This is my report on the PhD thesis of M. Jiří Adámek, titled "Behavior Composition in Component Systems".
This thesis is in the domain of Software Engineering, and more precisely in the area of formal methods for the specification and verification of software systems, that is currently very active. The growing trend of using software components for building complex applications and especially distributed applications makes the analysis of their behavior, and of component interactions, very important. At the same time, it brings in more difficulties, because of the non-functional and often dynamic features offered by component frameworks.

This thesis is organized in 7 chapters.

The first chapter is a short introduction of component models, of component behaviors, and presents the main results and the structure of the thesis.

The second chapter presents a review of background and of related works, starting with a detailed presentation of previous work by the SOFA team that sets up the formal framework that will be the basis for all the developments in the thesis. In particular the definition of the behavior-protocol specification model is not itself part of the thesis, but is used, specialized and extended in this work. Then the author goes through a number of other approaches to behavior specification, either quite recent (pNets, Component interaction automata) or older (Tracta, Wright, RADL), together with a short review of the fundamental semantic frameworks and verification techniques applicable to the domain concerned, namely process algebras and model-checking. Works around some of the most known models of the area, e.g. CCM, are not included, but may be they were judged to be too different from the author approach. My only regret is that the chapter does not include a real comparison between the listed approaches, or may be that later in the thesis we do not find any comparison between the achievements of the author with the other approaches.

The third chapter is a 4 pages summary of the problems addressed in the thesis, and of the results obtained and described in the following chapters. This is an original structuring artifact, and constitutes a clear and valuable introduction to the sequel.

The next four chapters each contains a published paper, with an introduction that links the context, goals, and results of the paper to the skeleton of the thesis. This structure is quite unusual for a PhD thesis. It has the disadvantage of offering a different introduction and background section in each of the chapter, that are repeated, without being identical, in the 4 papers. Moreover, the original papers are included untouched, and this does not allow the author to present them within an homogeneous, formally defined framework, nor to have the detailed formalism and proofs appearing in the thesis. Instead, full versions of the papers, published as technical reports, are pointed in the various bibliography sections, making the understanding of the thesis more complicated.
However, the level of the papers included is very unusual for a PhD thesis (one international and reviewed workshop, 2 international conferences, and one journal paper!), and each of them is clear and well presented, with significant scientific results.

The first included paper is the journal paper, dealing with the detection of errors by model-checking techniques. The errors identified here are errors occurring when assembling components for building more complex software units, and errors occurring when updating components inside an application at runtime. This is clearly an original contribution to the domain, and furthermore an idea that has a very high potential of being imitated in other contexts, and of being very useful in practice. It allows for detecting statically, before running the application, incompatibility between the intended protocols of connected components that could lead to deadlocks or other errors.

The next two papers are extensions of the basic method. They allow a more flexible application of the assembly specification, and thus a more precise detection of the errors, depending on the behavior of the rest of the application and of its environment.
The last paper extends significantly the type of applications that can be addressed by behavior-protocol specifications by allowing some restricted form of unbounded parallelism.

A short evaluation chapter reconsider the thesis work in perspective of the rest of the research done within the SOFA team, and in particular of relations of the methods developed by the author with software tools and with a real-size case-study. I would have appreciated getting more information and more concrete elements in this chapter. For example it would have been interesting to discuss which aspects of the theories were the most useful in the implementation of the tools and in the treatment of the case-study, or on the contrary which were the most difficult to implemented or to use...

Last a very short section with a conclusion, and the presentation of ongoing or future works, essentially in the direction of extending the application field of the last paper on unbounded parallelism. This section states a difficult problem that may require deep changes in the semantics underlying the behavior-protocol formalism, namely redefining the compliance relation in terms of parameterized protocols. I would be interested in hearing the author's ideas on this subject.

Back to the technical content of the thesis, there are a number of questions that are raised, that could either bring a better understanding of the author arguments, or open the discussion towards new developments. Let me state some of them:

- The first one may be thought either political or philosophical. The author states that software components "require" the usage of formal methods, and that many component models indeed are endowed with means of specifying and testing formally behavior correctness. This is alas not yet the case in the real world, despite the number of research efforts, conferences and publications on the subject in the last two decades; how could we encourage and foster more widespread and easy usage of our results ?

- In section 2.3.1, the author comments on the differences between the SOFA and Fractal component models, but not much on the relations between parameterized networks and behavior-protocols at the semantic level. In particular, what are the advantages or disadvantages of having a trace-inclusion based semantics, compared with the LTS/bisimulation based semantics of the pNets ? Also, pNets are able to express unbounded parallelism; how does this relates to your last paper ?

- The Update Atomicity feature of the consent operator deals with the update of a subcomponent that complies with the frame protocol of its predecessor. In real life, the upgrade of a component within a running application usually comes with new functionalities that will then have to be usable by other parts of the system. How do you imagine a verification scenario allowing adding features dynamically ?

- In the "future work" section of the journal paper (p57), is mentionned the difficulty of reasoning about the proper stopping of a (sub-) component. In fact this problem can be still more complicated in the context of distributed components, communicating in an asynchronous way. What changes in the model would this problem require?

- The CADP verification toolset is developed by the VASY team in Grenoble, in a context slightly different (branching time logics and bisimulation-based systems). It has a number of features that could be related to your approaches. In particular, the Projector tool computes behavior products in the context of a behavior of the environment that can be either manually specified, or automatically computed. How does this relate to your approach ? More practically, have you tried using the trace equivalence tool of CADP?

- In the conclusion of the Unbounded Parallelism paper (p 87) there is a reference to a technical report containing detailed development of the paper example. It would be interesting to give orders of size of the expanded system, for this toy example, or possibly for a more realistic case-study. Has the Airport example mentioned in chapter 8 given you some feedback on real-life unbounded parallelism ?

As a summary, methodological contributions in this work, and in particular those brought by the consent operator and to a lesser extend those relative to unbounded parallelism, are definitely novel approaches, and will have impact on the research community. The various facets of the consent operator developed in the first 3 included papers also have a very pragmatic impact on the way errors may be reported to developers in software development environments.
I regret that the method chosen for a "weak assembly" of papers in the corpus of the thesis does not allow for a self-contained formalization, which could provide a strong and homogeneous foundation, with all relative proofs, in the same document, and that separate formalizations and proofs must be looked at in separate technical reports.
This does not withdraw the quality of the creative work, and the high level of the published paper is a good measure of it.

Dr. Eric MADELAINE
INRIA
In Sophia Antipolis, july 6th 2006