Chapter 2

Model Expansion in Model-Driven Architectures

Tsung Lee
National Sun Yat-Sen University, Taiwan

Jhih-Syan Hou
National Sun Yat-Sen University, Taiwan

ABSTRACT

In this chapter, the authors introduce a model expansion method that is used in a new methodology of model composition and evolution for broad design domains. In the methodology, hierarchical model compositional relationships are captured in a model composition graph (MCG) as a schema of designs. An MCG schema can be used as a blueprint for systematic and flexible evolution of designs with three hierarchical model refinement operations: expansion, synthesis, and configuration. In this methodology, due to the need of hierarchical sharing in software and hardware domains, the authors designed an algorithm to achieve conditional and recursive model expansion with hierarchical model instance sharing that is not achievable in other expansion methods. Hierarchical model instance sharing complicates the design structure from tree structures to graph structures. The model expansion algorithm was thus designed with enhanced features of maintenance of MCG instance consistency, path-based search of shared submodel instances, and dependency preserving expansion ordering. The expansion specification and the expansion process are integrated with the MCG-based methodology. Model parameters set by designers and other refinement operations can be used to guide each expansion step of design models iteratively.

DOI: 10.4018/978-1-4666-0897-9.ch002
INTRODUCTION

In the coming era of cloud computing, vast amount of applications will be employed to provide a rich set of cloud services to various users. Such applications can be designed off-the-shelf or synthesized on-the-fly. Hence, there is a problem to enhance design productivity of tremendous amount of application services. Design reuse is considered extremely important to meet such need.

With kinds of support of design reuse, applications can be freely synthesized and deployed in cloud environments. Application designs can be synthesized at static time or on-the-fly. Model-driven architectures (MDA) (Majkut, 2003; Miller & Mukerji, 2003; Trujillo, Batory, & Diaz, 2007; Bhati & Malik, 2009; Perovich, Bastarrica, & Rojas, 2009) play a key role in reusing existing designs to generate application designs. Aspect-oriented programming (AOP) (Kuhlemann, Rosenmüller, Apel, & Leich, 2007; Ubayashi & Nakajima, 2007) joins clustered design aspects together with application-specific code into a main design. Aspects can be clustered in advance and reused dynamically. Feature-oriented programming (FOP) (Batory, 2004; Kuhlemann et al., 2007; Trujillo et al., 2007; Ubayashi & Nakajima, 2007; Apel, 2008; Apel, Kastner, & Lengauer, 2009) forms reusable features in advance and applies them to transform design via stepwise refinement.

In this chapter, we focus on introducing a design methodology that can support methodologies of MDA, FOP, and AOP. A model composition graph (MCG) representation was designed to represent hierarchically composed design models. It naturally reflects the model-driven architectures. We also designed the blueprint representation of model composition graph instances, called model composition graph schema (MCG schema). It can capture feasible design feature composition rules at static time and at dynamic time. Needed hierarchical feature composition models can be provided to compose features at any instant and thus support FOP and AOP models.

A design represented by a model composition graph instance (MCG instance) can be dynamically refined via three kinds of refinements: expansion, synthesis, and configuration. When MCG schemas are formed manually or automatically, designers can utilize these refinement operations to iteratively evolve a partial design or the initial design gradually into a desirable target design. Operations in the MDA methodology can be supported by our three refinement operations. Since all design steps can be classified in such three categories and integrated in this design methodology, it can thus support all design tasks in the life cycle of MDA designs.

In this chapter, we focus on introducing the techniques for one of the model refinement operations, namely, the model expansion method. Model expansion utilizes an MCG schema as a blueprint. Each model schema in the MCG schema specifies how submodels compose the model. Designers can specify the model composition in a procedure with conditional, repetitive, and recursive expansion of submodels. Multiple hierarchically composed model schemas thus form an MCG schema of a design domain.

In software and hardware system designs, we observed the need of hierarchical model instance sharing. Hierarchical model instance sharing exhibits that certain lower level submodel instance may be shared by intermediate model instances under certain higher level model instance’s scope. In practical designs, hierarchical model instance sharing can be classified into two types: sharing pure mechanisms without shared states and sharing non-pure mechanisms with state sharing. For example, a divider may be shared among multiple processors for the first kind of sharing in a processor chip. A level-2 cache may be shared among multiple processors for the second kind of sharing. Such hierarchical model instance sharing is not supported by previous expansion techniques.

We introduce a model expansion method with hierarchical model instance sharing and integrated the method with the MCG design process.