Complex Event Processing with Aspect Oriented Programming

Technion Reference COM1125

BRIEF
During the operation of a software system (or multiple systems in a distributed setup), many events of interest occur. Complex Event Processing (CEP) is an existing event processing concept that deals with the task of processing such events with the goal of identifying meaningful event sequences denoted as complex events. Applying CEP on top of existing software systems is challenging and the many existing implementation approaches for CEP are far from optimal. In our technology, the implementation is based entirely on Aspects (AOP), while clearly separating the identification of events from their treatment, and facilitating a reusable repository of such Aspects.

BACKGROUND
CEP is a technology to process events and discover complex patterns among multiple streams of event data, with the goal of identifying the meaningful event patterns within those streams, and thus deriving meaningful information. For example, if one person draws cash from two ATMs in New York and London within an hour, this is an odd event to be examined. Existing CEP implementations require the integration of external software components (e.g., event processing engines) into the existing software system, which reduces their prevalence. In addition, the set of events that can be observed directly in the system is not always under control, which may limit the expressiveness of the complex events defined by it.

METHOD
Our technology provides an aspect-oriented (AOP) implementation for complex event processing. Given a base system, an organization of specially designed event and response aspects is designed as a separated sub-system that is then integrated with the original code using aspect language compilation. This approach facilitates the identification of events from the base system, processing of the events in order to identify complex event patterns, and reacting to the identified events. Event aspects are used to identify both basic and complex events, as well as to signal their occurrence. Response aspects are used to define appropriate actions when an appropriate event is identified. Definition of complex events is facilitated by a framework called HighspectJ, and reuse is aided by an event repository containing event building blocks contributed by different parties.

ADVANTAGES
- No need for sophisticated integration - simple implementation based on AOP languages
- Flexible and straightforward event identification
- Implementations may be reusable for different systems

APPLICATIONS
- Managing distributed IT systems (e.g., banking systems),
- Managing and supporting the software development process within the development environment
- Automatic usability evaluation of user interfaces.
- Business process management and automation (process monitoring, BAM, reporting exceptions, operational intelligence)
- Finance (algorithmic trading, fraud detection, risk management)
- Network and application monitoring (intrusion detection, SLA monitoring)

Contact Ehud Shavit, es@dp.technion.ac.il
<table>
<thead>
<tr>
<th>Technological Keywords</th>
<th>Complex Event Processing, CEP, aspect-oriented programming, AOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market Keywords</td>
<td>banking, risk management, reporting exceptions</td>
</tr>
<tr>
<td>Main category</td>
<td>Software and Algorithms</td>
</tr>
</tbody>
</table>