Event-driven Reactivity
A Survey and Requirements Analysis

Kay-Uwe Schmidt, SAP
02.06.2008

Co-Authors:
Dakro Anicic, FZI
Roland Stühmer, SAP
1. Introduction
2. Survey of Event-triggered Reactivity
3. Requirements
4. Future Work
Introduction

Event-driven processing becomes ever important in various application domains
- Ranging from traditional business applications, like supply-chain management
- To the entertainment industry, like online gaming applications

The market value should increase tenfold by 2010 and should reach something like $4bn in total (IBM).

Key role of even-driven processing for making business more agile

Main benefit of "eventizing" business systems
- Event processing introduces a kind of reactive dynamics
- Enabling active responding on signals sensed/derived from the current context

Event-triggered Reactivity (EtR)
- Opens great opportunities for system/process improvements.
1. Introduction
2. Survey of Event-triggered Reactivity
3. Requirements
4. Future Work
First mention of the ECA paradigm by U. Dayal, A. P. Buchmann, and D. R. McCarthy in 1988

- “Rules are objects too: A knowledge model for an active, object-oriented database system”. In Lecture notes in computer science on Advances in object-oriented database systems, pages 129-143, New York, NY, USA, 1988. Springer-Verlag New York, Inc.
- This paper describes work in progress on the knowledge model (an extended data model that includes constructs for representing rules) of HiPAC, an active, object-oriented DBMS.

“Central to our knowledge model is the concept of event-condition-action (ECA) rules, which generalizes the many different mechanisms introduced previously in the literature to support active DBMS functions.

- The event part of an ECA rule specifies database operations, temporal events, or signals from arbitrary processes;
- the condition part specifies database queries;
- and the action part specifies a program.
- When the event occurs (is signaled, the condition is evaluated; if the condition is satisfied, the action is executed.”

**ON** event **IF** condition **DO** action
Event Processing

The two kinds of event processing:
- Complex Event Processing (CEP)
- Event Stream Processing (ESP)

Dealing with different problems in event processing using different approaches

ESP – extraction of simple events from a stream
- Events are totally ordered by time
- Emphasis of ESP on efficiency for high throughput and low latency
- Algorithmic stock trading

CEP – extraction of complex event patterns from a cloud
- Only a partial temporal order of events
- Other partial order of interest for CEP is for instance causality
- More time and memory needed
- Business Process Monitoring

CEP is a superset of ESP

CEP and ESP nowadays adopt each others approaches
Event Detection Languages

Development for active databases in the late 80’s and early 90’s
- use complex event specifications to facilitate database triggers
- not only listen to simple events but observe complex combinations of events until the trigger procedures are executed

Simple events carry a type, their occurrence time and possibly other parameters

Creation of complex nested expressions, using operators like And, Or, Sequence, and others

Complex events are detected from occurrences of one or more of simple or complex events

Structure of event patterns: event operators

A nested event operator might have several event types as arguments

An event detector for the given pattern functions as a stream pattern matcher and listens for events that satisfy the type constraints and together satisfy the semantics of the given operator, e.g. occurred in sequence
Complex Event Detection Algorithms

Finite State Automata
- Ode 1992
- Transformation of complex event expressions into deterministic finite automata
- Convenient model to define the semantics of complex event operators
- Downside no acceptance of overlapping occurrences of the same complex event

Colored Petri Nets
- SAMOS 1994
- Convenient model to define the semantics of complex event operators
- Also the detection of overlapping occurrences is possible

Graph-based Approaches
Graph-based Approaches

Sentinel based on Snoop, SnoopIB in the mid 90’s

Construction of the graph from the event expressions

Example: (E1,E2);E3

The graph is a directed acyclic graph and generally does not form a tree for two reasons: nodes may have several parents, when their represented expression is part of more than one complex events, and secondly there is no single root node, when there is no overarching, single most complex event.
Overview of the State of the Art in Event and Action Processing

ECA Rules

Event Processing
- Complex Event Processing
  - Graph-based Approaches
  - Finite State Automata
  - Colored Petri Nets
- Event Stream Processing
  - Pattern Matching in Streams then DB Tables, Sliding Windows
  - StreamSQL

Action Processing
- Production Rules
  - Rete (Forgy in 1982)
  - TREAT (Miranker 1990)
  - LEAPS (Batory 1994)
- Logic
  - IBM Tivoli Enterprise Console (Prolog)
  - Bry, Paschke 2007
1. Introduction
2. Survey of Event-triggered Reactivity
3. Requirements
4. Future Work
Requirements for Handling EtR

Efficient combination of complex event processing with action processing

Comprehensive framework capable to deal with
- Semantics of events and actions
- Termination of rule processing
- Rule ordering

Context and Situation
- Event + Condition + Context = Situation

Reasoning over all reactivity rule constituents
Vision Event-triggered Reactivity

Holistic approach: Event-driven Reactivity

Unique handling of the different constituents of an event-driven architecture
- events, actions, conditions, contexts and situations

Realization of next generation efficient and manageable event-driven (reactive) applications

Decrease the complexity of setting-up/evolving event-driven applications, that nowadays requires lots of manual work, especially in defining what an event is

Increase the benefits (added value) of such applications, which are currently constrained on the complex monitoring of events

Open new possibilities to apply them in highly dynamic and distributed environments
Vision Requirements

Efficient modeling of the sense-and-respond (reactive) nature of a system, especially its contextualization

Comprehensive management of the reactivity life cycle of a system, including automatic discovery of relevant situations, efficient detection of events reasoning about actions

Efficient implementation of the reactivity life cycle management
Context Detection

In fact, we argue that the ECA (event-condition-action, such as it is) model is too simple presentation of the (intelligent) event processing nature

Efficient context detection process is inevitable for the efficient event processing and is totally neglected in the literature

Unified mechanism for formal representation of all phases in the reaction cycle is needed for efficient and complex event processing

Usage of a richer conceptual model for describing reactions on events
Context as a first class citizen in the event processing is currently completely missing in the literature for event processing

Context-based event processing is the next "big thing"

- Shaping of the future of the computing
- Context-Driven Architecture (CoDA) is the most promising paradigm that will extend SOA

New possibilities for event triggered reactivity by reasoning about situations and context
Reasoning Services about situations and context

Opens new possibilities for event triggered reactivity

Situations as formal logic models; some very interesting reasoning services can support the whole event processing

The system can check formally the consistency of the system and backtrack if a conflict (meaning inconsistency in the system) is to appear.

Another service would be the synchronization of situations if we consider that two or more reactions will run in parallel, which is a quite natural assumption in the rich-event systems.
1. Introduction
2. Survey of Event-triggered Reactivity
3. Requirements
4. Future Work
Future Work

Development of a new conceptual model and architecture of event-triggered reactivity (EtR)
- Introducing novel concepts (situation and context)
- Its formal, logic-based representation

Development of a model for managing the whole life cycle of EtR, including
- Language for modeling EtR concepts (e.g. situations, context)
- User-friendly editor based on pattern modeling metaphor
- Methods for ensuring the consistency of such a rule base and its interoperability with other reactive systems
- New methods and tools for the automatic discovery of complex event and situation patterns from stream data by taking into account their evolution as well
- New algorithms for scalable ECA reasoning, based on the selected logic and its implementation in a new reasoning engine that will serve as the event-, condition- and action-handler in a reactive system.
- Realize, test and refine an integrated software framework for the management of EtRs life cycle, containing elements of the distributed event processing, that can be easily deployed in the selected legacy landscape
- Development of use cases, their implementation, testing and evaluation in real-world pilot studies in order to validate proposed model and framework.
Thank you!
# Definition and halftone values of colors

<table>
<thead>
<tr>
<th>Secondaries</th>
<th>100%</th>
<th>85%</th>
<th>70%</th>
<th>55%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB 145/175/170</td>
<td>RGB 163/162/161</td>
<td>RGB 171/156/160</td>
<td>RGB 175/154/159</td>
<td>RGB 179/152/158</td>
<td>RGB 183/151/157</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary colors</th>
<th>100%</th>
<th>85%</th>
<th>70%</th>
<th>55%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB 204/204/204</td>
<td>RGB 192/192/192</td>
<td>RGB 180/180/180</td>
<td>RGB 168/168/168</td>
<td>RGB 156/156/156</td>
<td>RGB 144/144/144</td>
</tr>
</tbody>
</table>


| RGB 158/48/57 | RGB 157/47/56 | RGB 156/46/55 | RGB 155/45/54 | RGB 154/44/53 | RGB 153/43/52 |

**Tertiary colors**

| RGB 158/48/57 | RGB 157/47/56 | RGB 156/46/55 | RGB 155/45/54 | RGB 154/44/53 | RGB 153/43/52 |

© SAP 2008 / Kay-Uwe Schmidt / Page 21
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned and associated logos displayed are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Einige von der SAP AG und deren Vertriebspartnern vertriebene Softwareprodukte können Softwarekomponenten umfassen, die Eigentum anderer Softwarehersteller sind.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge und andere in diesem Dokument erwähnte SAP-Produkte und Services sowie die dazugehörigen Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und in mehreren anderen Ländern weltweit. Alle anderen in diesem Dokument erwähnten Namen von Produkten und Services sowie die damit verbundenen Firmenlogos sind Marken der jeweiligen Unternehmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länder spezifische Unterschiede aufweisen.


SAP übernimmt keine Haftung für Schäden jeglicher Art, einschließlich und ohne Einschränkung für direkte, spezielle, indirekte oder Folgeschäden im Zusammenhang mit der Verwendung dieser Unterlagen. Diese Einschränkung gilt nicht bei Vorsatz oder grober Fahrlässigkeit.

Die gesetzliche Haftung bei Personenschäden oder die Produkthaftung bleibt unberührt. Die Informationen, auf die Sie möglicherweise über die in diesem Material enthaltenen Hotlinks zugreifen, unterliegen nicht dem Einfluss von SAP, und SAP unterstützt nicht die Nutzung von Internetseiten Dritter durch Sie und gibt keinerlei Gewährleistungen oder Zusagen über Internetseiten Dritter ab.

Alle Rechte vorbehalten.