Reference on the doctoral thesis

„Methods for effective querying of RDF data"

By RNDr. Jiriho Dokulila

Before I start my reference, I have to explain the background of the evaluation. It is based on a recent report prepared by the Research Evaluation Committee of Informatics Europe.

Research evaluation for computer science must be adapted to the specifics of this new discipline, namely, combining characteristics of science and engineering.

Computer science concerns itself with computing: processing information using algorithmic techniques. The term Informatics, popular in Europe, highlights the need for a broad perspective including human aspects of information technology. An important part of computer science research produces artifacts other than publications, in particular software systems. In measuring impact, these artifacts can be as important, as publications.

Dokulila's thesis is a typical example of a good quality research work in informatics.

It is a part of a team work. The publications are of many authors and published mainly in the proceedings of international conferences. The novelty of the research is demonstrated by large software systems developed by the team. The research quality and scientific merit should be evaluated first for the whole team, then for individual members.

Dokulila's work belongs to the Systems research category of computer science. In the Systems research field the relevance, the novelty, the complexity and the size of the developed system characterize the scientific impact.

The thesis summarizes a large part of the team work. The explanations clearly describe the research goals, the selected solutions and the key issues in the implementation. Dokulila's personal contributions are exactly given in Appendix A "Personal responsibilities".

The project goal is a demanding approach to solve one of the key challenges of IT: to develop tools for human access to the exponentially growing digital information.

The specific problem selected for research is the support for querying RDF data.

RDF (Resource Description Framework) itself looks simple, just put together from small three-tuples a graph representation of some semantic knowledge. In graph representation form, a labeled edge goes from the subject node to the object node, and the label is the predicate. It typically results in a very large graph that is good for computer interpretation, usually used in the background of application system, but hard to follow by human mind. The team's goal was to develop tools for the full spectrum of querying RDF graphs. This means: visual representation and navigation of RDF graphs, query language for RDF data, and a parallel query execution system.

The dissertation consists of 6 chapters and 3 appendices. After the introductory first chapter the second one briefly summarizes the basics of RDF.

The 3rd chapter describes the first large component of the system developed by the team: the RDF visualization. The selected solution and the implementation are carefully compared with many known solutions and proposals. The novelty is the triangle layout algorithm with the vertical range distribution. The nodes of the RDF graph are represented by rectangles containing lines for the descriptions of the nodes' children. In the detailed description of the algorithm the author proves the optimality of the layout, namely the size of the area used for rectangle placement is quadratic in the number of nodes. The author's main contributions are in RDF-related aspects. The first scientifically relevant solution is the node merging algorithm

which is important to represent nodes with large neighborhood. The second is in the detailed analyses of the possibilities selected in the implementation for navigation in large and fat graphs. The chapter demonstrates valuable research and development results of the team and the contribution of Dokulila.

The 4th chapter contains the second component: the TriQuery query language. One may ask, why create another RDF query language? It is clearly explained: since the low level RDF data format is relational, typical RDF query languages uses SQL based formalism. It is not effective for express complex semantics. The first attempt of the author was an algebra defined on the low-level RDF format, the TriQ algebra. (The specification of TriQ is given in appendix B. It is from the author's Master thesis.) But it is really just another –though expressive- RDF query language. The team selected another, much more promising approach: extend the XQuery language so that it helps to express RDF and relational queries easier and allowing to work with RDF and XML within one language. This extension doesn't influence the expressive power of XQuery, since it contains the Turing-complete XSLT version 2. The new W3C recommendation for OWL 2 (October 2009) proves that the direction was right: OWL 2 Ontology is should be mapped to RDF graphs and XML documents. What is more important, syntactic subsets (OWL 2 Profiles): OWL 2 QL enables conjunctive queries to be answered in LogSpace using standard relational technology; OWL 2 RL enables the implementation of polynomial time reasoning algorithms using rule-extended database technologies operating directly on RDF triples. So the extension of XQuery with record constructors, operators and functions was a god step to the right direction. The main problem with query languages is not in what but in how can we express our queries by them. So sublanguages are really important. Just a remark that come to my mind from OWL 2 Profiles: The syntactic restriction are given on the data structure, so in parallel to XQuery extension the extension of XSD with record data type might be useful.

The chapter contains a complete explanation of the solution, describes implementation issues, the RDF and relational support, and an empirical comparison with SPARQL on the SP²Bench. Also contains a comparison with related works. The grammar of the extended XQuery is given in Appendix C. The design of TriQuery was a common work with D.Bednarek, where J.Dokulila created the basic concept of the language.

The 3rd component of the system is Bobox, a parallel programming system and server for OLAP-like queries on RDF databases. The parallelism is on running a query in parallel on multi core, blade server systems. They use task level parallelism with schedulers on the threads of the processors. The nonlinear pipeline model consists of Boxes and Vias. Boxes contain the executable program codes and sends/receives data through Vias. Data level parallelism is solved by storing and communicating columns of relational data. The scheduler, the library and the parallel server are implemented and tested for working properly and experimentally evaluated. I propose to follow the development and performance modeling of the Bobox system. From the three members of the Bobox team J. Dokulil's contribution is the flow control, the data handling, log visualizer and the benchmark with SP² Bench and Sesame.

Chapter 6 is the Conclusion that clearly highlights the main contributions of the research team to RDF querying. The proposal for future work is promising.

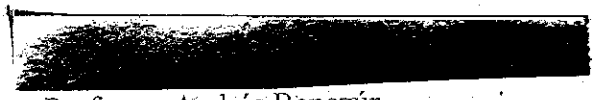
Summary.

Dokulila's personal contribution as I have referred his contribution to the team's work in each paragraphs are new scientific results. The team's work is an important contribution to technology development in the area of semantic databases and query systems. Though the focus of the work is RDF, the results are also important in graph visualization, querying new XML-databases and parallel query processing.

The form of the thesis shows a careful description and clear presentation of the results. The background and related works are given in appropriate details. The thesis describes the system as a whole, but the results of the author are clearly identified.

The range of the problems solved by Dokulila during the team work clearly proves his ability for creative scientific work in informatics.

Budapest, 30. 07. 2010.



Professor András Benczúr