

DESIGNING A WEB-BASED SYSTEM FOR MANAGING A SUPPLY NETWORK

Thomas Franz, University of Cologne, tfranz@clemsun.edu

Steve Davis, Clemson University, davis@clemsun.edu

ABSTRACT

This paper discusses design of a system to manage a supply network and its implementation in Web-based software. This design supports multiple supply networks, each of which may consist of a complex network of customer-supplier relationships. This project simulated production and shipping of products, allowing testing software in realistic situations and supporting research into strategies for supply network management.

Keywords: Supply chain management, information system, Web-based system, simulation

INTRODUCTION

According to Lee, Padmanabhan & Whang, firms that do not operate in a managed supply network may encounter severe problems responding to demand, leading to the “bullwhip effect” (5). Some researchers have investigated the benefits of information sharing-based supply chain partnerships (12), but very little research and development has been done to support inter-firm operations beyond linear supply chains. Yet there is a need to support a network consisting of multiple supply chains, where one firm may participate in more than one supply chain.

Although interest in forming supply networks has been increasing, relatively little guidance has been published on how to develop software system for managing a supply network. Published architectures provide helpful high level guidance but lack details such as database design (9,11). This paper suggests the requirements for such a system, proposes a design, and briefly describes an implementation. These ideas may be helpful to developers of management systems.

GENERAL SYSTEM REQUIREMENTS

The system should provide all the information needed by members of the supply network including a possible central coordinator. This information includes contract information and information from each participating company including its products and its inventory. Information sharing among participating firms is a fundamental capability of a system supporting supply chain management. For example, it could help to minimize the bullwhip effect, where the variability of demand seems to increase for firms farther up the supply chain. The system should have a simulation capability to allow testing for proper function before it is deployed. (Besides helping to test software intended for real-world application, the simulation feature also would support research into supply net management algorithms.)

SYSTEM DESIGN

System Architecture

We chose a centralized approach mainly for the following reasons. 1) It facilitates focusing on the functional requirements pertaining to supply network management rather than distributed implementation issues, 2) It reduces the information technology burden on participating firms, thus facilitating participation by small firms, and 3) It makes data synchronization easier, the most challenging problem according to the VICS Internet Standards Committee. A centralized system could be operated by one of the leading organizations in a supply network on behalf of the entire network or by an external company as a service to one or more supply networks.

We divided the system into tiers so that parts of the system can be easily exchanged or extended. For example changing the underlying database should be possible without requiring any significant changes to the application. Thus this project adopted a three-tier software architecture: user interface (client), business rules (process server and data persistence), and data management (data server). Compared to the two-tier approach it provides increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user (7).

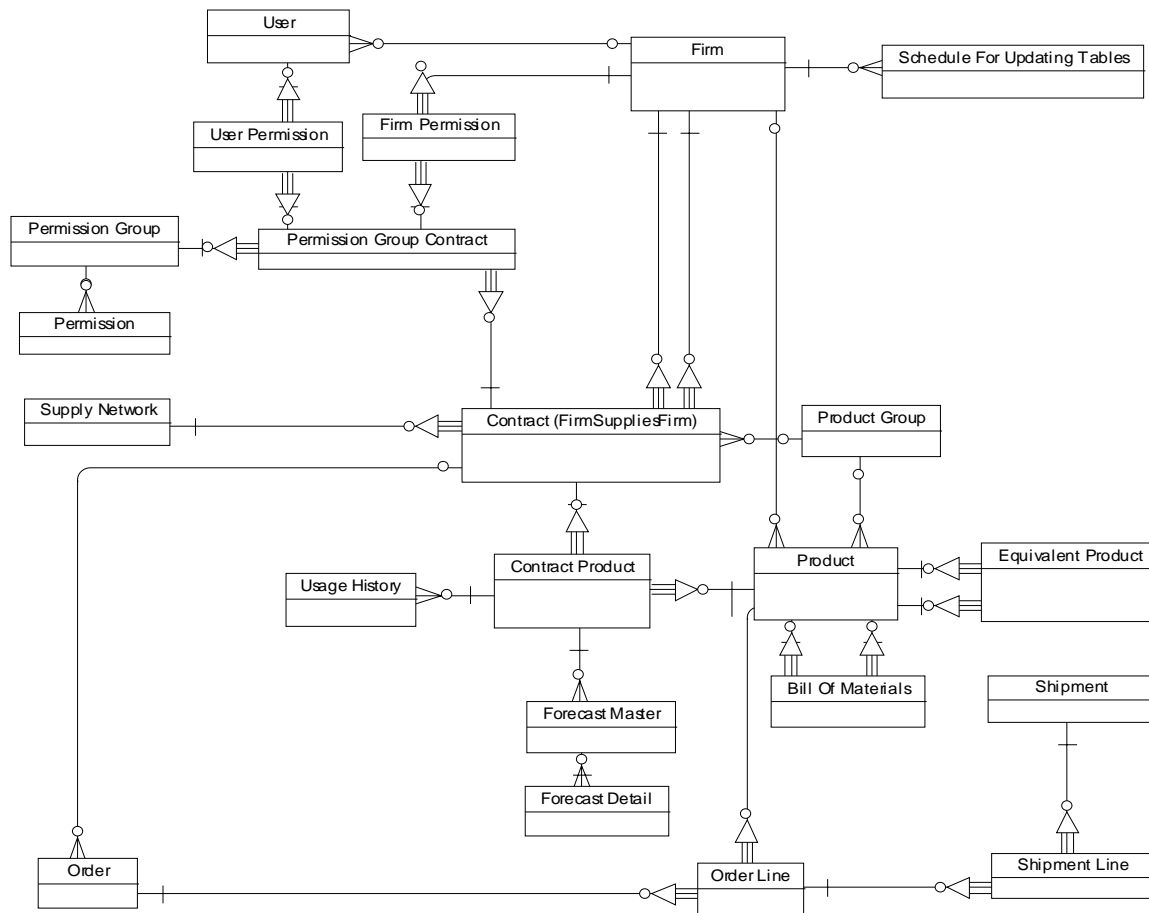


Figure 1. Database model

Database Design

The database was designed to support any number of supply networks. Within each network each firm may have many customers and many suppliers. The structure of the network is represented by contracts among firms. The simplified database model in Figure 1 shows the most important entities/relationships but eliminates attributes. The triangular symbols, either solid or bird's foot, at the end of a relationship line signify "many", for example, a firm is associated with many products. A solid triangle indicates the existence of such an entity is dependent on the existence of the entity at the other end of the relationship. For example, an instance of the *order line* entity is dependent on the existence of related *order* and *product* entity instances. For brevity we have not shown the physical model with database tables that correspond to entities in the logical model.

The table *schedule for updating tables* stores the policy for updating information for all the tables. Updating can be done on a "pull" or "push" basis and with various schemes for timing, depending on the wishes of supply chain members.

Security

A supply network system requires very detailed security, based on access rights for table rows as well as table columns. To achieve database system independence we developed security features in software rather than relying on features provided by a database management system. Data integrity in the security table requires all primary key values of tables demanding security to be globally unique. Therefore we set up a central global generator of unique primary key values that is used by all tables. Generating these key values in a sequence provides a history of record insertions that could be valuable for auditing. For example, if firm x has the key 5, contract y the key 6 and product z the key 7, we know firm x was introduced into the system followed by contract y, and then product z.

An Access table represents a relationship among the following, each of which is indicated by a foreign key value in the record: a permission group for a contract, a row, and a column of the table to be secured. For example a record of this table might represent that the Administrator Group for Contract X includes Write permission for the CompanyName column in the row having key = 6 of the Company table. This table supports a fine-grained security scheme but requires an operation on the Access table each time a record is added or deleted in a secured table. These operations could reduce performance but may be well worth the security benefits.

Contracts, Products, Orders and Shipments

In the database each contract between two adjacent firms is associated with a particular supply network and thus the database can represent multiple supply networks.

Detailed information about a product is needed to help firms decide what is the appropriate product for a contract or for an order, and to facilitate management of inventories throughout the network. Each product is associated with a firm (that may buy it or sell it). Because different firms may have different names for the same product there is an equivalence table to resolve

ambiguity. For each product there is a bill of materials that is similar to a list of ingredients for a recipe. Because the bill of materials is global (covers an entire supply network) it helps show the side effects of events. For example it helps determine the potential impact on suppliers at all levels in the network of a forecast change in demand for a finished product at retail. Ordering is represented in the typical way, with an order consisting of many order-lines.

A shipment consists of many ship lines each referencing an order line. Each ship line has a quantity attribute (so we can infer how much still has to be shipped) and a status attribute that shows, for example, that a ship line has been shipped or has arrived at its destination.

APPLICATION IMPLEMENTATION

Need for Web System with Database Interaction

Certainly an acceptable system must be Web-based and provide dynamic pages that are linked to a database. A number of software development tools are available to build such systems, but there are only two major types, those based on Microsoft Dot Net and those based on Java. We selected Java primarily because it can run on any operating system and processor architecture. Today's e-business applications have to span a variety of platforms and must or connect to different applications among the firms participating in a supply network.

Frameworks for Application, Persistence, and Rendering

For brevity we mention only a few implementation details. In accordance with the chosen architecture we set up the software in three tiers: user interface, business rules, and data management.

This project employed frameworks to reduce software development time and increase reliability, including servlet-based Turbine that encapsulates user interaction, security features and application flow. Turbine consists of five major modules. The Page module generates pages and contains the rest of the modules (Action, Layout, Screen and Navigation). The Action module performs tasks such as form validation and storing the form's information into a database, the others are self-explanatory. Also Turbine provides service modules for various tasks including scheduling, database access, request handling, and logging.

We employed IBM's Persistence Builder framework to help map objects, and relationships between objects, to information stored in relational databases and to control the commit or rollback of updates. Persistence Builder supports nested transactions that contribute to convenient user interfaces. For example, the user needs to enter information into a number of screens to complete a new product entry. If the user makes a mistake partway through, it is not helpful if the user has to go back to the very beginning to correct the error. With nested transactions the user can just correct the mistake because the system can roll back just a sub-transaction rather than the whole thing.

To simplify rendering we developed a custom framework that is capable of generic rendering and conversion of data to html based on given parameters. Using rendering classes afforded several advantages. Development of screens is faster and easier because it is easier to define parameters for the renderer than to develop code in HTML. Maintenance is easier. It facilitates consistency in the look and feel of screens that is important for a convenient user interface. One may embed authorization checks in just one place in the renderer so that every screen does not have to implement security checks.

Application Packages and Classes

Classes addressing persistence include SupplyNetScreen that extends the Turbine screen class to handle persistence and rendering and provide returnTable. It is filled with data by the renderPage method. To define parts of a web page SupplyNetScreen has methods including setHeader and setNavigation. The method isAuthenticated checks user authority.

The ScheduledJobs package simulates certain company actions, thereby allowing testing without connection to any company information systems. For example, the update of inventories for all firms in the system is automatically done by a job that is controlled by the cycle time for a product. Whenever the cycle time has run out the job increases the manufacturing firm's inventory level for the product. Likewise jobs also simulate shipments of products.

AuthorizationException provides for Screens to throw an exception when a user tries to call any action or screen that he/she has no permission for. BusinessLogicException throws an exception upon any violation of the business logic, for example if a user tries to order a higher quantity than the maximum quantity specified in the contract. The Sequence class provides primary key numbers for new inserted data, ensuring it does not deliver any key value twice. The Authorization class checks user permissions for data access and retrieves data associated with the current user, for example the firm associated with the user or groups the user is member of.

User Interface

On most screens the top navigation consists of (permission-based) links to FirmView, InventoryView, ProductView, ContractView for suppliers, ContractView for customers, the UserDetails page and the Administration page. One may add/edit if authorized. Each of these pages provides links to details.

With appropriate permission InventoryView allows viewing information at other firms in the supply network. The ProductView page shows links for starting contract negotiations for products not manufactured by the user's firm. The ContractView page displays brief information and the ContractDetail page shows details with links for changing a contract or accepting a proposed contract. A group of Orders pages provides traditional capabilities for creating/editing orders and order lines pertaining to a selected contract. The AdministrationView page displays all users and groups, each with a link for editing/deleting.

RELATED WORK

Recently researchers have begun to address extended ERP systems but most provide limited support for managing activities in a supply chain. Several researchers have proposed models of systems for supply chain management but none of them are a sufficient basis for developing a database or software (3). The Collaborative Planning, Forecasting and Replenishment Committee, an industry group made up of several retailers and manufacturers, developed models that provide valuable insight but do not provide sufficient details about the entities involved in the functioning of the supply chain (10,11). The Demand Activated Manufacturing Architecture Model for Supply Chain Collaboration provides a high level model that identifies many of the necessary functions for managing a supply network but does not provide details about the data structures needed to support the specified functions (1).

Humphreys, et al. described a framework for inter-organizational information systems for supply chain management but did not provide a detailed design (2). Sadeh et al. designed an agent-based system with blackboard architecture (6). Their report emphasized features for coordination of schedules among participating firms but did not address security or details of information sharing. Klen et al. described the Distributed Business Process Management System (4). Like the aforementioned systems it developed a coordinator focused on detecting problems in the collective plans of supply chain members and reacting to any problems discovered. This system pays little attention to proactive methods for scheduling production throughout the supply chain. Shin and Leem developed a reference system for inter-enterprise electronic commerce (8).

CONCLUSION

This work was partially supported by The National Science Foundation (grant DMI-0075608). This project produced an architecture that guided development of the system reported here. Other valuable management-support features are supported in the database but not yet implemented, including the awareness of equivalent products among firms, provision for special product attributes, and accounting for different units of measure for different products. Besides helping to demonstrate and test this system, the simulation feature could also support research into supply network management algorithms that may seek to optimize the benefit for the overall supply network rather than optimizing profits for just one firm.

REFERENCES

1. Chapman, L.D., Lathon, R., Petersen, M. (2000). DAMA Model for Collaboration, <http://www.dama.tc2.com/Pub/Model.html>.
2. Humphreys, P.K, Lai, M.K., Sculli, D. (2001). An Inter-Organizational Information System for Supply Chain Management, *Int. J. Production Economics*, (70), 245-255.
3. Kanji, G.P., Wong, A. (1999). Business Excellence Model for Supply Chain Management, *Total Quality Management*, (10:8), 1147.
4. Klen, A.A.P., Rabelo, R.J., Ferreira, A.C., and Spinosa, L.M. (2001). Managing Distributed Business Processes In The Virtual Enterprise, *Journal of Intelligent Manufacturing*, (12:2), 185-197.

5. Lee, H.L, Padmanabhan V. and Whang, S. (1997). The Bullwhip Effect in Supply Chains, MIT Sloan Management Review, (38:3), 93-102.
6. Sadeh, N.M., Hildum, D.W., Kjenstad, D., and Tseng, A. (2001). MASCOT: An Agent-Based Architecture for Dynamic Supply Chain Creation and Coordination in the Internet Economy,” Production Planning and Control, (12:3), 212-223.
7. Sadoski, D., Comella-Dorda, S. (2000). Three Tier Software Architectures, (http://www.sei.cmu.edu/str/descriptions/threetier_body.html) Carnegie Mellon Software Engineering Institute.
8. Shin, K. and Leem, C.S. (2002). A Reference System for Internet Based Inter-Enterprise Electronic Commerce, Journal of Systems and Software, (60:3), 195-209.
9. Siau, K. and Tian, Y. (2004). Supply Chains Integration: Architecture and Enabling Technologies, Journal of Computer Information Systems, (XLIV:3), 67-72
10. The Collaborative Planning, Forecasting and Replenishment Committee. (1999). Conceptual Model, www.cpfr.org/ConceptualModel.html.
11. White, A. (1999). The Value Equation: Value Chain Management, Collaboration and the Internet, <http://www.cpfr.org/WhitePapers.html>.
12. Yu, Z, Yan, H.Cheng, TCE. (2002). Modeling The Benefits of Information Sharing-Based Partnerships In a Two-Level Supply Chain, Journal of the Operational Research Society, (53:4), 2002, 436-447.