

CoCA: A Composition-Centric Approach to Requirements Engineering

Ruzanna Chitchyan, Ian Sommerville, Awais Rashid
Computing Department, Lancaster University, Lancaster, LA2 4YR
{rouza,is,marash}@comp.lancs.ac.uk

Abstract

In this paper we discuss how mutual influences (e.g. conflicts) of different stakeholder concerns can be detected and reasoned about through composition and stepwise refinement. Some concepts from the Aspect-Oriented Software Development paradigm are used to support the composition-centric approach.

1. Introduction

Requirements Engineering (RE) discipline has long appreciated the value of separation of concerns. For instance, viewpoints [1] differentiate requirements on the basis of subjective perspectives of stakeholders; use cases [2] separate per units of useful functionality, etc. However, the issue of integration of such separated concerns has not been fully addressed. In this paper we present the Composition-Centric Approach (CoCA) to RE which tackles the integration issues.

CoCA focuses on modularising concern relationships and using them for requirement composition. Conflicts and inconsistencies between requirements can be detected via composition and addressed early on [3], and an early understanding of relationships between artefacts of later development stages can be formulated. The modular representation of relationships is achieved by encompassing these relationships in composition operators. The concerns and their relationships are gradually refined into more detailed representations through an analysis process.

For the discussion we use an example of an on-line auction system where bids can be placed by registered users who have sufficient credit on their accounts. A security procedure must be observed while bidding.

2. Main Elements of CoCA

In this section we focus on main elements of the CoCA which are concerns, their representations, relationships, composition, and refinements.

2.1. Concerns and their Representations

A *concern* is defined as any matter of interest to the stakeholder with respect to the system (e.g., Security &

Bidding in Figure 1(a)). In RE, concerns are represented via requirements (e.g., Security in Figure 1(a) is made up of R1, R2, R3). Concerns and their representing requirements can initially be identified using any RE approach, such as, for instance, viewpoint analysis or use cases, etc. Concerns can be related to one another in many ways. We use these relationships for composition (e.g., *observe*, *between* in composition of Security and Bidding in Figure 1(a)).

2.2. Composition Elements

The semantics of composition are defined by the composition *actions* and *operators* derived from the concern relationships. The composition actions apply to concerns at specific points (*joinpoints*) at which concerns are bound together.

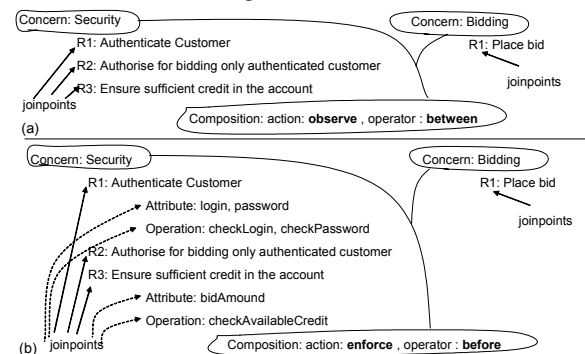


Figure 1: (a) Concern composition; (b) Refinement of concerns and compositions.

2.2.1. Relationships between requirements are identified using support of the WMATRIX semantic analysis tool [4]. It uses corpus linguistics and statistical analysis methods to identify meaning of the words in a given text and categorise them per semantic groups which collect together the words and phrases contributing a similar meaning. The categories used by WMATRIX are those of USAS Lexicon [5].

From the general USAS [5] categories we have identified a small subset of categories which can be classed as related to *effects of actions*. This sub-set of categories is used to analyse the effects of

requirements on each other. The initial set of effective categories is presented in Figure 2.

In CoCA, the relationships are represented as operators and actions (similar to [3]).

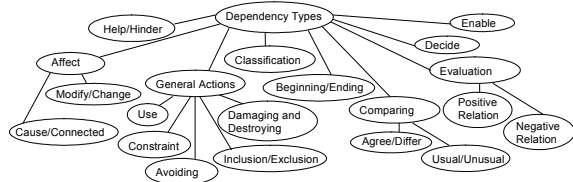


Figure 2: Semantic categories as relationships

2.2.2. Actions. The *effective categories* identified above are used to derive the actions for our composition semantics. These define what effect the composition should have on the composed requirements. For instance, when terms belonging to the “Help/Hinder” group (Figure 2) are identified in a requirement, it is likely that helping/hindering in that requirement occurs in relation to another requirement.

2.2.3. Operators. While actions define what should be done about concern interrelationships, the operators reflect the nature of these relationships. For instance, a temporal interval between two requirements (defined by *between* operator, e.g. Figure 1(a)) can be merely observed without any intervention (*observe* action) or intentionally enforced (*enforce* action, e.g., Fig. 1(b)).

In [3] Rashid et. al. defined a set of composition operators and actions using XML-based composition rules. We too follow a similar representation.

Although we aim to provide a comprehensive set of most used operators and actions, we anticipate that project-specific actions and operators will be required. So, an open set of actions and operators is maintained.

2.2.4. Joinpoints. The relationships between concerns do not always equally affect all elements representing a concern. Some representations can participate in a relationship fully, others partially or not at all. Thus, we need a mechanism for segregating the concern representations (and their sub-parts) that will participate in a certain relationships. This is where we draw on the power of Aspect-Oriented mechanisms such as *joinpoints* and *pointcuts*.

We characterise joinpoints as well defined points in the body of concerns. Since join points are clearly defined, we can reference them and employ other concerns to interact with a given concern at these points. For instance, a requirement in the concern representation is a clearly defined point in the body of a requirements document. Another requirement can be used to replace/augment it (e.g., R1-R3 in Figure 1(a)).

2.3. Refinement

In our model, both the *composition semantics* and the *joinpoint model* are *refinable*. At each refinement level the *joinpoint model* may expose finer-grained *joinpoints* defined for that given level; and the composition actions and operators may be elaborated to more precisely match the relationships of concern bound at the given joinpoints. Thus, composition is supported at different levels of refinement of concerns and their relationships:

- at the early stage of analysis, when only user-level requirements are available, we define composition using general composition operators with reference to requirement statement level joinpoints (in Figure 1(a) joinpoints are the requirements, composition operator is *between*, with *observe* action);
- as more detail is exposed through analysis, more precise operators are used to reference the finer-grained joinpoints (in Figure 1(b) additional joinpoints for attributes and operations are available, with *before* operator and *enforce* action);

3. Conclusion

The CoCA approach builds on [3] by providing the notions of composition operator and joinpoint refinement; enriching the semantics of composition operators with those of concern relationships; providing a principled way of deriving composition actions; and foreseeing concern relationships at later stages of development through composition.

Additionally, the gradual increase of detail in requirements and operators due to the CoCA supported refinement takes the requirements closer and closer to the design space, transparently bridging the gap between the two. Through this refinement, we also preserve the traceability of requirements, thus contributing to improved requirements management.

Acknowledgement: This work is supported by EC Grant IST-2-004349 (AOSD-Europe).

4. References

- [1] A. Finkelstein and I. Sommerville, "The Viewpoints FAQ," *BCS/IEE Soft. Eng. Journal*, vol. 11, 1996.
- [2] I. Jacobson et. al., *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992.
- [3] A. Rashid, A. Moreira, and J. Araujo, "Modularisation and Composition of Aspectual Requirements," in proceedings of AOSD 2003, Boston, Massachusetts.
- [4] P. Rayson, "WMATRIX," Lancaster University, URL: <http://www.comp.lancs.ac.uk/ucrel/wmatrix/>, 2005.
- [5] "UCREL Semantic Analysis System (USAS)," 2005.