Active Management Technologies in the e-business Management Services Offering

Technical Brief

Written by:
Opher Etzion, Dagan Gilat


IBM Research Laboratory in Haifa
Matam Advanced Technology Center
Haifa 31905, Israel
The leading edge:

This document summarizes the leading edge of the e-bMS service offering and discusses the technological novelties that make it happen.

The e-bMS follows the “autonomous computing” paradigm that IBM applies in different domain. It answers the following problem:

Given a set of goals in the business process level (e.g. performance, diversified service level, availability), and a given configuration that describes the relationships among entities (business processes, activities, applications, resources), construct a set of monitoring rules that determine when a goal may be violated, and activate a correcting action where applicable.

The aim is to ensure the health and continuity of e-business systems.

In order to support such an ambitious goal, a set of technological novelties is required:

(1). A powerful rule engine that can provide high performance, and support complex patterns over the event history.
(2). An inference mechanism that converts semantic descriptions of entities, dependencies and goals into a set of rules.
(3). A decision support system that can find the most appropriate action to be taken in real time.

We shall start by a short survey of the basic technologies, followed by the description of the e-bMS architecture components and their role.
Basic Technologies:

Many contemporary systems require active functionality. Active functionality is event driven, and it is either reactive, reacts to some phenomenon that occurs, or proactive, using current events and context information in order to change future events. The active technologies framework has been developed in IBM Research Laboratory in Haifa (HRL). It is intended to provide tools for development and maintenance of active applications. The technology is based on the observation that in many cases an action is not triggered by a single event, but by a possibly complex composition of events. These applications exist in many domains and are very useful for e-business applications (stock market, business opportunities, sales alerts, etc.), operations management, customer relationship management, command and control applications, and monitoring systems.

![Active Technologies Overview](image-url)
The main technologies are:

- **Amit - Active Middleware Technology**
  Resolving a major problem in this area: the gap that exists between events reported by various channels and the reactive situations that are the cases to which the system should react. These situations are a composition of events or other situations (e.g., "when at least four events of the same type occur") or content filtering on events (e.g., "only events that relate to IBM stocks") or both ("when at least four purchases of more than 50,000 shares have been performed on IBM stocks in a single week"). Amit is context-sensitive in the sense that each situation is relevant within a certain temporal or spatial context. Examples: during working hours, 5 minutes from the time of the last alert, within the boundaries of the 8th district of Paris.

- Enabling the application to detect customized situations without having to be aware of the occurrence of the basic events.

The main difficulties in current tools that Amit handles:

- **Effectiveness** (expressive power, capabilities, modeling), which for the user is the answer to the question: How much code outside the rule system should be written? How many rules should be defined? How easy is to debug such an application?

- **Efficiency**: how efficient the rule system is?

Amit has been created after a careful study of the state of the art in this area, including experience surveys about problems in the implementation and deployment of rule based systems, of a team of people who has expertise and experience in this area, and firmly believe that we have a leading edge in those issues.
• Adi – Active Dependency Integration.

Adi Main goal is creating domain dependent modeling tools that will be able to infer situation for monitoring rules from higher-level abstractions. This component provides model-based view, in which the model definition derives the monitoring needs. Adi was applied successfully to eBusiness Management including the definition of Business Process, Activities, Resources (Both IT and non IT), Constraints and dependencies. New entities and dependencies types can be defined to feet to any domain or application.

Adi rules are created automatically from the system model (the entities, dependencies and constraints). Those rules create complex hierarchical situations that automatically alert of situations such as Service level agreement violations (response times, activities failures, max failures by customer), activities and business process failures due to application failures or IT failures, automatic propagation of alerts between IT, and application and business processes. Based on the system model - business process, applications and IT resources as well as the connection among them-- a criticality is automatically calculated for each entity. This measure reflects the importance of the entity based on the entire model structure and using probabilities network. In addition many "what if" scenarios are supported, for example: What will be the total revenue for a business process if the availability of a web application will increase from 0.9 to 0.95?
• Arad – Active Run-time Automated Decision making.

*Arad* is a general framework for real-time decision-making that is triggered by active rules. It aims at actively generating an optimized solution and performing an action when a problem arises.

A *problem* is any diagnostic, possibly detected by a situation awareness tool (Amit) that requires a response (e.g. action).

*Arad* is a complimentary technology to Amit and Adi that handles the proactive part of the active management technologies *(enough complaining, its time do something about it)* and uses these technologies as tools in the problem resolution process by considering the system’s state (as monitored by Amit), the dependencies between system components (as modeled by Adi), and other constrains (such as SLA’s);

**The e-bMS architecture:**

The e-bMS set of tools is partitioned to two types of tools: off-line tools that are used to define an e-bMS engagement, and on-line tools that are used to act at run-time.

Figure 2 present the e-bMS offering offline overview.

Offline components includes Business process mapping, IT mapping, Event Mapping, IT discovery, automatic rules inference, criticality and cost calculation, rules authoring and event mining. We will describe each component in detail.
IBM Research Laboratory in Haifa

**Figure 2: eBM Offering Overview (Off line)**
(1.) Business Process Mapping

**e-bMS Business Process Mapping** builds on the IBM Business Decomposition methodology by adding a horizontal mapping of a transaction through business processes. The "Business Process Mapping" methodology examines traditional business services and processes and adds the dependencies of the external service providers to a business. These dependencies might be the suppliers of sub-assembly parts to a manufacturer, the finished goods of a clearinghouse, or a shipper of products or services. This methodology also includes non-monitored dependency resources such as facilities, people, and other infrastructure devices. It also identifies all applications required to support the business processes.

(2.) Automatic rules inference

**e-bMS Automatic rules inference** (Adi) – rules are created automatically from the system model (the entities, dependencies and constraints). Those rules create complex hierarchical situations that automatically alert of situations such as Service level agreement violations (response times, activities failures, max failures by customer), activities and business process failures due to application failures or IT failures, automatic propagation of alerts between IT, and application and business processes. For more details see Appendix B – HRL Active Technologies Management – Adi.

(3.) Criticality calculation and cost analysis [part of the rules inference]:

**e-bMS Criticality calculation and cost analysis:**
Based on the system model - business process, applications and IT resources as well as the connection among them -- a criticality is automatically calculated for each entity. This measure reflects the importance of the entity based on the entire model structure and using probabilities network. In addition many "what if" scenarios are supported, for example: What will be the total revenue for a business process if the availability of a web application will increase from 0.9 to 0.95?
(4.) **IT Mapping**

**e-bMS IT mapping** expands the idea of technology decomposition by adding data synchronization and platform dependencies. These dependencies extend to all inter and intra-enterprises identified to support the business processes. The business processes and associated applications are then linked to the appropriate IT touch points. This component also uses automatic IT discovery tools such as Ecora and IDD.

(5.) **Event Mapping**

**e-bMS Event Mapping** examines the data streams (events) generated by all levels: business processes (e.g. a business transaction of sale with the sale attributes: product, cost etc.), applications and middlewares (e.g. queue overflow in message queues..), and high-level event about resources technology that are reported through systems management software packages such as Tivoli, Computer Associates Uni-Center, and BMC Border Patrol. According to the desired constraints that the system traces, the events that should be monitored are mapped, and the monitoring tool (system management, instrumentation tool, enterprise’s database) is determined using this process.

(6.) **Event Mining**

**E-bMS Event Mining** – This component applies data mining techniques for event logs of various sources (system management logs, application instrumentation logs, and database update logs) and finds statistical correlations among events, meaning that if event A occurs, event B occurs within a certain time window. This is a predictor for a possible causality relationships among these events, thus event A can serve as a predictor for event B. The event mining technique will be able to detect more complicated patterns. This technology is used in e-bMS for two main purposes. (1). Verification and validation of the dependency model of the enterprise (2). Detection of causality dependencies that cannot be explained by the semantic model.
7). The data store

**e-bMS data store** - is a knowledge base implemented as a DB2 database. It contains all the knowledge accumulated during the business process, IT and event mapping, and the dependencies among them, which makes it a **Semantic Network**

It represents different types of semantics. These types of semantics should be classified, and the appropriate properties for each type of dependency should be formalized. Dependencies exist between entities, first we define the **entity**, **dependency** and **constraints** attributes and then define the active dependency concept.

Figure 3 present the e-bMS offering online overview. Online components include e-bMS filter, System management event sources, Business process event sources and Applications event sources, e-bMS consolidation gateway and the Active desktop. We will describe each component in detail.
Figure 3: e-bMS Offering Overview (Online)
(1.) **Consolidator**

**Consolidator / Consolidator**
The consolidator/Consolidator (/C) tier is used to consolidate, collect, and analyze all the events from all the Filters used in the engagement. Events flow into/C from/F, are inserted into a DB2 event queue, and are then transferred into Amit. Amit collects events and analyzes them to detect situations based on predefined rules, or Amit Metadata. Amit rules support higher-level abstractions, built-in support of temporal operators, context dependent operations. For example: the situation **IBM STOCK HAS GONE UP in at least 3 percent twice within the same week and MICROSOFT stock has gone down in at least 1 percent during that period.**

For more details on Amit, see Appendix B – HRL Active Technologies Management.

(2.) **Automated actions**

**e-bMS Automated Actions**
The subscriber/Subscriber (/S) tier is used to subscribe to and take actions based on detected Amit situations. There are three possible actions that can be taken upon receipt of a detected situation from Amit:

- Operator console message
- e-mail message
- Pager message

It is possible to invoke multiple actions for a single detected situation, and to send e-mail or pager messages to multiple interested parties.

In addition, proactive capabilities are part of the e-bMS service that enable real time decision making for predicted constraint satisfaction (such as: SLA violation for a certain transaction), load balancing, fail over and other possible actions to ensure business continuity. The actions can also be in the business process level (start or stop running business processes).
(3.) Filter

**e-bMS Filter**

The filter/Filter tier is designed to filter incoming events, translate them into a specific-specific format, and securely transport them to the second tier, where the Amit situation manager resides. In a typical e-bMS offering installation, one or more Filter servers will reside in a customer’s environment (that already utilizes BMC or Tivoli), while a single Consolidator server will reside within and be hosted by IBM. Events are extracted from the various event sources and inserted into a DB2 Trigger Table on/F. A java daemon running on/F reads the events and sends them to/C, via HTTPS, across multiple firewalls.

(4.) Active Desktop

**e-bMS Active Desktop**

The active desktop is the graphic display of the user’s console.

The data in the e-bMS Data store will be displayed graphically on the e-bMS Active Desktop. Any pertinent changes in the e-bMS Data store will be reflected on the active desktop.