Achieving Semantic Interoperability By Using Complex Event Processing Technology

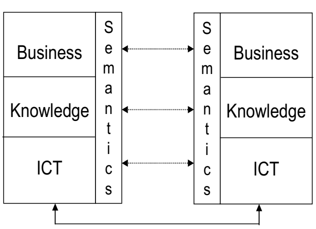
Margarete Donovang-Kuhlisch

Geschäftsbereich Verteidigung
IBM Deutschland GmbH
Gorch-Fock-Str. 4
D-53229 Bonn
mdk@de.ibm.com

Abstract: State-of-the-art interoperability between heterogeneous systems on a technical level builds on services oriented architectures (SOA) for a given problem space. Semantics can be introduced by the use of descriptive languages and reasoning algorithms, usually implemented within a message-based enterprise service bus (ESB), the messages triggering service invocation and consumption. Complex event processing (CEP) allows for the definition and detection of situations based on the correlated occurrence of a set of messages, rather than single ones. This paper introduces the technology and lists first references.

1 Problem Statement

Increasing collaboration beyond organizational boundaries and during long-lasting and complex processes is a global trend. Interoperability is usually defined as “the ability of two or more systems or components to exchange information and [meanfully] use the exchanged information”[1]. Up to now interoperability has mainly been dealt with in the information systems and communication technology (ICT) layer. In order to achieve seamless business interaction between organizations interoperability has to include semantic descriptions for the business and knowledge layers.



Necessary for the achievement of such enterprise interoperability[2] is the application and combination of latest research results in the three key areas of the interoperability problem space: enterprise modeling, ontology-based systems plus ontology modeling and mo-dern ICT solutions: Model-Driven Architecture (MDA), semantic web and Agent-Based Systems (MDA).

Picture 1: Layers of Interoperability

[1] F.e. EICTA Interoperability White Paper [EIC04]
[2] as targeted f.e. within the European Commission sponsored project ATHENA [IS07]

Approach

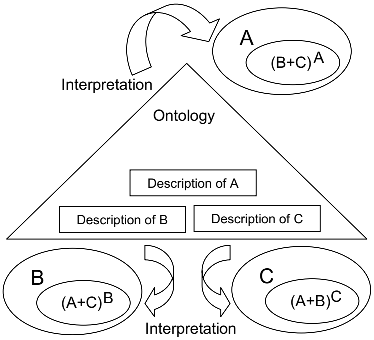
Modern ICT solutions thriving for interoperability are based on SOA and consist of a network of service providers. Services are invoked by client applications by means of messages that conform to descriptive schemas. In order to accommodate the business need for semantic interoperability, the message brokering infrastructure needs to be able to:

- provide independence of individual component technology and location
- support dynamic, automatic flow control
- perform data transformation between components
- supply a repository for storing definitions
- orchestrate components into business processes.

Most available enterprise service bus (ESB) products[3] provide a reasonable set of tools to deal with the definition and administration of business process activities. They use a metadata-driven approach and store definitions of components, characteristics, data formats, events, navigation paths and actions in a common directory. Message-based communication within the ESB's message brokerage can follow any combination of the following patterns: request / reply, fire & forget, publish / subscribe or event-driven.

The Role of Ontologies

An ontology can be considered as an intensional account of what exists in a certain domain [VL05]. Within a SOA, any service description is an ontology in itself: WSDL[4] allows the description of atomic and composite data types understandable as structures of logical predicates and constraints, which altogether specify a sort of intensional description.



The necessary interpretation of the various service providers' ontologies within the ESB depends on a set of predicates that is adopted, shared and understood by every node in the distributed environment. Basically, there are two alternatives to set up the schema integration mappings, that is any-to-any or any-to-one, and two ways to execute the integration logic: centralized on a single node or distributed among several nodes (decentralized). Within the Semantic Web Community, shared ontologies are focused upon.

Picture 2: any-to-one decentralized model for semantic interoperability

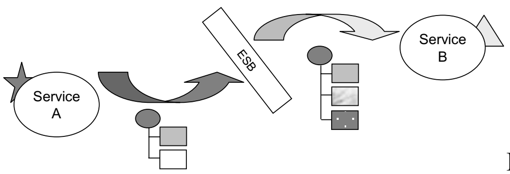
---

[3] f.e. Gartner Application Integration Magic Quadrant 2003
[4] Web Services Definition Language

Any-to-one decentralized systems therefore are peer-to-peer infrastructures in which endpoints directly communicate with each other, but which are provided by the ESB with service descriptions from a common shared business model. Each service models the other services with interpretations of ontology fragments that correspond to the union of their schemas (as shown in the nested ellipses in Picture 2).

Event-Driven ESB

For IBM the ESB is more than a set of software products, it also is a concept to achieve technical, procedural and semantic interoperability within SOA. Therefore the IBM ESB-related offerings include the implementation of the core enterprise services (CES) especially in the disciplines: messaging, discovery and mediation. Core technology is the WebSphere® Message Broker, which especially performs transport, event and mediation services to facilitate the integration of large-scale heterogeneous applications. Mediation can be performed both on the protocol (transformation) and the content level (transformation and augmentation).



Picture 3: scope of mediation in the ESB

Event services provide event detection, triggering and distribution capabilities. An event thereby can be anything from sensor input, message receipt to evolution of a specific situation.

Complex Event Processing

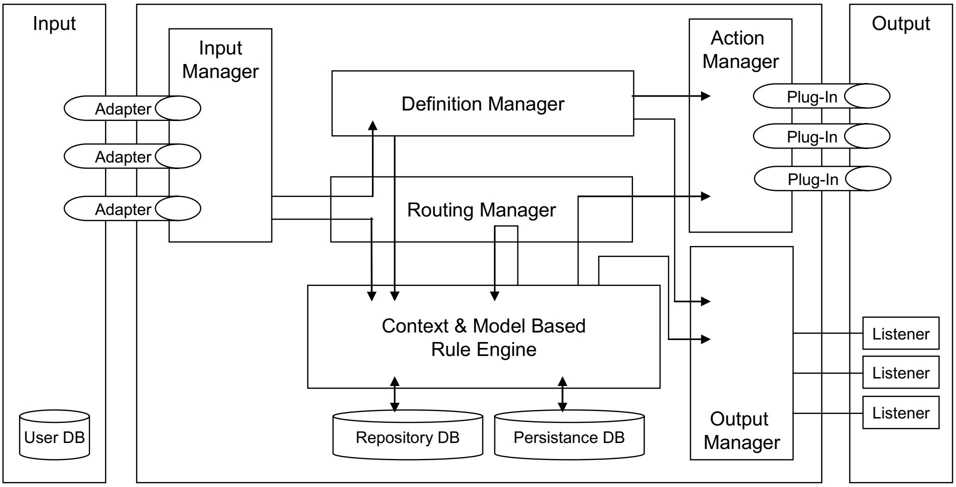
CEP [Sg06] is a technology based on the IBM HRL[5] research asset AMiT[6] and designed to detect complex situations that involve a context-sensitive composition of messages and events. It can enrich the mediation services of the ESB by analyzing the messages and monitor predefined relationships between messages in the flow. Application examples include: validation and compliance, aggregation and monitoring, exception detection and handling as well as routing. Gartner has recognized CEP as an emerging market.

CEP within WebSphere Message Broker is implemented as two nodes: the IntelligentFilter node performs a filtering or routing task by using CEP situations to determine a message's destination. The SituationManager node performs an observation task, providing notification whenever a situation is detected. A situation thereby can be determined by a particular sequence of events (messages) or the lack of such.

---

[5] Haifa Research Laboratory
[6] Active Middleware Technology

CEP [HRL07] is a lightweight and agile execution engine and includes an Eclipse based authoring tool for the user. At build time, rules and conditions are composed using a wizard. Simulation can be performed to test these definitions before export to the run-time engine. During runtime, events of different format and from multiple sources are fed in and analyzed based on current rules and conditions. Upon detection of the prede-fined patterns, alerts are sent and actions triggered as well as fed back into the engine (nested patterns). Picture 4 illustrates the component model of the runtime.



Picture 4: CEP Runtime Middleware Component Architecture

CEP Usage & References

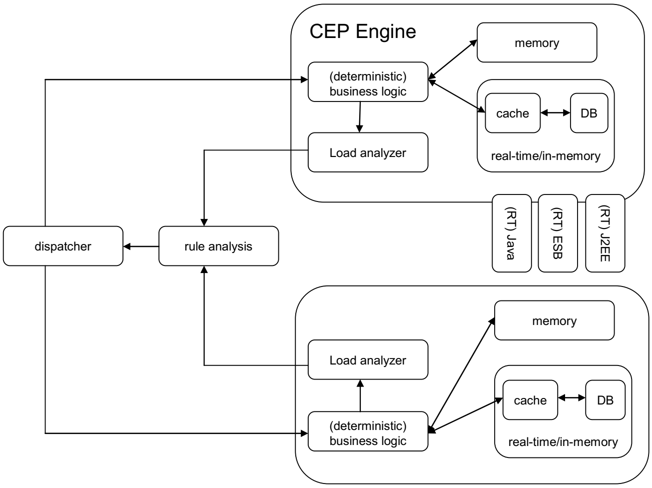
First applications of CEP technology include business activity monitoring for fraud de-tection, content-based routing or process invocation using situation detection to trigger actions. With any of these CEP can enhance the business process management capabili-ties of any organization. Besides its integration in IBM Middleware, the technology is also embedded in Bristol Technology®[7] and Synchron[8]. Ready-to-go solutions exist for the financial (Trade Processing – Transaction Monitoring and Exception Management) and insurance markets (CEP for Underwriting and Insurance Service Hub). Customer references include:

- Bank HAPOALIM (Israel)
- Bank Julius Baer (Switzerland)
- IBM Internal Audit Team (CEP for risk & compliance)
- University Hospital of Nice (Using RFID to track emergency room patients)
- Roche Pharmaceutical (CEP for Contract and Sales Validation).

[7] www.bristol.com
[8] www.synchron-is.de

Value Proposition of Realtime CEP [SO06]

IBM is working on an architecture for distributed realtime complex event processing, thus thriving to introduce the technology also in time-critical sensor-effector chains. To support both real-time and scalable event processing requirements, two methods for distributed processing will be used: state sharing and load balancing. The architecture will take advantages of both methods and enable different quality of services:



Picture 5: real-time & scalable event processing architecture (under construction)

This architecture is designed to support especially time-critical, security and safety related processes like: several mission specific sensors are following targets, also sending information updates on their own location and creating situational alerts – after applying an IFF[9] algorithm – thus triggering actions based on dynamic asset availability ("logistics' sensors").

Literaturverzeichnis

[EIA04] EICTA Interoperability White Paper , European Industry Association, 2004, www.eicta.org

[IS07] www.athena-ip.org

[VL05] Vetere G., Lenzerini, M.: Models for semantic interoperability in service-oriented architectures, IBM Systems Journal, 2005, www.research.ibm.com/jounal/sj/444/vetere,html

[Sg06] Sharon G.: Performing complex-event processing using IBM WebSphere Message Broker, Version 6, IBM HRL, G224-7572-00, 2006

[HRL06] http://www.haifa.il.ibm.com/dept/services/soms_ebs.html

[SO06] Shulman B., Oren D.: RT Complex Event Processing, IBM Event-based Middleware & Solutions Group, HRL, 2006

[9] Identification Friend or Foe