

Interests and drawbacks of AOSD compared to MDE

A position paper

C. Kaboré and A. Beugnard

1 Introduction

The quest for modularity is a long term research activity. Historically, functions were the first natural modules. Good practices were edicted to search for low coupling and high cohesion. But functions are often interleaved and module boundaries are difficult to identify. Software architectures, architecture styles are topological abstractions that help reasoning on modules, called components, and their interactions, called connectors. But, considering functional properties is not enough ; a lot of non-functional properties are worth trying to be isolated, placed in modules.

A few years ago, aspect oriented programming [1] was introduced to separate concerns. Concerns, also called aspects, are described independently of the functional part of the system, and are *waved* with it during the development process. Concerns (or aspects) are a way to describe another dimension of modularity. But, concerns are, as functions, highly interleaved [2]. To our knowledge, no standard classification is agreed today.

More recently, the object management group (OMG) has defined the model driven architecture (MDA™) [3] that proposes an other class of concerns : the platform. The platform describes the target environment with its own features. The principle of the MDA is to *merge* a platform independent model (PIM) with a platform description model (PDM) in order to obtain a platform specific model (PSM). Beyond this simple principle, the model driven engineering approach generalizes and suggests to define

more models and more models merging operations (called transformations).

We experienced the MDE approach to automatize the development of a special kind of components called communication components. These components have functional specification and we identified at least 4 concerns : data type implementation, data distribution, data replication, data representation. The full process is described in [4]. This experimentation leads us to a comparison of the AOSD and MDE approaches.

2 Aspect Oriented Process

Aspect oriented approaches rely on a description language and a weaving mechanism. The language allows to specify the different concerns. The mechanism offers operations that are used during the weaving to merge the concern with the program. The weaving mechanism defines the aspect technology approach ; which operators are available, which pointcuts can be used, etc. This defines the join point model. Knowing this model, designers have to specify their concerns using the aspect language.

When many concerns are defined one of the not yet solved problem is to select the order of their weaving. This question point out that weaving aspects (or concerns) is included in a process.

The AOP process can be summarized as follows : first, select a programming language ; second, define join points and a way to identify them (the join point model), and eventually defines concerns. For instance : using Java, and As-

pectJ, the designer defines a logging aspect and weaves it to the program.

3 Model Driven Process

The way model driven approach is used is less formalized than AOP, and still in construction. We used it with a special interpretation ; instead of having only a PIM and a PDM model to merge into a PSM, we keep the PIM (considered as an abstract specification) and define many models, one for each the concerns we were interested in. Each concern gives raise to a meta-model (a specific grammar) that was used to define variants of the same concern. And for each meta-model we defined a tailored transformation that injects this concern into the trunk model (program).

Having many concerns we had to choose their order of application. This is a design choice, that leads to adapted model transformations. Transformations are developed *knowing* the result of the previous concerns merging.

The MDE approach we used can be summarized as follows : first select a modelling language, second define a meta-model that can be used to specify the weaving transformations.

4 Discussion

Separation of concern is an essential design process. Two challenges are how to describe a concern and how apply it ?

The aspect approach makes the choice to offer an universal, generic, mechanism of weaving and requires that the concern designer adopt it and expresses concerns knowing this universal mechanism. All the flexibility is in the concern description.

On the contrary, the model driven approach offers more flexibility. In fact, the concern designers decides first the way he describes the concern, selecting a concern meta-model, and after, elaborates a transformation that injects concerns into the base model. No universal merging (weaving) transformation is required. Every

transformation is tailored.

We argue the MDE approach can be used to separate concerns in a more flexible way that the usual AOP does. Transformations implement automatized steps of the design process. Parts of this process are related to the woven concern and, hence, can be implemented thanks to model transformations.

We also argue that concerns must be selected, analyzed, specified, modeled prior to their weaving process. The concern model influence the weaving transformation, but the implementability of the transformation may also influence the concern model. This is why the flexibility offered by MDE is so important.

Références

- [1] Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingtier, and John Irwin. Aspect-oriented programming. In *European Conference on Object-Oriented Programming*, volume 1241, pages 220 –? 242, 1997.
- [2] Lawrence Chung, Brain A. Nixon, and Eric Yu. *Non-Functional Requirements in Software Engineering*. Springer, January 1999. ISBN-10 : 0792386663, ISBN-13 : 978-0792386667.
- [3] Architecture Board. Model driven architecture (mda). Technical report, Object Management Group, July 2001. <http://www.omg.org/cgi-bin/doc?ormsc/2001-07-01>.
- [4] Eveline Kabore and Antoine Beugnard. Conception de composants répartis par transformations de modèle (, antoine beugnard). In *Journées de l'Ingénierie Dirigée par les Modèles*, pages 117–131, Toulouse, France, 29–30 mars 2007.