HAL
open science

Modélisation centrée sur les processus métier pour la génération complète de portails collaboratifs

Amen Souissi, Pierre Boulet, Cedric Dumoulin, Michael Launay

To cite this version:

Amen Souissi, Pierre Boulet, Cedric Dumoulin, Michael Launay. Modélisation centrée sur les processus métier pour la génération complète de portails collaboratifs. [Rapport de recherche] 2011. inria-00638298

HAL Id: inria-00638298

https://inria.hal.science/inria-00638298

Submitted on 4 Nov 2011

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.

Modélisation centrée sur les processus métier pour la génération complète de portails collaboratifs

Amen Souissi* — **Pierre Boulet**[1]** — **Cedric Dumoulin**** — **Michael Launay***

** Ecréall*
Parc scientifique de la Haute Borne, 11 rue de l'Harmonie 59650 Villeneuve d'Ascq
{amensouissi, michaellaunay}@ecreall.com

*** Univ. Lille 1, LIFL*
Cité scientifique - Bâtiment M3 59655 Villeneuve d'Ascq Cédex
{pierre.boulet, cedric.dumoulin}@lifl.fr

RÉSUMÉ. *Les entreprises ont de plus en plus besoin de collaborer suivant des processus métier. Les portails collaboratifs apportent une solution orienté web à ce besoin de collaboration. Cependant, leur conception et leur maintenance n'est pas trivial. La société Ecréall propose de concevoir, générer et maintenir complètement des portails collaboratifs à partir d'une modélisation centré sur les processus métier, et en suivant une approche IDM. Cette article présente la solution proposée par Ecréall à travers son métamodèle MACoP et sa chaîne de transformation*

ABSTRACT.

MOTS-CLÉS : *IDM, métamodélisation, modélisation,transformation de modèles, génération de code, portail collaboratif*

KEYWORDS:

1. This work was partially supported by the HiPEAC european Network of Excellence

1. Introduction

La collaboration est une nécessité pour l'entreprise, que ce soit en interne, entre ses collaborateurs, ou en externe, avec d'autres entreprises ou avec des clients. Une collaboration se concrétise généralement par le partage de documents, de ressources, de services ou de connaissances. Ce partage se fait en suivant des règles métiers indiquant par exemple quel collaborateur peut créer ou manipuler le document ou la ressource, à quel moment, et enfin quelles étapes doivent être suivies afin d'aboutir au produit final.

L'informatique et les technologies de l'internet permettent de répondre de manière optimale à ce besoin de collaboration, sous la forme de portail collaboratif. Cet outil informatique, généralement constitué d'un serveur web, réalise l'interface entre les différents systèmes d'informations impliqués dans les collaborations. Ce portail permet à un acteur donné d'agir sur les informations d'une collaboration en suivant quatre grands périmètres fonctionnels qui sont la gestion de la connaissance, l'espace collaboratif métier, la plate-forme communautaire et la publication.

La société Ecréall[1] est spécialisée dans la conception, le développement et la maintenance de portails collaboratifs sur mesure. Forte de son expérience, Ecréall cherche à automatiser la conception et la génération de portail spécifique du métier de ses clients. Pour cela, elle a choisi une démarche Ingénierie Dirigée par les Modèles (IDM)(Favre, 2006)(IDM, n.d.), permettant de modéliser à un haut niveau d'abstraction les besoins de collaborations, puis de générer automatiquement le portail collaboratif opérationnel. Aucune intervention n'est nécessaire sur le code généré, d'ailleurs les modifications de ce code sont proscrites : elles doivent passer par le modèle de haut niveau.

Pour une entreprise, le portail collaboratif évolue suivant son système d'information et les nouvelles technologies. Cette évolution doit être rapide, efficace et à moindre coût. Toutefois ceci n'est possible que si l'entreprise est structurée, c'est a dire si elle a formalisé sa façon de travailler. Les formes de structuration les plus matures reposent sur la modélisation des entreprises. Convaincus que l'entreprise doit avoir le plus d'autonomie possible sur l'évolution de son système d'information, étant donnée qu'elle est la seule à le connaître précisément, nous avons choisi de privilégier la modélisation des entreprises « centrée processus métier » comme point de départ pour la modélisation de leur portail collaboratif.

Plusieurs outils et méthodes existent aujourd'hui permettant la modélisation et la génération des applications web en adoptant des démarches différentes avec des domaines allant de l'orchestration des processus à l'IHM. Néanmoins, aucun de ces outils ne permet de générer le portail collaboratif complet, opérationnel et adapté au métier de l'entreprise. À un niveau plus abstrait, l'étude de ces outils nous a permis d'identifier certains besoins et critères, au niveau de la méta-modélisation. Le respect

1. Ecréall est une société ayant six ans et composée de cinq collaborateurs, dont un chercheur CIFRE

de ces critères nous a conduits in fine au métamodèle MACoP (Modeling and Analysis of Collaborative Portal) accessible, compréhensible et pertinent.

Notre solution se distingue par le fait qu'elle s'appuie sur une modélisation des entreprises centrées processus métiers comme modèle de haut niveau afin de générer un portail collaboratif. Cette modélisation se concentre uniquement sur le métier, et ne contient aucun artefact technique en relation avec la cible technique visée. Cette modélisation est divisée en « points de vues » concentrés chacun sur un problème particulier. Dans cette modélisation nous essayons de répondre aux critères que nous avons identifiés en nous appuyant sur des techniques comme l'imbrication des processus métier. Le métamodèle MACoP n'est donc pas destiné uniquement aux informaticiens, mais aussi, et surtout, aux non-informaticiens, plus précisément aux différents acteurs de l'entreprise.

Dans cet article, nous commençons par nous positionner par rapport aux travaux relatifs au domaine de la modélisation des entreprises et des systèmes d'information en prenant la modélisation des entreprises comme base pour la modélisation des systèmes d'information. Ensuite nous résumons d'une manière générale l'étude que nous avons faite des outils et des méthodes de modélisation et de génération de code pour les applications web. Suit une synthèse des problématiques tirées de cette étude, et une proposition de critères que doit remplir notre proposition. Ensuite nous détaillons notre démarche de conception des portails collaboratifs basés sur le métamodèle MACoP que nous proposons. Après nous validons notre approche à travers l'exemple « Tracker » de gestion de projet de développement. Enfin, nous évaluons notre approche par rapport aux critères que nous avons proposés, et enfin nous concluons.

2. Modélisation des entreprises comme base pour la modélisation des systèmes d'information collaboratifs

Ecréall se sert depuis longtemps de la capture des processus métier de l'entreprise afin de concevoir des portails collaboratifs. Les recherches que nous avons effectué nous ont permis de trouver une démarche allant de la modélisation des entreprises au portail collaboratif, en passant par la modélisation des systèmes d'information collaboratifs. Pour bien comprendre cette démarche, nous commençons par donner une définition du système d'information collaboratif, correspondant à notre axe de recherche. Cela nous permet de positionner le portail collaboratif par rapport à l'entreprise et clarifie la relation entre portail collaboratif et système d'information. Ensuite, nous discutons de la modélisation des entreprises, base de notre démarche, et de la relation entre systèmes d'information et modélisation des entreprises.

2.1. *Systèmes d'Informations Collaboratifs*

Parmi les définitions existantes, nous avons retenu celle de (Morley *et al.*, 2006) (voire figure 1) dans laquelle le système d'information est mis en interaction avec le système informatique au sein d'une entreprise. Dans cette définition, le système d'in-

formation d'une entreprise est la partie du réel constituée d'informations organisées, d'événements ayant un effet sur ces informations, et d'acteurs qui agissent sur ces informations ou à partir de ces informations, selon des processus visant une finalité de gestion et utilisant les technologies de l'information (les processus métiers de l'entreprise). D'après la même source, le système informatique est un ensemble organisé d'objets techniques (matériels, logiciels, applicatifs...) dont la mise en œuvre réalise l'infrastructure du système d'information et lui permet de fonctionner.

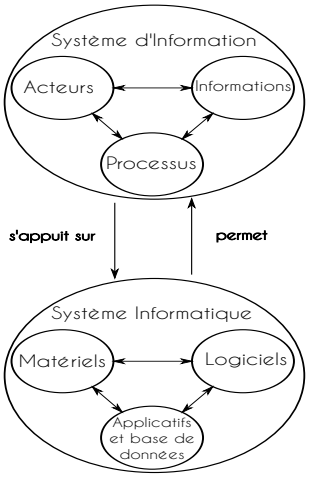

Figure 1. *Interaction entre le système d'information et le système informatique*

Selon (Touzi, 2007) un système d'information collaboratif (SIC) externe est un système d'information qui inclut les différentes parties publiques des systèmes d'information des entreprises participantes à une collaboration donnée. De même, un SIC interne est un sous ensemble du Système d'Information de l'entreprise impliqué dans la collaboration. Par extension, un Système Informatique Collaboratif est un système Informatique permettant le fonctionnement du SIC. Le Portail Collaboratif fait partie du Système Informatique Collaboratif et il peut jouer un rôle d'interopérabilité entre les différents systèmes d'information.

D'un autre point de vue, le SIC peut être perçu comme une spécification d'un cahier des charges et le système informatique comme son livrable. Par extension, une modélisation du SIC peut servir de cahier des charges pour le portail collaboratif.

Dans notre contexte, la modélisation des entreprises est prise comme base pour la modélisation des systèmes d'information, car elle permet de concevoir le système d'information. On peut donc passer de l'espace des modèles des entreprises à l'espace des modèles des systèmes d'information (Touzi, 2007).

2.2. *Modélisation des entreprises*

L'évidence de son fonctionnement mène l'entreprise à une compréhension et une communication efficace entre les différents acteurs qui la constitue. En conséquence, l'entreprise ne pouvait que structurer son organisation autour de processus, définis aussi clairement que possible, formalisés et autour desquels les différents métiers communiquent. Cela lui permet d'avoir une vision globale et précise de son métier et ainsi d'améliorer et optimiser ses processus.

Plusieurs courants de travaux scientifiques accompagnent cette forme de structuration, et des langages de modélisation, souvent graphiques, sont proposés pour obtenir une représentation formelle. La modélisation des entreprises est certainement l'une des formes de structuration les plus matures.

Les normes permettant la modélisation des entreprises sont nombreuses, certaines sont recensées dans le travail de recherche effectué par David Chen et François Vernadat (Chen *et al.*, 2001) .

Pour nos besoins, nous avons retenu les travaux de synthèse relatifs à un langage pour la modélisation des entreprises normés ENV 40003 (ISO, n.d.) et plus précisément la notion de « point de vue de modélisation ».

Un point de vue de modélisation est une vision particulière de l'entreprise qui met en lumière certains aspects, en augmentant leur niveau de détail, et rend transparents les autres, en diminuant leur niveau de détail. C'est une perspective particulière pour décrire, puis analyser, une même entreprise au moyen du modèle (Darras, 2004). Ceci permet une communication plus efficace entre les différents acteurs de l'entreprise.

La norme ENV 40003 reconnaît quatre points de vue de modélisation :

– la vue fonctionnelle qui fournit une représentation des processus métier de l'entreprise.

– la vue informationnelle qui fournit une représentation structurée d'un ensemble d'information de l'entreprise et des relations de dépendance entre ces informations ;

– la vue des ressources qui fournit une représentation de l'ensemble des moyens nécessaires pour mettre en œuvre les activités ;

– la vue de l'organisation qui fournit une représentation de l'organisation structurelle de l'entreprise.

Parmi les cadres de modélisation d'entreprise adoptant ces vues, nous citons CIMOSA (CIMOSA, n.d.)"Computer Integrated Manufacturing Open System Architecture" qui est un framework de modélisation des entreprises et vise à connecter et combiner les machines, les ordinateurs et les personnes au sein de l'entreprise. UEML (Anaya *et al.*, 2010) est le fruit d'un programme de collaboration à l'échelle européenne et a pour objectif de définir une interface standardisée entre les outils de la modélisation d'entreprise basés sur des modèles différents. Ce qui a conduit in fine à un certain degré de compatibilité entre les outils.

Nous venons de voir que le portail collaboratif fait partie du Système Informatique Collaboratif qui, de son coté, permet le fonctionnement du Systeme d'Information Collaboratif. Nous avons vu aussi que la modélisation des entreprises permet de concevoir le Système d'Information, qui lui, englobe le Systeme d'Information Collaboratif. Il existe donc un chemin permettant de passer de la modélisation des entreprises au portail collaboratif. Nous avons choisit d'exploiter ce chemin, et nous proposons de générer le portail collaboratif à partir de la modélisation des processus de l'entreprise. Cette modélisation s'inspire des travaux faits dans ce domaine qui adopte les notions de points vues de modélisation de la norme ENV 40003.

3. État de l'art et analyse

L'Entreprise 2.0 est un terme inventé par (McAfee, 2006) de Harvard Business School afin de désigner les entreprises qui collaborent précisément et rapidement, en utilisant les outils Web 2.0. Ces entreprises recherchent une communication fluide, un rapprochement des collaborateurs et un partage efficace des ressources, des services et des connaissances à travers l'utilisation de portails collaboratifs.

Un portail collaboratif est avant tout une application web. Nous avons commencé par étudier plusieurs méthodes et outils existants de modélisation et génération de code pour ce type d'applications. Cette étude nous a permis d'identifier plusieurs problématiques et nous a amené à énoncer differents critères qui nous serviront à évaluer notre démarche.

3.1. *Modélisation d'application WEB*

La conception et la modélisation des applications Web ont toujours été abordés sous plusieurs aspects ayant été mis en œuvre par l'utilisation d'une méthode et des modèles décrivant chacun une vue spécifique à l'application. Ces méthodes, comme WebML, OOHDM ou UWE (da K ADRI, 2009)(Rossi *et al.*, 2007), se préoccupent toute d'avantage de la mécanique de l'application que du métier des utilisateurs, et aucune ne part de la modélisation des entreprises centrée processus métier. En majorité, elles utilisent un métamodèle de description du métier cible de l'application sous forme de modèle de navigation entre les pages de l'application. Nous citons dans cette catégorie UML-based Web Engineering (UWE)(Kroiß*et al.*, 2008) qui génère une application web partielle depuis le modèle, en utilisant une chaîne de transformations de modèles. Le métamodèle UWE est une extension conservatrice du métamodèle UML 2.0. Ce qui signifie que les concepts du métamodèle UML ne sont pas modifiés. UWE permet une spécialisation d'UML pour les applications web en passant par le fond et la forme. Un autre exemple est le projet BluAge (BARBIER, n.d.) qui adopte une approche MDA pour générer le code des applications web à base de service depuis un modèle UML, profilé et structuré selon une structure spécifique. Dans ce projet, la partie visuelle est décrite par des templates en xHTML connectés au modèle métier.

D'autres outils existants permette la modélisation et la génération de code ou d'interprétation du modèle pour des applications web centrées processus métier. La plupart de ces outils est destinée à l'orchestration de ces processus, permettant, ainsi, de les exécuter à travers une application web fondée sur un moteur d'exécution de processus métier. Nous citons par exemple WebRatio (WebRatio, n.d.), Intalio et BonitaSoft (Fischer, 2010), qui se distinguent par leur maturité sur le marché. BonitaSoft est un outil Open-source qui propose un éditeur de processus métier respectant le standard BPMN2.0 (Silver, 2009)(Allweyer, 2010)(OMG, 2011). Cet outil propose un moteur d'exécution permettant d'exécuter les processus de l'entreprise, modélisés par l'utilisateur, à travers une application web proposée par l'outil. Qu'il aura connecté à son système d'information via les connecteurs proposés par l'outil.

Il existe d'autres outils ayant adopté une approche IDM et spécifiques à une technologie donnée, par exemple ArchGenXML (Pelletier, 2005) qui est une application permettant de générer des modules Plone (Knox *et al.*, 2009)à partir d'un modèle décrivant le plan documentaire et le diagramme d'états représentant le workflow documentaire de chaque type de contenu.

D'autres outils de génération de code pour des applications web prennent comme modèle d'entrée un modèle UML (avec ou sans Profile). Ces outils considèrent le métamodèle comme une notion implicite. Ce qui rend cette approche difficile à appréhender puisque UML n'est pas dédié à un métier donné. Je cite par exemple le travail fait dans (Roques, 2007).

Une des approches les plus répandues est celle de la génération de code pour des applications web à partir des modèles décrivant les IHM. Dans ce domaine la plupart des approches mettent en avant la partie visuelle et les interactions. Le noyau fonctionnel (NF) y est décrit d'une manière implicite et adapté aux besoins visuels. Dans cette approche la description du NF est souvent basée sur un modèle de tâches hiérarchiques simples ne permettant pas de décrire des processus métier et des workflows complexes. D'autres remplacent le modèle de tâche par un modèle décrivant les processus métier associé à la dynamique et les interactions de l'application comme, par exemple, dans (Brossard *et al.*, 2007). Dans le domaine de la malléabilité des IHM par exemple, dans la thèse de (Sottet, 2008) un travail bibliographique sur les outils de génération de code/interprétation de modèle pour les IHM a été fait et abordé largement.

Nous avons constaté que les modèles issus de l'application de ces méthodes ou orienté par les contraintes des outils associés sont souvent difficiles à comprendre. Nous allons montrer les critères que nous avons identifié pour améliorer la compréhension des modèles.

3.2. *Séparation des préoccupations*

Les préoccupations sont les différents aspects d'un système. Leur séparation améliore la lisibilité et permet l'évolution des modèles en traitant chacune des préoccupations indépendamment les unes des autres.

Les questions permettant d'identifier les préoccupations d'un système sont Qui, Quoi, Où, Quand, Comment, Combien, Pourquoi ? (QQOQCCP)(wikipedia, n.d.). Comme elles sont à un niveau méta-méta, on les appellera des méta-méta-questions. En permettant de poser les bonnes méta-questions sur le système à modéliser, un modèle peut être vu comme la réponse aux méta-questions formant le métamodèle.

Les méta-méta-questions QQOQCCP peuvent être posées à plusieurs niveaux de détails. Donc un sous ensemble des questions sur un niveau peut former une question sur un niveau supérieur.

La plupart des démarches citées si dessus mélangent les préoccupations. La forme la plus fréquente de ce mélange est celle de la fusion entre les processus métier et la partie visuelle qui se présente comme un modèle de navigation des écrans de l'application, par exemple dans le cas des méthodes objet UWE et BluAge ainsi que les approches IHM en général. Assurément, cette présentation exprime le métier de l'utilisateur d'une manière non structurée et implicite. Nous pensons que le modèle de navigation est une information explicite de trop, car le modèle de navigation peut être déduit du modèle de processus métier et de celui de l'aspect visuel. Nous pouvons comparer cette vision à celle de la marionnette à fils et du marionnettiste, ou la marionnette représente l'aspect visuel de notre application et le marionnettiste représente le métier. Il suffit de regarder les mouvements du marionnettiste pour en déduire les mouvements de la marionnette. Dans ce cas de figure le modèle de navigation est considéré comme le résultat de la fusion entre le modèle métier (NF) et le modèle de l'aspect visuelle (IHM).

D'autres formes de mélange des préoccupations existent comme celui entre la vue informationnelle et la vue fonctionnelle. En effet, dans la plupart des outils se basant sur des méthodes objet, par exemple UWE ou ArchGenXML, nous remarquons une fusion entre le « Quoi ? » et le « Comment ? » qui se présente comme une modélisation des objets de l'application tout en spécifiant les méthodes de chaque objet. Nous pensons que les méthodes des objets tiennent du « Comment ? » de l'application et peuvent présenter un détail technique qui ne doit pas être présent dans un modèle à haut niveau d'abstraction. Le modèle de la vue informationnelle doit décrire uniquement les objets de la collaboration et la relation structurelle entre ces derniers. L'aspect comportemental des objets doit être décrit dans la vue fonctionnelle de l'application en suivant l'exemple des méthodes non objet comme WebML.

3.3. *Remontée d'abstraction et aération du métamodèle*

La remontée d'abstraction consiste à simplifier le métamodèle en enlevant les détails techniques et technologiques et en s'approchant le plus possible du langage humain en utilisant les termes et les concepts du domaine à modéliser. Elle permet de rendre le modèle plus abordable et plus compréhensible par des non-informaticiens. Ce qu'ArchGenXML et les méthodes objet, en générale, ne font pas. En effet les méthodes objet présentent des modèles décrivant l'application à un niveau d'abstraction

élevé par rapport aux langages de programmation, mais pas assez par rapport à l'utilisateur final, qui lui n'a pas forcement connaissance des méthodes objet. On peut comparer le résultat des méthodes objet au plan d'un bâtiment, qui reste difficilement compréhensible par le propriétaire et ne peut lui donner une image explicite de son futur bâtiment puisqu'il est fait par un technicien. Par contre, une maquette reste un moyen très efficace et plus accessible pour décrire les besoins du propriétaire.

La remontée d'abstraction est fortement associée à l'aération du métamodèle. En effet, dans un métamodèle une information peut être présente d'une manière implicite ou explicite. Rendre explicites les notions du système réel peut saturer le métamodèle, ce qui le rend difficile à maîtriser et les besoins difficiles à exprimer. Cela va à l'encontre d'un des principes fondamentaux de l'IDM qui est la simplification. Opter pour une expression implicite des notions du système réel rend le métamodèle plus spécifique à un métier donné, plus lisible par le concepteur et donc plus accessible et plus aéré. Mais dans ce cas, les transformations jouent un rôle plus important dans l'analyse et la manipulation du modèle. Il faut donc mettre plus d'intelligence à ce niveau ce qui rend la maintenabilité des transformations plus difficile. En conséquence, un compromis est nécessaire.

La séparation des préoccupations peut être vue comme une solution à apporter à un métamodèle pas assez abstrait et/ou encombré. En effet, la remontée d'abstraction est favorisée par la séparation des préoccupations. Le passage d'un niveau d'abstraction à un autre converge vers la fusion des préoccupations et l'ajout des détails techniques et technologiques qui encombrent le métamodèle.

La plupart des outils et méthodes existantes présentent l'information d'une manière souvent explicite ce qui la met à un niveau de détail et d'abstraction bas. Par exemple, BluAge propose une modélisation structurée par avance et contraint l'utilisateur à rajouter des concepts comme les «Bisnesse Object » pour chaque entité métier modélisée ce qui encombre le modèle.

3.4. *Élargissement de l'espace d'expression*

L'une des problématiques les plus connues dans le monde de la modélisation est l'expressivité. En effet, il est difficile de tout exprimer avec le modèle puisqu'il est destiné à simplifier le développement d'une application. Un concept, ou un pattern, dans un modèle est considéré à un niveau d'abstraction et de détail très élevé par rapport aux langages bas niveau. On peut, donc, transformer un concept, ou un pattern, en plusieurs lignes de code. Nous pouvons comparer cette vision à celle de la grande brique et la petite brique. On peut construire un bâtiment avec des formes et une finesse plus parfaites avec des petites briques qu'avec des grosses briques. La question est donc, comment approcher la finesse d'un langage à bas niveau avec le modèle ?

La réponse est de construire les grandes surfaces avec les grandes briques d'une manière la plus optimale et faire la finition avec les petites briques. Il faut trouver des

techniques de modélisation et un langage abstrait pour augmenter le degré de finition et par conséquent élargir l'espace d'expression.

Les outils et les méthodes étudiés montrent un manque d'expressivité pour des raisons différentes partant de l'absence des implémentations des actions dans le cas de certaines méthodes objet comme dans UWE jusqu'à l'impossibilité de décrire des workflows complexes impliquant plusieurs données, par exemple, comme dans la plupart des outils et méthodes étudiés. Nous allons maintenant aborder notre proposition.

4. Conception des portails collaboratifs

L'IDM offre un cadre méthodologique et technologique prometteur permettant d'unifier différentes façons de faire en un processus homogène et de favoriser l'étude des différents aspects du système. L'IDM est une forme d'ingénierie générative qui aboutit au code du système décrit à haut niveau. Ici, nous allons tirer profit de l'IDM pour résoudre la complexité de conception des portails collaboratifs.

4.1. *Description générale de notre démarche*

Nous avons divisé la modélisation des portails collaboratifs en différents points de vue traitant chacun d'un problème particulier. Pour cela, nous avons distingué le fond de la forme, le noyau fonctionnelle (NF) de l'IHM. Nous avons séparé le fond en structures documentaires « Quoi ? »), en rôles (« Qui ? ») et en comportements (« Comment ? »). De même, nous avons séparé la forme en : aspect structurel brut (« Voir le Quoi ? »), style statique (« Comment voir le Quoi ? »), style dynamique (« Quand et comment voir le Quoi ? ») et enfin interactions (« Faire le Comment ? »). L'imbrication successive des questions du QQOQCCP, nous permet de créer notre métamodèle en limitant l'absence d'oubli, tout en respectant la notion des vues de modélisation de la norme ENV 40003. Ainsi, nous avons élaboré une méthode qui consiste à métamodèliser les portails collaboratifs en se basant sur les notions du système d'information (voire figure 2), en couvrant le point de vue fonctionnel pour les processus métiers, le point de vue informationnel pour les objets de la collaboration et le point de vue de ressource pour les rôles des acteurs de la collaboration. Le point de vue organisationnel n'est pas complètement couvert puisqu'il ne donne aucune information sur la conception du portail collaboratif.

Nous pensons que le modèle est un moyen de communication efficace au sein d'une organisation. En conséquence, notre approche propose plusieurs niveaux de détail pour une même modélisation. Cela permet à un intervenant d'agir sur plusieurs niveaux de détails selon sa compétence et ses rôles dans l'organisation.

Le but de nos travaux n'est que la modélisation du portail collaboratif du système informatique, nous nous sommes inspiré de ce qui a été fait dans le métamodèle de CIMOSA et celui de UEML.

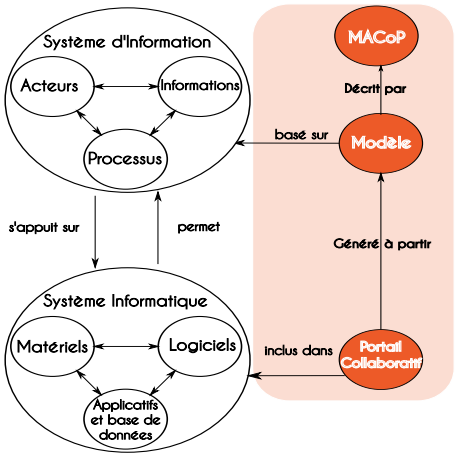

Figure 2. *MACoP et Système d'information*

Ce travail a abouti à notre métamodèle baptisé MACoP pour « Modeling and Analysis of Collaborative Portal ». MACoP nous permet de modéliser les portails collaboratifs à haut niveau d'abstraction, davantage basé sur les concepts des systèmes d'information que sur les concepts du web. Nous montrerons que les détails techniques et les concepts web peuvent être rajoutés automatiquement au modèle initial en appliquant des transformations qui ne se contenteront pas de réaliser de simples transformations un à un, mais joueront le rôle d'expert, car elles contiennent l'expertise et les connaissances des maîtres d'œuvre et des développeurs web. Le modèle initial ne présente donc que les besoins de l'utilisateur et le cahier des charges de l'application.

4.2. *MACoP*

Généralement, une application est formée de la partie métier qui décrit le noyau fonctionnel et de la partie visuelle qui décrit l'IHM de l'application.

Nous avons constaté que la dynamique de l'IHM peut être complètement spécifiée dans le NF ce qui permet de générer une IHM par défaut respectant le métier de l'utilisateur final. Néanmoins, modéliser la partie IHM peut permettre au modeleur de spécifier ses besoins visuels et ses types d'interactions avec le NF. En effet, le NF a besoin des données issues de l'application (base de données ou données calculées automatiquement) ou introduites par l'utilisateur final. Ces dernières peuvent être le résultat d'une interaction entre l'homme et la machine comme, par exemple, la saisie d'un champ de formulaire ou une commande vocale ou tactile. Dans ce cas, il faut un modèle décrivant comment interpréter les interactions de l'IHM en données compré-

hensibles par le NF, sinon le modèle est implicite et est l'identité (un champ décrit par le NF et généré tel quel).

MACoP est formé de trois parties qui sont le fond qui décrit les besoins métier de l'application (NF), la forme qui décrit l'aspect visuel et les types d'interaction de l'application (IHM) et finalement l'interpréteur qui décrit l'interprétation des données échangées entre le fond et la forme (voire figure 3). Bien qu'actuellement seul le fond

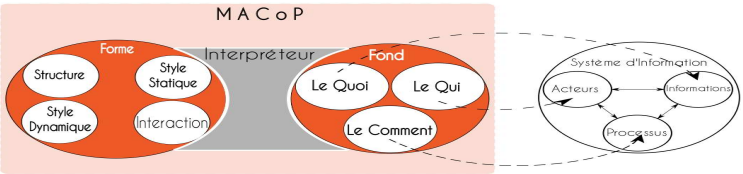

Figure 3. *MACoP*

est décrit, nous verrons par la suite que la partie visuelle est générée automatiquement. Toutefois, la description du métamodèle de l'interpréteur et celle de la partie visuelle sont en cours.

4.2.1. *Le métier*

Dans ce modèle l'utilisateur décrit ses besoins informationnels ; les objets de la collaboration (« Quoi ? ») qui peuvent être les informations à manipuler et/ou les profils qui présentent, virtuellement, les utilisateurs réels dans l'application. Les ressources : les rôles des acteurs qui vont agir sur les données de l'application (« Qui ? »). Et enfin, les besoins fonctionnels qui décrivent les processus métier (« Comment ? »). Ce qui suffit pour capter tous les besoins nécessaires pour le développement d'un portail collaboratif, les autres questions du QQOQCCP sont posées à des niveaux plus bas.

4.2.1.1. Vue informationnelle

Les objets de collaboration manipulés sur le portail collaboratif vont être manipulés dans cette vue. Les contraintes et les invariants sont exprimés par le langage OCL (OMG, n.d.) qui est un langage de requêtes standardisé permettant d'exprimer nos contraintes à un niveau d'abstraction élevé sans inclure les détails technologiques.

L'instanciation lors de l'initialisation de l'application est modélisée par l'utilisateur afin d'assurer le minimum requis par le métier.

Plan documentaire Pour méta-modéliser cette partie, nous nous sommes inspiré du métamodèle UML 2.0. En effet dans notre métamodèle (voir figure suivante) le concept ObjectOfCollaboration hérite du concept Classifier d'UML, duquel nous avons éliminé l'aspect comportemental du Classifier. ObjectOfCollaboration est équi-

valent au concept Class d'UML manipulable par des processus métier mais sans l'aspect comportemental.

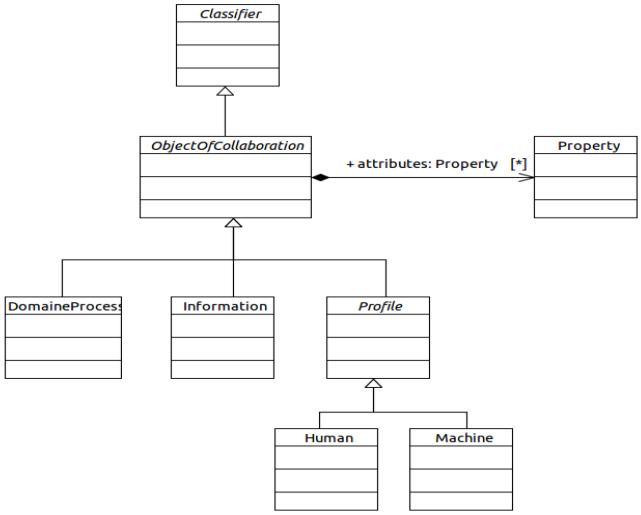

Figure 4. *Métamodèle des objets de collaboration*

Nous pouvons distinguer trois types d'objets de collaboration : Les *Informations* qui sont des objets de collaboration passifs permettant de décrire la structure métier de la collaboration (le plan documentaire) ; Les *Profils* qui sont des objets de collaboration passifs pouvant être des machines ou des humains, ils sont des représentations informatiques des acteurs physiques dans la collaboration. Et finalement, les *Processus Domaine* qui sont des objets de collaboration actifs et présentent la vue passive d'un processus métier.

À chaque objet de collaboration, il faut associer un ou plusieurs processus le manipulant, sinon un processus par défaut sera créé pour pouvoir le manipuler.

Configuration de l'application Le concept InstanceSpecification utilisé pour l'initialisation est équivalant à celui du métamodèle UML2.0. On peut, aussi, instancier les Profils avec ProfileInstance qui est un InstanceSpecification avec en plus la possibilité de lui attribuer des rôles.

Il est possible de créer des règles métier en écrivant des requêtes sur les InstanceSpecification. Si l'InstanceSpecification est généralisée (non contextualisée), elle se comporte comme une requête sélectionnant toutes les instances de même type et ayant la même forme, ce qui généralise les règles métier écrites.

4.2.1.2. Vue des ressources

Une fois les données (Objet de collaboration et leurs instances) modélisées et l'arborescence spécifiée, l'utilisateur modélise les rôles de la collaboration.

Les rôles sont les concepts clefs de l'organisation des individus. À un Rôle est attaché un ensemble d'activités confiées à une personne dans le cadre d'une responsabilité, d'une délégation, d'une charge, ceci en fonction d'une qualification ou de compétences. C'est la « fonction » professionnelle de cet individu. Chaque Rôle est rempli par un Acteur (une personne donnée). Plusieurs personnes peuvent tenir le même Rôle. Et une personne peut avoir plusieurs Rôles .

Une relation de supérieur/inférieur hiérarchique existe entre les rôles. Cette relation est équivalente à la relation d'héritage en UML. Enfin, un rôle peut être abstrait pour permettre l'héritage d'un ensemble de tâches confiées (voir figure 5). Pour pou-

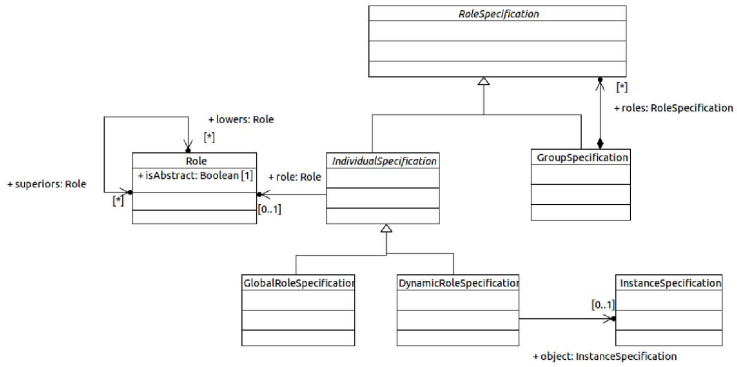

Figure 5. *Métamodèle des rôles*

voir spécifier un rôle, on utilise le concept RoleSpecification. Les rôles peuvent être individuels ou regroupés ce qui permet d'avoir des spécifications complexes. La spécification peut être dynamique, spécifiée par rapport à un objet de collaboration donnée (voir figure 5), ou globale, le rôle est spécifié pour toute l'application. En général un rôle est attribué à une instance de profil ou à un ensemble d'activités à effectuer dans la collaboration.

4.2.1.3. Vue fonctionnelle

Une application présente deux sortes de comportements : le local qui dépend des données de l'application et qui présente les processus métier, et le global qui ne dépend pas des données de l'application et qui est visible par l'application comme les services.

Processus métiers Une fois les données et les rôles modélisés, l'utilisateur spécifie les processus métiers (Godart *et al.*, 2009) qui vont permettre de décrire comment

les données seront manipulées par les acteurs. Nous avons écarté l'utilisation du diagramme d'activité d'UML peux adapté à la modélisation des processus métier. Nous lui avons préféré BPMN 2.0 qui est standardisé par l'OMG. Plusieurs études montrent que BPMN a une présentation plus puissante et plus abstraite, pour les processus métier, que le diagramme d'activité d'UML, citons celle de (Eloranta *et al.*, 2006), que nous trouvons la plus concluante. Ce travail présente une étude comparative entre le diagramme d'activité d'UML et le BPMN 1.1. Nous pouvons également citer les travaux de (Macek *et al.*, 2009) ayant proposé une transformation de BPMN vers le diagramme d'activité d'UML et qui ont rencontré des problèmes dus à la richesse de BPMN par rapport au diagramme d'activité d'UML .

Malgré sa puissance de présentation, BPMN reste incomplet pour la génération de code. En effet, BPMN présente une notation standardisée permettant d'exprimer les processus métier dans le but de les étudier et les analyser et non dans le but de les exécuter.

Pour remédier à ce manque, nous avons ajouté d'autres notions au métamodèle permettant une modélisation plus intuitive, ainsi que la génération de code. Les concepts rajoutés nous ont permis de répondre aux questions « Où ? » et « Combien ? » et de spécifier l'implémentation des actions. Pour cela nous avons ajouté la notion de contexte d'action, ce qui permet de spécifier sur quelle information on agit et à qu'elle information on accède. Il est formé de trois types qui sont :

– Le Contexte d'action : Ce contexte permet de spécifier sur quoi l'action est effectuée. Il définit : l'instance ou la collection, les attributs de cette instance ou collection qui sont manipulés directement par l'utilisateur dans le cadre de cette activité, les états dans lesquels l'instance devrait être et enfin la multiplicité pour spécifier le nombre d'instances nécessaires.

– Le Contexte d'étude : Ce contexte permet de spécifier ce qu'il faut étudier avant d'effectuer l'action, car l'acteur doit connaître l'ensemble des informations nécessaires à l'action. Nous pourrons en déduire le droit d'accès de l'acteur concerné. Dans ce contexte, la multiplicité n'a pas d'intérêt.

– Le Contexte de production : Ce contexte permet de spécifier le résultat de l'action et sa localisation.

Pour pouvoir donner à notre métamodèle la possibilité de décrire des comportements complexes, nous avons autorisé l'imbrication des processus métier et l'implémentation des actions à base de composants adaptables au modèle.

Dans les métamodèles CIMOSA et EUML, un processus métier est considéré comme une entité active faisant partie de la vue fonctionnelle d'un système d'information. Sachant qu'un processus métier est une information de l'entreprise passive avant d'être active nous pouvons le considérer comme faisant partie de la vue informationnelle du système d'information. En se basant sur cette constatation, le processus métier devient un objet de collaboration qui peut être manipulé par d'autres processus. Ce qui nous donne la possibilité d'imbriquer les processus et par la suite la possibilité de décrire des comportements complexes comme la supervision ou le contrôle des

processus. Il devient possible de décrire la communication entre les processus d'une manière précise, verticale (entre les instances d'un même processus) ou horizontale (entre les instances des processus différents).

Pour les actions et les services, nous avons choisi une implémentation à base de composants. BPMN ne spécifiant pas comment implémenter les activités, nous proposons une approche basée sur des composants adaptables au contexte et à l'application. Ces composants sont des propriétés intellectuelles, ils sont connectés les uns avec les autres et/ou connectés aux données de l'application et il est possible de les trouver sur étagère. Ces IP (Intellectuel Property) sont comparables à la notion des connecteurs de l'outil BonitaSoft mais en plus abstraits.

Le composant est définit par un ou plusieurs fichiers à partir desquels le code est généré. Ce code est généralement peu adapté au contexte, rendant l'application lente. Une solution consiste à rendre les composants configurables et adaptables à l'application au niveau du modèle. En plus d'avoir le même comportement qu'un composant ordinaire, le composant offre le code le plus adapté à l'application. Un composant peut aboutir à la génération de deux codes différents selon là où il a été connecté dans le modèle.

Nos IP ne sont plus de simples fichiers, mais des templates de génération de code permettant ainsi d'adapter le code par rapport aux paramètres de configuration spécifiés au niveau du modèle.

Pour une implémentation donnée, on trouve des composants connectés les uns aux autres et connectés aux données de l'application. Les données de l'application sont décrites par le langage OCL.

Services Un service est une application sous forme d'IP. Par exemple, un service Mail est une application pouvant être modélisée avec MACoP, mais, selon l'utilisateur, la manipulation de mail peut être vue comme un service ou comme une application faisant partie de son métier.

Un Service est un comportement visible par toute l'application, il peut être appelé à tout moment dans le modèle. L'utilisateur modélise les fonctionnalités de ses services avec le même mécanisme d'implémentation des activités des processus métier.

4.2.2. *Interpréteur et Aspect visuel*

L'utilisateur modélise la structure visuelle de son portail d'une manière indépendante du fond. La forme présente l'interface entre le fond et l'utilisateur réel et décrit comment présenter un Objet de collaboration et son interface de manipulation.

Nos futurs travaux reposeront sur ceux du domaine des IHM, car d'importantes problématiques sur la présentation et l'interaction entre l'homme et la machine ont déjà été traitées. Ces travaux constituent une approche intéressante pour la modélisation de la partie visuelle des applications web. Ce qui va nous permettre de nous abstraire du type d'interactions classique, comme les formulaires, et de pouvoir modéliser

d'autres types d'interactions plus adaptées à la plate-forme matérielle (commande tactile, vocale...).

Là aussi, nous avons séparé les préoccupations : le modèle de la structure brute décrivant les éléments visuels d'une manière basique sans spécifier leur aspect ou leur emplacement dans la page ; Le modèle du style statique qui décrit l'aspect statique de la page (couleur position...) ; Le modèle du style dynamique qui décrit l'aspect animé de la page (Liste/texte défilant...) ; Le modèle d'interaction qui permet de décrire le type d'interaction entre l'homme et la machine.

Un modèle intermédiaire permettant l'interprétation des données entre ces deux derniers est nécessaire. il permet de modéliser les flux de données entre les deux modèles comme pour le chiffrage ou les transformations des données.

5. Validation

Nous avons validé notre approche par la conception d'un cas d'utilisation, un outil de « Tracker », et par le développement de la chaîne de transformation permettant de générer du code opérationnel.

5.1. *Exemple du Tracker*

L'exemple que nous proposons illustre l'usage de l'outil « Tracker » dans le cadre d'une démarche agile de gestion de projet informatique. Le client y crée des tickets pour demander de nouvelles fonctionnalités ou pour signaler des anomalies dans le logiciel livré. Les tickets sont regroupés par itérations, qui sont des sessions de développements avec un nombre de jours-homme fixe. Chaque ticket possède un état parmi « Non confirmé », « Ouvert », « En recette », « En production »... De ce fait, à tout moment, le client peut visualiser l'état d'avancement du projet.

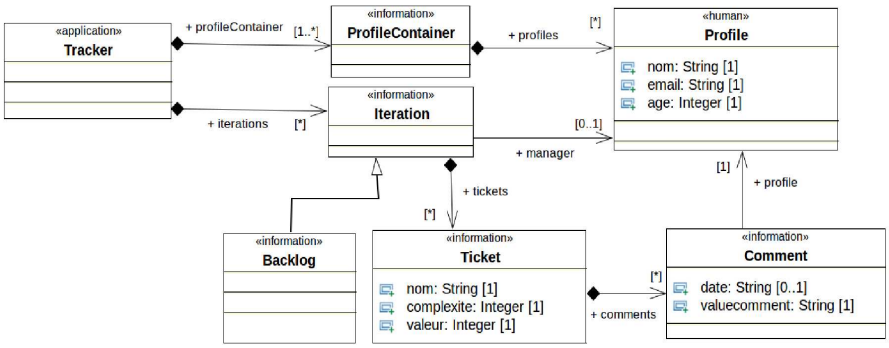

Figure 6. *Modèle du plan documentaire du "Tracker"*

Dans cet exemple (voir figure 6), nous avons commencé par modéliser la racine de l'application « Tracker », qui contiendra les itérations « Iteration » auxquelles seront assignés les tickets. Lors de sa création, un ticket est assigné par défaut à une itération spéciale nommée Backlog. Les utilisateurs « humain » seront représentés par des profils « Profile » qui seront stockés dans un conteneur qu'on appellera « ProfileContainer ». Enfin, un ticket peut être commenté.

À ce stade de modélisation, on ne décrie que les objets de collaboration de l'entreprise sans spécifier la dynamique de ces derniers. Sur ce modèle, nous pouvons spécifier deux types de contraintes : celles structurelles décrites par les relations entre les objets de la collaboration et celles fonctionnelles décrites par le langage OCL sur chaque objet.

Une fois le modèle des objets de collaboration défini nous pouvons configurer l'application. Ici, nous avons créé l'instance de notre application « monApplication ». Nous créons l'unique instance du « ProfileContainer » nommée « profileContainer » qui mémorisera les profils de l'application, ainsi que l'unique instance du « Backlog » nommée « backlog », itération par défaut pour les tickets (voir figure 7).

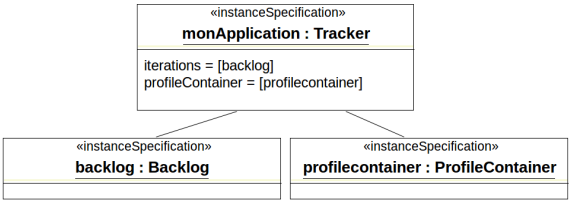

Figure 7. *Configuration du "Tracker"*

Après la modélisation de la partie statique de l'application (« Quoi ? »), nous modélisons « Qui ? » et « Comment ? » du Tracker (voir figure 8). En premier, nous définissons les rôles de l'application, dans notre cas, nous avons le Client, le Développeur, le Responsable du ticket et enfin le Manager des tickets qui est le supérieur du Client et du Développeur. Ensuite, nous décrivons les domaines d'activité et les processus domaine (Processus métier) à un niveau de détail simplifié. Cela permet d'avoir une idée globale des aspects métier impliqués dans la collaboration. Ensuite, nous raffinons chaque processus domaine en spécifiant le processus métier BPMN associé (voir figure 9). Ici, nous avons le domaine « TicketManagement » qui est composé de processus domaine « TicketCreation » et « TicketValidation ». Par souci de clarté, seul a été développé le processus domaine « TicketCreation ».

Nous devons maintenant raffiner chaque processus domaine. Ainsi la figure 9 décrit plus en détails le processus métier de création de ticket. Celui-ci peut être initié par le Développeur ou par le Client lors de la création d'un ticket. Si le Développeur est l'initiateur, alors il doit choisir un Client présent dans l'application comme Manager de ce ticket. Si le ticket est créé par le Client, il en devient le manager, le Développeur pourra alors commenter ou modifier la complexité de ce ticket. Le Client, quand à lui,

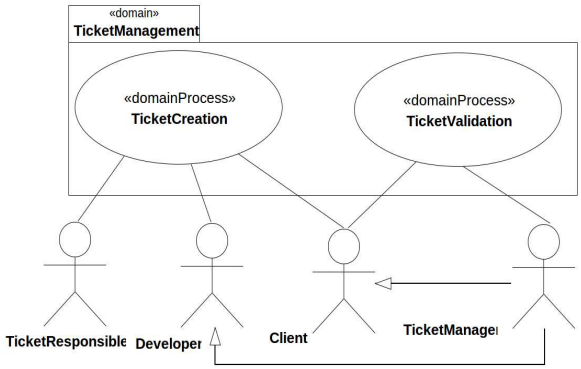

Figure 8. *Les rôles du "Tracker"*

pourra assigner le ticket à une itération et le valider afin que le développeur puisse produire les actions associées. Le processus métier est associé à un data-store contenant des instances des objets manipulés (en relation avec le processus). Nous y avons spécifié une instance de « Ticket ».

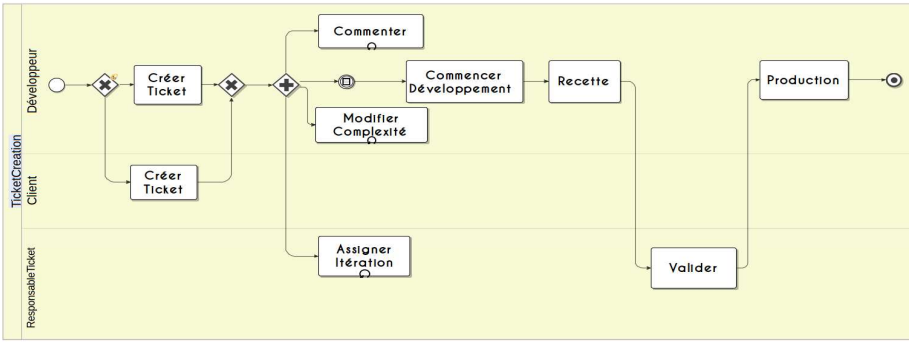

Figure 9. *Processus métier de gestion de ticket*

Les actions complexes ou spécifiques doivent être détaillées. La figure 10 présente l'implémentation de l'action « Créer Ticket » du Développeur. Cette implémentation contient un IP , « initTicketResponsible », permettant d'associer le rôle local TicketResponsible au client « c » vers le ticket « t » et des données, le ticket créé et le client. Le ticket est celui présent dans le data-store du processus. Dans cette implémentation, l'acteur, ici le développeur, doit pouvoir choisir le client parmi une liste de client potentiels. Ceci s'exprime par la requête OCL « **choice(| profileContainer.profiles->select(p| p.hasRole(Client)))** ». L'expression « **choice(| [Liste])** » permets de choisir un élément de la liste proposé en argument. Dans notre cas, la requête OCL permet de récupérer tous les clients présents dans l'instance « profileContainer ».

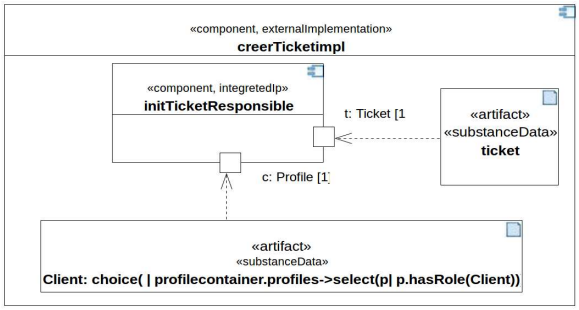

Figure 10. *Implémentation de l'action "Créer Ticket"*

5.2. *La chaîne de transformation MACoP->Portail Collaboratif (Dolmen)*

Pour valider notre approche nous avons développé une chaîne de transformation permettant de générer le code des portails collaboratifs pour la technologie Dolmen, qui est un ensemble de bibliothèques Python/Zope (voir figure 11).

La chaîne de transformation nous permet de passer d'un espace de modélisation à un autre. Chaque espace est à un niveau d'abstraction et de détail plus élevé que l'autre. Les modèles intermédiaires sont décrits par des méta-modèles permettant d'ajouter les détails techniques et technologiques nécessaires à la génération de code. Nous avons pris soin de faire apparaître le plus tard possible les détails technologiques propre à la cible visée, ici domaine. Ces détails n'apparaissent que dans les derniers modèles de la chaîne.

Les transformations captent les connaissances du maître d'œuvre pour l'analyse du modèle et celle du développeur pour les stratégies de développement.

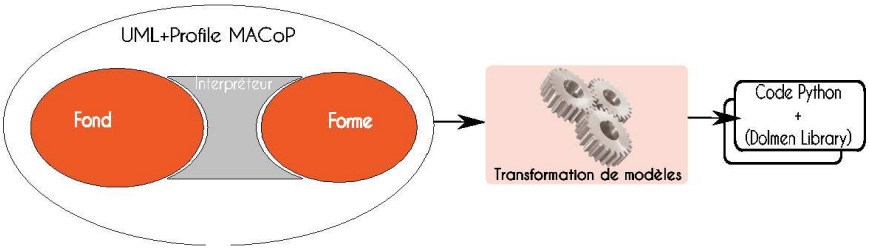

Figure 11. *La chaîne MACoP->Portail Collaboratif(Dolmen)*

La chaîne de transformation MACoP->Portail Collaboratif(Dolmen) nous permet de générer 100% du code fonctionnel. Le développeur n'a plus qu'à lancer l'application. Le portail collaboratif généré permet de manipuler les données de la collaboration selon les processus métier spécifiés au niveau modèle. Les processus métier sont exécutés d'une manière implicite en agissant directement sur les objets avec lesquels

ils ont une relation directe ou indirecte, ils sont eux-mêmes vus comme des objets de collaboration sur lesquels nous pouvons agir comme spécifié dans le modèle.

Les captures d'écran suivantes (voir figure 12) montrent le formulaire de création d'un ticket de l'application Tracker, générée à partir du modèle présenté précédemment, et le formulaire permettant de le commenter.

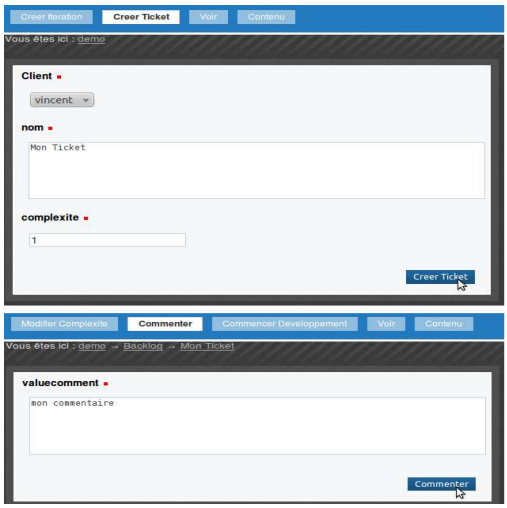

Figure 12. *Capture d'écran de l'application générée*

6. Évaluation de notre approche

La conception et la génération complète de l'application du « Tracker » nous ont permis d'effectuer une première validation de notre approche. Nous avons complété cette validation par une évaluation selon les trois critères identifiés dans la section 3 de cet article.

6.1. *Séparation des préoccupations*

Au contraire des méthodes et outils de modélisation et génération d'applications web existants, nous avons séparé les aspects et préoccupations des portails collaboratifs d'entreprise, tout en conservant la modélisation des entreprises centrée processus métier pour le noyau fonctionnel (NF) et en adoptant une approche dirigée par le NF pour la modélisation de l'IHM.

Nous avons aussi séparé les préoccupations du NF selon les vues proposées par la norme ENV 40003 pour la modélisation des entreprises, ce qui nous a permis de nous aligner avec la culture industrielle et son domaine.

La séparation des préoccupations de l'IHM concerne la structure brute des données de l'application, le style statique, le style dynamique et enfin les interactions. Cela nous permettra de décrire la manière de voir les données de l'application et leur interface de manipulation (tableau de bord) indépendamment de la mécanique du métier.

Sachant que les besoins sont avant tous des besoins métier, notre approche, basée sur la modélisation du NF, donne un rôle très important aux experts métier. Ce qui offre à l'entreprise suffisamment d'autonomie pour gérer et faire évoluer son portail collaboratif.

La séparation forte des préoccupations permet la réutilisation des modèles, elle diminue les coûts de développement et devrait permettre de concevoir de nouvelles applications par assemblage de modèles existants. Avec de tels outils, l'évolution verticale que représente toute modification du métier d'une entreprise devient possible à un prix modéré, puisqu'il suffira de mettre à jour les modèles pour générer le nouveau portail qui saura utiliser les données existantes.

6.2. *Remontée d'abstraction et aération du métamodèle*

Le fait de s'inspirer des méta-modèles décrivant les entreprises, comme UEML, CIMOSA et BPMN, nous permet de remonter en abstraction. En effet ces métamodèles permettent de décrire le métier de l'entreprise d'une manière puissante et abstraite. Des concepts techniques comme les méthodes sont décrits d'une manière implicite à travers les processus métier. Cette abstraction permet d'éviter la pollution des modèles par la stratégie d'implémentation définie par l'analyse des modèles d'entrée, les choix et les détails techniques issus des technologies employées.

Comme MACoP est issu des méta-modèles de BPMN, UEML et CIMOSA, les experts métiers le comprennent nativement. Ce qui permet de diminuer le risque d'erreur lors de la communication entre les experts métier pour la réalisation de l'application.

6.3. *Élargissement de l'espace d'expression*

Pour élargir l'espace d'expression de notre méta-modèle nous avons utilisé le langage OCL pour : décrire les contraintes et les requêtes, extraire les données de l'application, décrire des règles métier complexe. Nous avons ajouté des concepts comme les contextes d'étude, d'action et de production pour permettre la spécification du « Quoi ? », « Combien ? » et « Où ? ». D'autres techniques de modélisation nous permettent d'élargir l'espace d'expression : comme la possibilité de modéliser des processus métier imbriqués sur plusieurs niveaux, ce qui permet de modéliser des comportements complexes tels le contrôle et la supervision des processus. L'implémentation, à base d'IP adaptable au modèle, des actions nous permet de définir un comportement complexe purement métier.

Notre approche répond donc bien aux différents critères énoncés dans la section 3 et, en ce sens, possède plusieurs avantages sur les outils et méthodes existants dans le domaine de la modélisation des applications web.

Néanmoins, il est évident que nous devrons à l'avenir définir des grilles d'évaluation plus précises pour mieux comparer notre approche avec celles existantes.

7. Conclusion et perspectives

Dans cet article, nous avons présenté le métamodèle MACoP permetant de décrire les portails collaboratifs à un haut niveau d'abstraction, dans le but de générer le code en suivant une approche IDM. Dans ce métamodèle, nous nous sommes focalisés sur les concepts issus des systèmes d'informations centrés processus métier. En premier, nous avons positionné nos travaux par rapport au domaine du génie industriel en proposant la définition du système d'information la plus adéquate à notre démarche. Pour la spécification de notre métamodèle, nous avons pris la modélisation des entreprises comme base pour la modélisation des systèmes d'information. Ce qui nous a permis d'étudier et de nous inspirer des différents cadres de modélisation comme CIMOSA et UEML. Cela nous a aussi permis d'organiser notre métamodèle métier en respectant la notion de point de vue de modélisation évoquée par la norme ENV 40003. Notre approche nous à permis de répondre aux trois questions fondamentales concernant le fond de l'application, le « Quoi ? », « Qui ? » et « Comment ? », afin de constituer un cahier des charges complet sans rentrer dans la complexité des aspects techniques et technologiques du web. La réponse au « Quoi ? » est présente, dans MACoP, comme modélisation de tout ce qui peut être manipulé dans la collaboration (Vue informationnelle d'un système d'information). La réponse au « Qui ? » est présente comme modélisation des rôles et leur hiérarchie dans la collaboration (Vue des ressources d'un système d'information). Enfin, la réponse au « Comment ? » est présente comme modélisation des processus métiers permettant d'établir les relations entre les rôles et les données de l'application (Vue fonctionnelle d'un système d'information). Les réponses aux autres questions comme « Où ? » et « Combien ? » sont considérées comme faisant partie du « Comment ? ».

En plus de la modélisation du fond, MACoP permettra de décrire la forme de l'application à un haut niveau d'abstraction et d'une manière totalement indépendante du fond. Dans MACoP, nous avons résolu certains problèmes identifiés dans les outils et méthodes existants sur le marché comme dans la littérature. Par exemple la séparation des préoccupations, la remontée d'abstraction et l'aération du métamodèle et enfin l'élargissement de l'espace d'expression.

Enfin, pour valider notre approche, une première chaîne de transformation vers la technologie Dolmen (Python/Zope) a été développée. Cette chaîne nous permet de générer 100% du code fonctionnel pour les portails collaboratifs permettant d'exécuter les processus métier de la collaboration d'une manière implicite et transparente.

Dans l'état actuel de nos recherches, de nombreux travaux restent à faire allant de la modélisation à la génération de code. Par exemple, dans notre modélisation la notion de relation couvre presque touts les points de vues, mais elle est inexistante au niveau des paramètres des actions. Nous avons donc des relations entre objets de collaboration qui se présentent sous forme d'associations, de relations entre rôles qui se présentent sous forme de relation hiérarchique dans l'entreprise, de relations entre processus métier qui se présentent par l'imbrication des processus et enfin de relations entre actions qui se présentent sous forme de flots des données. Identifier les relations entre les paramètres de l'action nous permettra de décrire les dépendances et les relations dynamiques entre ces derniers. Cela donnera la possibilité de générer des formulaires à structure dynamique avec des dépendances entre champs. D'autres travaux doivent être réalisés comme le test et la vérification, la simulation et l'optimisation... Enfin au niveau de la génération de code nous allons écrire d'autres chaînes de transformation vers d'autres cibles technologiques ou de plate-forme.

En conclusion, nous pouvons dire que notre approche ouvre une nouvelle voie qui s'avère prometteuse pour une utilisation plus large des outils de modélisation et de génération automatique complète des portails collaboratifs. Nous pouvons dire aussi, à travers cette approche, que nous atteignons la finalité de ces travaux qui est de donner à l'entreprise la possibilité de faire évoluer son portail collaboratif verticalement lors des changements du métier et horizontalement lors des changements technologiques, et ce en toute autonomie. Le metamodèle et l'ensemble de la chaîne de transformation seront disponibles très prochainement sous licence GPL3 (`http://omegsi.ecreall.com/`)

8. Bibliographie

Allweyer T., *BPMN 2.0*, BoD, feb, 2010.

Anaya V., Berio G., Harzallah M., Heymans P., Matulevicius R., Opdahl A. L., Panetto H., Verdecho M. J., « The Unified Enterprise Modelling Language–Overview and further work », *Computers in Industry*, vol. 61, n° 2, p. 99-111, feb, 2010.

BARBIER M. F., « Enterprise Model-Driven Development with BLU AGE », n.d.

Brossard A., Abed M., Kolski C., « Modélisation conceptuelle des IHM. Une approche globale s'appuyant sur les processus métier », *Ingénierie des systèmes d'information*, vol. 12, n° 5, p. 69-108, dec, 2007.

Chen D., Vernadat F., « Standardisation on enterprise modelling and integration : Achievements, on-going works and future perspectives », INCOM, Vienna, 2001.

CIMOSA, « CIMOSA », n.d.

da K ADRI R., Une approche pour la Modélisation d'Applications Web a base de Composants Logiciels, PhD thesis, jan, 2009.

Darras F., « Proposition d'un cadre de référence pour la conception et l'exploitation d'un progiciel de gestion intégré. », , http ://ethesis.inp-toulouse.fr/archive/00000599/, oct, 2004.

Eloranta L., Kallio E., Terho I., « A Notation Evaluation of BPMN and UML Activity Diagrams », p. 1-45, 2006.

Favre J.-M., *Concepts fondamentaux de l'IDM. De l'ancienne egypte à l'ingénierie des langages*, 2èmes Journées sur l'Ingénierie Dirigée par les Modèles (IDM'06), Lille, France, 2006.

Fischer L., *2010 BPM and Workflow Handbook, Spotlight on Business Intelligence*, Future Strategies Inc, jun, 2010.

Godart C., Perrin O., *Les processus métiers : Concepts, modèles et systèmes*, Hermes Science Publications, sep, 2009.

IDM A., « Action IDM », n.d.

ISO, « Enterprise Integration - Framework for Enterprise Modelling », n.d.

Knox S., Stahl J., Aspeli M., Convent D., Hanning D., Newbery R., DeStefano J., Parker C., Clark A., Williams V., *Practical Plone 3 : A Beginner's Guide to Building Powerful Websites*, Packt Publishing, feb, 2009.

KroißC., Koch N., UWE Metamodel and Profile . User Guide and Reference, Technical report, feb, 2008.

Macek O., Richta K., « The BPM to UML activity diagram transformation using XSLT », p. 119-129, 2009.

McAfee A., « Enterprise 2.0, version 2.0 », 2006.

Morley C., Hugues J., Leblanc B., *UML2 pour l'analyse d'un système d'information : Le cahier des charges du maître d'ouvrage*, 3e édition edn, Dunod, jan, 2006.

OMG, « Business Process Model and Notation (BPMN) FTF Beta 1 for Version 2.0 », 2011.

OMG, « Object Constraint Language OMG Available Specification Version 2.0 », n.d.

Pelletier M., *Plone Live*, LL, 2005.

Roques P., *UML 2 : Modéliser une application web*, 3e édition edn, Eyrolles, apr, 2007.

Rossi G., Pastor O., Schwabe D., Olsina L., *Web Engineering : Modelling and Implementing Web Applications*, 1 edn, Springer, oct, 2007.

Silver B., *BPMN Method and Style : A levels-based methodology for BPM process modeling and improvement using BPMN 2.0*, Cody-Cassidy Press, jun, 2009.

Sottet J.-S., Mega-IHM : Malléabilité des Interfaces Homme-Machine Dirigée par les Modèles, PhD thesis, JOSEPH FOURIER - GRENOBLE 1, oct, 2008.

Touzi J., « Aide à la conception de Système d'Information Collaboratif , support de l'interopérabilité des entreprises », , http ://ethesis.inp-toulouse.fr/archive/00000606/, nov, 2007.

WebRatio, « WebRatio », n.d.

wikipedia, « QQOQCCP », n.d.

ANNEXE POUR LE SERVICE FABRICATION
A FOURNIR PAR LES AUTEURS AVEC UN EXEMPLAIRE PAPIER
DE LEUR ARTICLE ET LE COPYRIGHT SIGNE PAR COURRIER
LE FICHIER PDF CORRESPONDANT SERA ENVOYE PAR E-MAIL

1. ARTICLE POUR LA REVUE :

Technique et science informatiques

2. AUTEURS :

Amen Souissi[*] *— Pierre Boulet*[2**] *— Cedric Dumoulin*[**] *— Michael Launay*[*]

3. TITRE DE L'ARTICLE :

Modélisation centrée sur les processus métier pour la génération complète de portails collaboratifs

4. TITRE ABRÉGÉ POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :

Génération de portails collaboratifs

5. DATE DE CETTE VERSION :

4 novembre 2011

6. COORDONNÉES DES AUTEURS :

– adresse postale :
[*] Ecréall
Parc scientifique de la Haute Borne, 11 rue de l'Harmonie 59650 Villeneuve d'Ascq
{amensouissi, michaellaunay}@ecreall.com

[**] Univ. Lille 1, LIFL
Cité scientifique - Bâtiment M3 59655 Villeneuve d'Ascq Cédex
{pierre.boulet, cedric.dumoulin}@lifl.fr
– téléphone : 03 20 79 32 90
– télécopie : 00 00 00 00 00
– e-mail : amensouissi@ecreall.com

7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :

LaTeX, avec le fichier de style `article-hermes2.cls`,
version 1.23 du 17/11/2005.

8. FORMULAIRE DE COPYRIGHT :

Retourner le formulaire de copyright signé par les auteurs, téléchargé sur :
`http://www.revuesonline.com`

SERVICE ÉDITORIAL – HERMES-LAVOISIER
14 rue de Provigny, F-94236 Cachan cedex
Tél. : 01-47-40-67-67
E-mail : revues@lavoisier.fr
Serveur web : http://www.revuesonline.com