

## Review of dissertation thesis of RNDr. Roman Vaculín

The submitted dissertation thesis „Process Mediation Framework for Semantic Web Services“ addresses several intertwined research topics from the semantic web service field, which is nowadays considered as promising technology for software application integration.

The thesis consists of eight chapters (excluding bibliography). It starts with a comprehensive introduction (Ch. 1) and necessary preliminaries (Ch. 2). Then follow four chapters with the author's original contribution: new methods for process mediation (Ch. 3), data mediator discovery (Ch. 4), web service monitoring (Ch. 5) and web service recovery (Ch. 6). The thesis is concluded with a thorough review of related work (Ch. 7) and a conclusions chapter (Ch. 8).

I truly appreciate the high number of original research results contained in the material. Apparently, the research on the original topic (process mediation proper) led the author to consider additional problems related to the invocation of a semantic web service, namely, retrieval of its external components (namely, data mediators), monitoring of events arising during the execution of the service, and its recovery after a failure event. In each of these areas, new achievements have been made, which have also been recognised by the international community; this is manifested by the large number of peer-reviewed publications on these topics in recognised journals and conference proceedings. There are twelve such publications listed.

The original contributions consist in particular in

- an abstract framework for process mediation
- implemented algorithms for process mediation
- implemented algorithms for discovery of external services as well as their combination and composition; this part of work also includes derivation of algorithmic complexity
- an ontology of semantic events and a language for composite event patterns based on the event algebra
- a model for exception handling and recovery in semantic web services.

Most parts of the text describe the results in a mature and consistent form (a few issues that might in my opinion need revision are mentioned in the detailed comments below). I also appreciate the clear identification of restricting assumptions of the approach, which was done in Section 3.1.5.

Although the research is mainly at the level of theory and experimental implementation, the high degree of match between the functionality offered by (semantic) web service technology and practical problems tackled in day-to-day business application integration also suggests that some of the results could have industrial impact in the future. The two subject areas chosen for illustrative examples, flight reservation and electronic market, are adequate representatives for typical semantic web service application areas.

There is not much to be criticised on this work in general. As a minor flaw I see the uneven support of the provided methods and algorithms by experimental evidence:

- process mediation has been evaluated on a single (though relatively rich) artificial use case
- data mediation has been evaluated on a synthesised data collection with real-like features
- semantic monitoring has been evaluated on a single, real, semantic web service workflow; only the computational overhead was computed
- there is no evaluation of workflow recovery at all.

I did not find a systematic consideration of evaluation methodology that would relate these heterogeneous approaches to each other or justify their selection – even if at least the first two

evaluation settings are sufficiently explained each by itself. For the third setting (monitoring) I miss concrete information on the complexity of filter expressions etc.

Although the exemplification is solid in general, I would have welcomed references to running examples even within the explanation of basic concepts. For example, Section 3.2 only refers to an example once.

The thesis is written in a readable style. Its logical structure is transparent, and the writing style is coherent with international standards. The number of typos and syntax-grammatical errors is relatively low, and they do not significantly compromise legibility.

Detailed comments (all on Chapter 5):

- The event pattern ‘types’ listed in Section 5.2 are actually a strange mix of generic types and concrete examples.
- You use the notion of ‘interest’ for something that probably is ‘event of interest’ (Section 5.2, line 1; Section 5.3, para 2, line 2); is it a typo or a kind of convention?
- From the point of view of ontological engineering, I see the distinction of ‘event type’, ‘event instance’ and ‘event occurrence’ (which you adopted from the literature on database management?) as strange. If one ‘event instance’ is to be interpreted as a class of individual ‘event occurrences’ (this is probably the case), why not model it as a leaf class of the event type hierarchy? Notably, DL reasoning is generally stronger over classes than over individuals. Conversely, if one ‘event instance’ were to be interpreted as a single ‘event occurrence’, just formally separated from the time stamp, then I would not see a reason for such a separation.
- I am afraid that the explicit distinction of ‘instantaneous’ atomic events and ‘interval’ composite events makes the whole model unnecessarily complicated. Instantaneous events can simply be modelled as events with start equal to end.
- Event expression operators in Table 5.2 are somewhat opaque. It would definitely help to explain them in terms of e.g. Allen’s operators of temporal overlap (at least later, when event streams come into game!). I first intuitively interpreted “Conjunction. Occurs when both A and B occur irrespective of their order.” as interval intersection. However, then I see “Sequence. Occurs when A occurs before B.”, which cannot be straightforwardly interpreted in terms of an interval. This makes me rethink the whole set of explanations purely in logical terms (the ‘when’ connective in the explanations was the source of confusion?). However, in my opinion, then something is missing again: the event expression holds or does not hold in some context, presumably process, not just by itself out of the blank!
- Why are simultaneous occurrences of the same primitive event type not permitted? I would expect that two parallel processes can encounter the same type of event in the same time (at a certain level of granularity) in principle. This is probably related to the previous problem: a single process is implicitly assumed as context to the model.
- Definition 5.5 on event streams is a bit strange. It seems that  $\text{neg}(R,S)$  only excludes situations when the whole of  $S$  is contained in  $R$ . However, I am not sure if this is the most intuitive interpretation of  $R-S$  (“there is an occurrence of  $R$  during which there is no occurrence of  $S$ ”). The definition of  $\text{delay}(R,t)$ , mixing logical connectives, left arrows and the plus sign, is particularly opaque.
- In Definition 5.7 you should consistently use either ‘class name’, ‘object property name’ etc., or just ‘class’, ‘object’ etc.
- How is ‘restricted event types’, as in Definition 5.8, related to possible axiomatisation of the event ontology using class definition axioms in DL? Is it a formal layer that is in practice modelled in OWL (and processed by a DL reasoner), or is it an alternative, possibly with a different expressivity? Example 5.2 looks like a natural subject for a property restriction axiom.

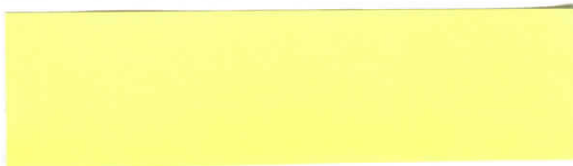
Additional questions to be answered during the defence:

- From the practical point of view, what kind of services can realistically be discovered and embedded into semantic workflows in runtime? Only data converters and similar kinds of elementary and testable services, or also services that e.g. provide financial information on companies or even affect the state of the real world? (Making orders, reservations etc.) You seem to build on the heuristic (Section 3.4) that reconciliation plans should be short, i.e. only simple services should be embedded in runtime. Is this heuristic to be generally applied for SWS?
- In Section 3.7, page 78 you mention the possibility of high-level modelling of desirable mediation process between a requester and a provider by means of rules. How could such rules look like in the flight reservation example? What (human) subject would specify them in real settings? What level of process model visibility would be needed for this kind of design effort?

As summary, the dissertation thesis of RNDr. Roman Vaculín addresses topical problems of computer science research, formulates ambitious goals that it also fulfils in a satisfactory way, by using novel models, methods and algorithms, and is also formally well written. It convincingly manifests the scholarly capacity of the PhD candidate. Any critical comments mentioned above are minor and do not affect the overall quality of the work.

Based on the above assessment, I consider the submitted thesis as **adequate for awarding the title “Doctor”, conditioned by successful defence.**

Prague, 24 August, 2009



Doc. Ing. Vojtěch Svátek, Dr.  
Department of Information and Knowledge Engineering  
University of Economics, Prague