JB 98-10

Entitled:

Web-It-Man: Web-based Integrated Tool for Manufacturing Environment

Authors:

R.K. Karne, J.S. Baras, D.S. Nau, M.O. Ball,
E. Lin, V.S. Trichur, S. Dandekar,
S. Poluri and J.T. Williams

Submitted to

American Society of Mechanical Engineers
(ASME)

Web-It-Man: Web-based Integrated Tool for Manufacturing Environment

Ramesh K. Karne[1], John S. Baras[2], Dana S. Nau[2], Michael O. Ball[2], Edward Lin[2], Vinai S. Trichur[2], Swati Dandekar[1], Sridhar Poluri[1], and James T. Williams[3]

Abstract

Exponential growth in Internet applications and a need for a global access for future manufacturing demands web-based tools that operate seamlessly in heterogeneous environments. We present a Web-based Integrated Tool for Manufacturing[4] that assists designers with a variety of CAD/CAM tools through a unified user interface. Web enabled system architecture is proposed for the future development of manufacturing tools. Design issues and research topics pertinent to this architecture are described. A prototype implementation based on this architecture and its current status is outlined. Finally, our research efforts in the development of this tool and some future research areas are identified.

Introduction

Product and process design in a manufacturing environment consists of variety of tools. Integrated product and process design, commonly known as IPPD, has been developed by many authors [2,4,8] and significant improvements have been achieved. Numerous researchers have developed decision support systems [1,13] for evaluating manufacturability of a product and formulating cost and quality models [6,8]. A variety of commercially available concurrent engineering tools [3,11] and CAD/CAM systems are in the market place. Most of the available tools for manufacturing a product focus on its own data and its manipulation and its storage. In general, these tools do not share any data and they operate as standalone boxes. A product or process designer has to deal with multitude of tools and data models and is challenged to integrate a variety of environments and data management tools.

When data management alone is considered, the above manufacturing environment is analogous to a heterogeneous database management system [5] that has been researched since the first inception of a database management system. Heterogeneity [9] can be classified in many ways and manufacturing environment fits well with heterogeneous systems. Many database management systems that store manufacturing data are heterogeneous and distributed in nature. Heterogeneity exists with respect to vendors, data models, platforms, hardware, software, user interfaces, and so on. In a manufacturing environment tools such as CAD/CAM are heterogeneous and do not communicate well with each other. We have observed this problem in many manufacturing enterprises and classified it as a database problem. We have been working with a major manufacturing company to study this issue and approach a solution with an integrated view and usage of a database as an integrator.

In addition to the database usage as an integrator for data [12], the explosion of the Internet and its role as a global information highway has provided us with a great motivation to integrate manufacturing environment

---

[1] Towson University, Towson, MD 21252
[2] Institute for Systems Research, University of Maryland, College Park, MD 20742
[3] Northrop Grumman, Electronics Sensors & Systems Division, P.O.BOX 746, Baltimore, MD 21203
[4] This project is funded by Northrop Grumman and Maryland Industrial Partnership (MIPS)

through the Web. Ideally, each and every tool must operate in the Internet environment and must be able to be accessed all over the world. This is not yet possible as most of manufacturing tools are proprietary in nature and they are based on their own data models. Most of tools in manufacturing interface with files. There is no standard application programming interface (API) available to interface with other applications. Considering this limitation in the real-world environment, we have developed a Web-based Integration Tool for Manufacturing (Web-It-Man) that works with a variety of tools in a heterogeneous environment. The Web-It-Man is developed in the Java programming language environment and suitable for integrating various manufacturing tools. In this paper, we will present a specific manufacturing environment that exists with our manufacturing enterprise client and illustrate the operation of the proposed tool.

The rest of this paper is organized as follows. The manufacturing environment section describes a real-world manufacturing environment and facilities to be considered for an integrated environment. The Web-It-Man section describes our proposed architecture. The design issues section identifies major issues and research areas related to the manufacturing environment, available tools, platforms, and data models. The prototype environment shows the building blocks or component tools used to integrate in the tool. The implementation section describes our implementation of the prototype in the Java programming environment. Finally, the conclusions section summarizes overall contributions of this paper.

Manufacturing Environment

Today's manufacturing environment is an emerging global enterprise as illustrated in Figure 1. Most of the United State's corporations are involved in diversifying their manufacturing in almost any place in the world where they can produce a product cheaper, better, and at a faster pace. In order to accomplish this goal, manufacturing enterprises need to design, develop, build, and rebuild their products based on the Internet serving as their information infrastructure. Users, customers, designers, developers, and sales people across the world must have a unified and quick access to their products and services. For instance, to design an automobile, designers may be located any where in the world and they must be able to share and manage the design data as they need. An end product resulting from one or more manufacturing plants may require hundreds or even thousands of parts and alternate parts. It will also require many processes and alternate processes that operate on parts and alternate parts. In addition, suppliers, marketing and sales personnel, and customers are located through out the globe.

As it is illustrated in the figure, there are many tools available to produce a product at a design level and also at a manufacturing level. These design automation tools use information including: parts, processes, vendors, design data, design knowledge, graphics, text, documents, undocumented knowledge from people, price information, inventory information, and so on to design or build a sub-component or a complete assembly. These tools work in collaboration with real-world manufacturing in the plant and help designers at every step of its manufacturing cycle. The information used in the tools must be as accurate as possible and it must represent models that simulate close to the real-world environment. The information may be stored in files, documents, databases, and other forms of knowledge and may be scattered through out the world. In addition, the information may be represented in many ways and may be heterogeneous with respect to manufacturers, data models, symbolic representations, formats, sizes, and so on.

Figure 1 helps us to capture the complexity of emerging manufacturing application domain and a desperate need for an architecture that copes with such diversified and heterogeneous environments. Using the Internet as an information infrastructure, we propose an architecture that addresses the above application domain and allows extensibility for future manufacturing applications.

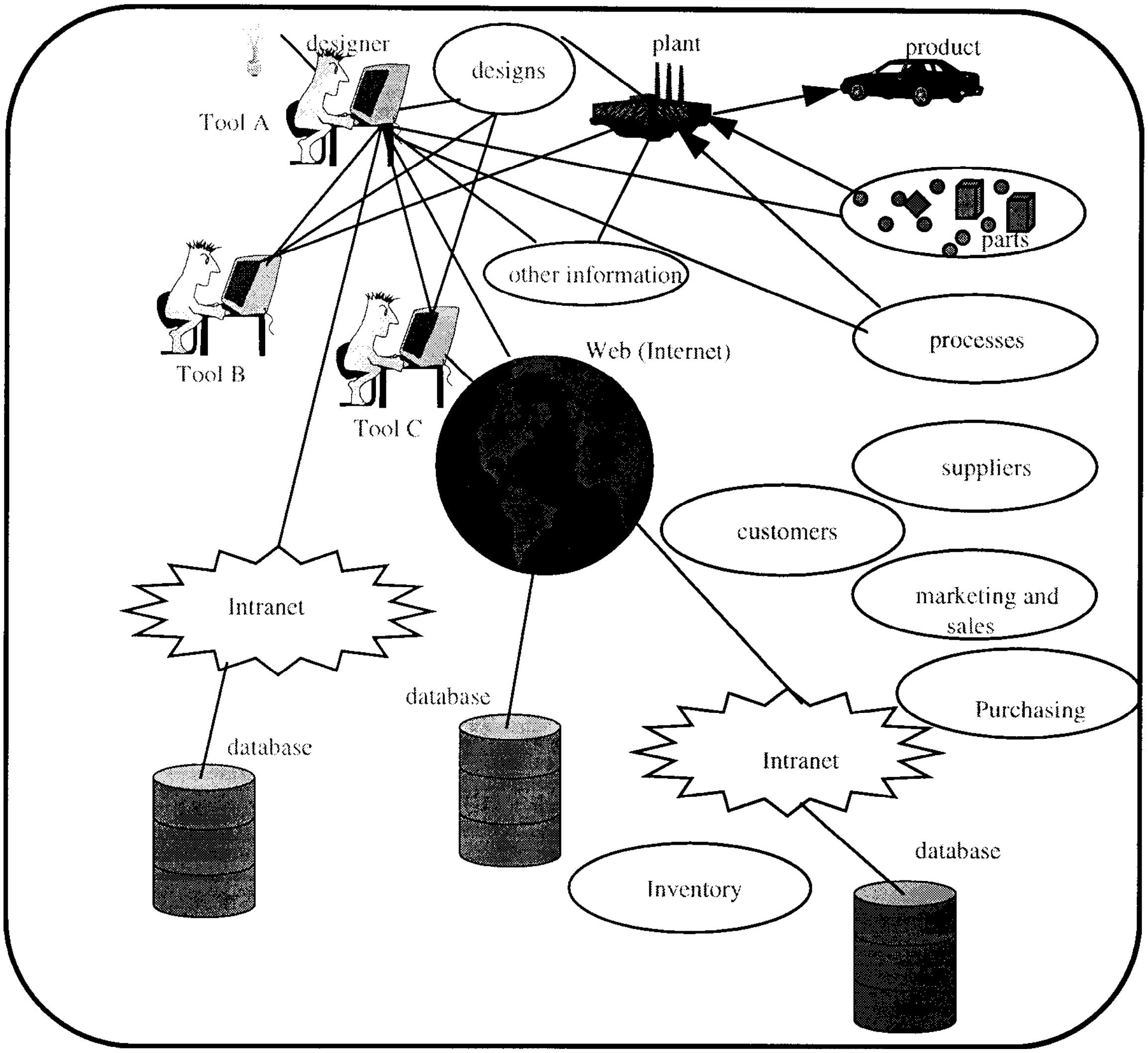

Figure 1: Emerging Manufacturing Environment

Web-It-Man Architecture

The Web-based Integrated Manufacturing (Web-It-Man) is based on a client/server approach that is natural to the Web or the Internet. Figure 2 illustrates the proposed Web-It-Man architecture. A client shown in the block is a client request and server shown in the box is a server responding to the client's request. A user can initiate a client request and a server responds to this request. Client and server programs can be located any where on the Internet and their exact location is transparent to the architecture. A user can be a customer, a designer, a developer, a tester, a vendor, a manufacturer, or a sales representative. A users' interface can be tailored based on their need and usage. For example, a designer of a product requires access to design information, where as a sales person may only need to browse the appropriate characteristics of a marketable product. The client graphical user interface (CGUI) allows the client (user) to work with the product and its related information. Appropriate access permissions and security issues must be addressed to provide such a unified interface and an access to the product information. Each client's requests are processed by a server located on the Web and it is implemented by using Java's remote method invocation (RMI).

In addition to the client/server approach, the object data models are used to model manufacturing objects and they can be stored in an object-oriented database management system (OODBMS). However, Java based "persistence" support of objects can also be used to store objects in files instead of an OODBMS. Other manufacturing data sources that are currently located in relational database management systems (RDBMS) can be accessed through JDBC. Overall, using the Java programming environment, the entire manufacturing environment can be architected as shown in Figure 2 for future manufacturing applications and tools.

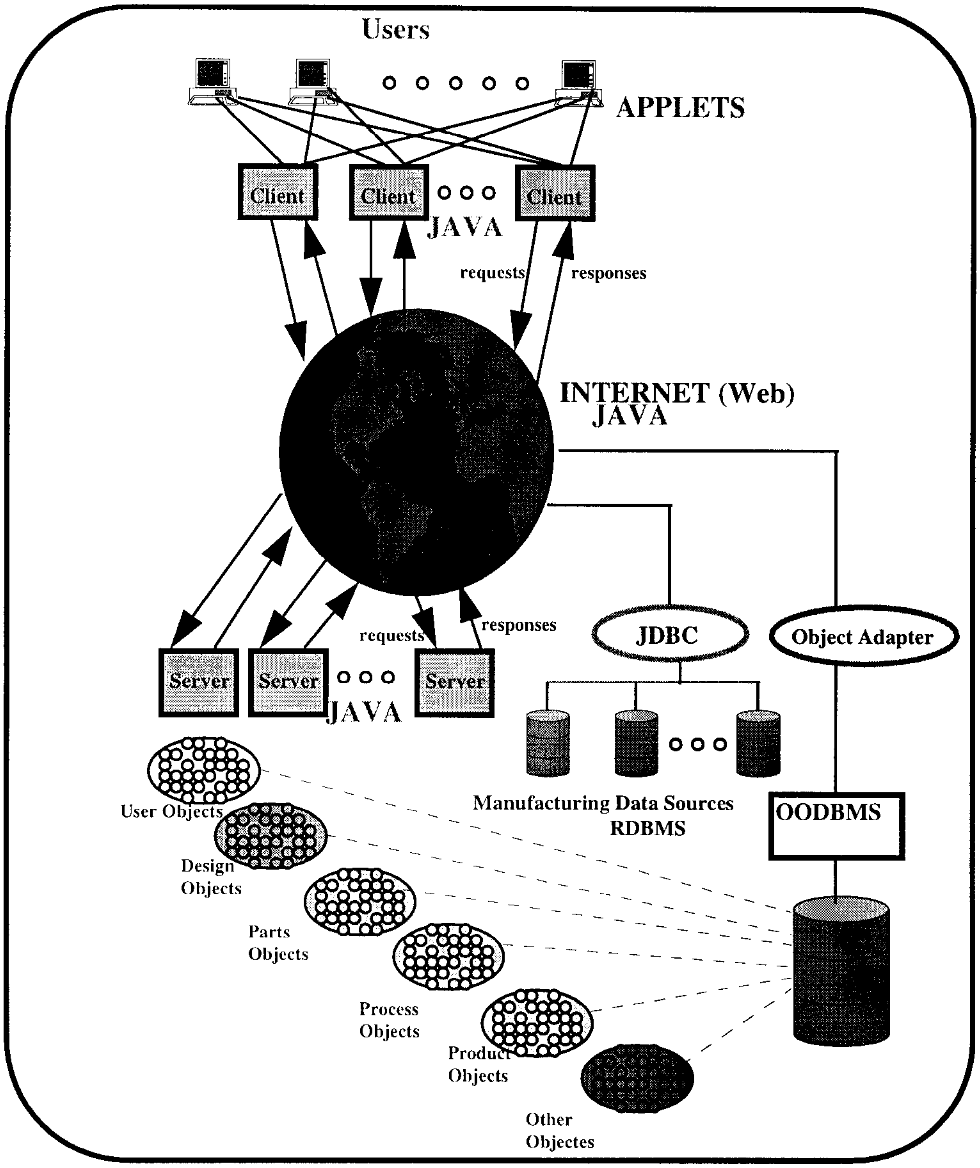

Figure 2: Web-It-Man Architecture

Design Issues

The Web-It-Man architecture proposed in this paper raises many design issues with respect to its feasibility and implementation. Some of the design issues are described here to identify and spur further research in this area.

Data Models: The information required in manufacturing is no different from data that is stored in a conventional database system. Most of the conventional database systems have been based on relational data models and the new database systems are following object-oriented and object-relational data models. Object-oriented modeling has been used in manufacturing applications by many researchers [7] and it is also being used in multidatabase systems [14]. Considering a variety of data modeling approaches for database systems, it is evident that object-oriented models are most appropriate to model manufacturing entities. However, there is no universal model that embraces all manufacturers and software developers. With the emerging popularity of the Java programming language, object-oriented data models are clearly a right choice for modeling manufacturing applications. In addition to data models, there is also a need for other models such as functional data models [8] to represent design logic, AND-OR models to provide searching capabilities [8,10], and logic-based models to serve reasoning and decision making applications.

Application Programming Interface: Most tools designed for manufacturing operate as standalone applications and do not provide any API for other tools to access data. For example, EEsof [3] is an electrical CAD tool and Microstation [11] is a mechanical CAD tool. Only file interface is possible between these two tools. There is no way to invoke one tool from another to build an integrated environment. A designer has to understand these interfaces in detail and manually exercise these tools to achieve a complete design for a given product. If a standard API is available, then it is much easier to integrate these tools in a single design automation environment.

Data Sources: Most of the tools in a manufacturing environment work with files and they have their own formats and attributes. That means, in an enterprise world, there is a redundant information in these files and there is no convenient way to integrate these data sources. Multidatabase techniques [5] such as global schema or a programming language approach such as open database connectivity (ODBC) can be used to interface with data sources. However, these approaches require large programming efforts or a need for a middleware tools.

User Interfaces: Each tool in a manufacturing application has its own user interface. The user interfaces vary in implementation and appeals. Visual Basic, Motif/X Windows, Java/Applets, Delphi, Developer 2000, Power Builder, and Microsoft Windows are some of the popular environments used to implement a graphical user interface. They all have different look and feel user interface and programming environment. Thus, a user of a manufacturing tool is faced with a variety of interface choices and conflicting environments.

Data Locality: In an enterprise environment, usually data is scattered all over the company in a variety of data sources. For example, purchasing information may be stored in a purchasing database in a division X, part information may be stored in a design database in division Y, and process information may be available through process engineers and some may be stored in a manufacturing database. As the information is scattered and spread across many data sources, identifying the location of data in a large enterprise is a complex problem. A given manufacturing tool may need to locate data and use it as needed to reduce cost, reduce inventory, and reuse existing components. Web-based data search engines are needed to locate and access appropriate data on the network.

Data Integrity: As the data is spread across many sites and data sources, the data integrity becomes very crucial to perform accurate analysis and design of products. A given entity in an enterprise must have consistent information to guarantee the data integrity.

Data Access and Performance: For mission critical applications, data access techniques and speed are very important for a proper operation. In today's multimedia world data, voice, and video require different data access mechanisms and latency requirements. Current Internet data access and its performance is not acceptable to many web-based design applications and tools. New technologies such as fast search engines, Asynchronous Transfer Mode (ATM) switches, and Fiber Optics are necessary to address communication performance issues.

Data Distribution: As the data in global environment is located remotely and needs to be accessed through a variety of networks and nodes, the data transmissions are prone to errors, delays, and catastrophic failures. Thus, there is a need to distribute data among many nodes for redundancy and also provide multiple paths to access the same data. This raises many research issues related to data migration and data coherency.

Data Security: As data is accessible to a variety of users including intruders, data authentication, privacy, and security must be insured to guarantee a proper usage of data. New and innovative security measures are needed in addition to simple encryption of data during transmission and storage.

Data Autonomy: There are number of owners of a data in this global environment. For example, a given part may be manufactured by different vendors. Each vendor must be able to update data asynchronously and the changes must be transparent to other users. The changes made by different vendors must be immediately reflected in the global data to be accessed by any user. Thus, local autonomy must be guaranteed while maintaining the data transparency to users.

Prototype Environment

The Web-It-Man architecture as proposed above is a complex system and requires tremendous resources to validate the entire environment. We have scaled this problem to a manageable prototype environment and focused the implementation based on a client global user interface (CGUI). This CGUI allows any user to invoke a particular tool and work with the tool to design a specific task. Figure 3 shows our prototype environment to achieve this objective.

CGUI is designed to work with many users and runs on either a workstation or on a PC platform. The portability is achieved by using the Java programming environment for implementation. EEsof is an electrical CAD tool that can be invoked through the CGUI. Actual system files and working directories for this tool can be located anywhere on the network and these files can be managed by our tool. For example, EEsof (ES) has .dsn, .msk, .igs, .pl, .hl files. These files have their own format and attributes. When a designer invokes this CAD tool, first, a working directory is created by checking out files from a persistent directory. When a design is complete, the files associated with the design are checked into the persistent directory. Thus, at any given time a file is owned by one user for usage and updates. However, many users can obtain the file for read only purposes. As the designers may be located any where on the Web, our tool provides a unified interface to the designer and easy access through the Web.

Microstation (MS) is a mechanical CAD package. In our prototype, it is used to design a substrate for a microwave module. It's invocation and file structure is similar to the EEsof tool, however, file formats are different and unique to this tool. For example, an "interface.dat" file contains dielectric and "via" attributes of a design file. EEsof 's output file has to be translated to an "interface.dat" file and "rho.dat (masking file)" to input to this tool. As it can be observed, there is a need to translate one file format to another file or files based on an input requirements of a tool.

Cost Advantage (CA) from Cognition Corporation will provide a means to evaluate a design. A designer may use this tool to evaluate designs from HP EEsof, or alternate designs and generate a "cost sheet." The output is a file with its unique format. Once again, this tool can be invoked through CGUI.

HTN Planner (HP) developed at Institute for Systems Research (ISR) takes, as input, information about the parts list, layout, and mechanical features of a microwave module. The process planner then does process planning and produces a description of electronic and mechanical processes required to manufacture a microwave module. This tool works with input and output files and has its own formats. This tool can be invoked through CGUI.

Trade-off analysis (TA) tool is a multi-objective optimization tool which can generate optimal design choices for microwave modules based on a given design and a set of alternate parts and processes. This tool is built at ISR and works with either file inputs or a database interface. The alternate parts and processes required to formulate a linear programming problem may be stored in a database or a file.

The database gateway (DG) is a database access tool which access data from enterprise database systems (ED) and also from databases that are located on the Web. The Local Database (LD) is a database system that stores the global schema required to model Web-It-Man tool. The data entry to LD tool (DE) is a front-end interface to LD to enter data using a user-friendly interface. The data integrator (DI) is the middleware that glues together all component building blocks or data sources.

This prototype environment shows different component building blocks ES, MS, HP, CA, TA, DG, and DE. These blocks need to be integrated through a common user interface CGUI. In the multidatabase terminology, these building blocks are analogous to component database systems (CDBs) [9] and they can be integrated together by using a global schema approach [5]. A global schema captures all the local schemas that are being used in each component database or a building block. In our prototypes, the building blocks as defined above have a file interface which contain entities that need to be stored in a global database. However, the TA tool can be directly interfaced with a global database, thus tightly coupled with a database. All other building blocks are loosely coupled with the database.

We have integrated the prototype environment using a global schema approach and developed an Entity Relationship (ER) model to represent this schema. A partial ER model is shown in Figure 4. The global schema is implemented in LD. Data can be entered to this database by using the DE tool. Any data that is pertinent to the global schema is stored in LD, that means, an appropriate entities from each input or output files will be stored in the database to guarantee the data integrity. As most of our building blocks except TA do not have any API, the user must invoke "update" activity from the CGUI to keep data up-to-date in the database.

Implementation

The prototype architecture is shown in Figure 5. The CGUI and DI constitute major software efforts implemented in the Java programming environment. We have coded all the objects in JDK 1.1.3 which supports persistent objects, remote method invocation (RMI), and Java database connectivity (JDBC). We have also used Visual Cafe Pro to develop our graphical user interface. An object-oriented methodology for software development is used through out the prototype development. Each building block or a component database system is interfaced with input object and an output object so that in case of an interface change in a tool, we only need to replace a corresponding object. The DI block consists of objects that are needed to access the global database and we partitioned these objects such that they are dedicated to serve each building block interface. The DI objects consist of JDBC code and methods that support the interface to building blocks, CGUI, and the Web. Commands from CGUI that do not go through a given building block can directly interface with the objects in DI. The CGUI consists of all the functions that are needed to perform tool invocation, file management, database interface, user functions, and network related operations. As the CGUI performs actions, all the relevant operations that effect the global schema must either access or update the data defined by the global schema in LD. We have used Oracle database as LD and interfaced the database with JDBC. This prototype is currently operational in our laboratory and the CGUI is shown in Figure 6. Figure 7 shows a check-in operation for a file management function and check-out looks similar to this. Other details of the implementation is not shown due to space constraints of this paper.

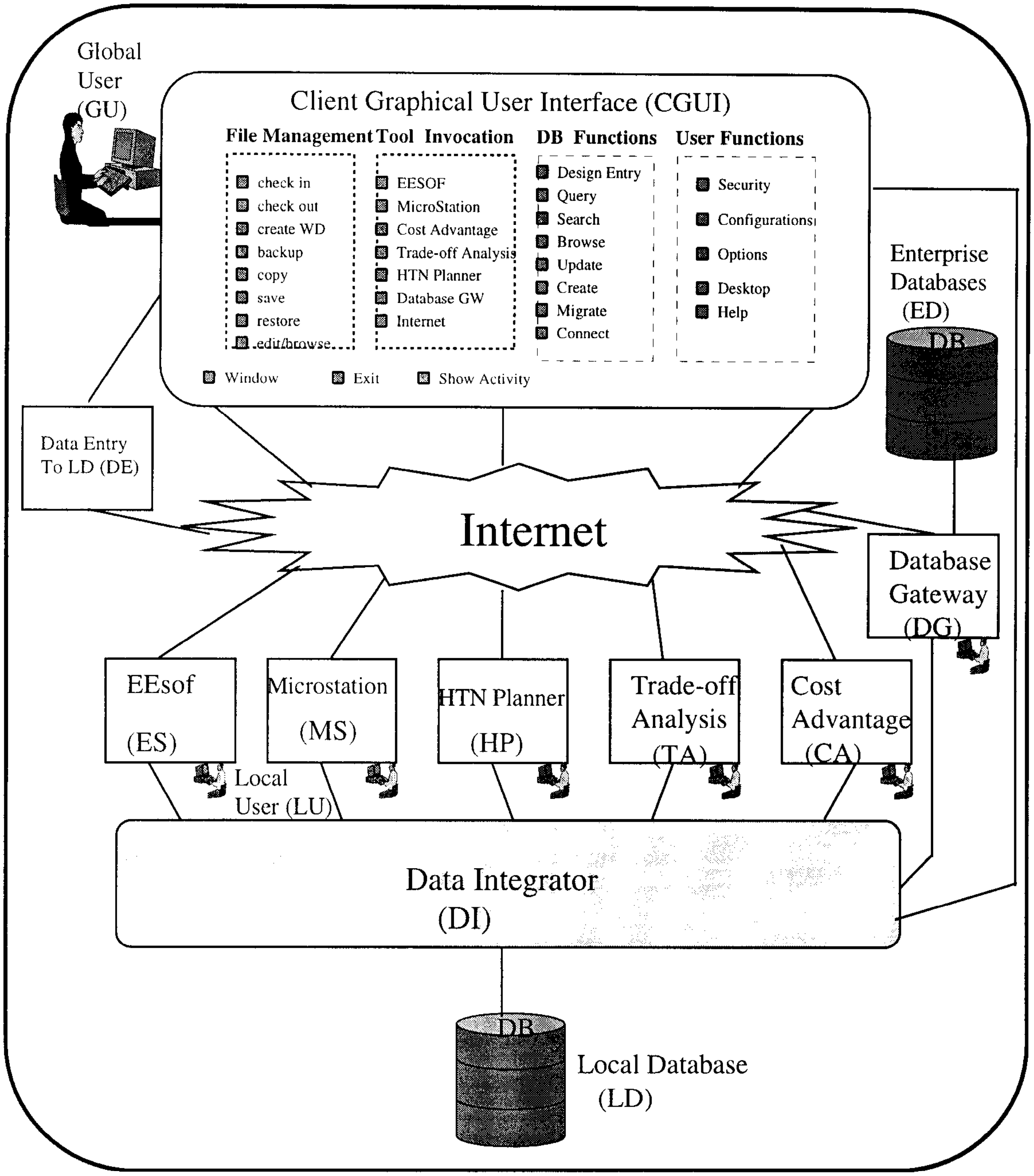

Figure 3: Prototype Environment

Currently, we are working on making the tool robust with enhancements such as user security, error recovery, object migration, and performance improvements. We plan to extend the tool to resemble Web-It_Man architecture by implementing objects using RMI.

Conclusions

We have presented a novel Web-It-Man architecture that can be used for development of any manufacturing application tool. We have identified some design and research issues pertinent to development of such Web-based tools. A real-world prototype architecture and implementation using Java programming environment is described. A heterogeneity problem that exists in manufacturing applications is identified as a multidatabase problem and a well-known technique such as a global schema approach is used to address this problem. Some real-world CAD/CAM tools such as EEsof, Microstation, Cost Advantage, Trade-off Analysis, and HTN Planner are integrated using the Java programming environment. We have identified and stimulated some research issues for building tools on the Web including user security, tool performance, object migration, data access, data locality, and data integrity which require further investigation and prototyping to validate the proposed architecture.

References

[1] Ball, M. O., Baras, J. S., Bashyam, S., Karne, R. K., and Trichur, V. S., On the Selection of Parts and Processes during Design of Printed Circuit Board Assemblies, INRIA/IEEE Conference, October 1995.

[2] Cutkosky, M. R., Engelmore, R. S., Fikes, R. E., Genesereth, M. R., Gruber, T. R., Mark, W. S., Tenenbaum, J. M., and Weber, J. C., PACT: An Experiment in Integrating Concurrent Engineering Systems, Computer, January 1993, pages 28-37.

[3] EEsof Series IV version 4. 1992. EEsof Inc., Westlake Village, California.

[4] Harhalakis, G., Minis, I., and Rathbun, H., Manufacturability Evaluation of Electronic Products Using Group Technology, In Proceedings of the 1993 NSF Design and Manufacturing Systems Conference, pages 1353-1360, Charlotte, NC, 1993.

[5] Hurson, A., R., Bright, M. W., Pakzad, S. H., Multidatabase Systems: An Advanced Solution for Global Information Sharing, IEEE Computer Society Press, 1993.

[6] Heng, C. A. S., and Gay, R. K L., Design for Manufacturability: Cost Analysis of Electronic Printed Circuit Board Assembly, In G.A. Gabriele, editor, Advances in Design Automation- Volume 1, pages 63-67, ASME, 345 East 47th Street, United Engineering Center, New York, NY 10017, 1991.

[7] Jorgensen, K. A., and Madsen, O., Object-oriented Modeling of Active Systems, IFAC Workshop on Intelligent Manufacturing Systems 1994, pages 129-133.

[8] Karne, R. K., Baras, J. S., Ball, M. O., Bashyam, S., Kebede, A., Williams, J., Trichur, V.S., Karir, M., Lai, H., and Dandekar, S., Integrated Product and Process Design Environment Tool forManufacturing T/R Modules, Journal of Intelligent Manufacturing, Volume 9, Number 1, February 1998, p9-15.

[9] Kim, W., Seo, J., Classifying Schematic and Data Heterogeneity in Multidatabase Systems, IEEE Computer, December 1991, p12-p18.

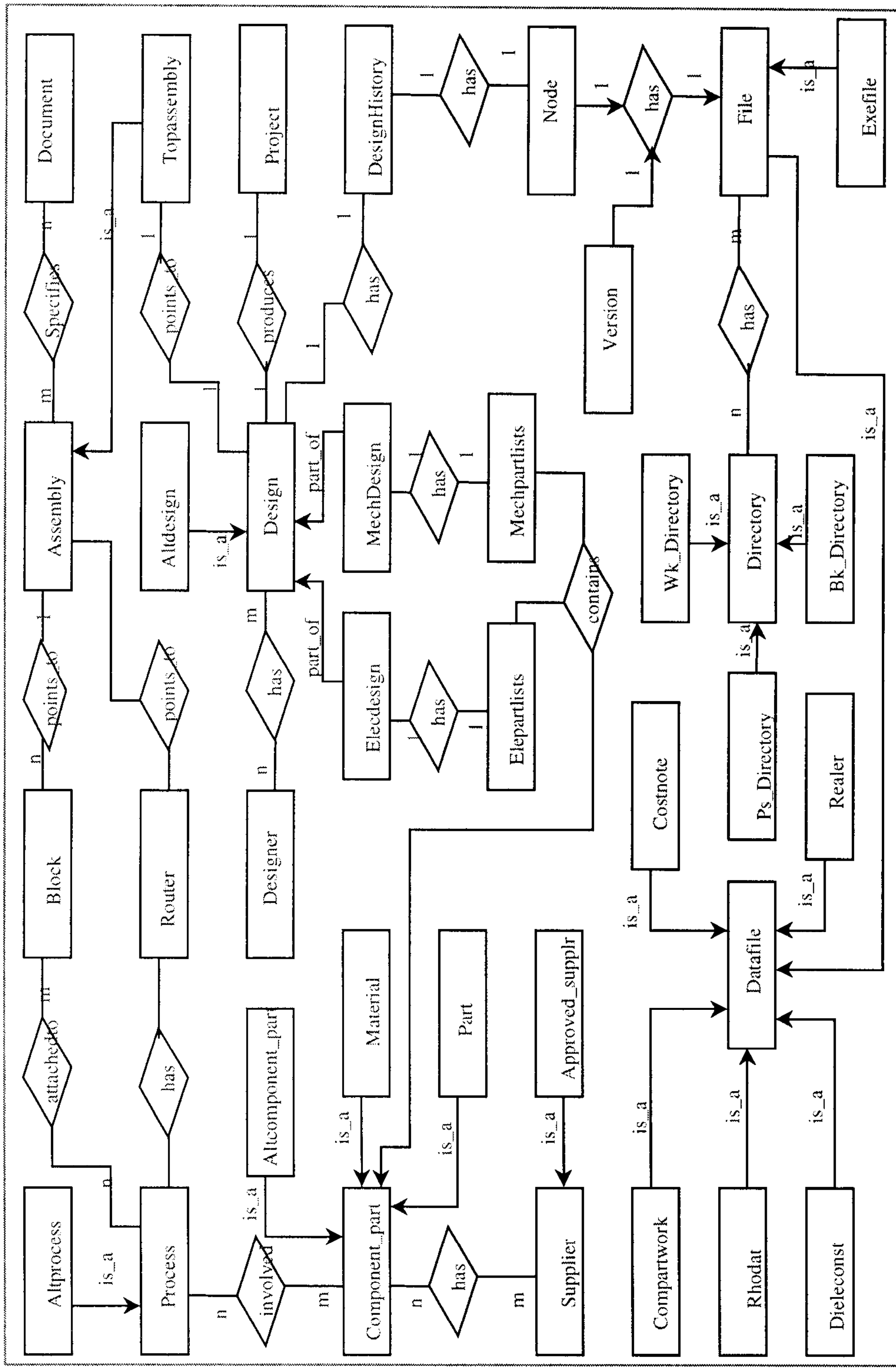

Figure 4: ER Model (Global Schema)

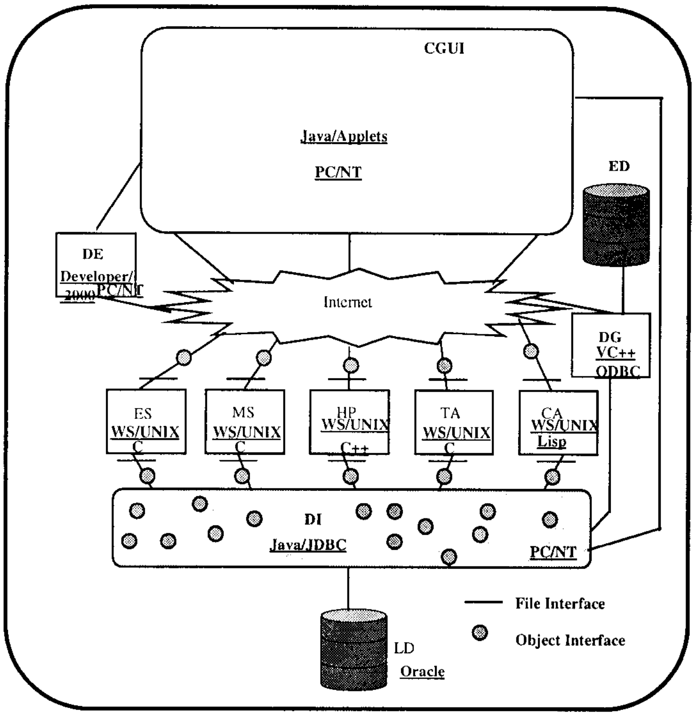

Figure 5: Software Architecture

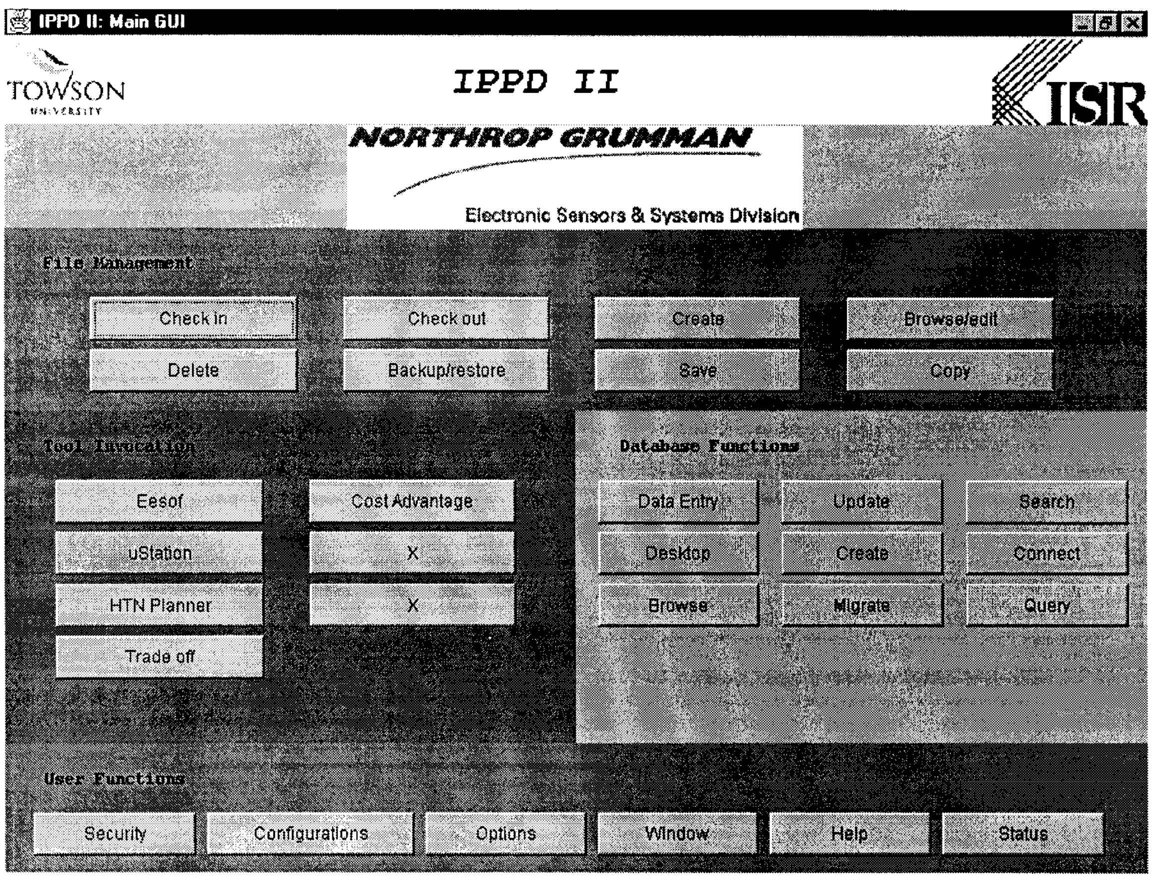

Figure 6: CGUI

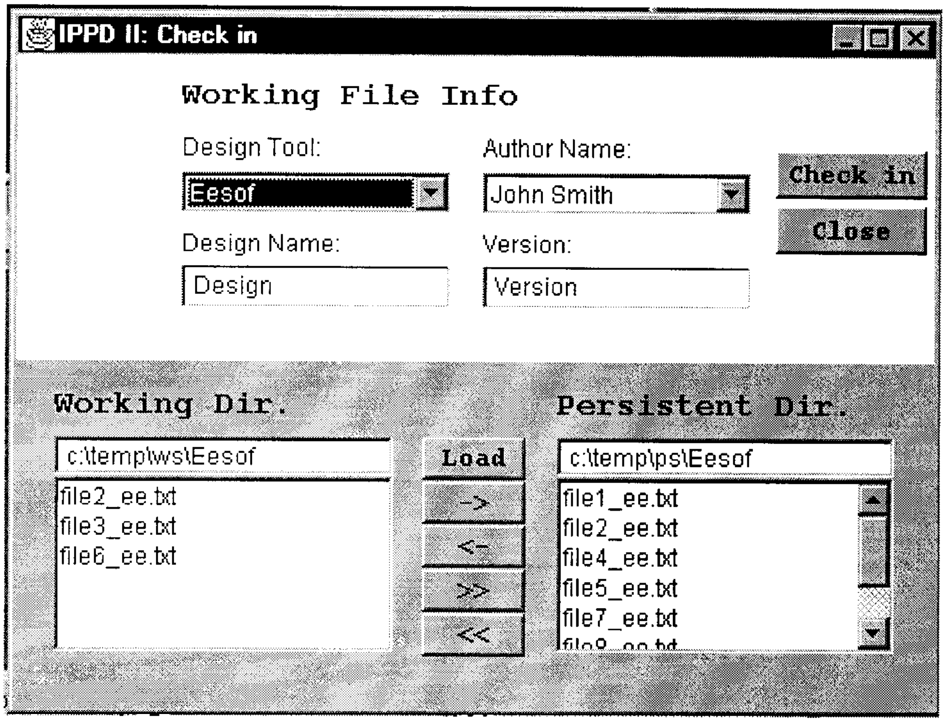

Figure 7: Check-in File Management in CGUI

[10] Kusiak, A., Szczerbicki, E., and Park, K., A novel approach to decomposition of design specifications and search for solutions, International Journal of Production Research, 29(7):1391-1406, 1991.

[11] Microstation version 5. 1995. Bentley Systems Inc., Exton, Pennsylvania.

[12] Nestler, A., System Integration and Technological Database, IFAC Workshop on Intelligent Manufacturing Systems 1994, pages 109-113.

[13] Oh., C. J., and Park, C. S., An Economic Evaluation Model for Product Design Decisions under Concurrent Engineering, The Engineering Economist, 38(4):275-297, Summer 1993.

[14] Pitoura, E., Bukres, O., and Elmagarmid, A., Object Orientation in Multidatabase Systems, ACM Computing Surveys, vol.27, no.2, June 1995, p141-p195.