

Instanciação de um Framework de Geração Dinâmica de Processos para Reconfiguração Arquitetural em Ambiente OSGi

Fernando Helton Linhares Soares¹, Adilson Barbosa Lopes¹ e
Carlos Eduardo Da Silva²

¹Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte (UFRN) – Natal, RN – Brasil

²Escola de Ciências e Tecnologia
Universidade Federal do Rio Grande do Norte (UFRN) – Natal, RN – Brasil

fernandosoares@ppgsc.ufrn.br, adilson@dimap.ufrn.br,

carlos.silva@ect.ufrn.br

Abstract. *Self-adaptive systems must be able to adapt their behavior or its own structure in response to changes in the environment. One way to adapt is to generate, at runtime, the process that will perform the adaptation. In this work, we present a solution to integrate a framework for the dynamic generation of adaptation plans to the OSGi environment.*

Resumo. *Sistemas autoadaptativos devem ser capazes de adaptar seu comportamento ou sua estrutura em resposta a mudanças ocorridas no ambiente. Uma forma de adaptação é gerar, em tempo de execução, o processo que irá executar a adaptação. Este trabalho apresenta uma solução para integrar um framework de geração dinâmica de planos de adaptação ao ambiente OSGi.*

1. Introdução

Uma forma de lidar com a elevada complexidade dos sistemas de software atuais é através de sistemas autoadaptativos [Cheng et al. 2009]. Sistemas autoadaptativos devem ser capazes de monitorar a si mesmos e ao seu ambiente, analisar os dados monitorados para determinar a necessidade de adaptação, decidir como a adaptação será realizada e, por fim, realizar as adaptações necessárias [Kramer and Magee 2007]. Uma das maneiras de realizar a adaptação de um sistema é gerar, em tempo de execução, o processo que irá executar a adaptação. Uma vantagem dessa abordagem é a possibilidade de levar em consideração características que só podem ser avaliadas em tempo de execução [da Silva and de Lemos 2009].

Da Silva define em [da Silva and de Lemos 2009, da Silva and de Lemos 2011] um *framework* para geração dinâmica de processos capaz de gerar planos de adaptação para reconfiguração arquitetural de sistemas autoadaptativos. Este *framework* utiliza técnicas de transformação de modelos, planejamento baseado em inteligência artificial (IA) e tecnologia de *workflow* para gerenciar a geração e a execução do plano de adaptação.

A tecnologia OSGi¹ é um conjunto de especificações que define um modelo de componentes dinâmicos para Java, reduzindo a complexidade do software através uma arquitetura modular para sistemas distribuídos de larga escala até sistemas embarcados. Aplicações OSGi podem ser compostas dinamicamente por diferentes unidades modulares chamadas *bundles*. Na nossa abordagem, consideramos *bundles* como componentes. A especificação OSGi define também, mecanismos para gerenciar o ciclo de vida dos *bundles* dinamicamente, permitindo instalar, desinstalar, iniciar ou parar *bundles*.

O objetivo deste trabalho é propor uma solução para integrar a geração dinâmica de processos com plataformas modulares desacopladas. Nossa solução visa a aplicação do *framework* de geração dinâmica de processos para gerar planos de adaptação sobre o ambiente OSGi. Nosso principal interesse é avaliar como a abordagem de Da Silva se comporta em um novo ambiente. Para isso, apresentamos neste trabalho uma solução para a integração do *framework* de geração de processos com o ambiente OSGi. O trabalho encontra-se em andamento e os componentes apresentados ainda não foram implementados. O uso de técnicas de transformação de modelos permite a reutilização da abordagem por diversas aplicações, uma vez que os modelos são automaticamente traduzidos para uma linguagem de planejamento. Além disso, o uso de tecnologias de *workflow* facilita a execução automatizada do plano gerado sobre o ambiente.

2. Sobre o Framework de Geração Dinâmica de Processos

O *framework* para geração dinâmica de processos proposto por Da Silva em [da Silva and de Lemos 2009, da Silva and de Lemos 2011] é composto por um processo de referência e uma infraestrutura de suporte que podem ser customizados para gerar planos de adaptação. A abordagem de Da Silva utiliza técnicas de transformação de modelos para traduzir modelos arquiteturais em problemas de planejamento. A descrição do problema de planejamento é utilizada como entrada para um planejador baseado em inteligência artificial (IA). Essa descrição é feita utilizando a linguagem *Planning Domain Definition Language* (PDDL) [Gerevini and Long 2006], que é uma linguagem padronizada para descrição de problemas de planejamento. O plano gerado a partir da descrição do problema de planejamento é então mapeado em *workflows*, que são executados através de um sistema de gerenciamento de *workflows* (WfMS).

3. Integrando o Framework ao Ambiente OSGi

Na Figura 1 apresentamos uma visão geral da solução proposta. O sistema de gerenciamento de *workflows*, representado pelo componente *WfMS*, é responsável por gerenciar a execução do processo de geração do plano de adaptação, e pela execução do plano gerado. O componente *Model Manager* é responsável por monitorar a aplicação que está sendo executada sobre o ambiente, utilizando *Probes*. O *Model Manager* deve manter um modelo atualizado da aplicação em execução (*App Bundles*) e dos componentes que estão disponíveis no ambiente através do *Bundle Repository*, que contém informações sobre os componentes disponíveis. Assumimos que quaisquer mudanças que ocorram no ambiente são refletidas no *Model Manager*.

O componente *Configurator* detecta a necessidade de adaptação através do *Model Manager* e inicia o processo de geração do plano de adaptação, passando para o *frame-*

¹www.osgi.org/

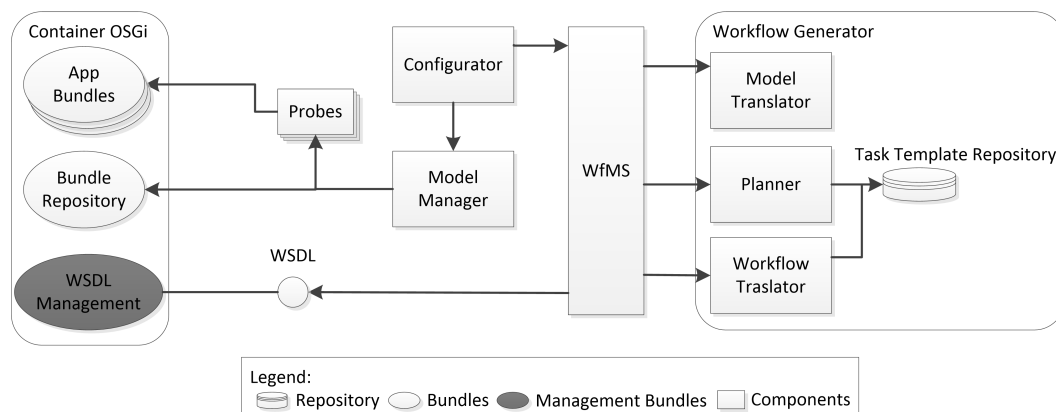


Figura 1. Visão Geral da Solução

work de geração de processos um modelo com a configuração atual da aplicação e a nova configuração que deve ser alcançada. Após receber os modelos do *Configurador*, o componente *Model Translator* é responsável por traduzir os modelos recebidos para uma especificação de problema de planejamento. Essa especificação é então passada para o *Planner* que utiliza as tarefas que estão em um repositório para gerar um plano de reconfiguração arquitetural. O plano gerado é então traduzido em um *workflow* pelo *Workflow Translator* e depois é executado pelo *WfMS*.

As tarefas do plano de adaptação são executadas através de um *bundle* (*WSDL Management*) de gerenciamento que deve estar instalado no ambiente OSGi. Este *bundle* utiliza os mecanismos de gerenciamento de *bundles* do *container* OSGi para executar as tarefas do plano gerado. O componente *WfMS* se comunica com o ambiente OSGi através de uma interface WSDL disponibilizada pelo *bundle Management WSDL*.

Para finalizar a instanciação do *framework* de geração de processos, é necessário definir outros artefatos, tais como: um metamodelo, utilizado para definir modelos arquiteturais; um modelo de domínio PDDL, que captura as características do metamodelo que são utilizadas para o planejamento; as regras de transformação para traduzir modelos arquiteturais em problemas de planejamento; e um conjunto de tarefas disponíveis que são utilizadas para gerar os planos de adaptação.

4. Trabalhos Relacionados

Di Benedetto [Di Benedetto and Werner 2012] também explora um planejador para geração de planos de adaptação. Os planos gerados são traduzidos manualmente para linguagem de *script FScript*, que realiza as adaptações sobre FraSCAti. Na nossa abordagem o plano gerado é traduzido para um *workflow* de maneira automática, e tecnologias de gerenciamento de *workflows* são utilizadas para executar os planos de adaptação.

Em [Ingstrup and Hansen 2009], Ingstrup e Hansen utilizam um planejador IA para gerar o plano de adaptação que é executado como um script. A principal diferença para nossa abordagem é que utilizamos modelos arquiteturais que são traduzidos para problemas de planejamento, permitindo o reuso da nossa solução para diversas aplicações, e os planos gerados são executados através de tecnologias de gerenciamento de *workflow*.

5. Conclusão

Neste trabalho apresentamos uma proposta para geração dinâmica de planos de adaptação sobre o ambiente OSGi. A solução proposta utiliza técnicas de transformação de modelos para traduzir modelos arquiteturais para especificações de problemas de planejamento, o que possibilita a reutilização da abordagem por diversas aplicações. Além disso, utilizamos técnicas de gerenciamento de *workflow* para executar o plano gerado sobre o ambiente, facilitando a execução do plano gerado sobre o ambiente.

Diversos desafios técnicos ainda existem e são base para a definição dos trabalhos futuros. Apontamos para futuros trabalhos: a criação do metamodelo de tal forma que qualquer aplicação OSGi possa se beneficiar da nossa abordagem; a criação do modelo de domínio PDDL, que deve conter as características do metamodelo que são necessárias para gerar os planos de adaptação; a definição das regras de transformação necessárias para traduzir os modelos arquiteturais em problemas de planejamento; e a definição das tarefas que podem ser executadas sobre o ambiente através do *WSDL Management*. Além disso, é necessário a realização de experimentos para demonstrar a viabilidade da nossa solução, além de avaliar sua performance e escalabilidade em um ambiente distribuído.

Referências

- Cheng, B. H., Lemos, R., and et al. (2009). Software engineering for self-adaptive systems. chapter Software Engineering for Self-Adaptive Systems: A Research Roadmap, pages 1–26. Springer-Verlag, Berlin, Heidelberg.
- da Silva, C. E. and de Lemos, R. (2009). Using dynamic workflows for coordinating self-adaptation of software systems. In *Proceedings of the 2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '09, pages 86–95, Washington, DC, USA. IEEE Computer Society.
- da Silva, C. E. and de Lemos, R. (2011). A framework for automatic generation of processes for self-adaptive software systems. *Informatica: An International Journal of Computing and Informatics*, 35(1):3–13.
- Di Benedetto, M. E. M. and Werner, C. M. L. (2012). A declarative approach for software compositional reconfiguration. In *Proceedings of the 11th International Workshop on Adaptive and Reflective Middleware*, ARM '12, pages 7:1–7:6, New York, NY, USA. ACM.
- Gerevini, A. and Long, D. (2006). Preferences and soft constraints in pddl3. In *ICAPS Workshop on Planning with Preferences and Soft Constraints*, pages 46–53.
- Ingstrup, M. and Hansen, K. (2009). Modeling architectural change: Architectural scripting and its applications to reconfiguration. In *Software Architecture, 2009 European Conference on Software Architecture. WICSA/ECSA 2009. Joint Working IEEE/IFIP Conference on*, pages 337–340.
- Kramer, J. and Magee, J. (2007). Self-managed systems: an architectural challenge. In *2007 Future of Software Engineering*, FOSE '07, pages 259–268, Washington, DC, USA. IEEE Computer Society.