de modélisation de MDSEA et les phases du cycle de vie du système de services (besoin utilisateurs, conception, implémentation) est établie. Une des innovations importantes dans MDSEA est de définir l'intégration entre les composants de domaine (IT, organisation, moyens physique et humain) au niveau du modèle de services aux entreprises (BSM) afin de s'assurer que ces aspects d'intégration seront répartis dans les autres niveaux. En ce sens, c'est donc considéré comme une adaptation et une extension de l'approche MDA/MDI au contexte de l'ingénierie des services liés aux produits dans l'environnement de l'entreprise virtuelle. Sur la base de MDA/MDI, la méthodologie MDSEA définit un cadre pour la modélisation de système de service autour de trois niveaux d'abstraction liés aux niveaux d'abstraction de MDA (CIM, PIM, et PSM) : modèle de services aux entreprises (BSM), modèle indépendant d'une technologie (TIM) et modèle spécifique à une technologie (TSM) tel que présenté dans la Figure 1.

La Figure 1 montre que l'intérêt de cette architecture est d'une part de concevoir et de développer un produit de service et d'autre part de produire le système de service dédié et cohérent avec les modèles de service aux entreprises (représenté en utilisant des modèles d'entreprise). En examinant les niveaux TIM et TSM, nous pouvons observer comment la méthodologie se différencie entre les trois types de ressources classées en informatique, moyens humains ou moyens matériels. La raison d'une telle catégorisation est de s'attaquer aux différentes exigences de ressources au niveau de la mise en œuvre du système de services. La mise en œuvre des ressources détaillées dans les modèles TSM permet la mise en œuvre de systèmes de services et de produits de service connexe à travers un nuage de services, c'est à dire un système dans lequel le fournisseur de services (une entreprise à l'intérieur du réseau) n'est pas toujours reconnu par le client qui est uniquement axé sur le service. La maintenance du service, et les activités de déclassements peuvent également être assurées par des sociétés différentes dans le réseau sans véritable connaissance par le client. Cependant, l'organisation virtuelle dédiée a également les droits de propriété sur les services fournis.

Il est important de mentionner que le système de service représenté à chaque niveau de la méthodologie MDSEA est le même système, mais avec plus ou moins de détails et en tenant compte des contraintes de mise en œuvre.

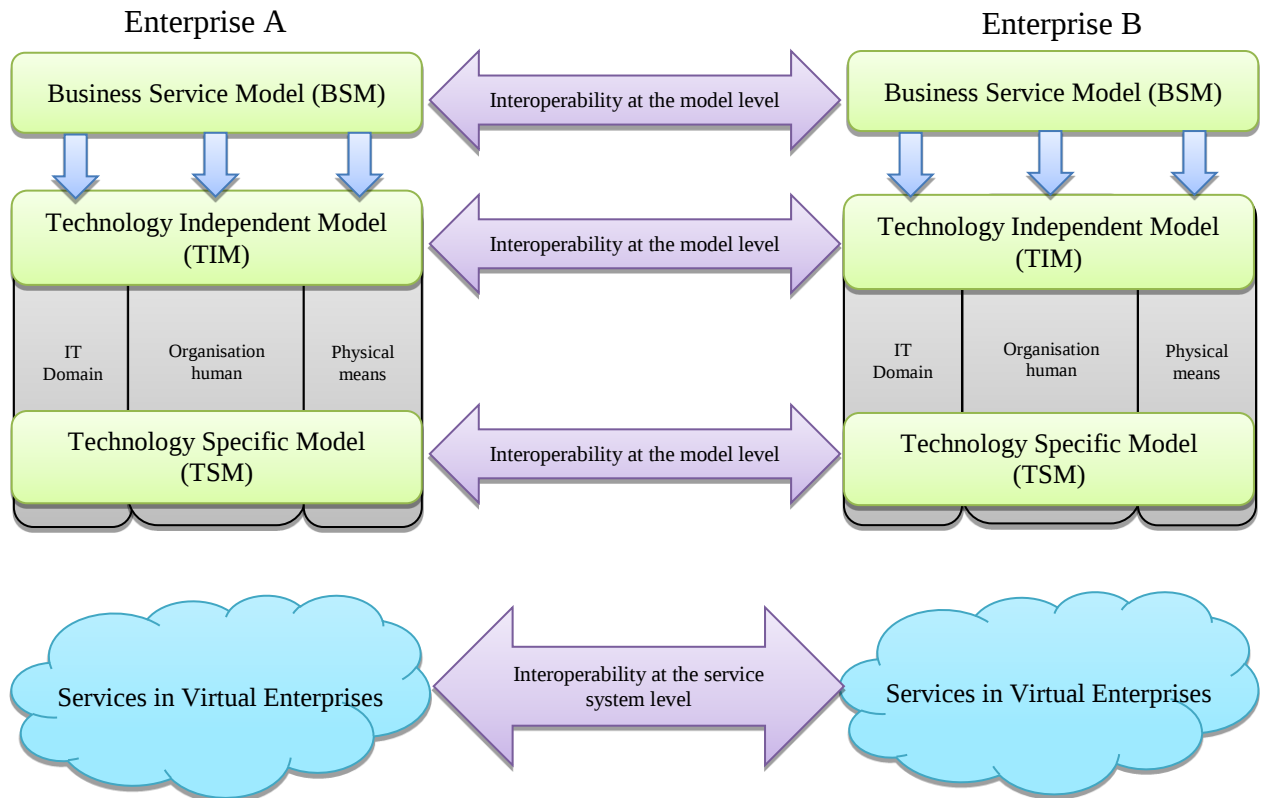


Figure 1 The MDSEA architecture applied in a service network of two enterprises

La Figure 1 montre deux entreprises collaborant ensemble afin de produire un service. La collaboration entre les différentes entités peut arriver à différents niveaux d'abstraction de la méthodologie MDSEA : BSM, TIM, ou TSM. Les modèles au niveau BSM permettent de représenter les modèles TO BE des deux entités et d'aligner les pratiques d'interopérabilité en termes de modèles de processus d'affaires et de décisions. Dans MDSEA, l'interopérabilité est un facteur clé pour la collaboration des entreprises. Les modèles d'entreprise assurent non seulement l'interopérabilité à un niveau pratique, mais aussi l'interopérabilité entre les ressources humaines et les systèmes informatiques qui soutiennent ces pratiques.

2.1 Business Service Model (BSM)

BSM spécifie les modèles, au niveau global, en décrivant le service fonctionnant à l'intérieur d'une entreprise ou d'un ensemble d'entreprises ainsi que les liens entre ces entreprises. Les modèles BSM doivent être indépendants des technologies qui seront utilisés et doit refléter le point de vue métier du système de service. En ce sens, le niveau BSM est utile, non seulement

pour comprendre et visualiser la problématique globale, mais également pour combler le fossé entre les experts métier et les experts en développement qui bâtiront le système de service. Le niveau BSM permet aussi de définir le lien entre la production de produits et la production de services.

2.2 **Technology Independent Model (TIM)**

TIM propose des modèles à un deuxième niveau d'abstraction indépendant de la technologie utilisée pour mettre en œuvre le système. Il donne des spécifications détaillées de la structure et des fonctionnalités du système de service sans donner de détails technologiques. Plus concrètement, il se concentre sur les détails opérationnels tout en cachant les détails spécifiques à toute technologie utilisée pour la mise en œuvre. Au niveau TIM, la spécification détaillée des composantes d'un système de service est élaborés par rapport à l'informatique, l'organisation et les moyens humains et physiques impliquées dans la production du service. Il est important de mentionner que, par rapport à l'approche MDA ou MDI ou SOMA (Service Oriented Modeling and Architecture), l'objectif de MDSEA n'est pas seulement orientée informatique et que cela nécessite la représentation des ressources humaines et techniques dès le niveau BSM. Au niveau TIM, les représentations doivent donc ajouter des informations par rapport aux modèles BSM.

2.3 **Technology Specific Model (TSM)**

TSM améliore les spécifications du modèle TIM avec des détails qui précise la façon dont la mise en œuvre du système va se baser une technologie en particulier (par exemple des applications informatique, ou une personne avec des compétences spécifiques). Au niveau TSM, les modèles doivent fournir suffisamment de détails pour permettre de développer ou d'acheter des logiciels appropriés, des composants matériels, recruter des opérateurs / gestionnaires humains ou établir des plans de formation interne, acheter et réaliser des machines spécifiques, afin de fournir les services aux clients. Par exemple pour les applications informatiques, un modèle TSM améliore un modèle TIM avec des détails technologiques et de mise en œuvre pour une plate-forme spécifique, ce qui inclus les middlewares, le système d'exploitation et les langages de programmation (par exemple Java, C++, EJB, CORBA, XML, Services Web, etc.). Basé sur les spécifications techniques définies au niveau TSM, l'étape suivante consiste en la réalisation et la mise en œuvre du système de service conçu en termes de composants informatiques (applications et services)

des moyens physiques (machine, composants, manutention) et des ressources humaines et organisationnelles.

3. Langages de modélisation proposées

Sur la base des niveaux de modélisation décrits précédemment, la méthodologie MDSEA propose d'associer des langages de modélisation pertinents à chaque niveau afin de représenter en toute confiance le système existant et le futur produit de service et système de service. Pour atteindre cet objectif, les normes pour la modélisation des processus gagnent de plus en plus d'importance, ce qui a donné lieu à plusieurs langages et outils de modélisation de processus pour améliorer la représentation des processus d'entreprise. Le niveau d'abstraction nécessaire est important de choisir le langage de modélisation approprié.

Il est évident à-dire que la première étape de spécification d'un service devant être établie entre deux partenaires est cruciale. Au niveau BSM, le langage de modélisation doit être simple à utiliser, puissant et compréhensible par les utilisateurs au niveau Business. En outre, ce (ou ces) langage doit couvrir les processus et la prise de décision avec des modèles cohérents. Le choix est affecté par la capacité du langage à proposer une décomposition hiérarchique (vue globale vers vues détaillés), ce qui est particulièrement utile à ce niveau. En effet, les décideurs des entreprises ont souvent une vision globale du fonctionnement du système et ont besoin d'un langage permettant de représenter cette vision globale avec seulement quelques activités de haut niveau (processus et décisions). Cette vision globale doit être complétée par des modèles plus détaillés des activités élaborées par les responsables de chaque secteur. Ces modèles sont reliés aux modèles de haut niveau de manière hiérarchique et inclusive. Ces principes de la théorie des systèmes doivent être pris en compte dans le choix des langages.

Mais il est également évident que le choix du langage de modélisation est aussi subjectif, en fonction de l'expérience des utilisateurs et de la diffusion de ces langages dans les entreprises.

En ce qui concerne la modélisation des processus au niveau des entreprises, il existe plusieurs langages. Extended Actigrams Star (EA*), une extension de GRAI extended Actigram [4] - qui a été lui-même dérivé de IDEF0 [5] – a été choisi pour modéliser les processus au niveau BSM en raison de son indépendance concernant les détails de mise en œuvre technique, sa décomposition hiérarchique et le fait qu'il est possible de modéliser les trois types de ressources : matérielles, humaines et IT. Le langage EA* a été développé en réponse aux

problèmes rencontrés avec le langage GRAI extended actigram concernant l'interopérabilité. EA* vise à capturer des modèles de processus d'affaires à un niveau sémantique élevé, indépendamment de toutes les spécifications technologiques. Service Oriented Modeling and Architecture principles [6] développé par IBM a également été examiné, mais les langages qu'il contient sont plus orientés IT et donc étaient loin de nos exigences. Par ailleurs, la grille GRAI [7] a été sélectionnée pour modéliser la gouvernance dans un système de service. La grille GRAI vise à proposer une cartographie des décisions de l'entreprise qui contrôle les processus d'affaires, tel que proposé par exemple dans la norme ISO 9000-2008. L'intérêt de la grille GRAI est de représenter toutes les décisions et leur coordination, du niveau stratégique au niveau opérationnel. Cette représentation est très importante pour les utilisateurs au niveau Business parce que les résultats de la prise de décision sont aussi à l'origine de l'évolution des performances et de la réalisation.

Au niveau TIM, BPMN 2.0 [8] a été choisi notamment parce que ce langage propose un large éventail de construction de modèle détaillée, y compris au niveau informatique. BPMN offre également l'avantage d'être interopérable avec de nombreuses plates-formes informatiques de BPM permettant le déploiement, la transformation automatique et l'exécution des processus BPMN. De plus, BPMN permet également de représenter les ressources humaines et techniques qui sont nécessaires dans MDSEA. BPMN présente aussi l'avantage de fournir un méta-modèle développé par OMG ce qui facilite la mise en œuvre du langage. Les réseaux GRAI sont proposés afin de détailler les processus de décision en cohérence avec les décisions identifiées dans GRAI Grid mais avec l'ajout d'informations techniques et d'organisation tel que les règles de décision, les décideurs, et les modules de support à la décision.

3.1 Extended Actigram Star (EA*)

Extended Actigram Star est un langage de modélisation de processus inspiré de :

- GRAI Extended Actigram: langage de modélisation de processus créé par les deux groupes de recherche: GRAI et GRAISOFT, à Bordeaux. Il n'offre pas un méta-modèle informatique standard ce qui rendra compliqué sa mise en œuvre.
- IDEF3: méthode destinée à comprendre la façon dont fonctionne un système particulier.

Extended Actigram Star est une proposition d'une nouvelle version, plus développée de GRAI Extended Actigram. Extended Actigram Star est basé sur une stratégie de développement spécifique:

- Garder les principaux concepts de GRAI Extended Actigram
- Ajouter de nouveaux concepts afin de favoriser l'abstraction et le développement de modèle informatique. Ces concepts vont ajouter des niveaux d'abstraction et ainsi fournir à l'utilisateur la possibilité de zoomer sur un modèle en fonction du type de détails requis. Le nouveau méta-modèle sera ainsi plus facile à mettre en œuvre dans un outil de modélisation.
- Modifier certaines relations entre les concepts. Dans Extended Actigram Star les relations entre les flux et d'autres concepts sont modifiés de façon à mettre en évidence le rôle d'un flux en tant qu'élément de liaison dans le langage.

Extended Actigram Star facilite la modélisation des processus d'affaires dans une entreprise offrant une vue dynamique du processus en cours de modélisation. En tant que langage de modélisation graphique, Extended Actigram Star fournit aux utilisateurs des normes pour visualiser les processus d'affaires dans l'entreprise, ce qui rend facile et compréhensive la manipulation de ces processus.

rôle des activités structurelles (étendu) et terminal (non étendu) est de faciliter le choix de zoom avant et arrière au sein d'un processus.

Une ressource peut être de trois types:

- Humaine
- Matériel
- IT

Ces trois types de ressources sont proposés afin d'adapter notre approche à l'architecture MDSE, qui différencie les ressources en fonction de ces types.

Un opérateur logique peut être de quatre types:

- Divergent ET
- Divergent OU
- Convergent ET
- Convergent OU

Les notions de synchrone et asynchrone ne sont pas mentionnées dans ce langage, car il a été constaté qu'elles ne couvrent pas de situation concrète dans la modélisation des processus.

Un connecteur peut être de trois types:

- Interne
- Externe
- Processus

3.2 BPMN

L'Object Management Group (OMG) a développé le standard Business Process Model and Notation (BPMN). L'objectif principal de BPMN est de fournir une notation qui est facilement compréhensible par tous les utilisateurs, des analystes qui créent les premières ébauches des processus, aux développeurs techniques chargés de la mise en œuvre de la technologie qui va exécuter ces processus, et jusqu'aux utilisateurs au niveau Business qui vont gérer et surveiller ces processus. Ainsi, BPMN crée un pont normalisé entre la conception de processus d'affaires et la mise en œuvre de processus (OMG 2011a).

BPMN représente le flux d'un processus de gestion de bout en bout. La notation a été spécialement conçue pour coordonner la séquence des processus et les messages qui circulent entre les différents participants au processus dans un ensemble d'activités connexe.

4. Transformation de modèle

MDSEA débute lors de la mise en place stratégique d'entreprises qui veulent évoluer vers des méthodes commerciales orientées vers le service. Le modèle Business spécifiques « To be » est spécifié et développé au niveau BSM. Plus tard, le modèle de définition fonctionnel détaillé est développé au niveau TIM, et un modèle de mise en œuvre pratique au niveau TSM. Le passage d'un niveau à un niveau inférieur devrait se faire en réutilisant et en enrichissant le modèle de plus haut niveau. Un problème majeur qui est fréquemment identifié dans les entreprises est l'écart entre les visions des protagonistes pour décrire un processus. Le processus peut être défini pour créer des produits physiques ou des services. Il oppose d'un côté la vue Business et de l'autre la vue technique. Des efforts visant à réduire cette opposition ont été fournis. Les modèles au niveau BSM varient d'un modèle au niveau TIM en raison des différents langages de modélisation utilisés dans chacun de ces niveaux. Les processus d'affaires au niveau BSM sont modélisés en utilisant le langage Extended Actigram Star alors qu'au niveau TIM les processus d'affaires sont modélisés en utilisant la notation BPMN. Ces deux langages de modélisation différents sont basés sur différents méta modèles, et donc différentes constructions spécifiques. Cela soulève l'importance de transformer les modèles BSM vers des modèles TIM et les modèles TIM vers des modèles TSM.

4.1 Approche méta-modèle

L'objectif est de transformer les modèles sources BSM en modèles cibles TIM. L'une des techniques de transformation les plus utilisées est « Metamodel Approach » [9]. La Figure 3 adapte l'approche «méta-modèle» au contexte de transformation de modèles sources EA* vers des modèles cibles BPMN2.0.

L'étape 1 de la transformation - et un prérequis obligatoire - est la formalisation des méta-modèles source et cible (respectivement EA* et BPMN2.0). Ecore (qui fait partie d'Eclipse EMF) est une implémentation d'une forme simplifiée de MOF [10] et est utilisé pour définir les méta-modèles source et cible. En outre XML Metadata Interchange (XMI) [11] est utilisé pour sauvegarder les modèles source et cible.

L'étape 2 de la méthodologie de transformation marque le début de la conception réelle des transformations en définissant les mappages de modèles qui relient les concepts de chaque méta-modèle. D'un point de vue syntaxique, le mappage est un morphisme qui doit assurer la cohérence des modèles source et cible, et est créé en reliant chaque élément de la source avec un élément correspondant dans la cible (1-1, 1-n, ou m-n). Lors de la transformation, le modèle source est transformé en utilisant une fonction qui applique un mappage pour produire un modèle cible différent. Cette fonction peut être exprimée soit explicitement, en utilisant des graphiques, des ensembles, des tuples ou des tables; ou implicitement dans la tête du développeur. Cependant, dans les deux cas il est nécessaire de les mettre en œuvre en utilisant un langage de transformation (étape 3).

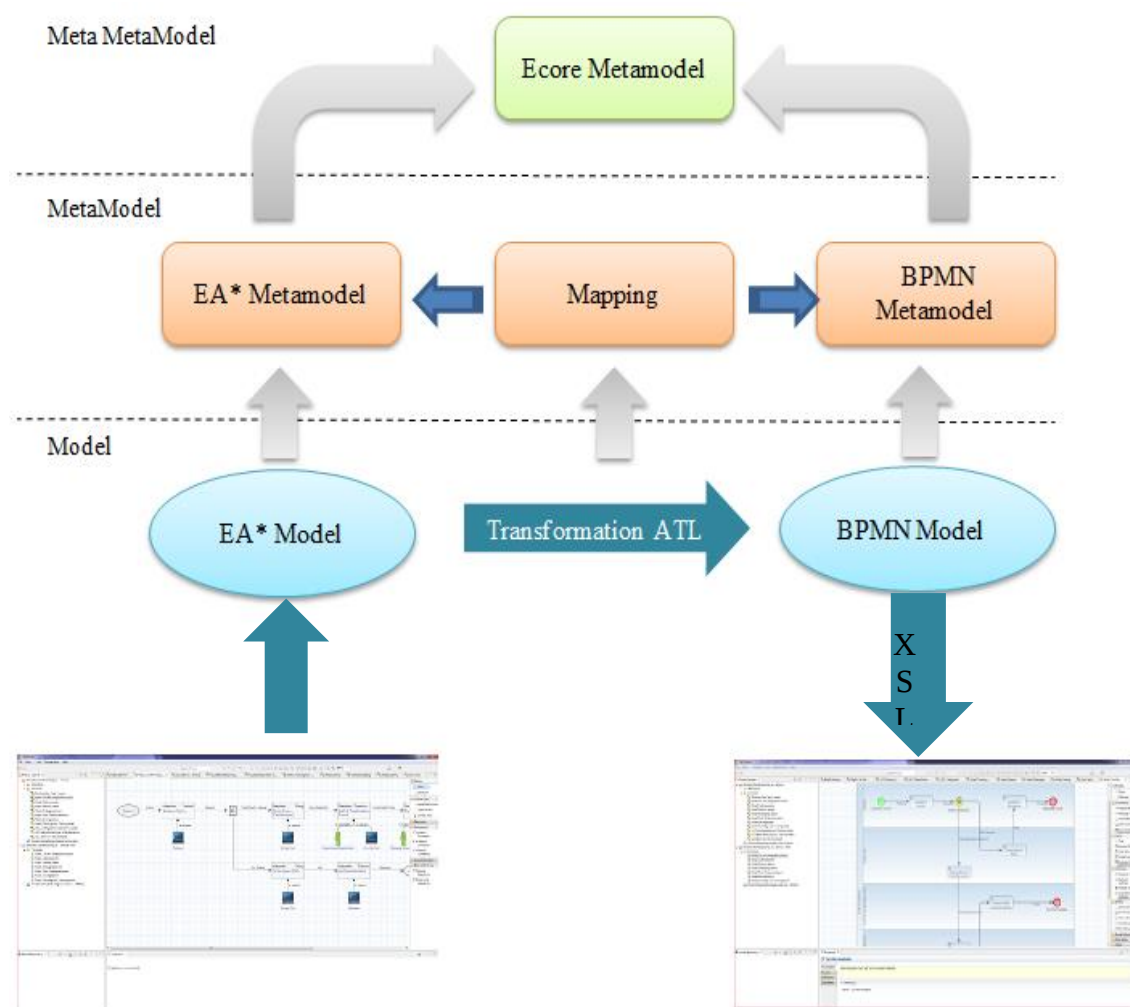


Figure 3 Transformation architecture of EA* to BPMN

La transformation de modèle est une activité importante en ingénierie dirigé par les modèles. L'OMG a reconnu cela en faisant un appel à contributions autour de Query/Views/Transformations (QVT) afin de trouver une réponse compatible avec la norme MDA. De nombreuses contributions ont été présentées, ce qui a conduit à plusieurs langages de transformation permettant la transformation automatique de modèles. Certaines d'entre elles sont basées sur l'Object Constraint Language (OCL), comme QVT lui-même et ATL, qui bien que n'étant pas une norme est l'un des plus utilisés, ayant une grande base utilisateur. Néanmoins, comme énumérés dans [12], d'autres peuvent également être utilisés et appliqués à la mise en œuvre des mappages de modèle, par exemple, Xtend / Xpand, UMLX, AToM3, MTL, etc. Comme analysé par [13], certains des langages ci-dessus sont idéales pour la représentation de mappage structurelles, d'autres pour le mappage sémantiques, offrant une bonne traçabilité humaine, tandis que d'autres sont plus formelle. Cependant, aucun ne donne la possibilité de traduire les mappages explicites en code exécutable. Les mappages mis en œuvre avec ces langages sont normalement statique et tout changement oblige à réécrire manuellement le code exécutable. Bénéficiant d'une bonne intégration JAVA, qui devrait permettre de résoudre les problèmes ci-dessus à l'avenir et d'avoir une quantité considérable d'aide et de ressources en ligne, ATL a été le langage choisi pour la mise en œuvre des mappages de MDSEA.

4.2 Transformer des modèles EA* en modèles BPMN

4.2.1 Diagramme de collaboration EA* vers BPMN – Mappage

Le tableau suivant résume le mappage des concepts EA* vers les concepts BPMN. Le mappage est assorti de conditions qui régissent la création des relations.

EA*	Condition	BPMN2.0
Model		Definitions
Process		Process, Participant
Extended Activity	Structural	Sub Process
	Atomic & supported by Human	UserTask
	Atomic & supported by IT (no human interaction)	ServiceTask
DivergingOr		Diverging Exclusive Gateway
ConvergingOr		Converging Exclusive Gateway
DivergingAnd		Parallel Gateway

ConvergingAnd		Parallel Gateway
MaterialResource	Material	Data Object
HumanResource	Responsible for	Lane
	Participates in	Resource (added to the list of resources of a task)
ITResource	Responsible for	Lane
	Participates in	Resource (added to the list of resources of a task)
Control Flow	If the source is an ExternalConnector or InternalConnector and target is an “atomic” ExtendedActivity	MessageFlow
	If the source is an ExternalConnector or InternalConnector and target is a “structural” ExtendedActivity	Catching Message Event, Message flow, and Sequence Flow
	If the source is a ProcessConnector or ExtendedActivity	DataObject, and associations
OutputInputFlow	If the source is an ExternalConnector or InternalConnector (and target is an atomic Extended Activity)	MessageFlow
	If the source is an ExternalConnector or InternalConnector (and target is a structural Extended Activity or LogicalOperator)	Catching Message Event, Message Flow, and Sequence Flow
	If the source is a ProcessConnector, ExtendedActivity, or LogicalOperator (and target is an ExtendedActivity or process connector or logical operator)	SequenceFlow
	If the source is a structural ExtendedActivity or logical operator (and target is an ExternalConnector or InternalConnector)	Throwing Message Event, Message Flow, Sequence Flow
	If the source is an atomic ExtendedActivity (and target is an External or Internal Connector)	MessageFlow
SupportFlow	If source is a material resource	Association
ExternalConnector		Participant (Pool)
ProcessConnector		Call Activity
InternalConnector		Participant (Pool) (Black Box)

Table 1 - EA* vers BPMN - Mappage (diagramme de collaboration)

4.2.2 Diagramme de collaboration EA* vers BPMN – Règles de transformation

ExtendedActivity atomique

- Une ressource humaine est responsable de la réalisation de l’Extended Activity. Dans ce cas, l'Extended Activity atomique est mappé à un UserTask.
- Une ressource informatique est responsable de la réalisation de l'Extended Activity. Dans ce cas, l'Extended Activity atomique est mappé à un ServiceTask.

Ressource

- La valeur de resourceRole est “responsible for”. Dans ce cas, la ressource (humaine ou IT) est mappée à une ligne, dans laquelle l’ExtendedActivity appartient à la ligne.
- La valeur de resourceRole est “participates in”. Dans ce cas, la ressource (humaine ou IT) est ajoutée à la liste des ressources de l’ExtendedActivity.

ControlFlow

- La source est un ExternalConnector ou InternalConnector et la cible est un ExtendedActivity « atomique ». Dans ce cas, il est mappé à un MessageFlow.
- La source est un ExternalConnector ou InternalConnector et la cible est un ExtendedActivity « structurelle ». C’est une relation "1 à n", dans lequel le "Control" de flux est mappé à une combinaison de MessageFlow, qui attrape des MessageEvent, et de SequenceFlow.
- La source est un ProcessConnector ou ExtendedActivity. C’est une relation "1 à n", dans lequel le "Control" de flux est mappé à une combinaison de DataObject et deux Associations.

OutputInputFlow

- La source est un ExternalConnector ou InternalConnector et la cible est une Extended Activity atomique. Dans ce cas, le flux "OutputInput" est mappé à un MessageFlow.
- La source est un ExternalConnector ou InternalConnector et la cible est une Extended Activity structurelle ou un LogicalOperator. C’est une relation "1 à n", dans lequel le "Control" de flux est mappé à une combinaison de MessageFlow, qui attrape des MessageEvent, et à un SequenceFlow.
- La source et la cible sont des ProcessConnector, ExtendedActivity ou LogicalOperator. Dans ce cas, il est mappé à un SequenceFlow.
- La source est une ExtendedActivity structurelle ou un LogicalOperator, et la cible est un ExternalConnector ou InternalConnector. C’est une relation "1 à n", dans lequel le "Control" de flux est mappé à une combinaison de MessageFlow, qui libère des MessageEvent, et à un SequenceFlow.
- La source est une ExtendedActivity atomique et la cible est un ExternalConnector ou InternalConnector. Dans ce cas, il est mappé à un MessageFlow.

SupportFlow

Condition1: la source est une ressource matérielle. Dans ce cas, elle est mappée à une Association.

4.2.3 Diagramme de processus de EA* vers BPMN – Mappage de modèle

Le tableau suivant résume le mappage des concepts EA* vers les concepts BPMN. Le mappage est assorti de conditions qui régissent la création des relations.

EA*	Condition	BPMN2.0
Model		Definitions
Process		Process
Extended Activity	Structural	Sub Process
	Atomic & supported by Human	UserTask
	Atomic & supported by IT (no human interaction)	ServiceTask
DivergingOr		Diverging Exclusive Gateway
ConvergingOr		Converging Exclusive Gateway
DivergingAnd		Parallel Gateway
ConvergingAnd		Parallel Gateway
MaterialResource	Material	Data Object
HumanResource	Responsible for	Lane
	Participates in	Resource (added to the list of resources of a task)
ITResource	Responsible for	Lane
	Participates in	Resource (added to the list of resources of a task)
Control Flow	If the source is a ProcessConnector or ExtendedActivity	DataObject, and associations
OutputInputFlow	If the source is a ProcessConnector, ExtendedActivity, or LogicalOperator (and target is an ExtendedActivity or process connector or logical operator)	SequenceFlow
SupportFlow	If source is a material resource	Association
ExternalConnector		Not mapped
ProcessConnector		Call Activity
InternalConnector		Not mapped

Table 2 - EA* vers BPMN - Mappage (diagramme de processus)

4.2.4 Diagramme de processus de EA* vers BPMN – Règles de transformation

Process

- Un processus EA* est mappé à un processus BPMN2.0, ce processus n'est pas représenté graphiquement par un pool (contrairement au diagramme de collaboration de BPMN2.0)

ControlFlow

- Si la source est un ProcessConnector ou un ExtendedActivity (atomique ou structurel), elle est associée à un flux de séquence.
- Si la condition1 n'est pas applicable alors un ControlFlow n'est pas mappé.

OutputInputFlow

- La source et la cible sont des ProcessConnector, ExtendedActivity ou LogicalOperator. Dans ce cas, elle est mappée à un SequenceFlow.
- Si la condition1 n'est pas applicable alors un OutputInputFlow n'est pas mappé.

SupportFlow

- Si la source est une ressource matérielle, alors elle est mappée à une Association
- Si la source est d'un autre type, alors elle n'est pas mappée.

Connector

Si le connecteur est un ProcessConnector alors il est mappé à un CallActivity. Tout autre type de connecteurs n'est pas mappé.

5. Conclusions

Le contenu de ce livrable permet de définir les spécifications des connecteurs MDI au travers des 3 niveaux de modélisation de MDSEA afin de pouvoir les appliquer au domaine du transport. Cela permet de définir au niveau BSM le modèle conceptuel nécessaire pour rendre deux systèmes de transport interopérables. Ce modèle conceptuel peut être transformé en un système d'information au niveau TIM grâce à la transformation de modèle basé sur l'approche méta-modèle.

D'un point de vue conceptuel et technique, le contenu de ce livrable a repris des résultats proposés par le projet Européen MSEE et montre le développement réalisé pour passer du niveau BSM au niveau TIM en appliquant l'approche méta-modèle proposé permettant la transformation de modèle EA* en modèle BPMN2.0

Il faut noter qu'il est nécessaire pour générer l'application informatique au niveau TSM de pouvoir utiliser des outils permettant de générer le code informatique adapté aux cas d'application.

Références

- [1] OMG, «Documents Associated With Business Process Model And Notation (BPMN) Version 2.0,» OMG, January 2011. [En ligne]. Available: <http://www.omg.org/spec/BPMN/2.0>. [Accès le 29 Aout 2014].
- [2] INTEROP (2005), 'Report on Model Establishment', Task Group 2. 'Model Driven Interoperability' of INTEROP NoE.
- [3] Y. Ducq, D. Chen et G. Doumeingts, «MSEE project, D 11.1, Services Modelling for SLM Project»,» Available online from MSEE Website <http://msee-ip.eu/>, July 2014.
- [4] Grangel, Seguer R., Cutting-Decelle A.F., and Bourey J.P. 2008. "A UML profile for transforming GRAI Extended Actigrams into UML." In Proceedings of the 17th IFAC World Congress, Session: Architectures and Software Tools for Enterprise Integration and Networking in Manufacturing, Paper ThB24.3, Séoul, Korea.

- [5] National Institute of Standards and Technology (NIST) 1993. “IDEF, Integrated DEfinition Methods” Accessed May 15 2013. <http://www.idef.com/IDEF0.htm>
- [6] Bell, M. 2008. Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley.
- [7] Doumeingts, G., Vallespir B., and Chen D. 1998. “GRAI grid, decisional modeling” In International Handbook on Architecture of Information System. Bernus P., Mertins K., Schmith G. eds 1998. Springer Verlag. 313-337.
- [8] OMG. 2011. “Business Process Model and Notation (BPMN) version 2.0” document number: formal/2011-01-03.
- [9] Bourey J-P, Grangel R, Doumeingts G, Berre A (2007), “Deliverable DTG 2.3 - Report On Model Driven Interoperability”, InterOP 2007 (report available at: http://interopvlab.eu/ei_public_deliverables/interop-noe-deliverables/tg2-model-driven
- [10] Object Management Group, Meta-object facility 2.0 core specification, <http://www.omg.org/spec/MOF/2.0/>, 2006.
- [11] OMG. 2000. “XML Metadata Interchange (XMI)”, document number: ad/2001-06-12
- [12] Czarnecki, K. & Helsen, S. (2006) Feature-based survey of model transformation approaches. IBM Systems Journal, 45 (3), p.pp.621-645. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5386627>
- [13] Agostinho, C., Sarraipa, J., Gonçalves, D., & Jardim-Goncalves, R. (2011). Tuple-based semantic and structural mapping for a sustainable interoperability. DOCEIS'11. Costa de Caparica.
