The PhD aims in developing a conceptual language for XML documents. The languages proposed so far are at the best at the logical level. They do not allow to survey the structuring of an XML document suite. They mainly describe which potential structuring is achieved after applying the grammar rules specified through XML schema, DTD or other languages. The funny assumption of the XML community is the automatic injection of semantics based on a partial meaning of wording. Typically XML documents are defined by application of list type constructor to basic types. Therefore, semantics is different from the case of typical database modelling languages. Additionally, XML documents have optional components. The notion of the key must thus be redefined. Handling, storage, maintenance and evaluation of XML document suites is far more complex due to the bulk semantics of XML data, due to the hierarchical structure of XML documents and due to various ways to define validity of a constraint in an XML document.

The proposal for an XML conceptual modelling language is separated into a native modelling language for XML and profiles for XML document representation. Profiles are similar to the styles for XML document suites such as Venetian blind or Russian doll (see Habib by M. Klettke). Profiles are already known for advanced ER models and their translation methods to logical database languages. This separation of concern is necessary since XML documents are mainly based on list type constructors whereas conceptual languages should be independent from the representation language that have been chosen. These translation and representation profiles are known for extended ER models (see HERM or Embley, Mok). They support to a certain extend independence of the logical schemata from the conceptual schemata.

Semantics is a wide area and should not be mixed with formal semantics. XML uses rudimentary lexical semantics and not at all any formal semantics. This is also true for the case of reference applications since enforcement support is not envisioned. I do not agree with the statement about maintenance of XML document suites. This problem is far harder than for usual software due to the chaotic development of XML documents.

The author decided to introduce the entity-relationship model on the basis of the functional data model. The introduction of the weak entity type must take into consideration cycles among determinants since ‘weak’ propagates through. I would advice not to use at all weak entity types. If you use them then why not to use also weak relationship types. The advantages of layered schema definitions are well known anyway. The same kind of critics applies to the notion of ISA relationship types. There is no reason why a relationship type cannot be an ISA relationship type defined on top of a relationship type. A similar observation can be made for generalisation called abstraction in the thesis. I will not comment on the UML nightmare. The only minor remark: UML uses look-through constraints instead of participation constraints. The author is using the model-driven development for architectural approach for his investigations. It is nice that he has been choosing this approach.
It seems that he could extract the useful side of this approach. I am not sure however whether he is aware of the bad sides of this approach, namely the consistency within document suites and maintenance of coherence during development. These problems are not yet solved and are neglected by the proponents of MDA.

Section 3 reviews references that seem to be of importance. The literature review is well prepared. The author shows that he did not only collect papers but has been reading and understanding them. The list of requirements to conceptual models seems to be appropriate. I think that the second requirement is biased. I prefer to have a visual representation for any specification. I met a lot of people who never want to see a graphical representation. There are more than a dozen pitfalls for graphical representation so far. M. Necaský surveys a number of proposals made for visualisation of XML structuring. My main criteria for all these proposals is that these proposals are only entry-level proposals. None of them has reached the maturity of an intermediate of full-featured language. A typical flaw within these language proposals is the optionality of list semantics beyond set semantics of collections. List semantics allows to have repetitions within a document that might make sense. Another not-so-important issue is that XML documents are inherently higher-order schemata (for instance if Russian doll is used for the document). It is sometimes useful to have such kind of schema since the coherence of objects and sub-objects must be maintained. I agree with the observation that most proposals that appeared so far are not based on real-life applications but rather toy example based.

M. Necaský thus decided to develop a language that supports all specifics of XML schema languages. The key concept is far more difficult for XML documents. There are at least three different competing and not comparable approaches to the key definition (Buneman and his group, Arenas, Yu/Jagadish) as strong, absolute, relative or context-dependent keys. The author decided to introduce the cluster type without the restriction of disjointness of component types. The component types are restricted to be entity types. The author uses the specific structuring mechanism of XML documents by trees to develop a specific from of constraints (called hierarchical projections). This mechanism might be nicely enhanced with the context injection method of Hartmann/Link (they used this method for the definition of keys). The rules for axiomatising hierarchical projections are similar to those that have been developed for hierarchical (join) dependencies. This similarity is only partial since XML documents have a deeper structure. The author also decided to introduce XML cardinality constraints. These constraints are rather difficult in the ER setting (nowadays I am in doubt whether they are the right form to express semantics although everybody uses them). They are far harder in the XML area due to the additional and hidden list constructor semantics. Therefore, I am not surprised to not find any axiomatisation effort in the PhD.

M. Necaský decided to introduce an explicit model for XML representation of conceptual schemata. This approach is similar to classical approaches used in the database area. I am personally not yet convinced that this approach is the only one (see for instance the Klettke approach). I would also see the similarity to classical sophisticated view techniques since XML documents might also contain subtrees that contain data that are derived from other parts. Reuters new agency uses, for instance, such an approach to limit repetitions in the document. The approach is in my eyes one of the main achievements made by the author. He uses his specific path constraints for well-specified XML document roll-outs. These specific XSEM-H representation can be used as an intermediate language
for translating XSEM-ER specs into the various XML dialects. The author also commits that his language might not be powerful enough to represent all issues through XML specs. That is correct but there is no universal world formula anyway. M. Nečaský decided to introduce a number of extensions such as complex attributes, irregularity catchers (that is one of the pitfalls of the XML W3C gurus since irregularity will never be properly supported by any mass data management), splitters (as long as splits are exclusive there is no harm - see the nightmare of Or-join in EPC or BPMN), mixins (as long as such are used it is ok), mirror structures (called re-used structures) (known already for hierarchical databases in the past but nowadays forgotten), recursive structures (as long as these form rational trees), and IsA hierarchies.

The claim that all modelling constructs introduced so far can be translated to XML dialects such as XML schema is handled in later chapters. So far the translation is based on an interpretation approach. There have already proposed a number of approaches to ER-XML schema translations, e.g. by Lipeck. The author uses a straightforward translation and succeeds due to his intermediate language.

I see a number of achievements which are valuable to the scientific community and which deserve the issuing of the PhD degree to the defender:

- The author succeeds in developing a conceptual modelling language for XML document suites.
- The author solves the representation variety problem by developing his own intermediate language that is solving main translation problems.
- The approach taken is practical, well settled and nicely applicable.

The approaches developed are novel. They have a high potential for further development or for application in real large projects.

The thesis is worth to issue the degree to the author. The ideas developed could however be the basis for a good seria of interesting research papers. The thesis should get some extensions in the future I would like to see:

- The theory of hierarchical projections can be enhanced by path constraints (Weddell) or by object-relational constraints (Biskup/Polle).
- The theory of cardinality constraints for XML documents might be enhanced by the approaches of Hartmann/Link.
- I would like to see a good paper that allows to handle explicitly the translation profile (or even portfolio that would also allow to treat the application cases in a different form) proposed by M. Nečaský. Whether the MDA approach is used or the author decides to use his model for another XML representation is a matter of taste and of the community to which such paper is submitted.
- The translation into XML dialect could also consider efficiency issues and take into account the derivation of optimised schema.
The author might also consider to have an integration with classical object-relational representation languages with his XML conceptualisation. This would allow an integration of database storage machines and XML play-in and play-out machines.

Let me finish my review with a personal observation. In the late 80ies and early 90ies I have been making a trial to integrate all different proposals for extensions of ER models. It took me years to discover which proposals can be consistently handled and integrated into a rich modelling language and which proposals should be refused. The XML community is as creative as the UML community in creating new and new constructs. I would like to advice M. Nečaský not to hunt for all these proposal but to settle his proposals, to found his constructs and to apply them within a large project.

The thesis is well-written and should be brought to a wider audience by publishing the work on its own.

I find this work really worth the degree and propose to issue the degree without any further restriction. The work meets the international standard for PhD and is a valuable contribution.

Therefore, the thesis conforms to doctoral level at international standards and should be accepted by the committee.