The Knowledge Engineering Review, Vol. 13:2, 1998, 195–199
Printed in the United Kingdom

The Third International Workshop on Rules in Database Systems (RIDS'97)

ANDREAS GEPPERT[1] and MIKAEL BERNDTSSON[2]

[1]*Department of Computer Science, University of Zurich, Winterthurerstr. 190, CH-8057 Zurich, Switzerland. Email: geppert@ifi.unizh.ch*
[2]*Department of Computer Science, University of Skövde, Box 408, 541 28 Skövde, Sweden. Email: spiff@ida.his.se*

1 Introduction

This report is a summary of the *Third International Workshop on Rules in Database Systems (RIDS'97)* (Geppert & Berndtsson, 1997) which was organized by the database groups at University of Zurich (Switzerland) and University of Skövde (Sweden), and was held at the University of Skövde, June 26–28 1997. Following the success of RIDS '93 (Edinburgh) (Paton & Williams, 1993) (*1st International Workshop on Rules in Database Systems*), and RIDS '95 (Athens) (Sellis, 1995), the successor RIDS '97 brought together researchers working on both theoretical and practical aspects of rules in database systems.

The field of rules in database systems encompasses active rules and deductive rules. Active rules (i.e., Event Condition Action (ECA) rules), specify and implement reactive behaviour of a database system. In response to events occurring in the database or its environment, a check is made on whether the condition holds, and—if so—the action is executed. Deductive rules are logic style rules, for example Datalog rules, that are used to deduce knowledge based on facts.

Although extensive work has been carried out on supporting active rules and deductive rules in database systems, many questions are still open concerning their incorporation into practical technologies. The aim of RIDS '97 was to examine the current state of the art, to present novel and innovative approaches to rule systems, to discuss experiences with operational systems and real-life applications, and to identify future trends and developments.

Subsequently, we summarize the technical presentations and general discussion.

2 Deductive database systems

At the workshop it was noted that the research field of Deductive Object-Oriented Databases (DOOD) is now becoming mature. Several working prototypes, and at least one commercial product exist. This was also pointed out in a survey of existing DOOD by Sampaio & Paton (1997). In the survey, DOOD proposals are categorized according to three language design strategies: (i) language extension, (ii) language integration, and (iii) language reconstruction. Based on this, DOOD proposals are compared with support provided for specific deductive and object-oriented features.

The architecture and implementation of the DOOD prototype FOLRE was presented by Mayol et al. (1997). In particular, they discussed details of how the prototype supports integrated treatment of advanced update capabilities, such as view updating and integrity checking.

In another presentation by May et al. (1997), a method was presented that allowed modelling of

dynamic aspects in DOOD, for example, dynamic objects and dynamic methods, by treating states as first-class citizens.

Although the basic technology of DOOD and deductive rules may now be well understood among DOOD researchers, some participants expressed concern over whether DOOD and deductive rules will be useful in practice for a wider community.

3 Construction and architecture of active database systems

The area of construction and architecture of ADBMS is concerned with questions of how to design and implement ADBMS. Thus, the relevant research issues are:

- What is the overall software architecture of ADBMS, and how are the "active" components related to the conventional, passive ones?
- What would a "construction theory" of ADBMS look like, i.e., how can a given description of the desired functionality be turned into an appropriate operational system?
- Which types of ADBMS are particularly useful for selected application domains? How can ADBMS be customized to optimally support these domains?

In the recent past, progress has been made mainly in the areas of prototype implementations as a proof of concept, and in implementation techniques for specific ADBMS parts. The RIDS papers focusing on architectural questions mainly address the customization of ADBMS for specific application domains and the systematic construction of specialized ADBMS.

Asynchronous Trigger Processing. In Hanson & Khosla (1997), Asynchronous Trigger Processors (ATP) are introduced, and TriggerMan is presented as a prototypical ATP. In ATP, the rule execution component is "unbundled" from a DBMS, i.e., the trigger processor is extracted from the ADBMS and runs separately on top of one or more database systems or other data sources. In TriggerMan, data sources can be registered, and triggers can be defined on top of these data stores. A data source has to inform TriggerMan about (committed) database updates, to which TriggerMan then reacts by executing triggers. In Hanson & Khosla (1997), special attention is paid to condition evaluation, and an approach to parallelizing this task using Gator networks is presented.

The appealing aspect of ATP is that it can be put on top of different (relational or object-relational) database systems or other types of data source. An ATP thus needs to be designed and implemented only once, while conventional trigger processors need to be implemented repeatedly if they are hard-wired into DBMS.

Construction of Customized ADBMS. In Bouzeghoub et al. (1997), an approach is presented to developing customized active DBMS. The motivation for this work is that many applications need specialized active functionality, e.g., data warehouse and workflow management. Bouzeghoub et al. introduce a generic rule execution model that can be customized towards concrete requirements. Functional abstractions (e.g., one that determines which rule instances to create) are combined to specify a generic rule execution model. A concrete, customized rule execution component is then obtained by specializing these functional abstractions.

Bouzeghoub et al. also outline the design of a toolkit that supports the customization of rule execution models. The toolkit prescribes the collection of modules needed, which when combined form a rule execution component. Specialized modules can be stored in libraries, and a specialized ADBMS can then be built by plugging appropriate modules (from the libraries) into the skeleton formed by a common core.

The work by Schlesinger & Lörinze (1997) also has the objective of building customized ADBMS. ALFRED allows the modeling and simulation of active add-on layers using specialized (Coloured) Petri nets. Such a layer designed in ALFRED can then be put onto a passive DBMS, thereby transforming it into an active one.

Specialized Active Systems for Workflow Management. In Geppert & Tombros (1997), ECA-rules are used for workflow enactment. A Workflow Management System (WFMS), in addition to its common tasks, has to support logging of workflows and post-mortem analysis of past workflow executions. The former is a critical requirement in applications that need to archive their business operations (e.g., banks). The latter is critical in the context of optimization and maintenance. Analysis allows the detection of bottlenecks such as activities whose executions need more time than expected, processing entities that are chronically overloaded, etc.

In Geppert & Tombros (1997), it is shown how a specialized active system can support the tasks of workflow logging and analysis. In this approach, any activity pertaining to a workflow execution is represented as an event. The key concept to logging and analysis is thus the event history (synonymously, event log) that stores all event occurrences that have happened within some workflow execution. In this way, it is straightforward to use the event history as a workflow log. Analysis is supported by allowing queries against this event history.

4 Analysis and validation

Some of the open questions in the area of design and analysis refer to how to predict and validate the behaviour of active rules. At the workshop, research progress was reported in the three areas of:

- *rule termination analysis*, which checks whether rule execution terminates, i.e., does not cause indefinite rule cycles (Comai & Tanca, 1997; Tschudi et al., 1997; Vaduva et al., 1997);
- *rule confluence analysis*, which—in the case that multiple rules can be fired upon the occurrence of an event—determines whether rule execution reaches a unique final state regardless of the execution order of rules (Comai & Tanca, 1997; Wang et al., 1997),
- *rule testing and validation*, which develops and adapts software techniques for testing and validating active database applications (Chan et al., 1997; Fraternali et al., 1997).

Comai & Tanca (1997) use properties of Datalog to prove termination and confluence in active databases. The expressiveness of the chosen rule model is restricted to action languages which only allow insertions and deletions. With regard to event specification, the composite event operator disjunction is considered. Briefly, ECA rules are translated to logical clauses (Datalog rules). Existing knowledge and experience about confluence and termination from work on deductive rules is then applied to prove termination and confluence for ECA rules.

Tschudi et al. (1997) report on details of the implementation of the Refined Triggering Graph (RTG) for termination analysis in the context of the active DOOD language CDOL (Comprehensive, Declarative, Object Language). This work extends the formal definition of the RTG method by considering execution constraints (e.g., coupling modes, rule priorities). Additionally, previous work on rule simplification is used. A sample active database application has been used in order to evaluate and test the implemented RTG method.

Vaduva et al. (1997) investigate termination of rule execution for expressive rule languages. In particular, they address termination analysis for rule languages that allow specification of composite events. A rule termination analysis tool based on the ideas presented in the paper has been implemented for the active database system SAMOS.

Wang et al. (1997) investigate the problem of confluence of rule executions. They address confluence under the presence of "meta rules", which can express for example, rule ordering. They derive a sufficient condition to be met by active rules in order to ensure rule confluence, and show that under specific assumptions this condition is also necessary.

Chan et al. (1997) introduce a software testing technique for active rules in the context of the active DOOD language CDOL (Comprehensive, Declarative, Object Language). The technique is based on previous work on software testing using control flows. Details about a test data generator for active rules are presented.

Fraternali et al. (1997) introduce a technique for active rule validation based on planning. The approach is based on two key concepts: first, the generic *core model* has been adopted as a

framework for expressing updates and active rule semantics. Second, techniques from planning have been applied for validation of rules. In detail, a set of desired rule properties (triggerability, condition satisfiability, joint applicability, rule coverage, rule cascading, postcondition satisfaction) are expressed as declarative first order goals. Both the goals and an abstract description of the semantics of the active database are submitted to a planner. The planner investigates whether a plan can be found for a given goal (rule property). Whenever the planner can find a plan for a given goal then the corresponding property holds.

Summarizing, the RIDS papers on rule analysis show that progress has been made in this area. However, it seems that more research is necessary to turn the approaches into practicable tools. In particular, RIDS participants have commented that

- research in analysis often assumes much simpler active rule models than those found in real systems;
- analysis suffers from the fact that actions can be defined in languages that are computationally complete (so that either simplifying assumptions have to be made or results of analysis are overly pessimistic)

It was also suggested by participants that analysis tools should be developed on top of real database systems (instead of standalone tools), in order to foster insights into the practicality of the underlying approaches.

5 Conclusions

At the end of the workshop a panel session was chaired by Klaus R. Dittrich (University of Zurich) on the subject ''The Future of Rules in DBMS: Is There Anything Left?'' The panel initiated a discussion on whether researchers in the RIDS community are ''polishing round balls''. It was noted that there has now been more then ten years of research on rules in database systems, but the impact and transfer of results to products and real-world applications is limited. Hence the subject of the panel was rephrased by the panel chair to ''The Future of Rules in Database Technology: Is There Any?''

Most participants suggested that there is still a need for research in, and product-support for, rules in database systems. However, to strengthen the impact on products and real applications, the panelists encouraged research groups to investigate the following issues:

- evaluating new concepts and systems with real application scenarios,
- methods and tools for designing active database systems, and
- measuring rule performance and developing tuning techniques for active database systems.

In other words, the panelists agreed that there are still balls which it is worth polishing. It was also noted that the future of rules may not necessarily be restricted to database technology. For example, events and active rules have been proposed (and used!) in other research areas, such as real-time systems, distributed computing, cooperative information systems, and workflow management. It might well turn out that such areas benefit form active rules as available in active database systems, i.e., that—from the viewpoint of active database technology—they are examples of the aforementioned ''real applications''.

Acknowledgements

The authors would like to thank the participants of RIDS '97 for their contributions, as well as Björn Lundell, Henrik Engström, Anca Vaduva, Klaus R. Dittrich, and Brian Lings for their comments on an earlier version of this report.

References

Bouzeghoub, M, Fabret, F, Llirbat, E, Matulovic, M and Simon, E, 1997. "Active-design: A generic toolkit for deriving specific rule execution models". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Chan, H, Dietrich, S and Urban, S, 1997. "On control flow testing of active rules in a declarative object-oriented framework". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Comai, S and Tanca, L, 1997. "Using the properties of datalog to prove termination and confluence in active databases". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Fraternali, P, Teniente, E and Urpi, E, 1997. "Validating active rules by planning". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Geppert, A and Berndtsson, M (eds) 1997. *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Geppert, A and Tombros, D, 1997. "Logging and post-mortem analysis of workflow executions based on event histories". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Hanson, E and Khosla, S, 1997. "An introduction to the triggerman asynchronous trigger processor". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

May, W, Schlepphorst, C and Lausen, G, 1997. "Integrating dynamic aspects into deductive object-oriented databases". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Mayol, E, Pastor J, Teniente, E and Urpi, E, 1997. "Folre: A deductive database system for the integrated treatment of updates". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Paton, N and Williams, M, (eds) 1993. *Proceeding of the 1st International Workshop on Rules in Database Systems*, Workshops in Computing. Springer-Verlag.

Sampaio, P and Paton, N, 1997. "Deductive object-oriented database systems: A survey". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Schlesinger, M and Lorinze, G, 1997. "Rule modelling and simulation in alfred". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Sellis, T (ed) 1995. *Proceedings of the 2nd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 985. Springer-Verlag.

Tschudi, M, Urban, S, Dietrich, S and Karadimce, A, 1997. "An implementation and evaluation of the refined triggering graph method for active rule termination analysis". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Vaduva, A, Gatziu, S and Dittrich, K, 1997. "Investigating rule termination in active database systems with expressive rule languages". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.

Wang, X, You, J-H and Yuan, L 1997. "On confluence property of active databases with meta-rules". In: A. Geppert and M. Berndtsson (eds), *Proceedings of the 3rd International Workshop on Rules in Database Systems*. Lecture Notes in Computer Science 1312. Springer-Verlag.