Asaf Adi
Suzette Stoutenburg
Said Tabet (Eds.)

LNCS 3791

Rules and Rule Markup Languages for the Semantic Web

First International Conference, RuleML 2005
Galway, Ireland, November 2005
Proceedings

Springer

Asaf Adi   Suzette Stoutenburg
Said Tabet (Eds.)

Rules and Rule Markup Languages for the Semantic Web

First International Conference, RuleML 2005
Galway, Ireland, November 10-12, 2005
Proceedings

Springer

Volume Editors

Asaf Adi
Haifa University, IBM Research Lab in Haifa
Mount Carmel, 31905 Haifa, Israel
E-mail: adi@il.ibm.com

Suzette Stoutenburg
MITRE Corporation
1155 Academy Park Loop, Colorado Springs, CO 80910, USA
E-mail: suzette@mitre.org

Said Tabet
RuleML Initiative
24 Magnolia Road, Natick, MA 01760, USA
E-mail: stabet@ruleml.org

Library of Congress Control Number: 2005935452

CR Subject Classification (1998): H.4, H.3, I.2, C.2, H.5, K.4, F.3

ISSN 0302-9743
ISBN-10 3-540-29922-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29922-6 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11580072 06/3142 5 4 3 2 1 0

Preface

RuleML 2005 was the first international conference on rules and rule markup languages for the Semantic Web, held in conjunction with the International Semantic Web Conference (ISWC) at Galway, Ireland. With the success of the RuleML workshop series came the need for extended research and applications topics organized in a conference format. RuleML 2005 also accommodated the first Workshop on OWL: Experiences and Directions.

Rules are widely recognized to be a major part of the frontier of the Semantic Web, and critical to the early adoption and applications of knowledge-based techniques in e-business, especially enterprise integration and B2B e-commerce. This includes knowledge representation (KR) theory and algorithms; markup languages based on such KR; engines, translators, and other tools; relationships to standardization efforts; and, not least, applications. Interest and activity in the area of rules for the Semantic Web has grown rapidly over the last five years. The RuleML 2005 Conference was aimed to be this year's premiere scientific conference on the topic. It continued in topic, leadership, and collaboration with the previous series of three highly successful annual international workshops (RuleML 2004, RuleML 2003 and RuleML 2002). The theme for RuleML 2005 was rule languages for reactive and proactive rules, complex event processing, and event-driven rules, to support the emergence of Semantic Web applications.

Special highlights of the RuleML 2005 conference included the keynote address by Sir Tim Berners- Lee, Director of W3C. His talk, titled "Web of Rules", discussed whether knowledge can be represented in a web-like way using rules, so as to derive serendipitous benefit from the unplanned reuse of such knowledge. The two other invited papers were by Dr. Opher Etzion of IBM Haifa Labs entitled "Towards an Event-Driven Architecture: An Infrastructure for Event Processing" and by Dr. Susie Stephens of Oracle USA on the application of Semantic Web technologies in life sciences entitled "Enabling Semantic Web Inferencing with Oracle Technology: Applications in Life Sciences".

We would like to thank our Steering and Program Committees as well as all colleagues who submitted papers to RuleML 2005. We would also like to thank the organizers of ISWC and the OWL Workshop for their cooperation and partnership. Special thanks go to IBM Haifa, Israel, for sponsoring RuleML 2005. Finally, we would like to thank GoWest for their conference planning services.

November 2005

Asaf Adi
Suzette Stoutenburg
Said Tabet

Organization

Conference Program Co-chairs

Asaf Adi, IBM, Israel
Suzette Stoutenburg, The MITRE Corporation, USA
Said Tabet, The RuleML Initiative, USA

Conference General Co-chairs

Harold Boley, National Research Council and University of New Brunswick, Canada
Benjamin Grosof, Massachusetts Institute of Technology, USA

Steering Committee

Asaf Adi, IBM, Israel
Grigoris Antoniou, University of Crete, FORTH, Greece
Harold Boley, National Research Council and University of New Brunswick, Canada
Benjamin Grosof, Massachusetts Institute of Technology, USA
Mike Dean, BBN Technologies, USA
Dieter Fensel, Digital Enterprise Research Institute (DERI),
National University of Ireland, Ireland
Michael Kifer, State University of New York at Stony Brook, USA
Steve Ross-Talbot, Pi4 Technologies, USA
Suzette Stoutenburg, The MITRE Corporation, USA
Said Tabet, The RuleML Initiative, USA
Gerd Wagner, Brandenburg University of Technology at Cottbus, Germany

Program Committee

Grigoris Antoniou, University of Crete, FORTH, Greece
Nick Bassiliades, Aristotle University of Thessaloniki, Greece
Harold Boley, National Research Council and University of New Brunswick, Canada
Mike Dean, BBN Technologies, USA
Dieter Fensel, Digital Enterprise Research Institute (DERI),
National University of Ireland, Ireland
Benjamin Grosof, Massachusetts Institute of Technology, USA
Guido Governatori, University of Queensland, Australia
Michael Kifer, State University of New York at Stony Brook, USA
Sandy Liu, National Research Council and University of New Brunswick, Canada
Jan Maluszynski, Linköping University, Sweden
Massimo Marchiori, W3C, MIT, USA and University of Venice, Italy

Donald Nute, University of Georgia, USA
Royi Ronen, Technion, Israel
Michael Sintek, DFKI, Germany
Bruce Spencer, National Research Council and University of New Brunswick, Canada
Said Tabet, RuleML Initiative, USA
Dmitry Tsarkov, University of Manchester, UK
Carlos Viegas Damasio, Universidade Nova de Lisboa, Portugal
Gerd Wagner, Brandenburg University of Technology at Cottbus, Germany
Kewen Wang, Griffith University, Australia

Sponsors

IBM Haifa Labs

Table of Contents

Towards an Event-Driven Architecture: An Infrastructure for Event Processing Position Paper

Opher Etzion

IBM EDA Initiative Leadership Team
opher@il.ibm.com

Abstract. Multiple business factors have emerged to accelerate the necessity of event-driven functionality and make it part of the main-stream computing, instead of a niche technology. Consequently, there is now focus on using high-level software constructs to build these applications. This paper presents a vision for such high-level features and architecture. This paper explains why "event-driven applications" becomes an emerging area, explains the basic terminology of EDA, explains the relationship to business rules, and sets some directions for the future of this discipline.

1 Introduction and Motivation

Event-driven applications are those which respond to the occurrence of events. This type pf processing is not new, and can be found over the history of computing, starting from exception handling in programming languages, passing through concepts and disciplines such as: active databases [1], publish/subscribe systems [2], network and system management [3] and business activity management [4]. Recently there is an increase in the interest in industry in this area, indicated from analysts' reports, from the sharp increase of start-ups in this area, and product announcements by application middleware and database vendors. This is an indication that event-driven programming moves from being used at some niches to the main stream of programming, and thus it is cost-effective to construct general tools that enable easy construction and maintenance of such applications. The contemporary business drivers for these directions are:

1. Enforcement of compliance with regulations inside the process (some times in "right-time" fashion)
2. The drive for expense reduction in back offices that increase the demand for more automation (e.g. automated exception handling)
3. Increasing complexity of inter process integration that require agility and flexibility;
4. Technology developments such as RFID that increase the scale and scope of event based data
5. Industry trends such as Business Activity Management, Real-Time Enterprise, Business Performance Management that place a demand on software infrastructure to deliver the event data to drive these high level objectives.

The rest of this position paper is structured in the following way. Section 2 explains the type of applications, and shows a case study. Section 3 explains the principles of EDA, Section 4 discusses relation to business rules technology, and Section 5 concludes the paper with some future predictions.

A. Adi et al. (Eds.): RuleML 2005, LNCS 3791, pp. 1–7, 2005.

2 Event-Driven Applications

Event-driven functionality is an enabler for the IBM's vision of the "on demand" enterprise, it enables enterprises to make "just in time" reactions to eliminate "worst case" expenses, it enables enterprises to improve control over their operations, and eliminate getting to critical situations, it enables to save cost by providing automation to exception handling, and it is enables loosely coupled integration among processes and systems, improving the agility of application integrations.

Table 1 shows classification of these applications along with the associated business value.

Table 1. Classification of event-driven applications

Type	Examples	Business Value
1. Agile Process Integration	➢ Time Critical Targeting (Military) ➢ EAI Integration hub (Telco) ➢ Just-in-time car rental (Travel and Transportation) ➢ Trade processing (Financial Markets)	Providing integration between various systems based on event input. Enables to support applications that require dynamic composition of various business processes, based on event processing. Enables to perform operations just-in-time and not in advance, thus eliminates excessive cost.
2. Autonomic behavior in business cases	➢ Straight Through Processing (Financial Markets) ➢ Automatic policy setting (Operational Resilience) ➢ Loan and mortgages decision support (Banking) ➢ Automated shipping and receiving based on RFID (Distribution) ➢ London Congestion billing (Travel and Transportation)	Reducing expenses in back offices by automating exception handling processes. Improving business decision process, by linking decisions with business objectives.
3. Awareness to Business Situations	➢ Anti Money Laundering (Banking) ➢ Fraud Detection (multiple industries) ➢ e-Pedigree (Pharmaceutical) ➢ Promotion Evaluation (Retail)	Providing the ability of timely identification of Business situations that requires reaction, and avoid critical situations. This is an enabler for run-time enforcement of regulations.
4. Change Management/ Impact analysis	➢ Design collaboration (PLM / Automotive) ➢ Authorization management (Security) ➢ Compensation management (Insurance)	Provides the impact of a change, allows automatic propagation of system and ensuring consistency throughout systems.

Table 1. Continued

Type	Examples	Business Value
5. Delivery of information services	➢ Information services in mobile devices (Telco) ➢ Stock market information (Financial Markets) ➢ Customer notification system (Banking)	Enables highly personalized information services, extending the capabilities of publish/subscribe systems.
6. Management of services and processes	➢ Information service delivery management (Telco) ➢ Management of Billing and charging (Telco)	Management of service quality agreements and key performance indicators. Event-driven functions are enablers for this type of applications.
7. Proactive systems	➢ Check volume prediction and management (Banking) ➢ Design validity check (PLM/Automotive)	Enabler for proactive behavior in which a system can eliminate possible problems due to predictor's analysis.

Note, that this classification is not a partition; an application can combine application integration, awareness to business situations and automatic behavior in a single application. The idea is to build all the required event-driven functionality in a seamless fashion.

3 Toward an Event-Driven Architecture

Fig. 1 shows an example of event processing within a trade example. Table 2 explains the various event processing artifacts, we call event processing mediations.

Table 2. Event Processing mediations

Event Processing mediations ID	Explanation
EP1	Enrich: Add attributes to the order event (e1) based on a query in a database result in enriched order event (e3)
EP2	Validate: Perform validation test on the enriched order event (e3), may result in an alert event (e4).
EP3	Aggregate: Match the enriched order event (e3) with the appropriate allocation event (e2), creating an allocated order event (e5)
EP4	Validate: Check credit for the allocated order event (e5), possibly revise order to get confirmed order (e6)
EP5	Route: Make a decision to which exchange the confirmed order (e6) should be sent.
EP6	Route: Send time-out alert (e8) for not getting ack. From the exchange (e7) for the confirmed order (e6)
EP7	Compose: Match seller and buyer (both confirmed orders) according to fairness criteria and create a settlement event (e10)
EP8	Route: Retrieve historical order events (e9) as part of the compliance process.

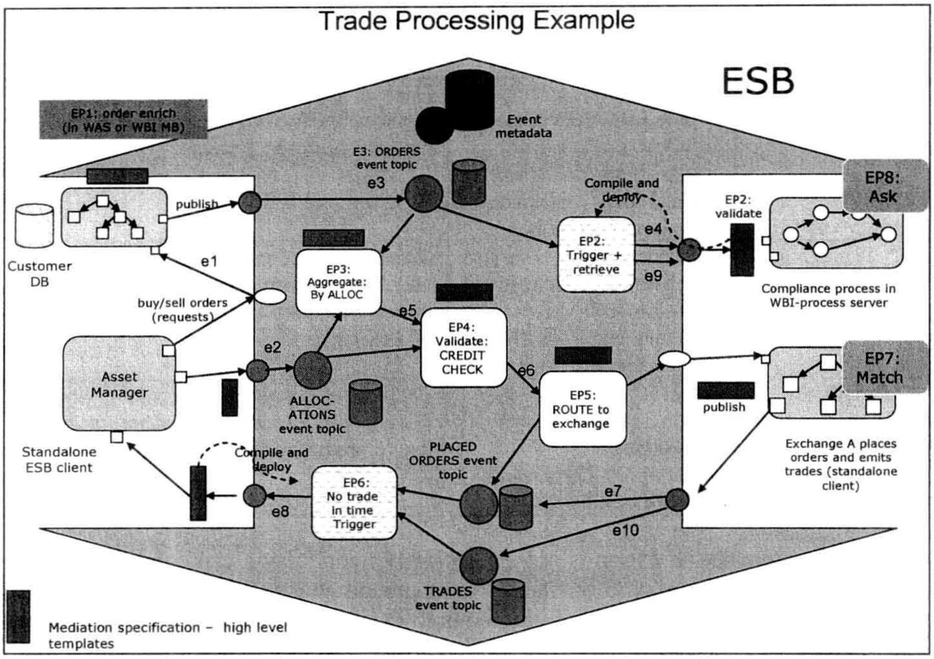

Fig. 1. An example of an event-driven application

The main challenges in setting up this type of applications are:

1. Providing architecture with standard interfaces so that:
   a. Event sources can emit events in various formats easily
   b. Event consumers can obtain events easily
   c. Different event processing components can plug in and interoperate
   d. On-line and historical events can be seamlessly processed
2. A standard event processing language will be used as a basis for all processing, and will enable to make it main-stream computing, independent in proprietary languages.
3. Develop tools that will enable to perform part of it as appropriate for user computing.

Our architecture composed of sources and consumers of events and to the event processing mediations. An event processing mediation is a triplet <selector, processor, emitter>, where:

1. Selector: selects the events that participate in the actual processing, usually by satisfying some pattern on multiple event streams (example: select a pair that consists of buy event and sell event that satisfy a complex pattern).
2. Processor: validate, enrich, compose etc..., creates new event.
3. Emitter: a decision process about the destination of the produced event.

Event processing is a descendent of concepts like "composite events" [5] and "situation detection" [6], but also has some partial overlap with the area of data stream management [7].

4 Relation to Business Rules

Some of the "frequently asked questions" about event processing refers to the relationship between event processing and business rules. This section will revisit the main features of business rules and position event processing in that context.

4.1 Introduction to Business Rules

The name "business rules" refers to a programming style that is intended to take some of the business logic out of the ordinary programming, to achieve more agility.

Barbara Von Hale [8], one of the industry business rules leaders, classifies rules according to the way they are activated. The two rule classes are:

- Service-oriented rules which are activated by explicit request. Business rule products such as: Blaze Advisor, and ILOG's Jrules are examples.
- Data-change-oriented rules which are activated by change in data. Business rule products such as: Versata and USOFT are examples.

Another classification is based on what the business rule does. This classification was done by the Business Rules Group:

- Derivation rules: a statement of knowledge that is derived from other knowledge in the business. This is further classified into:
  - o Mathematical calculations
  - o Inference of structural assertions
- Action assertion: a statement of a constraint or condition that limits or controls the actions of the enterprise.

4.2 What Is the Relationship Between Event Processing and Business Rules?

Event processing functionality can be expressed by the programming style of business rules.

Event processing can roughly be partitioned to two types of functionality:

- Event derivation: complex events are derived as a function of other events (selection + composition)
- Triggering: actions are triggered by (possibly complex) event.

While these type of processing can be expressed in the rule style, the major difference between event processing and other rule types is that the subject matter are events rather than data or structural assertions.

Table 3 compares traditional business rules and event processing.

Note that regular business rules can also be embedded in event processing, example: routing decisions can be done by decision trees, validation decision can employ rules.

5 Future Directions

The area of event processing now is in its early phases, there are some products, and some applications, but it is still climbing the hill towards prime time. Several developments will accelerate it use:

Table 3. Comparison between event processing and business rules

Dimension Name	Traditional business rules	Event Processing
Rule input	Facts - predicates in first order logic, typically relations among entities (e.g. John is Jim's manager) Data-elements – values of attributes (e.g. John's salary value).	Events – as defined in this document (e.g. John's salary promotion). The input is mostly event, but event processing may consult with facts and data-elements.
Rule output	Knowledge creation: Inference for facts (e.g. if John is Jim's manager, and John is Dan's manager, and peer is defined as two employees reporting to the same manager the system can infer that Jim and Dan are peers) Derivation for data (e.g. the account-balance = the sum of all deposits minus the sum of all withdrawals) Behavioral: Enacts an action (IF-THEN) example: IF there is a traffic jam than re-calculate the route Prevents an action example: if a transaction updates the salary of an employee to be higher than the salary of his manager, then REPAIR THE TRANSACTION to the maximal raise that does not violate the constraint	Complex Event Detection – (e.g. the complex event – at most two bike sells within the last hour) In event processing the created knowledge is in form of new event instances. Behavioral rules where the condition is a complex event. Both type of behavioral systems can be event-driven
Rule invocation	By Request: Rule is activated by specific request or as part of another request (e.g. query). Called: backward chaining (in inference rules, e.g. while answering the question about finding all the peers of Dan), lazy evaluation (in derivation rules, e.g. when calculating the final price). By Trigger: Rule is activated when there is some change in the universe in the rule's scope. Called: forward chaining (in inference rules, when a fact is added), eager evaluation (in derivation rules, when there is a withdrawal from the account).	Event processing functionality can be invoked both by trigger and by request. By trigger is detecting the complex event anytime that the condition is matched. By request --- process events in retrospect .
Rule processing type	Snapshot processing: The rule processes a single snapshot of the universe (regular SQL). Temporal processing: The rule processes a set of changes in the universe done over time (e.g. OLAP tools for data)	Temporal processing is one of the main characteristics of event processing.

1. Standardization: The ability to have standard event schemata, standard APIs, and standard event processing language will make events available and event processing services a major service.
2. Event extraction: There are many sources (such as: news streams, video streams, Blogs, audio files) that include events, but the extraction of these events to form in which events can be processed is very preliminary today. This area will be more mature in a few years.
3. Embedding event driven behavior in programming paradigms: being part of modeling and case tools, introduction into business process management tools.
4. Creation of highly scalable infrastructure that supports high rates of event processing.

References

1. Jennifer Widom, Stefano Ceri: Active Database Systems: Triggers and Rules For Advanced Database Processing. Morgan Kaufmann 1996
2. Roberto Baldoni, Mariangela Contenti, Antonino Virgillito: The Evolution of Publish/Subscribe Communication Systems. Future Directions in Distributed Computing 2003: 137-141
3. T. T. Mai Hoang: Network Management: Basic Notions and Frameworks. The Industrial Information Technology Handbook 2005: 1-15
4. Jun-Jang Jeng, David Flaxer, Shubir Kapoor: RuleBAM: A Rule-Based Framework for Business Activity Management. IEEE SCC 2004: 262-270
5. Shuang Yang, Sharma Chakravarthy: Formal Semantics of Composite Events for Distributed Environments. ICDE 1999: 400-407
6. Asaf Adi, Opher Etzion: Amit - the situation manager. VLDB J. 13(2): 177-203 (2004)
7. Arvind Arasu, Brian Babcock, Shivnath Babu, Mayur Datar, Keith Ito, Itaru Nishizawa, Justin Rosenstein, Jennifer Widom: STREAM: The Stanford Stream Data Manager. SIGMOD Conference 2003: 665
8. Barbara Von Halle. Business Rules Applied: Building Better Systems Using the Business Rules Approach Wiley, 2002.

Enabling Semantic Web Inferencing with Oracle Technology: Applications in Life Sciences

Susie Stephens

Oracle, 10 van de Graaff Drive, Burlington, MA 01803, USA
susie.stephens@oracle.com

Abstract. The Semantic Web has reached a level of maturity that allows RDF and OWL to be adopted by commercial software vendors. Products that incorporate these standards are being used to help provide solutions to the increasingly complex IT challenges that many industries face. Standardization efforts for the Semantic Web have progressed to the point where efforts are starting in the integration of ontologies and rules. This paper showcases the implementation of a Semantic Web rulebase in Oracle Database 10g, and provides examples of its use within drug discovery and development. A more detailed paper is currently being prepared with Dr. Said Tabet of the RuleML initiative where a more detailed design and specification is provided explaining the

1 Introduction

The current Web is an environment developed primarily for human users. The Semantic Web intends to extend the use of the Web by making documents machine-accessible and machine-readable [3]. A number of standards have been recommended to achieve this goal, including Extensible Markup Language (XML), Resource Description Framework (RDF), and Web Ontology Language (OWL) [6, 20, 21]. As these standards reach maturity, commercial software vendors begin to incorporate the technologies into their products. This trend is illustrated by support for RDF and OWL in the Oracle Database and the integration of RDF metadata in Adobe images.

Having a language for sharing rules is often seen as the next step in promoting data exchange on the Web [2]. Semantic Web rules would allow the integration, transformation and derivation of data from numerous sources in a distributed, scalable, and transparent manner [10]. Rules would themselves be available as data on the Web, and therefore would be available for sharing.

RuleML was the first initiative to bring rules to the Web with support for XML and RDF [5]. It provides a modular lattice of rule sub-languages that have various levels of expressiveness. The Semantic Web Rule Language (SWRL), acknowledged by W3C as a member submission in 2004, is a result of continued efforts in this area [17]. SWRL is layered on existing W3C standards, and integrates OWL and RuleML [4].

The life sciences industry has been taking advantage of rules in drug discovery and development for many years. The use of rules covers areas as broad as the identification of siRNA for gene knockouts [30], the prediction of oral bioavailability of chemical compounds [18], the prediction of human drug metabolism and toxicity of novel compounds [11], decision support rules for patient diagnosis and clinical trial patient selection [26], the determination of the progression of cancer during treatment

A. Adi et al. (Eds.): RuleML 2005, LNCS 3791, pp. 8–16, 2005.