Pergamon

Computers ind. Engng Vol. 35, Nos 1–2, pp. 255–258, 1998

Printed in Great Britain
PII: S0360-8352(98)00043-6 0360-8352/98 $19.00 + 0.00

VIRTUAL MANUFACTURING INFORMATION SYSTEM USING JAVA™ AND JDBC™

Sameer S. Pradhan and Wilfred V. Huang
Alfred University, Alfred, NY 14802

ABSTRACT

The programming language -- Java, from Sun Microsystems, promises to transform the internet into a medium of secure on-line and real-time business transactions, that are independent of operating systems. Also, with the development of Java Database Connectivity (JDBC) API, it is possible to create secure, three-tier, virtual database systems. This paper deals with the implementation of a three-tier database system, using Java and JDBC, and creation of an innovative, intelligent "interactive ordering system" on the Internet. It goes a step further to prove the flexibility of the system, by using different database engines at different nodes.

KEYWORDS

Information Systems; Database; JAVA; JDBC; Networks.

THE INTERNET/ INTRANET COMPUTING REVOLUTION

Internet signifies the vast collection of inter-connected networks that all use the TCP/ IP protocol and that evolved from the ARPANET of the late 60's and early 70's. In the past decade, the Internet has experienced exponential growth and was having 15,000,000 hosts across the world as of 1996 (Zakon, 1997). It was not till the past couple of years that the corporate world started to probe into the power of this huge network. Many connected their computers to this ever expanding network to achieve a global presence.

In short, the operating architecture involves a client application, which is a Java applet that communicates with a server application via Internet using some standard Internet protocol. The Java applet accepts user input, and depending on its type, either takes a decision locally, or sends the required data to the server for processing. The processed data is then returned to the user screen in the required format. All the user needs to have is a connection to the Internet over a modem or a T1 line (depending on whether he is a consumer, sending or receiving small packets of data or an enterprise user querying a distributed network of databases) and a Java enabled browser, as all the application libraries are downloaded at runtime. This browser is playing the role of a graphic user interface (GUI) and the execution environment, therefore operating system does not matter much. A two-phase security structure can be used to prevent hackers from intruding the proprietary data or applications. The first phase can be designed to ensure server security and prohibit unauthorized access to data or

applications, by having a hierarchy of configuration profiles which control the system configuration, the customer site configuration as well as the individual user. The second phase can secure the data transferred over the Internet, using Secure Sockets Level protocol (SSL™), browser based encryption technologies which are designed to prohibit unlawful tapping of data.

VIRTUAL MANUFACTURING INFORMATION SYSTEM

Many manufacturers now realize that a more effective structure that facilitates enterprise-wide integration of information and systems is a customer-oriented supply chain that demands real-time access to information for proactive decision support. The manufacturing industry is on the verge of a major paradigm shift, which will take it away from mass manufacturing, way beyond lean manufacturing, into agile manufacturing, typically comprising a nation-wide factory network in which a large base of diversified suppliers will be linked electronically. This will promote concurrent engineering between corporate partners, characterized by the rapid production of high quality customized goods, by a knowledgeable, empowered work-force, and an information infrastructure that links computers, marketers, engineers and robotics in a unified electronic web (Song and Nagi, 1996).

With the advent of Java, it became quite clear that the dependency on individual operating system would become very marginal. The problem of varying data-structures and even database systems was consequently solved by the introduction of the Java Database Connectivity (JDBC) API. Thus a long-term solution to this problem is the use of Java, JDBC and a properly selected middleware. These three together can eliminate the bottlenecks due to heterogeneity in the system and drastically reduce the implementation time.

OVERVIEW OF THE SYSTEM

In order to demonstrate this, a prototype information system has been designed. The aim of this system is to manipulate different databases in the organization in order to supply the required information, on-line, real time, and to anyone having access to the Internet. As indicated earlier, it caters a broad range of users, including the customers. At the core of this system lies the Internet programming language Java, its JDBC API and dbANYWHERE™ Server. In our example, the customer can get some specific information, for example, the delivery lead-time of a product or products and its price, depending on a selected priority level. A similar logic can be applied to modify this system so that it can incorporate several other functions. The prototype information system that we will be discussing is a modified order-entry interface, which has much more features built in it than a normal static interface, seen most prominently on the web-pages. Depending on the input, this system has the capability of modifying an existing production schedule or creating a new schedule.

The system is basically an applet that is written in the Java programming language. This applet accepts orders from customers (who ever has an access to the Internet) on-line. In today's Internet ordering scenario, a customer enters the order details on the respective web-site and a back-end CGI application stores it in some remote database. It can be said that the process is quite static, but with the advent of Java, the same application can be made more comprehensive. In the current scenario, the customer is left with an approximate duration within which he should expect the product. This delivery lead-time is a one time calculation and the actual might vary considerably, if the demand pattern is not the one predicted while calculating the same, which is quite likely the case. In some cases this uncertainty can be ignored without any far-reaching consequences. However in the case of a corporate ordering scenario, a lot of schedules

and commitments down the line depend on the accuracy of the intermediate procurement lead times. If these delivery lead-times were calculated dynamically and online, taking into consideration the inventory levels, production schedules and the order priority, then it would make the planning more realistic. This process can help a great deal in providing the goods to the right customers at the right time, instead of goods getting sooner to the customer who does not need them so urgently and vice-versa. Infact, the exact times of delivery might help the customer decide whether or not to place the order. This can help improve the image of an organization in the minds of customers. One loophole in this system is that all the customers can indicate a fake high priority, and invalidate the basic foundation of the system. The way to deal with this is adopting a variable pricing strategy, which is beneficial to both the buyer as well as the supplier. The ramifications of this system can be as far-reaching as the initialization of a production schedule at the respective manufacturing facility.

After the user enters all the required information, the program will check the availability of goods at the nearest (depending on the distance) Branch, Warehouse and Factory, in that order. However the starting point will be determined by the priority assigned by the user. For a high priority case, the program will start from the nearest branch, for a normal priority, it will start from the nearest warehouse and finally for the low priority, it will start from the factory. Depending on the place where the required quantity of the product is found, the lead time will be calculated and conveyed to the user along with the respective price, which in turn is queried from the database at the respective location. If the user confirms the order, the program marks the product as sold to that user and at the same time creates a dispatch plan for the product.

SYSTEM CONFIGURATION

The databases used in our system are Sybase SQL Server and Microsoft Access, each of which is a representative of the two major types of RDBMS, viz., a industrial strength server and a medium sized database engine, supporting the ODBC interface. The system can incorporate other RDBMSs like ORACLE Server, INFORMIX, Paradox, dBASE etc. by making minute changes indicated later in this section. The middleware chosen is dbANYWHERE™, from Symantec Inc. Following are the different tiers in the system.

THE CLIENT TIER (FIRST)

Client hardware can consist of any workstation running a Java enabled browser. Such browsers include Netscape Navigator and Microsoft Internet Explorer, making virtually any personal computer or workstation suitable as a dbANYWHERE™ client. The client workstation requires (or obtains during runtime) the following other software:

1. Client Java Application, typically downloaded from server
2. dbANYWHERE™ JDBC classes, downloaded dynamically from a web server where dbANYWHERE™ resides.

THE dbANYWHERE™ TIER (SECOND)

The dbANYWHERE™ middleware runs on a Windows 95 computer and manages the transactions between the client and the back-end database server, i.e. Sybase SQL Server, via. a type-3 Sybase JDBC driver and MS Access via. ODBC driver. A client connection is maintained to each database to which the dbANYWHERE™ server is attached. So, typically the dbANYWHERE™ Server requires the following software:

1. dbANYWHERE™ middleware, provided by Symantec
2. dbANYWHERE™ native database driver, provided by Symantec

3. Database configuration file, provided by the database vendor, configured by database administrators
4. Optional ODBC driver, provided by third parties to support dozens of different database servers
5. Client database driver, provided by the database vendor

THE DATABASE SERVER TIER (THIRD)

Any type of hardware from Windows based microsystems to UNIX based mainframes can be used as the back-end database server. These systems communicate with the dbANYWHERE™ server via the vendor provided database driver installed on the dbANYWHERE™ server. A wide variety of database software can be used on the database server.

CONCLUSION

The system that was discussed in the earlier sections is by no means a complete one. As said earlier, this system was specifically chosen to demonstrate the unique capabilities of Java and its JDBC API. As the system is beneficial to the external customer, so it is to the internal customer. However, this aspect was not discussed in detail earlier owing to the fact that much of what is possible to the internal customer is possible even with the use of existing technology, although Java makes it easier and more reliable. There are recent trends in using Java for performing discrete-event simulations (Buss and Stork, 1996, Nair and Zhang, 1996). Data is queried from distributed databases, thus simplifying the data-collection process. Also, its inherent object-oriented-ness facilitates generation of simulation models. A major addition to the order entry applet can be to use it as a tool for performing real-time simulations, i.e., the production plan can be prepared based on some quick simulation using the snapshot of data existing in the system. This can further increase the intelligence of the system as a whole.

REFERENCE

Buss A. and Stork K. (1996) "Discrete event simulation on the world wided web using Java," Proceedings of the Winter Simulation, Conference, December 1996.

Nair R., Miller J. and Zhang Z.(1996) "Java-based query driven simulation environment Proceedings of the Winter Simulation conference, December 1996.

Song and Nagi R. (1996) "A Virtual Information System for Agile Manufacturing Enterprises," Conference on Agile and Intelligent Manufacturing Systems, Troy, NY, June 1996. Available at http://www.acsu.buffalo.edu/~nagi/pubs/song.ps

Zakon R. (1997) Hobbes' Internet Timeline, The MITRE Corporation, July 1997. Available at http://info.isoc.org/guest/zakon/Internet/History/HIT.html