"Towards Thread Aware Component Behavior Specifications"

submitted by
Tomas Poch,
Faculty of Mathematics and Physics, Charles University in Prague

To master the complexity of big software systems the partitioning of software architectures into appropriate subsystems is indispensable. Based on this requirement component-oriented software development has emerged as an important subdiscipline of software engineering. Software components represent functional units which collaborate with their environment via distinguished interfaces. Of course, for the final system to work it is crucial that component implementations respect their specifications and collaborate with their environment without communication errors. Thus component-oriented development involves not only the practical side of implementing a component in a particular programming language but has also a strong theoretical flavor directed towards the study of the correctness of components and component communications. Obviously, the latter can only be achieved by means of formal specification and analysis techniques. This, however, opens a gap between the skills of the implementor of a component who is familiar with the programming language constructs used for component implementation but may have less experience in formal specification and analysis. On the other hand, formal methods themselves often tend to high level abstractions such that the relation to concrete programming practice is often neglected. This discrepancy is not sufficiently solved yet. The goal of this thesis is to tackle this problem and to propose a solution in terms of a specification language for components which, on the one hand, is close to usual imperative programming practice while, on the other hand, is subject for formal analysis.

The thesis starts, in Chapter 1, by discussing convincingly the motivation of this work leading to the research goals just mentioned. In Chapter 2, a felicitous overview of the current state of the art in component modeling is given which highlights again the scientific value of the research objectives.
The central results of this work are presented in Chapter 3. In Section 3.1 the author introduces the syntax of threaded behavior protocols (TBPs) which provide a specification language for components. Essential ideas of this language are: 1) to allow imperative constructs for the description of dynamic behavior, 2) to support a clear component structure which separates provided methods and their behavior (reaction) from the actively running threads of a component, 3) to provide specifications (called provisions) determining rules under which a component can be (correctly) used by an environment. The imperative part of TBPs is closely related to the constructs used in object-oriented programming languages, in particular Java. Hence, it meets the requirements of the thesis to support usability by practitioners. On the other hand, the provisions of a component provide the formal basis for stating and checking communication correctness and refinement, thus meeting the requirements for formal analysis.

Section 3.2 focuses on the semantics of TBPs and the analysis of properties for closed and open systems. The semantics of a TBP is defined in two stages: First, a TBP specification is translated into a TBP model where imperative constructs are removed by translating reactions and threads into particular labeled transition systems with assignments (LTSAs). On this stage also the composition of TBP models is defined. On the second stage, a closed TBP model is translated into a single labeled transition system (called closed computation), which describes the integrated behavior of all concurrently running threads as specified by a TBP specification. The semantics formalizes the ideas of concurrent object-oriented programming following Java's paradigm for synchronous method invocation. To split the semantics of TBP specifications into two layers was certainly a good idea. Also the technical details of the semantics are convincing, apart from some inaccuracies which appear in the discrimination of provided, required and internal labels complicating sometimes the understanding.

The semantic notions provide a strong formal basis for the subsequent analysis. First, the author defines adherence/violation of provisions for closed TBP models which is further elaborated by distinguishing different kinds of activity errors (e.g. no activity, bad activity) which may happen in a closed TBP model. To avoid these kinds of errors is indeed an important issue in system development, so the precise formalizations presented here are appreciated. How to check freeness of communication errors is, however, not discussed in the thesis; it is mentioned that a tool is under development. The thesis continues in Section 3.2 to elaborate a framework for refinement of TBP models. The refinement notion is driven by the idea that freeness of communication errors on the specification level should be propagated to the implementation level. Hence, according to the different kinds of communication errors, different kinds of refinements are defined. For instance, refinement w.r.t. bad activity requires that if a specification S is put in an arbitrary environment E such that no bad activity occurs, then also a refinement I (of S) put in the same environment E should not produce any bad activity. Thus refinement is defined as a transitive relation between open TBP models. For proving refinement the author proposes to use a parametric form of alternating simulation, originally introduced by de Alfaro and Henzinger for interface automata, which can be tuned by the parameter predicate to take into account different communication errors. The (parametric) alternating simulation is applied to the observation projection of the provision-driven computation associated to open TBP models. More precisely, the idea of the provision-driven computation is to complete an open TBP model according to what can be expected by an arbitrary environment satisfying the given provisions. Since, in principle, an environment may execute arbitrarily many threads, the author draws the correct conclusion to apply his formalism only under the assumption of a limited number of threads. Finally, the observation projection is applied to abstract from internal actions in a way such that potential erroneous communication can be revealed. The main theorem of the thesis shows that with the technique of parameterized alternating simulation refinement w.r.t. bad activity can be verified, and a similar result holds for refinement w.r.t. no activity. It is pointed out that the particular advantage of using alternating simulations lies in the decidability whether an alternating simulation holds (for finite state
automata as considered here). This is true but, on the other hand, alternating simulations are used in this thesis as a criterion for refinement and not for the refinement definition itself such that decidability of refinement (as defined in the thesis) seems to remain an open question. The theorems stating the conditions under which particular refinements hold, provide a new and important result for component-based development. To accomplish them with tool support, including the computation of observation projections, would be desirable and seems to be under development.

Chapter 3 finishes with an appropriate comparison of the proposal with related work and with a small case-study which highlights the features of TBP specifications but, unfortunately, does not further illustrate methodological issues like refinement and hierarchical development.

**Evaluation summary:**

This thesis presents a significant and novel contribution to the state of the art in component-based development. Without doubts Tomas Poch has proven to be able to develop new and scientifically very valuable results based on his impressive knowledge of the subject. The discussions in the thesis are convincing and the introduced concepts are always well motivated. Although the technical definitions are usually quite clear, sometimes more explanations and illustrations would have been useful, in particular when inaccuracies occur. Altogether, I strongly recommend the acceptance of this thesis.

(Prof. Dr. Rolf Hennicker)

**Attachment:** List of selected questions for the Ph.D defense