

Chapter X

Rules Capturing Events and Reactivity

Adrian Paschke

Freie Universität Berlin, Germany

Harold Boley

National Research Council, Canada

ABSTRACT

Event-driven reactive functionalities are urgently needed in present-day distributed systems and dynamic Web-based environments. Reaction rules constitute a promising approach to specify and program such reactive systems in a declarative manner. In particular, they provide the ability to reason over events, actions and their effects, and allow detecting events and responding to them automatically. Various reaction rule approaches have been developed, which for the most part have been advanced separately, hence led to different views and terminologies. This chapter surveys and classifies the wide variety of rule-based calculi approaches, engines and languages for event, action and state processing, and describes their main features. Founded on the original formalisms, major lines of development are traced to the present and extrapolated to the future.

INTRODUCTION AND MOTIVATION

Event-driven applications based on reactive rules and in particular ECA rules, which trigger actions in response to the detection of events, have been extensively studied during the 1990s. Originating from the early days of programming language

where system events were used for interrupt and exception handling, active event-driven rules have received great attention in different areas such as active databases, which started in the late 1980s (Paton, 1999) (Widom and Ceri, 1996), real-time applications, and system and network management tools which emerged in the early 1990s, as well

as publish-subscribe systems, which appeared in the late 1990s (Baldoni et al, 2003). Recently, a strong demand for reactive functionalities has come from the Web community, where data sources are enriched with reactive behavior to make them active and allow dynamic behavior (as opposed to the conventional query-driven passive Web). Additionally, there is an increased interest in industry and academia in event-driven mechanisms such as Complex Event Processing (CEP) (Luckham 2002) and high-level Event-Driven Architectures (EDA) (Chandy, 2006). (Pro-)active real-time and just-in-time reactions to events and relevant situations are a key factor in upcoming agile and flexible IT infrastructures and distributed service oriented environments. Among many other industry developments, Real-Time Enterprise (RTE), Business Activity Management (BAM) and Business Performance Management, as well as closely related areas such as IT Service Level Management (SLM) (Paschke, 2007a), are business drivers for this renewed interest.

A great variety of approaches have been developed for reaction rules, which have for the most part evolved separately in their respective domains, and which have led to different views and terminologies. Some detect events as they occur and directly react to them, as e.g. in active databases or implicitly in production rule systems which are fired by changing conditions in the working memory. Others, e.g., KR event / action logics, focus on the development of axioms to formalize the notions of actions and causality, and on the inferences that can be made from the fact that certain events are known to have occurred or are planned to happen in future.

This chapter defines the basic concepts of reaction rules and event processing technologies. It introduces a general multi-dimensional classification scheme for the major families of reaction rule languages, with the four dimensions *event*, *condition*, *action*, and *processing and reasoning* for the classification of the space of (possible) language features. It also gives a survey of the major

lines of reaction rule approaches and systems of the past decades. The objectives are:

- to provide a common set of foundational terms and definitions;
- to classify and survey the wide variety of reaction rule languages and event, action and state processing approaches;
- to provide a common vocabulary to describe the language features of different reaction rule languages.
- to provide a common frame of reference as a basis for understanding and analyzing the core properties of a reaction rule language and a rule-based reactive system architecture;
- to support the selection of an appropriate reaction rule approach and language for engineering a rule-based system in a particular application domain.

This chapter is structured as follows: Sections 2 and 3 introduce foundational definitions and a multi-dimensional classification scheme for the major families of reaction rule languages. The central section 4 gives a survey of the main families of reaction rule languages, their core properties and features, and the major developments in each field in the past decades till now. Section 5 highlights future trends in event processing and reaction rule technology. Finally, section 6 concludes this chapter.

FOUNDATIONAL DEFINITIONS

This section introduces the foundational notions and definitions of the reaction rule and rule-based event processing technology.¹

Atomic Event

An atomic event (also raw event or primitive event) is defined as an instantaneous (at a specific

36 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/rules-capturing-events-reactivity/35861

Related Content

Agile Development of Rule Systems

Joachim Baumeister, Dietmar Seipeland Frank Puppe (2009). *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches* (pp. 253-272).

www.irma-international.org/chapter/agile-development-rule-systems/35862

Supplementing UML with Concepts from ORM

Terry Halpin (2001). *Unified Modeling Language: Systems Analysis, Design and Development Issues* (pp. 168-185).

www.irma-international.org/chapter/supplementing-uml-concepts-orm/30578

Rules Verification and Validation

Antoni Ligezaand Grzegorz Nalepa (2009). *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches* (pp. 273-301).

www.irma-international.org/chapter/rules-verification-validation/35863

Systematic Design of Web Applications with UML

Rolf Hennickerand Nora Koch (2001). *Unified Modeling Language: Systems Analysis, Design and Development Issues* (pp. 1-20).

www.irma-international.org/chapter/systematic-design-web-applications-uml/30568

Linking UML with Integrated Formal Techniques

Jing Liuand Jin Song (2001). *Unified Modeling Language: Systems Analysis, Design and Development Issues* (pp. 211-224).

www.irma-international.org/chapter/linking-uml-integrated-formal-techniques/30580