Leizerowicz, W., Lin, J., and Fox, M. S., (1996), "Collaborative Design Using WWW", Proceedings of WET-ICE '96, CERC, University of West Virginia.

Collaborative Design using WWW

William Leizerowicz (leizero@ie.utoronto.ca)
Department of Industrial Engineering, University of Toronto, M5S 1A1

Taner Bilgic, Jinxin Lin, Mark S. Fox
Department of Industrial Engineering, University of Toronto, M5S 1A1

Abstract

Large projects require the use of concurrent engineering in order to save time and money. In order to achieve this effectively, the KAD group at the University of Toronto have built tools to coordinate the interaction between people and information. This paper describes the KAD project and its use of the WWW for the coordination of collaborative design. Baseline data from the SRMS space arm project was used in order to demonstrate improved collaborative design through information access, information consistency, design reuse, rapid redesign, and information visualization using a WWW interface. Information accessibility and consistency were found to be the central issues for achieving coordination.

1 Introduction to Industry Needs

Today's market pressures require industries to compete at every level in order to reduce costs and development time. Concurrent engineering is a method which helps achieve these goals through parallel design of components. Most large projects are examples of concurrent engineering and consist of design teams which work independently within an interdependent framework. If this framework is not coordinated properly then problems arise such as frequent oversights in component compatibility, time consuming engineering change notifications (ECNs), delay of procurement, refabrication of prototypes, long assembly and test periods, high reject rates and high maintenance costs[1]. These problems arise because designers are unable to clearly see out of their design domain into the interdependent framework, and are unable to access the right information when it is needed. Spar Aerospace realized the difficulties of coordinating large projects, and has chosen to cooperatively develop new tools to avoid these coordination problems. This paper concentrates on how concurrent engineering relates to the challenges of knowledge representation, access, and consistency, and how the WWW helps solve them.

1.1 KAD Objectives

University of Toronto's Enterprise Integration Laboratory is developing a Knowledge Aided Design (KAD) system. KAD is designed to support concurrent engineering design and enhance the degree of awareness, cooperation, and coordination among engineering team members. Some objectives are to:

- Provide a representation (ontology) that stores, integrates and manages design knowledge.
- Ease the access to and acquisition of information and knowledge from the representation.
- Provide a shared environment in which engineers can explore space of alternative designs and communicate their design in a uniform manner to the shared design.
- To reuse past designs and design experiences.
- To manage the systems engineering process for large design projects.

2 Knowledge Management

2.1 Knowledge Access

The SRMS project (Shuttle Remote Manipulator System or, Canadarm), a gift from Canada to the NASA shuttle program, was selected as the design set for KAD system because it brought together many disciplines and required much collaboration for its design. Parts, requirements, parameters, functions, constraints, descriptions and pictures from the SRMS were entered into the integrated knowledge base (IKB) as baseline data. Requirements and constraints which represent interdependencies where also entered into the project decomposition. Figure [1] demonstrates how components are decomposed within any project over many iterations of concept-analysis-design-requirements.

Each iteration creates a new set of components that can be designed and constructed independently by a design team. Each component, however, is interdependent with the rest of the project, and other components. Without proper knowledge access and visualization tools, it is difficult for a team to determine how components affect them and how their component affects others. As a result, revisions to a component may have disastrous effects because engineers are unable to see the effect of their changes on other systems.

Picture too large (7MB) will reduce

Figure 1 The V-model shows iterative decomposition of a project into components [2]. The unseen interdependencies between components, however, may be complex and subtle.

Other aspects of information accessibility include access to current and previous project information by the design engineers. A study has shown that design reuse reduces defect density and increases productivity, and a second study has shown that searching for the right information takes 60% of a design engineer's time, and is rated as the most frustrating of an engineers activities [3]. If the information cannot be found quickly, the engineer will give up searching, and duplicate the design (possibly along with the mistakes) at great cost in time and money. In these cases, the industry's information repository of filing cabinets and isolated databases is unaccessible and ineffective. A corporate system allowing engineers to query data repositories is essential and valuable. Without data coordination and query tools, old designs may never be explored or reused, resulting in lost opportunity, time, effort and money.

Other valuable information sources include the problems, solutions, and dead-ends encountered in previous projects. Although this information is extremely useful, it is seldom accessible because it is locked up in people's heads. Companies often find themselves repeating common mistakes. There are two methods of getting advice that are explored through the KAD system. The first allows direct querying of a repository of knowledge that is collected in an unobtrusive manner. The second is to have a system recognize symptom patterns and automatically warn a user of the risk of an occurrence.

2.2 KnowledgeConsistency

Coordination problems also arise when CAD drawings, requirements documents and specification documents get out of sync. The cause of this problem is that a kluge of systems allows identical or dependent parameters to acquire inconsistent values. These problems can only be overcome if there is a central information repository or an information switch that allows applications to access and update information. The root source of this problem is that the user is made to enter information more that once. Not only is this tedious and annoying, but it the cause of expensive mistakes in space history. A typical example of how information trickles into the final product can be seen in Figure 2, requiring duplication at every stage.

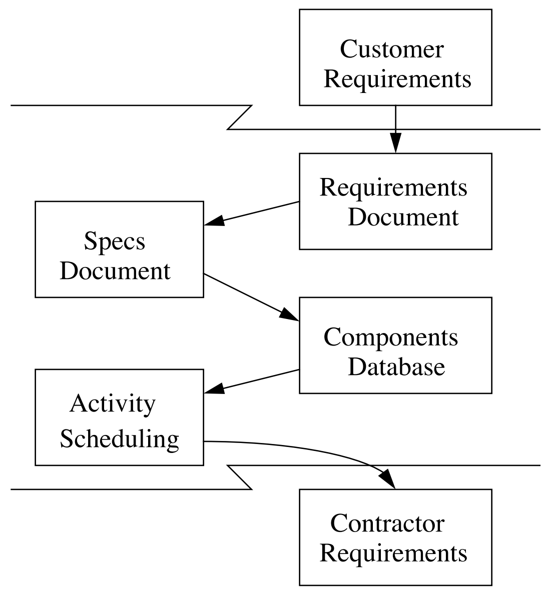

Figure 2 How information is passed down from document to document; hand to hand. This style of project tracking and development is redundant and time consuming.

The information is read from one document, and entered into another, resulting is a large delay in change implemention. Also, as revisions are made, all effected documents must be checked for consistency which creates a large overhead for revisions. The results of this revision difficulty is the oppression of creativity and sub-optimal designs because of the enormous efforts required to make even small changes. Solution to these problems are actively being explored by EIL and are discussed in this paper.

3 Solutions through the WWW

The availability of a universal client server architecture provided by the WWW has greatly facilitated the development of tools to solve the geographically and departmentally distributed problems of coordination. Benefits of using the WWW include a complete lack of concern for the development, maintenance, and distribution of GUIs for different platforms, development of communications protocols, or compatibility with other distributed applications. Other benefits noticed by using the WWW architecture is that separate programs could be merged seamlessly to appear as one application. This was facilitated by the netscape frames extension that allowed unanticipated programs work together as if planned. These benefits plus the low cost per seat make the WWW an ideal architecture for distributed business and engineering applications.

3.1 Transforming Data into Knowledge

DMS (Distributed Management System), was the first of the coordination tools developed on the WWW architecture with Oracle RDBMS as a backend. DMS is a set of CGIs that comprise a tool which improves on the deficiencies of email. The problem with email is that it is received personally, and when that person leaves the company, their repository of communications is lost with them. Usually email conveys important pieces of information such as goals, schedules, problems, solutions, dead-ends, and milestones. Although a person can be trusted to keep track of their own mail, this information is inaccessible to current and new team members. DMS was developed to be an information coordinator that categorized, archived and distributed messages. The advantage is that these messages were transformed into knowledge, and could be viewed by category/thread, indexed by keyword, queried by content, or shown in chronological order. The result was a permanent repository of information that remains long after users have left. The tool is now used by many teams, the most distributed of which consists of 12 members from different companies, all collaborating on a business process reengineering project. Because the system was developed on the WWW, it allows collaborative work to be performed in companies located across Ontario. This tool is generic and can be applied to any collaborative project. This tool serves the purposes of both allowing people to collaborate, and communicate, as well as providing a permanent store of information that can be powerfully queried using regular expressions and recursive logic. Also because the information is collected as it is send out as mail, the collection of this data is unobtrusive, and after many months it is surprising to see the amount of useful information that has been collected.

3.2 Perspectives for Multi Role Access

The next phase of software set out to tackle the problems that specifically face design engineers in a concurrent development environments. The web interface was chosen again because of its ideal distributed architecture. The vision was a system that could extend information about the SRMS throughout and beyond a company into the customers and contractors offices. The problem is that some information is sensitive, and should never be seen by unauthorized roles. There was a need to provide only the information that was required to prevent unwanted clutter and release information. This ability was implemented by perspectives, which allows certain people access to specific information. A perspective is the information that can be seen or edited from a certain role as shown in figure [3]. This uses the same concepts of roles in SQL, however, the security is handled through WWW authentication and access control rather than through the database itself.

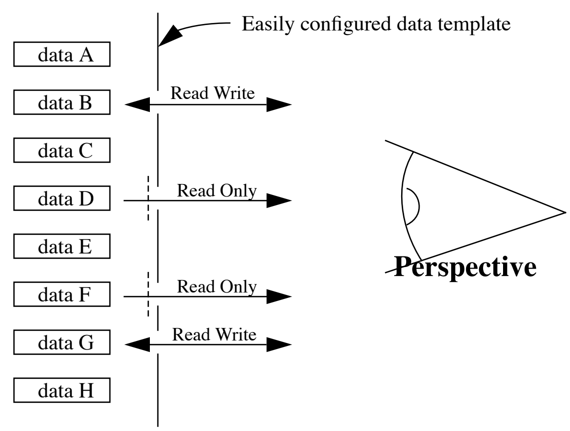

Figure 3 This diagram illustrates the concept of perspectives that allows information to be shared, while also restricting access.

The templates provide many levels of access to the data so that any administrator can easily configure perspectives. For example the design perspective uses this template fragment:

```
Version: @tnu::version_num!
Name: @tsu::name!
Token: @tsu::token!
Designer: @tsu:kb_GetValues(@@,prt_dsgn,XX)!
```

The @ and ! characters are delimiters which indicate the beginning and end. The first of the three characters allows for a flexible definition of the display type. Display types automatically handle pictures, large text areas, links and many other types. The second character indicates edit permissions to define what can be altered by the user. The third character selects the data source allowing for query routing. The rest is either an object attribute or a query. When the template is displayed, values are printed in place of the query statement. Future plans may implement the following perspectives:

design	receiving
manufacturing	purchasing
partner	testing
customer	quality_control
supplier	payroll
maintenance	customer_support
supply	marketing
scheduling	sales
accounting	public
human_resources	tutorial
shipping	

This set of perspectives would allow companies to coordinate the important operations so that individual departments can operate more collectively. This also permits access to better indicators of how the company is performing. The distributed nature of the WWW can function as a universal platform to allow authorized users to interact with their company from any location in the world, and will allow for an effortless move to branch plant or home office access.

3.3 Enforcing Consistency

Data inconsistency is unacceptable in collaborative environments because it causes anything from lost efforts to mission failure. Distributed environment are conducive to inconsistencies because each independent domain may appear self consistent, and so errors are not easily detected. This project has attempted to tackle two sources of information inconsistencies through a WWW interface. The first source of inconsistency occurs when independent teams make modifications causing two or more dependent parameters become inconsistent. The second is when a revision is made, and support information (i.e. documents, CAD drawings, etc.) become inconsistent with the design. Although conceptually they share the same results, the solutions are different.

The first source of information inconsistency is solved by a constraint management layer which propagates changes made using mathematical or conditional expressions (constraints). For example, the total weight of an object should not be entered every time the weight changes. Instead, the system should dynamically calculate the total weight by summing the weight of all the component of the object. This simple example can be used to keep very complex systems of parameters in sync with each other. This constraint management allows designers to define the constraints of reality, customer requirements or safety through a WWW interface. Users are now able to immediately see what consequences of changes have arisen. More importantly, from an interface point of view, the users do not have to be concerned about modifying all dependent parameters that require changes. If the design violates a safety constraint, then the offending change will be identified, even if it is buried through many systems interactions. Each simple interaction can be defined through the WWW, and the resulting constraint network would result in capturing complex behavior. The network itseld can also be visualized through the WWW. The effects of changes can now be caught in minutes instead of months of manual checking, expensive prototyping or disastrous mission failures.

The second source of information inconsistency is the decentralized method of trickling information through a project as shown in Figure 2. The solution is to completely redesign the way documents are produced by dynamically generating them as needed through a WWW interface. This has been accomplished by producing reports using a report template which extracts information from a knowledge base as it is needed. The report generation template supports any HTML capability including calculations, table generation, text display/entry, pictures and links. The benefit of dynamic documents are instant global accessiblity, up to date information, and no required time and effort to produce. If a parameter is changed directly or indirectly, then the recompiled document is available on request. The reports are not just limited to object attributes, but are able to traverse relations to show any stored or inferred information about the object. Changes in design may result in immediate changes in the document ensuring that reports are always consistent with the design and other documents. The contractor would also have the ability to (with permission) see how the requirements decomposition is going and make changes themselves. Approprate approval for these changes can be handled through existing PDM systems.

Plans are under consideration to integrate other systems that have independent information stores that may become inconsistent with eachother. Such information stores include isolated databases, scheduling systems, and parametric CAD systems. These plans are being considered because an integrated design environment will facilitate collaboration and result in improved potential productivity.

3.4 Knowledge visualization using the WWW

Knowledge visualization is vital for the faster understanding of a design. This essentially transforms data into understandable knowledge so that users can more quickly see objects, their attributes and their relationships. The graphical nature of the WWW allows these concepts to be conveyed and manipulated and is superior over textual information. This system has implemented two different types of viewers to visualize hierarchical and networked information shown in Figures 4 and 5.

Figure 4 This picture shows the use of a table based hierarchy viewer using consistent color and shading symbology to help identify objects.I

The first uses tables to display the hierarchical nature of data and symbology to quickly identify the different types of objects (components, parameters, constraints, etc.) Symbology is a system of colors, shades, textures or shapes to convey information. This system uses a highly flexible symbology module which allows consistent background colors, icons and ID bars to be defined by a single configuration command. The colors are transformed through computation so that all colors are mapped in a consistent way to the WWW browser. The result is a highly configurable consistent color scheme that identifies each object type uniquely.

Figure 5 This picture shows the java network browser. This applet is animated and allows drag & drop capabilities of objects. (source from Sun Microsystems)

Because of the need for interactivity to navigate and manipulate graphical displays, viewers have been implemented using java. Java allows programs to run on the client so that information can have a quicker response (more interactive). Future systems under consideration may use 3 dimensions or hyperbolic view transformation to make better use of the limited computer display space. The most important viewer requirements are to allow the user to see as much relevant information as possible without being cluttered or confusing. Graphical viewers have the advantage of being able to display the relations between objects. The relations allow engineers to visualize dependencies between objects and understand how the item is affected and how it affects others. Collaboration is facilitated because all engineers can see the interactions with other components in the project. This understanding allows engineers to design with more knowledge which is predicted to improve design quality and reduce unexpected conflicts found at the final stages of assembly.

3.5 Active Advice through Symptom Detection

A problem faced in todays industries is that conflicts are not detected until they have already happened. For example, one may not know that a project is over its weight limit until all the parts are ready for assembly. Although this simple example is more lack of planning, there are other more complex and subtle interactions that are not as easily budgeted. In order to help deal with this problem, a symptom management agent (SMA) was developed to inform engineers and managers of problems before they occur.

As changes are made through the WWW interface, the SMA checks to see if the data matches patterns and warns of possible future problems. An example might be to warn the manager if 60% of the budget has been spent before 40% of the schedule time has passed. In contrast to the passive information store (DMS) that must be queried, this system actively advises users as the information is entered.

3.6 Rapid Redesign

Once the information about a project has been entered, there is exists an information repository that allows schematics, requirements documents, schedules, budgets, etc. to be accessed. Accessibility, however, is more than just the presence of information. Without proper retrieval tools, the information is lost in its own clutter of quantity. Old designs cannot be reused if they can not be found. In order to truly make information accessible, the IKB also has a powerful set of retrieval mechanisms that allow for parametric and regular expression searches. The tools also has the capability taking requirements and searching (case based retrieval) all previous designs[4] to find a match. This powerful matching capability makes it more likely that a design will be found and reused. Design reuse has been shown to reduce costs, development time and defect densities[3].

Once the information has been retrieved, the WWW interface allows the designs to quickly be altered to match all the requirements as necessary. The engineers is guided by a violated requirements list that gives direction to the changes that must be made. The result as better reuse of existing designs.

4 System Architecture

This research project used the most ideal programming languages available for the required tasks. The working version of this project included C, perl, java, javascript, Eclipse prolog, Clips, Oracle RDBMS, ROCK RDBMS, and Netscape, and implemented HTML, KIF, and KQML through CGIs and agents. This style of multiple language development was personally found to be superior for development ease. The time need to learn new languages and develop interconnectivity was well offset by the reduced complexity and coding time. This attitude allows a richer selection of applications, languages and platforms to be considered, and may be of interest to groups that insist on using only one language for an entire application.

One of the problems with WWW development is that CGIs are not persistent, and are started and terminated with every query. A solution is to pass information from one CGI to another, however, with large projects, this results in unmanageable programming requirements. Both data management difficulties and document size grow exponentially with respect to the system complexity. These challenges of WWW programming were solved by using a data management layer shown in figure [6]. This layer unified all CGIs and allowed variables to have automatic persistency without any further programming efforts. As a result, WWW development speed and programming complexity were considerably improved.

<table>
<tr><td colspan="3">HTTP Server & Access Control</td></tr>
<tr><td colspan="3">Session Management Layer</td></tr>
<tr><td colspan="3">Data Management Layer</td></tr>
<tr><td>CGI A</td><td>CGI B</td><td>CGI ..</td></tr>
<tr><td colspan="3">IKB Backend</td></tr>
</table>

Figure 6 This diagram shows the layers under the HTTP server that allow for a unified and simplified WWW development environment.

Other issues exist when multiple users require access. Each user should be free to change their environment, keep setting, and maintain continuity throughout their session independently of other users. The solution was a session management layer (figure [6]) that transparently provides the CGIs and data management layer with independent workspaces. This ensures that each users actions does not affect another in an adverse or unexpected way.

5 Summary and Conclusion

Collaborative design allows for reduced costs and development time for large projects, but is prone to inconsistencies. Inconsistencies can be managed by implementing information coordination. Such a system has been implemented through the WWW that is able coordinate information and eliminate inconsistencies by propagating changes to all dependent parameters. The system also handles inconsistencies by eliminating redundant data entry and by generating documents dynamically.

Engineering design cost and time can also be reduced through reuse of old designs. In order to effectively reuse designs, old design information must be accessible. Information can only be made **accessible** by capturing it, and then providing tools to search and retrieve it. Without these tools, the information will be lost over time, and become inaccessible. It is concluded that a system, such as this one on the WWW, can reduce design cost and time by ensuring information consistency and accessibility.

Acknowledgments

The systems described in this paper are the collective effort of Taner Bilgic, Jinxin Lin, John Chionglo, Lokesh Gupta, Jacek Gwizdka, and William Leizerowicz under the direction of Mark S. Fox. I would like to thank my colleagues for their inspiring dedication, relentless effort and keen application of knowledge to produce this system.

References

[1] Taner Bilgic. “Concurrency in Engineering Design” http://www.ie.utoronto.ca/EIL/DITL/ProjectManagement/Coordination/concurrent.html

[2] P. Ludikar, “Systems Engineering Management Plan Handbook”, SPAR-HDBK.4 Issue A., Spar Aerospace Ltd, Advanced Technology Systems Group, 9445 Airport Rd, Brampton, Ontario, L6S 4J3.

[3] Crabtree, R.A., Fox, M.S., Baid, N.K.,, “Towards an Understanding of Collaborative Design Activities” Journal of *Research in Design Engineering*[3]

[4] Taner Bilgic, and M. S. Fox, “Constraint-Based Retrieval of Engineering Design Cases: Context as Constraints”, Artificial Intelligence in Design ’96, Stanford University, June 24--27, 1996, (to appear in S. Gero and F. Sudweeks (eds) AI in Design ’96, Kluwer Academic Publishers).