

EVA4Fractal : Adaptation de composants Fractal basée sur des événements

Julien Dormoy*, Aloïs Dreyfus*, Olga Kouchnarenko*

* Université de Franche-Comté - LIFC,
16, route de Gray - 25030 Besançon, France
prenom.nom@lifc.univ-fcomte.fr

Résumé. EVA4Fractal (pour *Event-based Adaptation policies for Fractal components*) est un outil dédié à la reconfiguration dynamique des systèmes à composants Fractal. Le défi est de construire des systèmes à composants dignes de confiance et de simuler leurs exécutions en considérant des exigences aussi bien fonctionnelles que non-fonctionnelles. La prise en compte de politiques d'adaptation tenant compte d'événements issus de l'environnement du système considéré constitue la principale nouveauté de cet outil.

1 Introduction

Le travail présenté dans cet article s'est déroulé dans le cadre du projet ANR TACOS¹. L'objectif de ce projet est de proposer une approche par composants pour la spécification de systèmes sûrs, depuis l'expression des besoins jusqu'à une spécification formelle, en utilisant ou adaptant des langages et des outils existants. Le domaine d'application est celui du transport terrestre, en particulier le concept CyCab, un véhicule électrique à conduite automatique ou semi-automatique pouvant se déplacer seul ou dans un convoi de véhicules. Les systèmes considérés, à la fois distribués et embarqués, nécessitent l'expression de propriétés fonctionnelles et non-fonctionnelles, incluant entre autres des contraintes temporelles et de disponibilité.

Le point de vue sur les composants adopté et supporté par notre outil est celui de Szyperski (1999), où un composant est une unité de composition logicielle décrite par ses interfaces. EVA4Fractal utilise et soutient le modèle à composant Fractal classique Bruneton et al. (2006). Nous nous appuyons, dans ce travail, sur une extension de Fractal proposée dans Chauvel et al. (2009), appelée Tangram4Fractal, qui permet à l'utilisateur de définir des politiques d'adaptation de haut niveau. Pour spécifier les politiques grâce à des aspects qualitatifs, cette extension est basée sur la logique floue Zadeh (1975). Dans le but d'exprimer d'autres aspects non-fonctionnels, EVA4Fractal exploite les travaux sur les logiques à base d'événements Lee et al. (1999); Gonnord et Babau (2009). Il est à noter que notre travail s'inscrit dans le cadre des travaux pour supporter la flexibilité par changement, en accord avec la taxonomie de flexibilité donnée dans Schonenberg et al. (2008).

Dans la section 2, nous présentons l'architecture d'EVA4Fractal. La section 3 décrit les différentes parties de l'interface graphique de l'outil avant de conclure dans la section 4.

1. ANR-06-SETI-017, <http://tacos.loria.fr>.

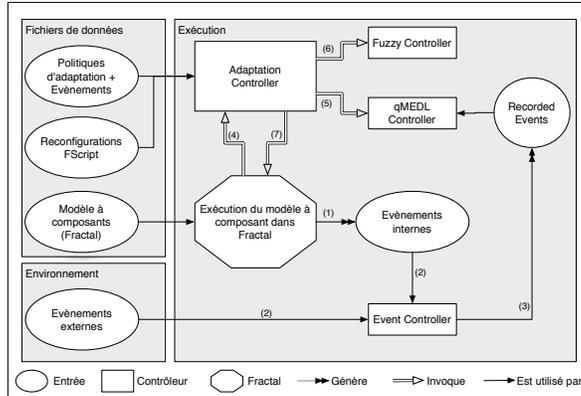


FIG. 1 – Structure interne de l'outil

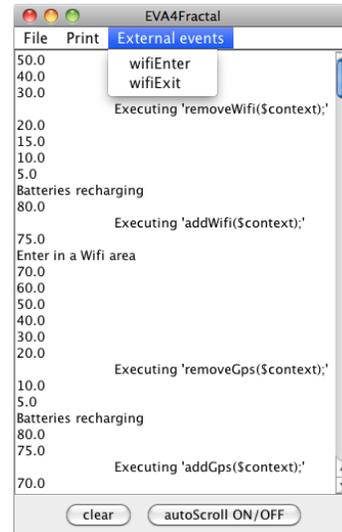


FIG. 2 – Exécution avec l'interface graphique

2 Architecture de EVA4Fractal

Dans cet article, nous décrivons l'outil supportant les propositions d'extensions de Dormoy et Kouchnarenko (2010). Plus précisément, nous proposons d'étendre le langage des politiques d'adaptation de Chauvel et al. (2009) en exploitant une logique à base d'événements, appelée qMEDL (Gonnord et Babau (2009)). En plus des événements spécifiés dans EDL (pour *Event Definition Language* (Lee et al. (1999))), cette logique permet de spécifier des aspects non-fonctionnels quantitatifs et temporels dépendant, en particulier, des événements.

Lors de la simulation d'un modèle à composants, EVA4Fractal permet la prise en compte d'événements provenant à la fois de l'environnement mais aussi des événements internes de ce composant (appels à des sous-composants par exemple). En fonction de ces événements, l'outil active, désactive, ajoute ou supprime un composant en fonction de politiques d'adaptation définies par l'utilisateur. L'architecture de l'outil est présentée dans la figure 1.

2.1 Simuler le modèle à composants

Pour simuler le modèle à composants, nous considérons deux types d'événements : externes et internes. Les événements externes sont déclenchés par l'environnement des composants, ils ne sont pas spécifiés dans le modèle à composants. Par exemple, les événements externes permettent de simuler l'entrée et la sortie des zones signalées par des capteurs. Les événements internes correspondent aux appels aux interfaces des composants. Fractal instancie alors le modèle à composants et simule son exécution. Pendant la simulation, le modèle à composants et l'environnement déclenchent respectivement des événements internes (1) et externes. L'EventController les détecte (2), et les sauvegarde dans une table d'événements

(3). Pour garder une trace des occurrences des événements, cette table contient les noms des événements et la valeur de chaque attribut des composants. Pour pouvoir simuler les événements externes, nous avons développé une interface graphique permettant à l'utilisateur de les déclencher lors de l'exécution (voir figure 2).

2.2 Adapter dynamiquement le modèle à composants

Lors de la simulation, l'`AdaptationController` est soit appelé par un composant, soit exécuté périodiquement (4). L'`AdaptationController` parse le fichier de politiques d'adaptation et appelle le `qMEDLController` pour évaluer les expressions qMEDL apparaissant dans les politiques (5). Comme qMEDL est une logique quantitative et temporelle basée sur les événements, le contrôleur a besoin de la table contenant les événements enregistrés. Le `FuzzyController` calcule l'utilité de reconfigurer le modèle (défini dans les politiques d'adaptation) qui va permettre de sélectionner la reconfiguration la plus importante à exécuter (6). L'`AdaptationController` simule la reconfiguration du modèle à composants en exécutant un code FScript² donné par l'utilisateur (7).

3 Interface Graphique

L'interface graphique³ (figure 3) associée à EVA4Fractal, développée en Java, permet de charger un projet EVA4Fractal en détectant les politiques d'adaptation et les reconfigurations. Elle peut être aussi exploitée pour charger les modèles Fractal sans politiques d'adaptation.

Une fois le projet chargé, le modèle à composants apparaît sous la forme d'une arborescence dans la partie *Components* de la figure 3. A partir de l'arborescence, les différents fichiers représentant le modèle à composants peuvent être ouverts afin de les visualiser et/ou les modifier. Les parties *Adaptation policies* et *Reconfigurations* listent et permettent respectivement d'ouvrir les politiques d'adaptation et les reconfigurations définies dans le projet Fractal. Le menu *Compilation* permet de lancer l'exécution du modèle à composants.

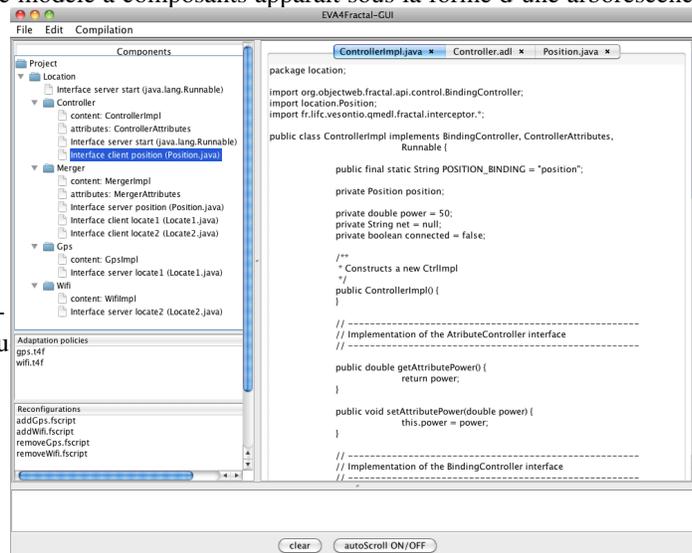


FIG. 3: Interface graphique de l'outil

2. <http://fractal.ow2.org/fscript/>

3. L'interface graphique est indépendante de celle présentée dans la section 2 qui se déclenche lors de l'exécution du modèle à composants.

4 Conclusion

L'outil présenté dans cet article permet aux utilisateurs d'adapter et/ou de reconfigurer les modèles à composants Fractal en tenant compte des différents événements internes et externes aux composants. La caractéristique principale de l'outil consiste au respect du principe du traitement séparé propre aux approches de développement des systèmes à composants. En effet, la prise en compte des événements dans les politiques d'adaptation ne nécessite pas la modification de composants ni de l'architecture globale du système. De cette manière l'outil supporte bien le principe de réutilisation de composants. Nous sommes en train d'étudier la faisabilité et les particularités d'un développement similaire pour l'implantation Fractal dédiée aux systèmes embarqués.

Une étude de cas issue de Dormoy et Kouchnarenko (2010) permet de reconfigurer un composant de localisation d'un CyCab en fonction de l'énergie dont il dispose et de la zone dans laquelle il se trouve. L'outil ainsi que l'étude de cas concernant le composant de localisation sont disponibles à l'adresse suivante : <http://lifc.univ-fcomte.fr/~jdormoy/index.html>.

Références

- Bruneton, E., T. Coupaye, M. Leclercq, V. Quéma, et J.-B. Stefani (2006). The Fractal component model and its support in Java. In *Software Practice and Experience, special issue on Experiences with Auto-adaptive and Reconfigurable Systems*, Number 36(11-12) in LNCS.
- Chauvel, F., O. Barais, N. Plouzeau, I. Borne, et J.-M. Jézéquel (2009). Composition et expression qualitative de politiques d'adaptation pour les composants Fractal. In *Actes des Journées nationales du GDR GPL 2009*, Toulouse, France.
- Dormoy, J. et O. Kouchnarenko (2010). Event-based Adaptation Policies for Fractal Components. In *AICCSA2010*. To appear.
- Gonnord, L. et J.-P. Babau (2009). Quantity of resource properties expression and runtime assurance for embedded systems. In *AICCSA2009*.
- Lee, I., S. Kannan, M. Kim, O. Sokolsky, et M. Viswanathan (1999). Runtime assurance based on formal specifications. In *PDPTA*, pp. 279–287.
- Schonenberg, H., R. Mans, N. Russell, N. Mulyar, et W. M. P. van der Aalst (2008). Process flexibility : A survey of contemporary approaches. In *CIAO! / EOMAS*, pp. 16–30.
- Szyperski, C. (1999). *Component Software Beyond Object-Oriented Programming*. Addison-Wesley and ACM Press.
- Zadeh, L. (1975). The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences 1*, 119–249.

Summary

EVA4Fractal is a tool dedicated to the dynamic reconfiguration of Fractal component-based systems. The challenge is to build trustworthy systems and to simulate their executions when considering both functional and non-functional requirements.