9
Conclusions

Modularity occupies a pivotal position in the design of software systems. Yet the task of considering the multi-dimensional facets of modularity remains a deep challenge to designers. Building modular software architectures is a challenging task mainly because the designers need to reason and make decisions with respect to a number of concerns that drive the design, from early architectural decisions to detailed design stages. Systematic assessment is essential to support designers in order to build modular designs. Metrics are traditionally fundamental mechanisms for assessing design modularity.

A plethora of software metrics and metric-based heuristic rules have been proposed for assessing architecture and detailed design modularity. Most of these metrics are defined upon module or component abstractions and are targeted at quantifying traditional modularity attributes, such as coupling and cohesion. Nevertheless, existing conventional metrics are not sensitive to the driving architectural concerns. Without concern-specific modularity indicators, it is not possible to quantify the impact of contemporary modularisation mechanisms, such as aspect-oriented software development, on system’s concerns.

Research on concern-based analysis of source code is limited to support the identification and tracing of concerns (e.g. Robillard & Murphy (2007) and Marin et al., (2007)). There are only a few concern-sensitive metrics available in the literature (Ducasse et al., 2006; Wong et al., 2000; Eaddy et al., 2007), but they are solely dedicated to quantifying concern scattering. In particular, metrics quantifying different dimensions of concern dependencies are missing, thereby raising a number of problems:

- The identification of non-localized concerns, i.e. concerns that are spread over several components and do not have well defined boundaries, are not accurately done (Robillard & Murphy, 2007).
- Many of these non-localized concerns are identified in earlier development stages, such as requirements engineering, and are projected into elements of

PUC-Rio - Certificação Digital Nº 0410867/CA

the design artifacts without explicit documentation; as a consequence, even though they have similar relevance to a concern entirely encapsulated into a module, their impact on design modularity cannot be computed with conventional metrics.

- The identification of dependence among concerns is hindered, because conventional metrics are only able to quantify the dependence among concerns whose boundaries are well defined by components. Existing metrics are not able to assess the dependence among non-localized concerns.
- The identification of unstable design elements (components, classes, and so forth) is impaired. Conventional metrics are not able to quantify the influence of non-localized concerns on design elements. A design element is typically unstable if it is influenced by a high number of concerns, as changes related to several concerns impact that design element (Greenwood et al., 2007a; Figueiredo et al., 2008b; Eaddy et al., 2008a).

PUC-Rio - Certificação Digital Nº 0410867/CA

9.1. Contributions

We claim that quantitative assessment of architectural and detailed design modularity must not only be rooted on the component abstraction and traditional modularity attributes. Design modularity assessment must be oriented by the concerns that drive the design conception and evolution. This research work defined a measurement approach that promotes concern as an explicit measurement abstraction in order to close the gap between quantitative modularity assessment and the concerns that drive the design. The contributions of this work, as stated in Chapter 1, are:

1. *A suite of concern-driven architectural metrics.* These metrics are defined upon the concern abstraction. They allow the identification of architectural design flaws and degeneration caused by the poor modularization of architecturally-relevant concerns. They also allow the comparison of alternatives of architecture design solutions in terms of how well architecturally-relevant concerns are modularized. They can be applied to the architecture description of systems, more specifically to a component-and-connector view of a system’s architecture (Clements et al., 2003). The

architectural metrics were presented in Chapter 4, and they can be applied to all types of software architecture, including aspect-oriented software architectures.

2. *A suite of concern-driven detailed design metrics.* As the architectural metrics, the detailed design metrics were uniformly defined upon the concern abstraction. They can be applied either to object-oriented or aspect-oriented design, or to compare both designs. A number of the metrics that constitute our detailed design metrics suite were proposed in previous studies (Sant'Anna et al, 2003, 2004; Garcia et al, 2006b, Cacho et al., 2006a). The contribution of this thesis is the extension of the existing set of metrics by defining new metrics. The new metrics focus on quantifying the contribution of certain concerns on the coupling and size of modules in detailed design, such as classes and aspects. The detailed design metrics were presented in Section 5.3.
3. *A suite of concern-driven design heuristics.* A design heuristic rule is a composed logical condition based on metrics by which design fragments presenting specific problems can be detected. The proposed heuristics rules support the interpretation of the concern-driven detailed design metrics by pointing out specific design modularity-related problems, i.e. design fragments that are negatively affected by the poor modularization of concerns. The design heuristic rules were presented in Section 5.4.
4. *Concern templates.* The notion of concern templates is a notation for documenting architecture elements related to each key concern considered in the measurement process. Concern templates, as well as concern metrics, are paradigm agnostic in the sense that they can be applied to designs structured according to different software decomposition paradigms. A language to support the description of concern templates was presented in Section 6.3.
5. *Concern-Oriented Measurement Tool (COMET).* COMET is a tool that supports the concern-driven measurement at the architecture level. COMET supports: (i) the importation or definition of the architecture description of a system, (ii) the assignment of concerns to design elements, and (iii) the application of the concern-driven architectural metrics. Section 6.2 described each of these elements and the architecture of the COMET tool.

PUC-Rio - Certificação Digital Nº 0410867/CA

6. A series of empirical studies that evaluated the usefulness and applicability of the proposed concern-driven measurement approach. Three studies were undertaken in order to evaluate the architectural metrics (Section 7.2, Section 7.3 and Section 7.4). Four different systems were involved in these studies. The concern-driven architectural metrics were used to perform modularity comparisons between conventional and aspect-oriented architecture of the systems. One study was undertaken to evaluate the applicability and accuracy of the design heuristic rules in order to detect flaws in both object-oriented and aspect-oriented designs. Six systems were used in this study (Section 8.1). Finally, a study was carried out to compare the effectiveness of conventional and concern-driven detailed design metrics in order to identify specific design flaws (Section 8.2).

PUC-Rio - Certificação Digital Nº 0410867/CA

9.2. Future Work

These contributions represent a first effort at supporting a concern-sensitive assessment of modularity from the architectural to detailed design stages. In spite of the benefits identified in the use of the proposed approach, there are many ongoing and future work, some of which are described in the following.

Additional studies

The proposed measurement approach has been evaluated using representative systems from different domains (Chapter 7 and Chapter 8). The evaluation studies provided strong evidence of the benefits and usefulness of the metrics and heuristic rules. However, it is necessary to undertake additional studies in order to assess the metrics and rules in different contexts. In almost all the studies, the metrics were used to compare conventional and aspect-oriented design. However, the metrics are not restricted to the aspect-oriented context. Hence, it is necessary to use them for comparing design alternatives apart from aspect-oriented ones. For instance, the metrics could be used to compare designs of the same system built on the basis of different design patterns. The studies involving architectural metrics focused only on architectures obtained from reverse engineering of the source code. As the source code represents the

PUC-Rio - Certificação Digital Nº 0410867/CA

projection of the architecture design, the studies are important. However the results could have been biased by information that would not be available if the architecture had not been derived from implementation artifacts. Therefore, new studies involving architecture design derived from requirements specification are needed.

Concern assignment

The process of mapping concerns to design elements is critical to the success of the proposed measurement approach. In the undertaken studies, we followed a specific guideline in order to make the concern assignment task more systematic: assign a concern to a design element if the complete removal of the concern requires with certainty the removal or modification of the element. We need to further elaborate on several issues related to this subject. We need to perform controlled experiments in order to evaluate the feasibility and reliability of the concern mapping process in general and, in particular, the concern mapping process supported by the mentioned guideline. This challenge also applies to other researchers working on concern-driven software analysis (Robillard & Murphy, 2007; Eaddy et al., 2008a). In order to make the process of concern assignment more accurate and less time-consuming, we also need to incorporate to our measurement approach more sophisticated techniques and tools for mapping concern to design elements, such as the one proposed by Robillard & Murphy (2007) and Eaddy et al (2008b). However, most of these techniques and tools focus on source code. Therefore, they need to be adapted to architecture and design level.

Detailed design heuristic rules

The suite of detailed design heuristic rules we proposed (Section 5.4) are not exhaustive. Other heuristic rules should be defined to identify different modularity-related flaws or specific bad smells. For instance, God Class (Riel, 1996) and Divergent Change (Fowler, 1999) are bad smells to which heuristic rules may be defined.

PUC-Rio - Certificação Digital Nº 0410867/CA

Architectural heuristic rules

Although some architectural heuristic rules are already supported by COMET (Section 6.2), we need to elaborate more on this issue. The supported architectural heuristics are, in fact, adaptation of some detailed design heuristic rules. We need to carry out empirical studies to evaluate whether they are useful at the architecture design, and, if it is the case, define heuristic rules specific to architecture modularity assessment.

Tool support

A number of improvements are needed to make COMET (Section 6.2) usable in practice, such as (i) making its graphical interface more user-friendly, (ii) providing support for comparison among different releases of the same architecture, and (iii) incorporating more sophisticated techniques for mapping concern to design elements. In addition, we only focused on providing tool support for architectural design measurement so far, because the detailed design measurement is already partially supported by an existing tool called AJATO (Figueiredo et al., 2006). However, AJATO only supports few of the metrics of our suite of detailed design metrics. Therefore, having the experience of building the current version of COMET, we plan to extend it for supporting detailed design measurement as well.

Empirical validation of the metrics

Controlled experiments are needed for empirically validating the proposed concern-driven metrics. Roughly speaking, empirically validating metrics consists of establishing a relation of cause and effect between internal quality metrics, such as our metrics, and external quality attributes, such as maintainability, reliability and reusability, which are measurable after a certain time of use of the software. In fact, Eaddy et al (2008a) have recently carried out three experiments which involved two of our detailed design metrics: Concern Diffusion over Components (CDC) and Concern Diffusion over Components (CDO). Their experiment aimed at testing the hypothesis that the more scattered a concern's implementation is, the more likely it is to have defects. They found a moderate to strong correlation between our metrics and defects for all three experiments, which suggested that scattering may cause or contribute to defects.