Un processus de développement pour contrôler l'évolution des processus métiers en termes de QoS

Alexandre Feugas, Sébastien Mosser, and Laurence Duchien

Inria Lille-Nord Europe, Univ. Lille 1, LIFL (UMR CNRS 8022)
SINTEF IKT, Oslo, Norvège
{Alexandre.Feugas,Laurence.Duchien}@inria.fr
Sebastien.Mosser@sintef.no

1 Motivations

Les systèmes informatiques sont de plus en plus complexes. Leur développement doit être rapide et réactif afin de pouvoir répondre à des besoins utilisateurs en constante évolution. Les architectures orientées services offrent des éléments de solutions pour palier ce problème en proposant le concept de *service*, l'encapsulation d'une fonctionnalité donnée, et en offrant un couplage lâche des services constituants le système. Toutefois, si ces caractéristiques permettent de faire évoluer facilement le système, il n'en reste pas pour autant difficile de comprendre concrètement l'effet d'une évolution sur le système, notamment en terme de Qualité de Service (QoS) [2]. Nous présentons ici un canevas de développement nommé *Service Modeling for Impact of evoLution framEwork* (SMILE), qui, associé à un processus de développement, permet de contrôler l'évolution de processus métiers en terme de QoS.

2 SMILE : un Processus de Développement Outillé

SMILE est un processus de conception et de mise au point qui offre la possibilité d'étudier les valeurs de propriétés de QoS d'un système orienté service. L'étude du système s'effectue à la conception, en l'analysant pour déterminer les valeurs des propriétés, et à l'exécution, en calculant les informations manquantes qui n'ont pas pu être déterminées. En connaissant la valeur de QoS du système étudié, SMILE vérifie si le système remplit les exigences des utilisateurs. Enfin, notre approche donne la possibilité de contrôler l'évolution d'un système (en cas de changement dans les spécifications ou de violation de la QoS) par l'estimation de son effet sur la valeur de la propriété de QoS.

SMILE repose sur la coopération de deux profils d'acteurs, *l'expert QoS* en charge de la définition des propriétés de QoS du système, et *l'architecte du processus métier* en charge de la définition du processus global, tout au long des cinq étapes détaillées ci-dessous.

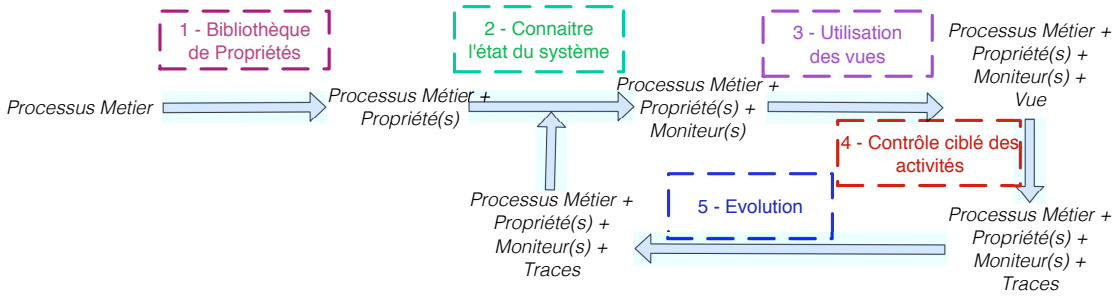

Figure 1. L'approche SMILE

1 - Une Bibliothèque de Propriétés - Le rôle de l'expert QoS est de définir les propriétés de QoS utilisée, de la conception à l'évolution du système. Cela se traduit par la description des analyses de processus métier pour (1) évaluer à la conception la valeur de QoS du processus métier et par composition du système complet, (2) mettre en place les informations de contrôle du système pour une propriété à l'exécution, et (3) gérer les effets de l'évolution. Ces outils sont utilisés dans les étapes suivantes.
2 - Connaître l'état du système en terme de Qualité de Service - La première étape nécessaire pour raisonner sur la QoS consiste à connaître la valeur de QoS du système étudié. Nous nous concentrons dans le cadre de SMILE sur les propriétés de QoS observables, pour lesquelles une formule est disponible pour calculer la valeur de QoS d'un système à partir de la valeur de QoS de ses composants [3]. Pour une propriété donnée, SMILE analyse le processus métier afin de calculer la formule associée. Nous pouvons alors appliquer cette formule afin d'obtenir les valeurs réelles des activités et du processus complet. Lorsque les informations nécessaires sont uniquement disponibles lors de l'exécution (comme par exemple le temps de réponse d'un service), SMILE enrichit le processus métier avec un mécanisme de contrôle pour capturer les informations manquantes à l'exécution et permettre le calcul a posteriori. Ainsi, après avoir déployé le processus métier, l'architecte du processus métier est en mesure de savoir si l'implémentation actuelle respecte les exigences utilisateurs.
3 - Utilisation de vues pour raisonner sur la qualité de service - Une vue [1] est une abstraction permettant de masquer les détails hors du champ d'étude de la propriété pour se concentrer sur les éléments du processus ayant un effet sur la valeur de la propriété. Cela permet notamment de s'abstraire de la granularité fine des valeurs de propriété des services pour pouvoir raisonner au niveau du système de manière plus simple. Cette séparation en termes de vues permet d'isoler la propriété et de mettre en évidence, lors d'une violation des exigences de l'utilisateur, quel est le facteur ayant causé cette violation.
4 - Contrôle ciblé des activités - SMILE essaie de déterminer lors de la conception les valeurs de propriétés. Si le calcul des valeurs nécessite des informations mesurables à l'exécution, SMILE va, de manière sélective, récupérer à l'exécution des informations de contrôle du processus pour capturer les informations manquantes. De cette manière, un sur-coût en terme de capture est évité.
5 - Calcul de l'effet d'une évolution sur la Qualité de Service - SMILE permet d'exprimer l'évolution sous formes d'opérations applicables aux processus métiers. Il analyse le résultat de l'évolution dans le but de déterminer pour une propriété donnée quelles seront les activités affectées directement ou par effet de bord. Par exemple, modifier une variable dans une activité peut entraîner des changements dans d'autres activités. Connaître ce sous-ensemble d'activités du processus métier permet de re-vérifier uniquement la valeur de QoS de ces activités, faisant gagner un temps non-négligeable dans la re-vérification. De plus, en identifiant l'ensemble des éléments affectés, l'architecte des processus métiers est en mesure de comprendre plus facilement quel a été l'effet de son évolution.

Notre approche permet d'avoir une analyse fine de la QoS d'un système de la conception à l'évolution, permettant de s'assurer du maintien de la QoS dans le temps. SMILE offre l'automatisation de l'identification des parties du système à re-vérifier, grâce à une modélisation fixe des propriétés et des systèmes étudiés. Dans un futur proche, nous souhaitons alimenter notre bibliothèque avec d'autres propriétés, faciliter l'expression des moniteurs dans la description des propriétés, et intégrer une approche auto-adaptative prenant en compte les informations de SMILE pour planifier une adaptation permettant de corriger une violation de la QoS.

Références

1. Clements, P., Garlan, D., Little, R., Nord, R. et Stafford, J. : Documenting software architectures : views and beyond. In : 25th International Conference on Software Engineering, 2003.
2. Cheng, B., de Lemos, R., Giese, H., Inverardi, P., Magee, J., et al. : Software Engineering for Self-Adaptive Systems : A Research Roadmap. In : Software Engineering for Self-Adaptive Systems,2009.
3. Crnkovic, I., Larsson, M., Preiss, O. : Concerning Predictability in Dependable Component-Based Systems : Classification of Quality Attributes, In : Dependable Component-Based Systems, 2005.