



GRIMACE: GeneRIc MetAmodel for domain Component modElling.

Rahma Bouaziz

► To cite this version:

Rahma Bouaziz. GRIMACE: GeneRIc MetAmodel for domain Component modElling.. 14th European Workshop on Dependable Computing (EWDC 2013), May 2013, Coimbra, Portugal. pp. 181-184. hal-01233221

HAL Id: hal-01233221

<https://hal.science/hal-01233221>

Submitted on 24 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 12563

The contribution was presented at EWDC 2013 :
<http://ewdc2013.dei.uc.pt/>

Official URL: http://dx.doi.org/10.1007/978-3-642-38789-0_16

To cite this version : Bouaziz, Rahma *GRIMACE: GeneRlc MetAmodel for domain Component modElling*. (2013) In: 14th European Workshop on Dependable Computing (EWDC 2013), 15 May 2013 - 16 May 2013 (Coimbra, Portugal).

Any correspondance concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

GRIMACE: GeneRIc MetAmodel for Domain Component modElling

Rahma Bouaziz

IRIT, University of Toulouse
118 Route de Narbonne, 31062 Toulouse, France
rahma.bouaziz@irit.fr

Abstract. Component Based Software Engineering (CBSE) is a popular and widely adopted software engineering paradigm that has proven his usefulness and success to increase reusability and efficiency in various application domains. In this paper, we propose a common metamodel to support CBSE requirements taking into account the specificities of each domain. The resulting modeling framework serves primarily to capture the basic concepts of concerns related to component systems development based on the clear separation between the development process, interactions and the domain knowledge.

1 Introduction

Component-Based Software Engineering (CBSE) [1] has emerged as a promising key technology for developing and maintaining complex systems. CBSE focuses on building large software systems by integrating previously existing software components. The system is constructed by the composition and the connection of these components. It is a good solution to optimize time and cost of software design while still guaranteeing the quality of the software [2].

Various component models have been proposed to deal with system complexity in industrial and academic domains. Variety of those model's applications in constructing systems has proved their usefulness and success. Among these approaches, we can find general-purpose software component models such as Enterprise Java Beans (EJB) [3], CORBA Component Model (CCM) [4] which are well-established for CBSE in generic problem domains. On the other hand, to address a specific domain challenge, specific component models like KOALA [5] are proposed to deal with specialized domains like distributed, embedded or real time systems.

Our first objective is to combine these two approaches – generic and specific component models – in order to propose a metamodel that overcome some of drawbacks and take advantage of each approach. In other terms, we propose a common representation of generic and specific component models taking into account domain specific concerns at the design level.

Model-Driven Engineering (MDE) [9] is also another emerging approach in system development. The use of models has become a major paradigm in software engineering. Its use represents a significant advance in terms of level of abstraction,

continuity, generality, and scalability. In this paper, we deal with these two technologies—CBSE and MDE—to propose a model-based component framework to get a common representation of component for several domains. The main motivation of this work is that the reuse of knowledge and expertise at high level is fundamental to guarantee quality systems. Inspired by the MDE methodology in which software is developed by constructing high level models, we propose a generic component metamodel—to capture generic concepts of CBSE approach –and then separate domain independent aspects of component model from those that are domain specific. The remainder of this paper is organized as follow. In Section 2, we present the proposed approach and describe the GRIMACE metamodel. Section 3 concludes the paper and presents ongoing work.

2 The GRIMACE Approach

The key idea presented in this work is to propose a common representation to target several domains of systems applications (e.g. distributed, embedded, real time, etc.). This representation allows to work at a higher abstraction level, which may significantly reduce the cost of system engineering. Our goal is to define the component based systems as easily and quickly as possible. To do this, the modeling framework must include simple abstractions known by the software developer. Our proposed architecture is based on models, which specify different levels of abstraction, helping developers to manage the inherent complexity of applications and facilitating the communication between the different contributors of software development.

2.1 An Overview

The approach is based on three levels of abstraction (see Fig. 1): GRIMACE metamodel, DICM model and DSCM model. In the following, we detail each of these models.

GeneRIC MetAmodel for domain Component modElling (GRIMACE). In This level – Metamodel level (M2) –we present a generic component metamodel that represents as its name suggests the abstract concepts of component based approach proposed by a large set of component models to describe software architectures. It provides the basic modeling elements for component based system: Component, Connector, Interface, Ports, etc. These elements are the basis for instantiating different component model.

Domain Independent Component Model (DICM). This model is an instance of GRIMACE. This level is intended to generically represent component independently from the application domain. Hence, we will focus on the representation of generic component concepts, interaction and connection between component conforming to the component metamodel (GRIMACE). A DICM model is conform to the GRIMACE Meta-model. Therefore each element of DICM is associated with an element of GRIMACE.

Domain Specific Component Model (DSCM). This model corresponds to a refinement of the DICM model. It combines component specification presented in DICM and the domain specification such real-time constraints. This model is, at the same time, an instantiation of the Generic Component model (GRIMACE) and a refinement of DICM. In this stage the DICM is tailored to a particular application domain e.g. real time embedded or distributed system. For example we can build from the Fractal model (DICM) several specific component models (DSCM) like Fractal distributed, Fractal real-time or Fractal embedded component models.

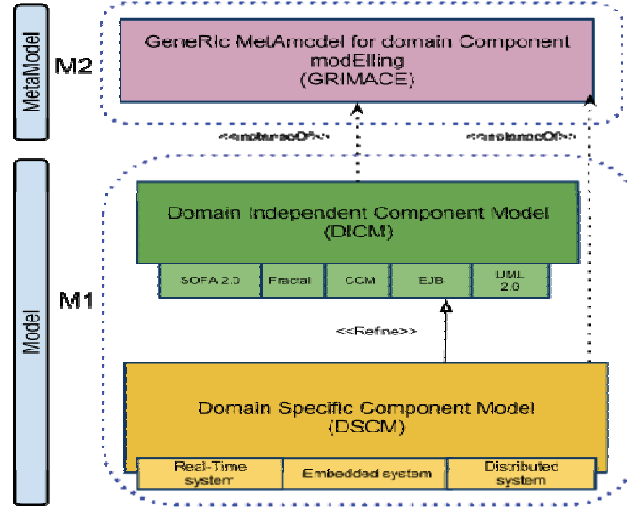


Fig. 1. Overview of the proposed Architecture

2.2 GRIMACE: GeneRiC MetAmodel for Domain Component modELLing

Based on a large study of general purpose and specific component models, we have defined the GRIMACE meta-model. It provides a component model state-of-the-art offering most features available in the domain of component-oriented software architectures.

GRIMACE describes concepts and their relations needed to represent component systems. The proposed meta-model is based on CBSE principles, containing the basic entities *Component*, *Interfaces* and *Connector*. Fig. 2 shows the proposed meta-model. These components will be used to describe both Domain-Independent (DI) and Domain-Specific (DS) models i.e., the originality is the use of the domain independent model and the domain specific model which are reflected in the meta-model level. In the proposed meta-model, a system is described by a set of components that represents the units of computation and data. The communication mode and coordination between these components is encapsulated in connectors. GRIMACE meta-model separates the computation concept represented by components and the interaction concept presented by connectors.

As shown in Fig.2 the generic meta-model defines the abstractions of component based concepts and allows the design of generic component based applications. In addition to these classes and associations, relations can be more precisely defined by adding constraints. Because a lack of space we will not present OCL constraints in this paper.

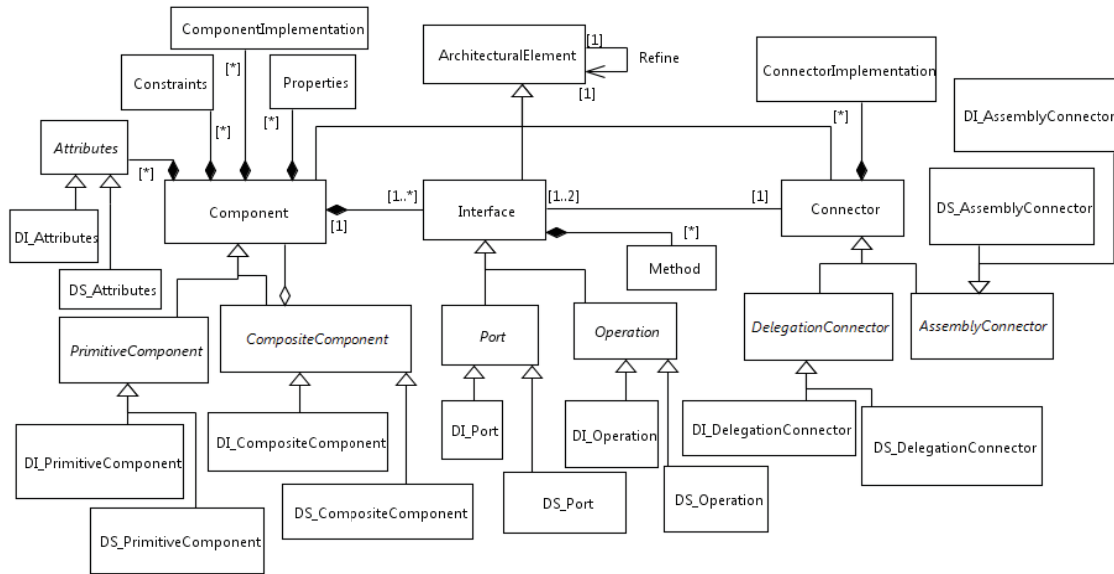


Fig. 2. GRIMACE: GeneRiC MetAmodel for domain Component modElling

3 Conclusion and Future Work

In this paper, we have proposed an overview of concepts—of several component based approaches among academic and industrial fields—to build common representation of component models for several domains. In this way this work gives several ways to go towards universal concepts to model components systems. Among these concepts, we can cite the following: (1) high level of abstraction to capture all the facets of a component assembly, (2) the domain independent design and (3) the domain specific design. Future work includes defining secure models (as instances of GRIMACE metamodel) by using security patterns [6].

References

- [1] Brown, A.W., Wallnau, K.C.: The current state of CBSE. *IEEE Software* 15(5), 37–46 (1998)
- [2] Szyperski, C.: *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley/ACM Press, Wesley, New York (2002)
- [3] DeMichiel, L.G. (ed.): *Enterprise JavaBeans Specification, Version 2.1.*, November 12. Sun Microsystems, Inc., Santa Clara (2003)
- [4] OMG, CORBA Component model, <http://www.omg.org/technology/documents/formal/components.html>
- [5] van Ommering, R., van der Linden, F., Kramer, J., Magee, J.: The Koala Component Model for Consumer Electronics Software. *Computer* 33(3), 78–85 (2000)
- [6] Bouaziz, R., Hamid, B., Desnos, N.: Towards a better integration of patterns in secure component-based systems design. In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O. (eds.) *ICCSA 2011, Part V. LNCS*, vol. 6786, pp. 607–621. Springer, Heidelberg (2011)