Synchronous Control of Reconfiguration in Fractal Component-based Systems – a Case Study

Tayeb Bouhadiba, Quentin Sabah, Gwenaël Delaval, Eric Rutten

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Synchronous Control of Reconfiguration in Fractal Component-based Systems – a Case Study

Tayeb Bouhadiba — Quentin Sabah — Gwenaël Delaval — Eric Rutten

N° 7631

May 2011

ISSN 0249-6399 ISRN INRIA/RR--7631--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
GRENOBLE – RHÔNE-ALPES

Synchronous Control of Reconfiguration in Fractal Component-based Systems – a Case Study

Tayeb Bouhadiba , Quentin Sabah , Gwenaël Delaval , Eric Rutten

Theme :
Équipe-Projet SARDES

Rapport de recherche n° 7631 — May 2011 — 32 pages

Abstract: In the context of component-based embedded systems, the management of dynamic reconfiguration in adaptive systems is an increasingly important feature. The Fractal component-based framework, and its industrial instantiation MIND, provide for support for control operations in the lifecycle of components. Nevertheless, the use of complex and integrated architectures make the management of this reconfiguration operations difficult to handle by programmers. To address this issue, we propose to use synchronous languages, which are a complete approach to the design of reactive systems, based on behavior models in the form of transition systems. Furthermore, the design of closed-loop reactive managers of reconfigurations can benefit from formal tools like Discrete Controller Synthesis. In this paper we describe an approach to concretely integrate synchronous reconfiguration managers in Fractal component-based systems. We describe how to model the state space of the control problem, and how to specify the control objectives. We describe the implementation of the resulting manager with the Fractal/Cecilia programming environment, taking advantage of the Comete distributed middleware. We illustrate and validate it with the case study of the Comanche HTTP server on a multi-core execution platform.

Key-words: Component-based systems, synchronous programming, reconfigurable systems, discrete controller synthesis.

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Synchronous Control of Reconfiguration in Fractal Component-based Systems – a Case Study

Résumé : Dans le contexte des composants pour systèmes embarqués, la gestion de la reconfiguration dynamique devient de plus en plus importante. Le modèle à composants Fractal et son implémentation MIND, fournissent des moyens de contrôle de cycle de vie des composants ainsi que des moyen pour le contrôle des architectures. L'utilisation des architectures intégrées de plus en plus complexes, rend la gestion des opérations de reconfiguration difficile à maintenir par le programmeur. Cette gestion devient plus complexe quand des propriétés globales sur le systèmes doivent être assurées.

Nous proposons d'utiliser des langages synchrones réactifs, reposant sur des modèles comportementaux sous la forme de systèmes de transitions. De plus, notre approches, qui produit un manager synchrone pour la reconfiguration dynamique profite des techniques formelles comme la Synthèse de Contrôleurs Discrets.

Ce papier décrit l'intégration concrète d'un manager synchrone pour la reconfiguration de systèmes-à-composants Fractal. Nous détaillerons notre approche en commençant par la partie modélisation du problème de contrôle sous forme d'espace d'états de configurations, ainsi que la description des propriétés de contrôle. Ensuite, nous aborderons la partie implémentation du manager résultant en Fractal/Cecilia et son intégration dans des applications Fractal distribuées en utilisant le middleware Comete. Nous validerons notre approche au moyen d'un cas d'étude sur le serveur HTTP Comanche sur une plateforme d'exécution multicoeurs.

Mots-clés : Systèmes à base de composants, Programmation Synchrone, Systèmes reconfigurables, Synthèse de contrôleurs discrets.

Contents

1 Introduction

In the context of component-based embedded systems, the management of reconfiguration in adaptive systems is an increasingly important feature. The Fractal component-based framework, and its industrial instantiation MIND, provide for support for control operations in the lifecycle of components. Nevertheless, the use of complex and integrated architectures make the management of this reconfiguration operations difficult to handle by programmers. To address this issue, we propose to use synchronous languages, which are a complete approach to the design of reactive systems, based on behavior models in the form of transition systems. Furthermore, the design of closed-loop reactive controllers of reconfigurations can benefit from formal tools like Discrete Controller Synthesis (DCS).

Using DCS, integrated in a programming language, provides designers for support in the correct design of controllers. This method is different from the usual method of first programming and then verifying. It involves the automated generation of part of the control logic of a system. Discrete control has until now been applied to computing systems only very rarely [20]. An important challenge is to integrate this formal reactive systems design in actual, practical operating systems. An open issue is the identification and correct use of the practical sensors and monitors providing for reliable and significant information; the control points and actuators available in the API of the OS, enabling enforcement of a management policy; the firing conditions for the transitions of the automata.

In this paper we describe an approach to concretely integrate synchronous reconfiguration controllers in Fractal component-based systems. Our contribution is: (i) a synchronous model of the behavior of the reconfigurable components, in the form of the state space of the control problem, and the specification of the control objectives (Section 4); (ii) a component-based architecture for the implementation of the resulting controller with the Fractal/Cecilia programming environment, taking advantage of the Comete middleware. (Sections 5 and 6). We validate it by the case study of the Comanche HTTP server deployed on a multi-core execution platform.

2 Background

2.1 The Fractal Component Model

We introduce Fractal [3][9], a hierarchical and reflective component model and Cecilia [1], a component-base software engineering framework providing a C implementation of this model. Fractal defines components as entities encompassing behaviours and data. A component can be dismentled in two parts: a *membrane* and a *content*. The content is either a set of operations or a finite number of sub-components, which are under the control of the enclosing membrane. Components can be nested at an arbitrary level in a recursive fashion.