

## **IncidentWeb: An Incident Management System for Enterprise Systems** **Bina Ramamurthy**

**CSE4/587 Information Structures**

**Due Date: 2/24/2003 by mid-night.**

Events play an important role in many enterprise systems. A typical enterprise system can be modeled as combination of a core services it provides, the applications that enable the services, the data and the resources used by the applications. The modern enterprise systems differ from legacy systems in that the services are mostly event-driven and transaction-oriented. An incident can be defined as an application-domain-dependent event. Incidents can be coarsely classified into expected and unexpected incidents. A robust enterprise system should have policies, mechanisms procedures and processes to handle these incidents in an efficient and timely manner. The overall effectiveness and success of a system may depend on how effectively these incidents are handled. The effectiveness can be measured in terms of system availability, cost savings for the organization, lost revenue minimized, lives saved in a possible accident, infrastructure saved in a disaster and so on. For example, the system can also be common factory floor and the enterprise system supporting its operation. In such a system any malfunction or failure of any of the processes need to be reported as incidents so that appropriate remedial process can be applied to keep up the production at the expected levels. IncidentWeb will play this role. The effectiveness of the IncidentWeb will be measured in terms of availability of the system and the amount of lost revenue saved. Transaction-oriented systems such as a stock exchange and automatic teller machine could also benefit from an IncidentWeb. We will consider a transaction-oriented system in this project. IncidentWeb will coexist along with the main transaction-oriented application.

The project consists of two major components: (i) the main transaction processing system (let us call it TPS) and (ii) the IncidentWeb. For the main transaction-oriented system you may use the CheapBooks.com implemented using Enterprise Java Beans (EJB) in the J2EE tutorial. But DO NOT implement the EJB version but implement a simple client-server version using Java RMI or Java Servlets. The IncidentWeb component (i) records the normal transactions in a system (ii) creates a transaction server summary (availability, patial failure, etc.) report at the end of every 24-hour period (iii) detects and reports any incidents in the system, (iv) implements a workflow controller to follow up on the incidents with email to people and messaging to applications to take appropriate actions. There will be a data server layer that stores all the enterprise data associated with the system. The integration will result in the processes forming an automated workflow that will eventually store all the relevant information in a database for further analysis (data warehouse). Clients who transact (interact) with the system will access the main application TPS through the Internet (web clients). The users of the IncidentWeb are (i) system administrators who monitor the operation of the system so that remedial actions can be authorized to handle any incidents (ii) automatically triggered applications, (iii) corporate decision makers who may read the reports and analyze the reports to planning purposes. User will not have direct access to the IncidentWeb as it is a background application integrated to the TPS.

Your task is to design and implement the TPS, its workflow, define a set of possible incidents for the main problem and design and implement its IncidentWeb. Test and illustrate the working of the integrated system (TPS+ IncidentWeb) using sample scenarios.

