Amit - Active Middleware Technology

Overview

IBM Research Lab in Haifa
Active Technologies
October 2002
OUTLINE:

The Active Technologies

Amit – Active Middleware Technology

Related Active Management Technologies

Amit Language

Amit User Interface

Amit at Runtime

Conclusions
Active Technologies
What are active applications?

- Many systems contain active (event-driven) components, including:
  - Command and control systems
  - Service oriented systems
  - Awareness systems

- Area examples:
  - Proactive management
  - Monitoring
  - Problem determination...
  - SLAs enforcement
Active Management in a Nutshell

1st sting is harmless, if you are not allergic.

2nd sting is most probably harmless, but beware of the 3rd one - use protection remedy!

3rd sting will most probably harm!

- Single event ("a sting") is often meaningless, but Events composition ("3 sequential stings") - is meaningful !!!
- Events correlation is important ("consider stings related to the same person").
- Timing is important ("3 stings which happened within at most 3 weeks").
- Conditions are important ("if you are not allergic").
- Events may have Uncertainty degree ("most probably harmless").
- Context is important ("in Africa probability is lower than in Europe").
- Reactive behavior ("after the 3rd sting") may be too late, but Proactive ("beware of the 3rd sting after the 2nd one") - desirable !
- Different Strategies may be suggested ("use ointment or spray").
- Predictions may be done ("if more than X persons suffered in the same area, the area is dangerous!").
Active Applications

- **Reactive**: React to something that happens in the system
- **Proactive**: Use predictive methods to redirect the system towards better results and or eliminate problems (or warn in advance)
What is the main idea of Amit?

- In many cases, a single event is meaningless to an application, however a (possibly complex) combination of events is required.

- Example: Alert me if the IBM stock has gone up in 3 percent within two hours, and the Dow Jones did not go up in more than 1 percent at the same period.
What is the main idea of Amit?

More Examples:

- Alert me if three memory problems occurred during the last hour.

- Alert me if the same request was reassigned to three agents, and no answer was given to the requester.
Run Time

Amit
Rule Engine

Detected Situations

events

Definitions
Amit Architecture

- System designers
- Authoring tool (Amit GUI)
- Subscription and action manager
- Event sources
- Event adapter
- Definitions
- Events
- Situation alerts
- AMIT
- Users
Application Types

- Customer Relationship Management
- Policy Management
- Multi-sensor diagnostic system
- System Management
- Network Management
- Active Services in Wireless Environment
- Maintenance Management
- Business Processes Management
- Monitoring Systems
- Service Management
- Personalized publish/subscribe
- Command and Control systems

Active Management Technologies
Amit Main Advantages

- Effective - Expressive, general purpose language for situation definition

- Efficient - Fast Algorithms
Amit Advantages

- Ability to get events from various external sources, various types of events and various types of events reporting (push, pull)
- Controlled filtering at the client level (events’ source level).
- Modeling (using ADI) of all the enterprise activities and components
- Given the dependency modeling – automatic derivation and inference
  The result: rules detecting situations that are implied from the dependency model
- Monitoring the events that are applied to the enterprise activities and components using the result of the automatic derivation and inference
Amit Advantages – Cont.

- **Supporting internal and external functions** relating to any type in the entire monitoring process.
- **External definitions** (expressing the functionality). Nothing is “hard-coded”.
- **Platform independent** - using Java all over the system.
- Amit is a key technology for SLAs' / Q.O.S measurements evaluation and enforcement.
- Can be integrated with location-base services in order to get a location-based decision support system.
The Dependency Graph

Business Process Level:
- BP #1 E-Store
- BP #2 Credit Card Company
- BP #3 Supplier
- BP #4 Shipping Company

Application Level:
- App. #1.1 Store Product Management
- App. #1.2 Customer Management
- App. #2.1 Bank Poling
- App. #2.2 Bank Credit Management
- App. #3.1 Supplier Accounting
- App. #3.2 Supplier Inventory
- App. #4.1 Shipping Orders Management
- App. #4.2 Shipping Scheduling

Resource Level:
- Res. #1.1 Disk A
- Res. #1.2 Disk B
- Res. #1.3 Phone
- Res. #2.11 Communication Center
- Res. #3.11 Server
- Res. #3.12 Database
- Res. #4.11 Database

Component X may impact component Y
## Performance - Benchmark Results

<table>
<thead>
<tr>
<th>World</th>
<th>detected situations / incoming events</th>
<th>% of events participate in situation definition</th>
<th>events per second</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standby World</td>
<td>0%</td>
<td>no situations definition</td>
<td>120,000</td>
</tr>
<tr>
<td>Noisy World</td>
<td>0.3%</td>
<td>1% of events participate in situation definition. No reference events</td>
<td>47,000</td>
</tr>
<tr>
<td>Middle World</td>
<td>4.5%</td>
<td>25% of events participates in situation definition. Average of 0.3 reference events per incoming instance</td>
<td>22,000</td>
</tr>
<tr>
<td>Complex World</td>
<td>24%</td>
<td>75% of events participates in situation definition lots of reference events complicated situations</td>
<td>2,200</td>
</tr>
</tbody>
</table>

- **Complex World**: 75% of events participate in situation definition lots of reference events complicated situations
- **Middle World**: 25% of events participates in situation definition average of 0.3 reference events per incoming instance
- **Noisy World**: 1% of events participate in situation definition. No reference events
- **Standby World**: 0% no situations definition
**Amit Current Status**

- Amit used as a detection technology in the e-bMS (e-business Management Services) offering announced in October 2001.
- Amit was released as a support-pack in MQSI.
Related Active Management Technologies
More active technologies

- run-time monitoring
- automatic action creation
- dependency and inference
- More active technologies

Amit
Active Middleware Technology

Arad
Active Real-time Automated Decision making

Adi
Active Dependency Integration
What?
- creating domain dependent modeling tools that will be able to infer situation for monitoring rules from higher level abstractions.

Why?
- getting the development of active application to be "programmer less" process (well, to a certain degree, at least)...

What is the starting point?
- the notion of dependency: dependencies among entities, which leads to dependencies among events, which lead to impact analysis.
Inferred situations

- Situations are inferred from constraints and dependencies.
- A simple example:
  - We define a constraint:
    ```
    department.budget > sum (employee.salary)
    ```
    such that employee belongs to the department
  - A situation that entails a constraint evaluation is:
    ```
    any (new employee hiring, reassignment of employee, salary raise, budget reduction).
    ```
  - This example can be inferred
Arad main challenges

• Developing a general framework for Real-time Decision making that is triggered by active rules
• Finding solution to constraint enforcement, such as SLA
• A complimentary technology to active detection
• Selecting appropriate action when any decision may affect the entire system, using the fact that we maintain current view of the system for monitoring purposes (including: “What-if” capabilities)
• Handling the proactive part ("enough complaining, time to do something about it")
• Start and learn:
  from specific cases - schedule and reschedule of constrained resources:
  - Agents in call centers
  - IT resources load balancing

to the general cases:
  - Dynamic services
Amit Language
The language issue

- Styles of languages:
  - An SQL like language (superset)
  - XML for data exchange
  - A Wizard

AMIT Framework:
  - Events definitions
  - Rules definitions
  - Import/Export
  - Multiple sources
  - Validity Checking
  - Links to other tools
  - More…
Customer Relationships monitoring Example

- Some of the basic events for a contact are:
  - open contact
  - open request
  - assign request to agent
  - reply to customer
  - close contact
Monitoring Situations:

1: An open request with priority > 3 did not receive any reply for 2 hours

2: A preferred customer sent request with higher priority for the same contact
   Report: customer name

3: A request was reassigned at least 3 times.
   Report: request, action : "e-mail manager"
First step: define events (a metadata entity)

- open-contact:
  new-contact(number), customer(string)

- open-request:
  new-request(number), contact(number), priority(number), customer-type(string)

- assign-request:
  request(number), assigned-agent(string)

- reply-request:
  request(number), agent(string)

- close-contact:
  closed-contact(number)
The key primitive

- A key is a collection of event attributes that are used to match different events.
- These attributes must have syntactic compatibility, and are usually semantically equivalent.
Second step: define matching factors (keys)

- **key-contact**
  - open-contact: new-contact
  - open-request: contact
  - close-contact: closed-contact

- **key-request**
  - open-request: new-request
  - assign-request: request
  - reply-request: request

- **key-agent**
  - assign-request: assigned-agent
  - reply-request: agent
Third stage: define lifespan

- For each situation you want to define, think what is the time interval in which the situation is meaningful.
- Remember, a lifespan is:
  - Bounded by **initiator** and **terminator**
  - Initiator: event and/or startup
  - Terminator (optional): event and/or expiration time.
  - A lifespan may have a key (identical attribute value for all the situation participating events)
Lifespan

- Bounded by two events: initiator and terminator.
- Initiator / Terminator: Event, time, Situation, automatic (after …), etc.
Situation

- General Information
- Participating events
- Matching keys
- Attributes (can be derived)
- Lifespan boundaries
The Building Blocks

Joining Operators

all, sequence

basic events

counting operators

atleast
atmost
nth
operator types:

temporal operators

every, after

absence operators

not, unless
Situation example #1:

Requirement: An open request with priority > 3 did not receive any reply for 2 hours

situation request-opened-for-2-hours
not, expiration-time = 2 hours,
from reply-request
initiator open-request where priority > 3
terminator close-contact
key-by key-request
Situation example #2:

The customer sent request with higher priority for the same contact
situatation preferred-customer-is-mistreated sequence
from open-request as first-request
   open-request as second-request
where
customer.customer-status = "preferred" and
second-request.priority > first-request.priority
key-by key-contact
Situation example #3:

A request was reassigned at least 3 times
situations too-many-assignments
atleast 3
from assign-request
key-by key-request
attributes request, max (detection-time)
Amit User Interface
Amit GUI

- Multiple workspaces
- Step by step wizards that support the creation and editing of meta-data definitions
  - event
  - situation
  - lifespan
  - key
  - database
- View an XML version of the definition
Amit at Run time
<table>
<thead>
<tr>
<th>Trade_Start 0 Monday</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm_comp_p 5 111</td>
</tr>
<tr>
<td>disk_p 2 222</td>
</tr>
<tr>
<td>sun_comp_p 1 234</td>
</tr>
<tr>
<td>sun_comp_p 2 444</td>
</tr>
<tr>
<td>disk_p 4 222</td>
</tr>
<tr>
<td>disk_p 9 222</td>
</tr>
<tr>
<td>application_report_p 222.0 disk_p 9.0 INFLUENCED</td>
</tr>
<tr>
<td>sun_comp_p 3 333</td>
</tr>
<tr>
<td>sun_comp_p 5 333</td>
</tr>
<tr>
<td>computer_network_p 3 675</td>
</tr>
<tr>
<td>many_memory_dependent_p ibm_comp_p 111.0 sun_comp_p 333.0 computer_network_p 3 675</td>
</tr>
<tr>
<td>sun_comp_p 3 333</td>
</tr>
<tr>
<td>computer_network_p 3 675</td>
</tr>
<tr>
<td>computer_network_p 5 998</td>
</tr>
<tr>
<td>sun_comp_p 6 234</td>
</tr>
<tr>
<td>application_report_p 234.0 sun_comp_p 6.0 SECONDARY_INFLUENCED</td>
</tr>
<tr>
<td>sun_comp_p 5 234</td>
</tr>
<tr>
<td>memory_p 9 555</td>
</tr>
<tr>
<td>application_report_p 555.0 memory_p 9.0 INFLUENCED</td>
</tr>
</tbody>
</table>

| Trade_End 24 Monday |

| application_report_p 222.0 disk_p 9.0 INFLUENCED |
| sun_comp_p 222.0 sun_comp_p 6.0 SECONDARY_INFLUENCED |
| sun_comp_p 5.0 |
| memory_p 9.0 |
| application_report_p 555.0 memory_p 9.0 INFLUENCED |
Amit at runtime - Example
Amit - Active Middleware Technology

Advanced
Event Model
Example: Problem Resolution

- Monitoring hardware components and software applications problems
- There are dependencies between components
- Receives:
  - Various problem / symptoms events
- Detects:
  - SLA (service level agreement) violation
  - Predicted problems (impact analysis)
  - Root cause of a problem.
The Event Model

- Every event belongs to unique event class
- There are connections between the event classes that enable to classify event to additional classes

- Connection types:
  - generalization
    - strict generalization - subset hierarchy relationship among two event classes
  - conditional generalization - generalization that is contingent upon additional conditions
Event Model (cont.)

- **cross section**
  - relationship between two classes and a conditional expression.
  - creates additional virtual class (has no events that are directly classified to it)
Event Model (cont.)

- Dependency modeling
  model dependencies between event classes

  gives support for:
  - problem resolution
  - impact analysis
  - root-cause analysis

- The dependency connection may include:
  - condition
  - certainty
Event Model (cont.)

- **Influence on**
  - In the problem resolution context:
    A problem represented by one event class, can be propagated, and cause another problem that is represented by other event class.

- **depend on**
  - In the problem resolution context:
    The cause for a problem event, might be other problem event.

```
Memory Failure

influence-on ("severity>3", cer=0.8)

depend-on (cer=0.5)

Computer Failure
```
Dependency Example

- **disk farm**
  - InfluenceOn/dependOn relationship with **application** (cer 0.3, svr > 8)

- **application**
  - InfluenceOn/dependOn relationship with **computer network** (cer 0.6, svr > 5)

- **computer network**
  - InfluenceOn/dependOn relationship with **SUN computer** (cer 0.2)
  - InfluenceOn/dependOn relationship with **memory** (cer 0.8)

- **memory**
  - InfluenceOn/dependOn relationship with **IBM computer** (cer 0.8)

- **IBM computer**
  - InfluenceOn/dependOn relationship with **disk** (cer 0.4, svr > 3)

- **disk**
  - InfluenceOn/dependOn relationship with **disk farm**

- **SUN computer**
  - InfluenceOn/dependOn relationship with **memory** (cer 0.8)

**Certainty and Severity Conditions**

- **certainty = 0.6**
  - condition: severity > 5

**Relationships**

- InfluenceOn/dependOn
- Generalization
Composite situations

- Situation detection is considered as an event signal, i.e. it is reported back to the system as if an event instance has occurred.
- Like event, situation is defined using attributes.
- Each attribute has name, type and expression that specifies how to compute the attribute's value.
Database Primitive

- **Database connection** - enables database queries as part of situation definition.
  - used when not all the necessary information is supplied by incoming events
  - database queries can be written in any condition (example: the ‘where’ condition of a situation)
- The database primitive defines the database name and the database connection protocol.
Situation features

- **Detection mode**: decision when the situation is detected.
  ➔ Possibilities:
  - *immediate*: when the conditions are satisfied.
  - *delayed*: the calculation is done at the end of the life span.
  - *deferred*: the calculation is done when the conditions are satisfied, but is reported at the end of lifespan.

- **Repeat mode**: if a situation was detected, should it be detected again (under the same life span with the same key value).
  ➔ Possibilities: *once*, *always*.

- **Condition**: condition over the situation operands.
Operands Quantifiers

- There can be multiple occurrences of events.
- The quantifier answers the question which instance(s) should we select.
- Possibilities:
  - **first**: the first instance that satisfies the conditions
  - **last**: the last instance that satisfies the conditions
  - **each**: each instance that satisfies the conditions
Advanced Capabilities
Temporal Issues

- **Problems:**
  - The detection time may not be the actual event occurrence time.
  - Different sources may not have synchronized clocks.
  - The order of events detection may not reflect the real order of event occurrence.

- **Solution Directions**
  - Using techniques from the temporal database area.
  - Using techniques of uncertainty handling.
Uncertainty issues

- Events with certainty value.
- Situation detection with certainty values.
- Flexible meaning and propagation
Recovery and Fault Tolerance

- **Problems:**
  - recovery of the situation manager
  - recovery of a source
  - recovery of a destination
  - communication time-out
Active Technologies and Location-Based Services

The main goal: Develop applications in the area of active services in wireless environment. This proof of concept will demonstrate utilization of Amit (Active Middleware Technology) extended with location-based context.

- Support wireless sources of events and knowledge to the system.
- Support wireless destinations of broadcasting situation as: conclusions, alerts, required actions.
- Support real-time decisions based on tracing and correlating multi events from various sources.
- Support real-time spatio-temporal situations / conclusions / alerts.
Spatio-Temporal

- Events that relate to combination of time and space.
  - Support mobile commerce applications
  - Support for "location-based" situations.
    - awareness: "person X is heading towards you"
    - location-temporal alerts: "traffic jam in your route"
    - location related assignments: cab drivers, delivery vehicles