Review of a PhD Thesis

Pavel Ježek: Hierarchical Component Models – "A True Story"

Thesis advisor: Prof. František Plášil, Charles University, Prague

Reviewer: Doc. Přemysl Brada, University of West Bohemia, Pilsen

Overview

The presented doctoral thesis of Pavel Ježek is concerned with the identification of component model weaknesses and finding ways to improve the research ones aiming at their better practical relevance. These aims are achieved by an analysis of a selected set of models and by introducing modifications of behavioural specifications for hierarchical models, based on experiences with large case studies and practical applications.

Main Research Findings and Practical Results

The candidate's key contributions can be summarised as follows.

1. Application of behaviour protocol verification in realistically large applications, and related modifications of methods/models employed. This is a valid and worthy contribution, albeit not an overwhelmingly strong one in terms of originality.
2. Formal modeling of component environment and/or interface based on linear temporal logic principles, applied on Microsoft Windows driver verification. This is both formally strong and practically usable achievement, although it would deserve deeper explanation and validation.

The thesis in addition presents the following topics, developed to a lesser depth.

3. An approach to model a subset of architectural reconfigurations at runtime while preserving ability to verify behavioural correctness. This is apparently a wider team effort which reduces its importance in the context of the thesis.
4. Component model classification and conceptual discourses related to various aspects of research and industrial component models, in an attempt to bridge these two worlds. The goal is important, approach interesting and the author shares many good insights, but I was somewhat disappointed with the results because the analysis is shallow and does not argument well the need for hierarchical components – in contrast to the thesis title.

Altogether this shows that the candidate's work covers a commendable breadth of areas, both on the theoretical and practical sides of the component based software engineering discipline.

Most parts of the work are backed or validated by some form of practical realization, as evidenced by concrete performance results and experiences from implementing the case studies. The key papers that are related to the thesis were presented at two international conferences, two workshops with electronic journal post-proceedings, and in a reviewed book chapter; this by itself is a strong record.

Formal and Methodological Issues

Most of the work is the result of team effort which is understandable and common in the area. The thesis nevertheless should have made it more clear which parts are candidate's most direct contributions.

The goals of the thesis are defined in an awkward way. They start in section 1.3 on p.18 as three points which cover rather wide areas -- in fact so much so that especially goal (2) alone is hard to achieve in a single thesis. The revised wording in section 2.7 p.53, narrowed down even further by points in section 3.2.7 p.71, then does not really introduce *goals* (stating open issues to address and by what approaches) but rather *specific achievements* presented up front – "provide examples of two industrial domains...", "evaluate hierarchical component models in two major case studies...", "provide ... approach to deal with complex error traces ...". Thus the fact they were achieved unfortunately looks like a self-fulfilling prophecy.

There is only a weak link between component model analysis in the first part of the thesis and the behavioural verification of hierarchical component models in the latter one. In particular:

- The analysis of component models does not answer questions "why are hierarchical models superior" or even "what really are hierarchical models" with sufficient precision. The key relevant parts, i.e. sections 2.2-2.5 and 3.1, present opinions rather than scientific evaluation and lack clearly conclusive outcomes (desiderata listed in 3.1.5 are broad, imprecise). This is a major issue considering that the core of the thesis contribution relates to hierarchical component models.
- Very little concrete and convincing arguments can be found that deal with problems caused by lack of formal, in particular behavioural specifications in mainstream component models. Chapter 2 omits this issue, chapter 3 does not provide strong research backing to this need; the term "correctness verification techniques" used in 3.1.1 needs precise definition in this context.

Related work does not cover research related to verification by other contract levels and means - type consistency (e.g. McCamant), non-functional property verification (e.g. Tilly, Koziolek), software testing using both standard and stochastic methods.

Some topics would deserve deeper explanations or more extensive examples, augmenting the published papers on which the respective chapters are based so as to make the thesis self-contained. This is the case of chapter 6 (should have covered analysis of the complete CoCoME verification) and chapter 8 which does not discuss the whole environment model using DeSpec and how it is used in verification.

Selected Concrete Issues

Chapter 2: The criticism of Szyperski's definition (section 2.1.1 p.23, also p.27) is in some parts unjustified. Firstly, the definition starts with "A *software* component is…" which clearly refutes the objection that "a unit does not necessarily imply any connection to software engineering (so it can define a hardware component as well)". Secondly, the author suggests that it's a problem of the definition that many (especially run-time) frameworks are not conformant with it. Insufficient consideration is given to a reverse view: that such systems should actually not claim to be component-based. See for reference "architectural components"as defined by Bachmann, definition by Taylor et al or even the foundational understanding by McIlroy from 1968 (all these rather key references are missing in the thesis).

Sections 5.3.1 and 7.5: The "Token dynamism problem" is disputable – in my view it is more a manifestation of incorrect design than a essential problem. The reason is that Token need not be considered a component (that provides services to its clients, cf. sec 2.5) but rather a data element as its name suggests. A TokenManager or similar service-based component could be used in its place, which would provide token state management and validity checking to AccountDatabase and Arbitrator. Also, more realistic cases of architectural reconfigurations could have been considered (What if [a component for] Lufthansa Fly Ticket Database should be added at runtime? What if a new Arbitrator needs to be installed which uses additional VIPCustomerDatabase as an authorization source through a newly added IVipCustomerAuth interface?).

Presentation and Writing Style

In most parts, the text is written in a clear and dense style and is well understandable. Only some sections of chapters 2 and 3 are of lower quality, with too long paragraphs which makes it hard to grasp ideas in the text.

The list of contributions in section 1.4 and in the Conclusion is overly detailed or rather, it includes items which are not scientific contributions but rather just partial steps needed to achieve goals (e.g., "overview of diversity [of] CBSE concepts," "introduction of a case-study," or "provided several motivational examples").

Chapter 2 is missing introductory paragraph stating its goals and contents. There are many forward references in Chapter 3, not necessary in its context and hindering understanding. In particular, the term "dynamic entity" used several times in section 3.2.5 is not defined in place, only later in chapter 7.

Not too frequent but still noticeable are typos, spelling and grammar errors in the text ("lays', "learnt", first example on p.15, p.17 bottom sentence "To communicate the CBSE advantages...", several sentences in 2.2.2 p.31). Occasionally, non-updated section or text references can be found (e.g. text in 6.2 p.118 refers to sec 3.2 apparently from original paper, intro paragraphs to ch.8).

Further Questions to Address

On p.18 you state "This thesis aims at helping the CBSE community to be able to compete with the main stream of software engineering and bring the interesting CBSE oriented systems' ideas to real life." Leaving aside the issue why research should *compete* with industrial mainstream, how do your contributions help in

concrete terms make a concrete component model more relevant in industrial setting, were they (or at least how can they be) turned into practical realizations? Point (f) in Conclusion p.157 is too unspecific in this respect.

Define the terms "application architecture" and "dynamic reconfigurations in applications structure". Explain how the reconfigurations should be "properly captured in application's architecture" and how the verification techniques proposed apply when such reconfigurations occur. (Cf. point (b) on page 68).

Summary and Judgement

Overall, the candidate clearly has good "feeling" for and has participated in important research areas, worked towards their practical relevance, and is able to argue for both. It is a pity that his original contributions are not completely clear from the thesis and that the methodical issues in its first part seem to indicate weaknesses in the ability to approach a problem methodically and thoroughly.

I therefore recommend the candidate defends the thesis, addressing the issues raised above, and if successful be awarded the degree of Ph.D.

Pilsen, 3rd August 2012

Doc. Ing. Přemysl Brada, MSc. (Sheffield UK), Ph.D. (Charles Uni., Prague)