Implementation of Web-based Event-driven Activity Execution in CapBasED-AMS

Patrick Hung
Hong Kong University of Science and Technology

Kamalakar Karlapalem
Hong Kong University of Science and Technology

Follow this and additional works at: http://aisel.aisnet.org/amcis1998

Recommended Citation
http://aisel.aisnet.org/amcis1998/340
Implementation of Web-based Event-driven Activity Execution in CapBasED-AMS

Patrick C. K. Hung
Cyberspace Center
Kamalakar Karlapalem
Database Laboratory
Department of Computer Science
The Hong Kong University of Science and Technology

Abstract

The CapBasED-AMS (Capability-based and Event-driven Activity Management System) is a workflow system developed in [5] deals with the management and execution of activities. A Problem Solving Agent (PSA) is a human, or a hardware system, or a software system having an ability to execute activities. An activity consists of multiple inter-dependent tasks that need to be coordinated, scheduled and executed by a set of PSAs. The activity execution is based on the occurrence of events. That is, a PSA after completion of a task (atomic activity) generates events, which are captured by the activity management system, for initiating the execution of the next task. In this paper, we describe three-tier system architecture to implement Web-based event-driven activity execution of CapBasED-AMS.

Introduction and Motivation

Many application domains like, office automation, planning, medical diagnosis, manufacturing systems, etc., need interaction between humans and systems, and among humans for conducting day-to-day work. Moreover, workflow systems are becoming very popular and are being used to support such activities in large organizations. One of the major problems with workflow system is that they often use heterogeneous and distributed hardware and software systems to execute a given activity. With the popularity of Internet technologies, this is a need for web-based workflow system, which is platform independent. In this paper, we describe three-tier system architecture to implement Web-based event-driven activity execution in CapBasED-AMS.

Architecture of Secure CapBasED-AMS

Activity management consists of decomposition of activities into tasks, coordination and data sharing among multiple PSAs executing the activity, and monitoring, scheduling and controlling the execution of multiple tasks of an activity. A software system that facilitates the specification, maintenance, and execution of activities is known as an Activity Management System (AMS).

Capability-based activity specification and decomposition: Each PSA has its competence defined by set of capabilities it has to execute tasks. Each activity requires a certain competence from the PSAs specified as a set of needs for executing all of its tasks. Each activity is decomposed into a set of tasks by using the property that each task must be executed by exactly one PSA, further, each task is matched to a PSA by selecting a PSA that has the capabilities to meet the needs of the task. In AMS, we use tokens to model the capability/need of a PSA/task respectively. This is the new idea which has not been applied in related works, because most of the related works model this concept by applying role based or hard-coded solutions. We present a new philosophy of applying capability-based approach as the bridge mechanism for matching a PSA to a task. A token is more flexible to handle and organize in the Capability Database, since a token does not belong to any fixed architecture. The match making of each task to an appropriate PSA can be done automatically by simple SQL queries on the Capability Database. The specification of activities, sub-activities, tasks and PSAs are all user-driven. We define this aspect of activity management as capability-based activity specification and decomposition [4].

Event-driven activity execution: An activity consists of multiple inter-dependent tasks that need to be coordinated, scheduled and executed. The dependencies between tasks can be expressed by means of a uniform framework of events, as i) system event - models data/control dependencies among tasks of an activity within the AMS, e.g., receipt of data from a PSA or completion of a task denotes the occurrence of a system event, ii) external event - models external dependency which affects the execution of an activity from a stimuli outside the AMS, and iii) temporal event - models temporal dependency. For each task, there are two types of events associated with it, namely In-events and Out-events. In-events are the situations in which the task is initiated to execute whereas Out-events are the effects after the execution of the task. Both In-events and Out-events can be one or more of the event categories described above. Note that both In-events and Out-events can correspond to receipt and delivery of documents processed by a task. Based on the events raised, an Event-Condition-Action Rule [2] (ECA Rule) is
triggered that lead to initiating the execution of the relevant tasks. When events are raised, conditions in the corresponding ECA rules are evaluated, and corresponding actions are triggered if the conditions are satisfied. Note that the ECA rules are also used to specify the security authorization requirements imposed on a task, and are used to authenticate the execution of a task by a PSA. Therefore, the activity execution is based on the occurrence of the events. We define this approach to execute activities as event-driven activity execution [7].

Figure 1 shows the CapBasED-AMS architecture, which consists of two parts: (1) capability-based activity specification and decomposition, and (2) event-driven activity execution. The capability-based activity specification and decomposition component consists of an Activity Specification Language (ASL) processor, an Activity Decomposer (AD) and a Match Maker (MM). ASL facilitates the description and specification of the activities. The Document Management System facilitates the specification and management of document involved in ASL. AD decomposes an activity into tasks. MM identifies the PSAs to execute the tasks. AD and MM generate a specification of coordination plan using a Task Specification Language (TSL) as addressed in [4,5]. The event-driven activity execution component deals with propagating information, coordinating PSAs, scheduling and monitoring execution of tasks of an activity. In AMS, the activity execution is supported by four modules, namely the Document Manager (DM), Event Manager (EM), Activity Generator (AG) and the Activity Coordinator (AC). The EM manages the event transmission to the PSAs. The DM manages the document flows and enforces the security specification on document during activity execution. The AG acts as a pre-compiler. It takes a specification in TSL (the ActivityID.tsl) in Figure 1 to generate the activity graph, ECA rules and defines runtime data schema, which are stored in Active Database [2]. The AC acts as an execution manager for activity execution and monitoring security enforcement by ECA rules when executing tasks. We refer to [4,7] for further details on activity specification, decomposition and execution.

A Prototype Implementation of the CapBasED-AMS

The first prototype was implemented in the C programming language and Embedded SQL of Sybase DBMS with the XT Motif application running on X windows environment and TCP/IP protocol communication network under the UNIX operating system using Sun SPARCstation 20. This prototype was demonstrated at 1996 ACM SIGMOD conference [3] in Canada and the other MS Window version prototype was also implemented by Visual Basic and Access last year. We are currently implementing the CapBasED-AMS [6,8] on Web in three tier system architecture by using Java Applet, Apache Web Server, Sybase DBMS and Weblogic T3 Server as shown in figure 2.

As Java computing is widely used for Internet applications with security features, we develop the system by using Java Applets. Three-tier system architecture consists of the client application, the application server and the database server as shown in figure 3. The client application as PSA interface refers to Java Applet embedded in HTML. It can be loaded and started up by trusted client by IP address with a Java enabled Web browser such as Internet Explorer or Netscape Communicator. The multithreaded application server (CapBasED-AMS) running on the web server is listening to Java Database Connectivity (JDBC) requests from the clients. This middle-ware can be closely controlled by the administrator, e.g. user authentication by Discretionary Access Control (DAC) or Mandatory Access Control (MAC) [8], secure data transmission by Secure Socket Layer (SSL) with private and public key, and it can also perform load balancing in multi-activity execution. The Database Server (Sybase DBMS) supports the knowledge and information storage and retrieval.
Conclusion and Future Works

The first version of Web-based event-driven activity execution of CapBasED-AMS prototype will be launched in April 1998. After that, we will develop the Web-based Capability-based activity specification and decomposition of CapBasED-AMS. We are also currently working on the security features of the Secure CapBasED-AMS [6,8].

References
References available upon request from author (cshck@cs.ust.hk).