



**HAL**  
open science

## Déterminer l'impact d'une évolution dans les processus métiers

Alexandre Feugas, Sébastien Mosser, Anne-Françoise Le Meur, Laurence Duchien

► **To cite this version:**

Alexandre Feugas, Sébastien Mosser, Anne-Françoise Le Meur, Laurence Duchien. Déterminer l'impact d'une évolution dans les processus métiers. Journées sur l'Ingénierie Dirigée par les Modèles (IDM'11), Jun 2011, Lille, France. pp.71-76. hal-00596428

**HAL Id: hal-00596428**

**<https://hal.science/hal-00596428v1>**

Submitted on 27 May 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.

---

# Déterminer l'impact d'une évolution dans les processus métiers

**Alexandre Feugas — Sébastien Mosser — Anne-Françoise Le Meur  
— Laurence Duchien**

*Equipe-Projet INRIA ADAM, Université Lille 1, LIFL UMR CNRS 8022  
INRIA Lille - Nord Europe  
40 Avenue Halley - Park Plaza - Bât A  
59 650 Villeneuve d'Ascq  
prenom.nom@inria.fr*

---

*RÉSUMÉ. Avec l'augmentation de la taille et de la complexité des systèmes à bases de services, la tâche du concepteur dans l'évolution des processus métiers est devenue ardue. Une simple modification localisée du processus peut entraîner une dégradation des propriétés non fonctionnelles (temps de réponse, sécurité, taille des messages, etc.) de l'application toute entière. Pouvoir déterminer l'impact d'une évolution sur l'ensemble du système aiderait le concepteur à se rendre compte a priori des conséquences potentielles de l'évolution. Nous proposons de construire un canevas de conception et d'évolution à base de modèles pour aider le concepteur à déterminer l'effet d'une évolution sur le système en s'appuyant sur des analyses d'impact.*

*ABSTRACT. With the increase of service-based systems size and complexity, the software designer's task within the business process evolution became harder. A simple local modification of the process may bring to a degradation of the entire application's non functional properties (response time, security, size of messages, etc.). Being able to define the impact of an evolution on the entire system would help the designer to consider a priori the potential consequences of the evolution. We propose to build a design and evolution model-based framework to help the software designer to define the effect of an evolution on the system, based on impact analysis.*

*MOTS-CLÉS : Processus métier - Impact - Qualité de Service - SOAs*

*KEYWORDS: Business process - Impact - Quality of Service - SOAs*

---

## 1. Introduction

De nos jours, les systèmes informatiques doivent répondre à une évolution fréquente des besoins de l'utilisateur. Ces systèmes de plus en plus grands et complexes sont difficiles à maintenir. A cette problématique de conception et de maintenance, les Architectures Orientées Services (SOAs) proposent de construire des applications par compositions de services. Ainsi, ces architectures constituées de processus contenant la logique métier des tâches à effectuer profitent d'une modularité et d'un couplage lâche des éléments. Toutefois, ces processus peuvent être composés de plusieurs centaines d'activités et il est donc difficile pour le concepteur de raisonner sur les conséquences d'une évolution, notamment en terme des performances. En effet, le niveau global des propriétés non fonctionnelles telles que le temps de réponse, la taille des messages échangés, *etc.* peut être détérioré par une mise à jour localisée d'un processus métier. Ce papier propose un canevas de conception basé sur les modèles pour aider le concepteur à localiser l'élément à l'origine d'une dégradation des propriétés non fonctionnelles.

## 2. Exemple de scénario d'évolution

Nous considérons dans cet article le système SOA JSEDUITE<sup>1</sup>, qui met en œuvre une application ubiquitaire supportant la diffusion d'information au sein d'établissements scolaires. Cette application, développée depuis 2005, est actuellement déployée dans trois institutions, et se compose de 23 services Webs et 24 processus métiers. Le processus métier de récupération d'information agrège les informations disponibles, et les délivre sur des écrans publics. L'application évolue de manière régulière, pour intégrer les nouvelles demandes des utilisateurs, comme par exemple l'ajout de nouvelles sources d'informations, ou l'amélioration de l'expérience utilisateur. Ces évolutions localisées peuvent nuire aux performances en impactant l'ensemble du processus.

Pour illustrer ce point, nous restreignons JSEDUITE à l'un de ses sous-ensembles caractéristiques : PICWEB (?). Cette application met en œuvre un processus métier de récupération d'images sur le Web. Dans une première version décrite en fig 1(a), le processus métier récupère les images disponibles sur la source de données exposée par le service de gestion d'images Picasa. L'évolution immédiate demandée par les utilisateurs fut de rajouter dans le processus une invocation du service concurrent Flickr, afin d'augmenter le nombre d'images récupérées par le système (voir fig 1(b)). A l'usage, les utilisateurs ont ensuite demandé que les deux jeux d'images soient mélangés aléatoirement, afin de rendre moins prévisible les images diffusées sur les écrans de diffusion (fig 1(c)). Or, cette évolution, bien que minime par rapport à l'évolution précédente, a provoqué un ralentissement significatif global de toute l'application sur les trois sites en production lors de son déploiement, obligeant un arrêt rapide et un redéploiement de la version antérieure.

---

1. <http://www.jseduite.org>

Pour comprendre l'origine du problème, nous devons raisonner sur le nombre de messages échangés au sein du système. Une approximation asymptotique du nombre de messages échangés au sein du système pour les deux premières versions est de l'ordre de  $O(n)$ , où  $n$  est le nombre d'images obtenues des sources de données. Or, le service de mélange aléatoire est mis en œuvre par un processus métier qui va procéder à  $n^2$  appels au service d'échange d'éléments pour obtenir un tableau convenablement mélangé. La complexité asymptotique de ce service est de l'ordre d' $O(n^2)$ . Cette complexité s'est «viralement» propagée au processus PICWEB, ainsi qu'à tous les processus reposant sur lui. De fait, la complexité asymptotique du système a changé d'échelle, générant ce ralentissement massif et rédibitoire sur la totalité de l'architecture. Nous aurions préféré nous rendre compte avant le déploiement du nouveau système que `Shuffle` était la cause du ralentissement du système.

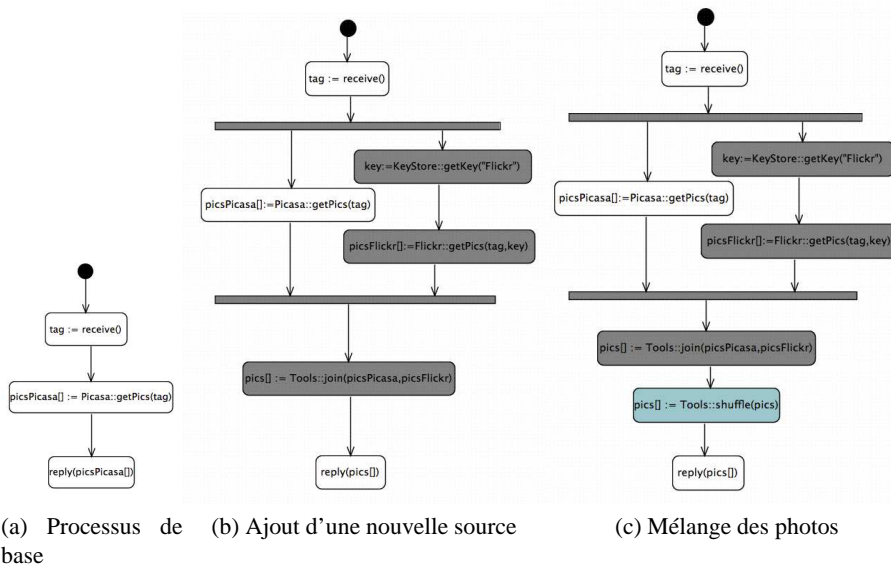


Figure 1 – Différentes versions de PICWEB

### 3. Evaluer l'impact d'une évolution

Pour offrir la possibilité de raisonner sur l'évolution d'un système tel que JSE-DUITE, nous proposons un canevas à base de modèles pour représenter l'application lors de sa conception mais aussi et surtout à l'exécution. Il s'agit non seulement d'avoir une vue du processus métier mais aussi de modéliser les différentes propriétés de qualité de service que l'on souhaite évaluer. Cette évaluation peut se faire en utilisant des valeurs approximées, lors de la conception par exemple, ou des informations de contexte puisque beaucoup de propriétés de qualité de service (comme le temps d'exé-

cution d'une requête par exemple) dépendent de valeurs connues seulement à l'exécution. En mesurant des propriétés de l'application et de son contexte, il est possible de raffiner, confirmer ou contredire l'approximation établie à la conception et de plus finement évaluer l'impact d'une évolution. C'est sur cet ensemble de données que nous voulons raisonner.

La figure 2 est un métamodèle de processus métiers. On représente les processus comme un ensemble d'Activity et de Relation. La métaclasse Property associe une valeur à une activité du processus ou à l'ensemble du processus. Cette valeur peut être calculée pendant une analyse (Computed), précisée par le concepteur (UserSpecified) ou mesurée (Monitored).

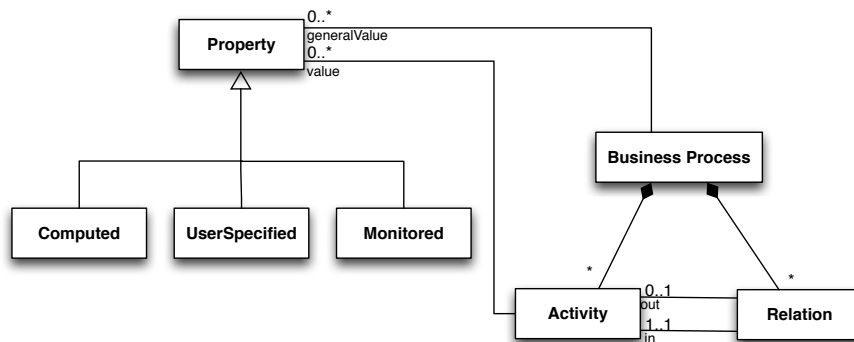


Figure 2 – Proposition de métamodèle pour représenter les propriétés non fonctionnelles sur des processus métiers.

L'approche que nous souhaitons mettre en œuvre repose sur les différentes valeurs calculées et mesurées. Il faudra être vigilant à ce que les valeurs provenant du contexte et les approximations ne se contredisent pas, afin d'avoir toujours une vue consistante du système. Ce genre de situation pourrait se produire si une approximation est erronée ou pas suffisamment précise. Dans ce cas, c'est au concepteur de détailler son approximation ou de ne considérer que les informations de contexte. Nous devons également nous poser la question de la fréquence des mesures du contexte d'exécution, afin d'être suffisamment précis sans alourdir le système par des mesures trop fréquentes.

Nous nous baserons sur l'ensemble de ces informations pour savoir si le système répond aux exigences. Cela implique l'expression de contrats de propriétés non fonctionnelles. Si après une modification le système viole une propriété non fonctionnelle, le canevas tentera d'identifier quelle en est la cause. Il s'agira d'identifier si le problème provient de la modification elle-même ou des effets de cette modification sur le reste du système. Pour cela, une analyse d'impact identifiera les effets de la modification. Le concepteur pourra alors déterminer en quoi sa modification a pu indirectement violer la propriété. L'analyse d'impact peut également nous être utile pour ne pas avoir à revérifier tout le système. En effet, en vérifiant uniquement

les éléments impactés par la modification, on économiserait une vérification complète pour chaque modification. Cela serait particulièrement bénéfique dans le cas où  $T_{calcul}(\Delta) + T_{calcul}(Impact) \ll T_{calcul}(Syst)$ . Enfin, nous étudierons le cas où l'impact d'une modification sur une propriété entraîne la modification d'une autre propriété. Dans cette situation, l'élément initiateur n'agit pas directement sur la propriété en défaut, ce qui cache la cause réelle du problème. Nous voulons trouver comment cet impact en chaîne peut être détecté afin de remonter à la source du problème.

#### 4. Travaux connexes

La problématique de la détermination de l'impact d'une modification est étudiée dans différents domaines, comme par exemple la maintenance logicielle ou la visualisation de programme. Nous donnons ci-dessous deux travaux connexes portant sur la détermination de l'impact dans le domaine des modèles et des processus métiers.

Blanc (?) propose de réduire le nombre de règles de consistance d'un méta-modèle à vérifier au cours de son évolution. Pour cela, Blanc construit une matrice d'impacts stockant pour les actions de base de construction du modèle (tel que `addElement`, `addProperty`, ...) les règles à révérifier. En analysant l'incrément, on peut ainsi sélectionner les règles étant impactées par la modification d'éléments. Son approche étant applicable à tout méta-modèle, elle pourrait s'appliquer sur un méta-modèle de processus métier. Nous nous différencions de cette approche en apportant une connaissance spécifique au domaine : dans notre exemple, nous pouvons directement éliminer les éléments du modèle antérieurs au point d'insertion de l'incrément. Cette spécificité devrait nous permettre d'être plus efficace que la méthode générale.

Krogmann (?) se focalise sur la propriété de performance dans les processus métiers. Ces travaux s'appuient sur les résultats obtenus des concepts d'ingénierie des performances logicielles (SPE), permettant de prédire en analysant l'architecture, les ressources, etc., les performances en terme de temps d'exécution et de charge. Krogmann utilise ces données brutes pour visualiser dès la conception les problèmes sur l'architecture logicielle. Cette approche se concentrant sur la performance ne cherche pas à identifier la cause d'un problème, mais à synthétiser les informations provenant des analyses de SPE pour aider le concepteur.

#### 5. Conclusion

Dans ce papier, nous avons présenté les prémices d'une approche s'appuyant sur les modèles pour déterminer l'impact de l'évolution d'un système pour une propriété donnée. Notre approche repose sur l'utilisation de modèles pour représenter l'application à la conception et à l'exécution. Notre but est de pouvoir apporter une aide à l'utilisateur pour identifier les problèmes que pourraient poser une évolution. Dans un futur proche, nous allons implémenter un prototype pour vérifier la faisabilité de notre approche. Nous étudierons ensuite d'autres propriétés dans le but d'identifier leurs

possibles interactions, potentielles sources de conflits et pouvant s'impacter les unes sur les autres.

### **Remerciements**

Ce travail a été soutenu par le Ministère Français de l'Education Nationale et de la Recherche, le Conseil Régional du Nord-Pas de Calais et le FEDER via le *Contrat de Projets Etat Region* (CPER) 2007-2013.

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

1. ARTICLE POUR LA REVUE :

*7ièmes Journées sur l'Ingénierie Dirigée par les Modèles – 06/2011*

2. AUTEURS :

*Alexandre Feugas — Sébastien Mosser — Anne-Françoise Le Meur —  
Laurence Duchien*

3. TITRE DE L'ARTICLE :

*Déterminer l'impact d'une évolution dans les processus métiers*

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

*Impact de l'évolution d'un process métier*

5. DATE DE CETTE VERSION :

*3 mai 2011*

6. COORDONNÉES DES AUTEURS :

- adresse postale :  
Equipe-Projet INRIA ADAM, Université Lille 1, LIFL UMR CNRS 8022  
INRIA Lille - Nord Europe  
40 Avenue Halley - Park Plaza - Bât A  
59 650 Villeneuve d'Ascq  
prenom.nom@inria.fr
- téléphone : 00 00 00 00 00
- télécopie : 00 00 00 00 00
- e-mail : Roger.Rousseau@unice.fr

7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :

$\LaTeX$ , avec le fichier de style `article-hermes.cls`,  
version 1.2 du 03/03/2005.

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](mailto:revues@lavoisier.fr)  
Serveur web : <http://www.revuesonline.com>