CBSE: a Quantitative Approach

Miguel Goulão

Departamento de Informática – FCT/UNL
2825-114 Monte de Caparica, Portugal
miguel.goulao@di.fct.unl.pt

Motivation

Component-Based Software Engineering (CBSE) is concerned with improving Component-Based Development (CBD) practices. In particular, CBSE aims to provide developers with final software system properties predictability based on the analysis of its constituent components. Therefore, we need to develop effective ways for evaluating both software components and component models.

CBD claims include reductions both in development costs and time and the increase of the developed software system's overall quality [1]. These improvements result from reuse: building software from existing well tested building blocks is more effective than developing similar functionalities from scratch. It is important to analyse the reuse related improvement achieved through CBD. The bulk of component related research has been mainly focused on the functionality and composability of components, as well as their specification. The research niche of component quality assessment, with an emphasis on quantitative approaches, is fairly unexplored. The few existing evaluations are performed at a qualitative level, which makes the assessment of the true impact of the above-mentioned claims harder. In our research, we are particularly interested in the quantitative evaluation of software components and their impact on the systems they are integrated in.

Evaluating components is useful in many ways. In project management, internal component complexity evaluation can help estimating the effort related to activities such as the construction, evolution or debugging of software components. For the components' producers, it is a way of promoting the best components in what concerns quality aspects such as reliability and reusability, along with the functionality richness. A well specified detailed contract seems to be an important factor, as it has a positive influence on the component's understandability. Component's procurement can benefit from the same assessment quality aspects, since it helps to select the best candidates from the available components. For the component consumers, their adaptability and reusability are also key factors. Those quality factors can be accessed from the component contract's specification.

Research goals and methodological approach

To solve these problems, we need to develop quantitative assessment practices on components and component-based architectures. This will include:

- The definition of a representation of the components and component-based architectures that is independent from a specific platform – we will work on the definition of a components meta-meta-model, as this is the appropriate level of abstraction to define such a representation. Mappings can then be done to the more specific component meta-models. Both the meta-meta-model and the mappings to meta-models will be defined using a combination of UML and OCL, so that we can combine the understandability provided by the UML graphical notation with the precision of OCL. Although OCL is a formal language, its syntax is close to the one of programming languages, making it fairly easy to understand to UML users. With OCL, we can build-in the appropriate well-formedness rules in our meta-meta-model, thus following the currently well-accepted idea of adding precision to modelling activities [2].
- The formal definition of metrics to evaluate components and component architectures – the metrics will be defined to evaluate data from instantiations of the mapped meta-models, using OCL, and building upon our previous experience with OO design metrics collection. Providing a formal and, therefore, non-ambiguous definition of metrics is essential to allow for their third party validation: other research teams can correctly implement the metrics collection on their tools, as well as having access to the ones developed in the realm of this project. Metrics validation will also be performed by us, through controlled experiments.
- The definition of a component-based software quality model that includes the quality evaluation of the components themselves – the model will support different weights for the assessed quality attributes, as well CBD quality heuristics. The calibration and validation of the model will be performed through repeatable controlled experiments. The ultimate goal is to use the model to help detecting design flaws, as well as producing recommendations for improvements on the evaluated components and component based systems. In order to define such a model, one must previously find answers to the following research problems: (i) Which quality attributes should be measured? (ii) How should they be measured? (iii) Who should measure them? (iv) When should they be measured? (v) What is their relative weight? (vi) Which is the appropriate generality level for a quality model to allow a good balance between our ability to compare different artefacts and an easy mapping to those artefacts?

All produced tools will be made available to the community, so that other researchers can try them out and make their own experiments with them. Whenever possible, we expect to conduct joint case studies and experiments with our peers, as these can provide a suitable form of independent validation.

References

[1] Szyperski, C.: Component Software: Beyond Object-Oriented Programming. New York: ACM Press / Addison-Wesley (1998)

[2] Clark, T., Evans, A., Kent, S., Brodsky, S., and Cook, S.: A Feasability Study in Rearchitecting UML as a Family of Languages using a Precise OO Meta-Modeling Approach (version 1.0). pUML Group / IBM (2000)