ABSTRACT

This paper presents an integrated but open product development environment for distributed and collaborative design. The web-based framework, called DOME, allows designers to build integrated models using both local and distributed resources and to collaborate by exchanging services. Thus, an integrated model can be created while each participant focuses on their own area of expertise. A design model is created by connecting modules, each of which can represent specific components, analysis capabilities/software, disciplines, or organizations relevant to the problem. The modules interact with each other using service exchanges based upon the CORBA standard communication protocol for distributed objects. The resulting module network forms a concurrent model in which changes propagate through service exchanges. Modules can simultaneously function as both clients (using services from other modules) and servers (providing services to other modules). A Java applet-based user interface provides cross-platform and distributed user access to DOME module servers throughout the network.

Keywords: World-Wide-Web, network-centric, object-based distributed modeling, integrated product design, collaborative design.

INTRODUCTION

Accelerating time to market and improving product quality is one of the driving forces encouraging companies to streamline product development processes. However, it is common to hear the sentiment that, although there are now numerous high quality specialized tools and models, it is very difficult and to understand and design the integrated performance of product systems. This paper presents a development environment that is intended to facilitate the rapid construction of integrated design models.

The approach is based upon the assumption that integrated design requires the skills of many designers and experts and, that each participant creates models/tools to provide information or simulation services to other participants given appropriate input information. It is our goal that collectively, the network of participants exchanging services will form a concurrent model of the integrated design. The concept can be described through analogy. At present, researchers and experts publish information on web pages [1]. Individual web pages are then connected together to form larger topical databases (for example, [2]). In the future, experts and designers may publish models on the world-wide-web. These models will operate when they receive information from clients or other servers. Individual web-based models will be connected together so that they can exchange services to form larger integrated system models.

An object-based modeling environment called DOME [3-6] provides the basis for this concept. DOME is intended to link heterogeneous design models and tools, assisting designers in evaluating the system performance of design alternatives, visualizing trade-offs, finding optimal solutions, and making decisions.

Building on DOME, an architecture that allows models/objects to be published and connected over the World-Wide Web to form integrated models is described. The paper begins with an overview of requirements for network-based design tools and then reviews related work. A brief description of the DOME system is also provided. Then, a system architecture for supporting different types of collaborative
design activities in a distributed design environment is proposed. Its implementation exploits CORBA and Java technologies to achieve an interoperable and easily accessible environment. A case study demonstrating the functionality of the architecture is also provided, illustrating how designers in different teams and organizations may participate in the design of a central district heating plant.

CHARACTERISTICS OF NETWORK-ORIENTED DESIGN TOOLS

This network-centric distributed design environment is intended to facilitate modeling interactions between many designers and experts participating in a product development effort. In this section key characteristics of this environment and its software support requirements are discussed.

Open & interoperable

In network-oriented computing it is almost impossible to have a homogeneous environment. Each expert may have their own preferred computer system and specialized software tools. Therefore, it is crucial for a network-centric design system to support a heterogeneous computing environment. Further, design problems grow and participants change over time. Correspondingly, an open environment is needed so models from new participants can be added, allowing the design model and tools to evolve with the design problem.

Distributed object technology, such as CORBA (Common Object Request Broker Architecture) and DCOM/ActiveX can be used to address this issue [7, 8]. A computer platform and language-independent interface definition allows software applications to communicate with each other provided a neutral interface has been agreed upon. For example, the World Wide Web has gained its popularity and momentum through a platform-independent protocol (i.e., HTTP) and a language-independent scheme (i.e., HTML) for presenting information.

Distributed & decentralized activities and design resources

Computer tools for product design are generally stand-alone applications. However, design activities may involve many participants from different disciplines. Thus, there is a need to support and coordinate highly distributed and decentralized modeling activities. Distributed design systems might have two distinct forms: distributed designers with access to centralized resources, or distributed designers with distributed resources (e.g., engineering models, databases, software applications, etc.). This work focuses on the later architecture. Decentralized means that the coordination between design participants and models is not centrally modeled or controlled (analogous to the WWW). This is important because centrally controlling the interactions of all distributed resources may restrict system growth and flexibility.

If there was a centralized control over the WWW for linking the hyper documents, it could not evolve so rapidly.

Concurrency

Concurrent engineering (or simultaneous engineering) emphasizes the early consideration of downstream requirements [9]. Critical to concurrent engineering is the maintenance of information consistency between participants and the rapid handling, exchange and propagation of design information or events. It is imperative that an integrated design system addresses these issues. Further, in order to reduce training costs and support a broad spectrum of designers, the integrated system needs to provide a single system image that gives users a consistent and easy to use interface [10]. The DOME framework adopts a web-browser interface for this reason.

Knowledge encapsulation & object-oriented approach

While many individuals and organizations may provide services so that an integrated product model can be constructed, it is not likely that each participant will disclose the full details or structure of their proprietary models and data. Providing a means for encapsulating expert knowledge or know-how is essential. An object-oriented approach provides a framework for such knowledge encapsulation [11]. Furthermore, an object-oriented architecture is also highly suited to a distributed computing environment [12].

Collaboration

Most design processes are a collaborative activity? designers from different disciplines need to communicate and interact with one another. We assume that problems are decomposed into subproblems and that each team member focuses on a particular portion of the subproblem. One of the key roles of an integrated modeling system is to mediate the information-flow between the participants.

Trade-offs analysis and decision support

It is important to provide designers with a means to compare design solution alternatives and support the decision making process. In a distributed design environment many designers may be working in collaboration, so there is a need to balance tradeoffs between competing viewpoints. Some design systems try to resolve the conflicts by notifying designers with conflicts so they can negotiate neutral solutions. In DOME, a multiple attribute decision method is used to capture preferences and evaluate design alternatives from different viewpoints.

NETWORK-CENTRIC DESIGN SYSTEMS

Many researchers are working on enabling technologies or infrastructure to assist product designers in the computer network-centric design environment. Some are intended to
help product designers to collaborate or coordinate by sharing product information and manufacturing services through formal or informal interactions. Others propose frameworks that manage conflicts between design constraints and assist designers in making decisions.

**Collaborative design information systems & manufacturing services**

The SHARE project by Cutkosky et al. supports design teams by allowing them to gather, organize, re-access and communicate design information over computer networks to establish a shared understanding of the design and development process [13]. While SHARE is primarily directed towards interaction through integrated multimedia communication and groupware tools, the NEXT-LINK project incorporates agents to coordinate design decisions affected by specifications and constraints [14]. Another network-centric design system using interacting agents to integrate manufacturing services available over the network is also under development [15]. The motivation and vision presented in this paper has similar themes but emphasizes mathematical modeling, decision-making and search/optimization.

The Electronic Design Notebook is an interactive electronic document which maintains the look and feel of an engineering document to provide an integrated user interface for computer programs, design studies, planning documents, and databases [16]. Manufacturing tools and services are encapsulated in the hypertext documents and distributed through servers using HTTP [17].

A design information system proposed by Bliznakov et al. incorporates a hybrid model for the representation of design information at several levels of formalization and granularity. It is intended to allow designers in a large virtual organization to indicate the status of tasks assigned to each designer or team so that other designers can follow their progress [18, 19]. A central database manages pointers and access methods for product and process information in the distributed environment.

Hardwick et al. proposes an information infrastructure architecture that enhances collaboration between design and manufacturing firms [20]. This architecture uses the WWW for information sharing and the STEP standard [21] for product modeling. It utilizes the CORBA standard for interoperability between software applications in the virtual enterprise.

N-dim is a computer-based collaborative design environment for capturing, organizing and sharing data [22]. The system is a base on which applications can be added for the purpose of history maintenance, access control and revision management. The primary focus of this research is information modeling. The system provides way for defining information types that capture the relations between data or models.

There are also national-level efforts involving university and industry collaboration to make a variety of engineering services available over the Internet [23-25]. The RaDEO program is concerned with comprehensive information modeling and design tools needed to support the rapid design of electro-mechanical systems. It supports engineers by improving their ability to explore, generate, track, store, and analyze design alternatives. The National Industrial Information Infrastructure Protocols (NIIIP) Consortium is attempting to develop open industry software protocols that will make it possible for manufacturers and their suppliers to effectively interoperate as if they were part of the same enterprise. The NIIIP goal is to help suppliers assemble virtual enterprises.

The CONSENS project (Concurrent & Simultaneous Engineering System) is a European project that is exclusively devoted to concurrent engineering [26]. It focuses on information infrastructure. Process, product, and project data management are supported by several engineering software applications integrated with the SIFRAME framework.

**Design rationale and conflict resolution**

Case and Lu's Discourse Model provides software support for collaborative engineering design by treating interactions as a process of discourse [27]. The model captures design commitments as opinions subject to review and revision by other designers. It also utilizes agents to identify conflicts between designers and to negotiate the resolution of conflicts.

A computer-based design system developed by Sriram et al. provides a shared workspace where multiple designers work in separate engineering disciplines [28]. In their DICE (Distributed and Integrated Environment for Computer-aided Engineering) program, an object-oriented database management system with a global control mechanism is utilized to resolve coordination and communication problems. Design rationale provided during the product design process is also used for resolving design conflicts [29, 30].

**OVERVIEW OF THE DOME SYSTEM**

In this section the basic principles of the DOME (Distributed Object-based Modeling and Evaluation) system are reviewed. The Web-based modeling and evaluation environment presented in this paper will build upon this framework. Details may be found in [3].

The DOME framework asserts that multidisciplinary problems are decomposed into modular subproblems. Modularity divides overall complexity and distributes knowledge and responsibility amongst designers. It also facilitates the reuse of modeling elements [11]. DOME allows designers to define mathematical models/modules and integrate/interconnect them to form larger system models.

Modules represent knowledge related to different aspects of the design in the form of variables and relations (Figure 1).
Load Characteristics
Power Supply

Example Motor Module

Figure 1  Example DOME module. The variables contained in the module are represented as interconnected circles (the directed arcs imply dependency). Outputs and inputs to the module constitute the interface of the module. Modules can be interconnected through interfaces.

Custom created computer programs and third party applications, such as domain specific analysis tools or CAD systems, can be embedded into a module. Modules interact with each other by exchanging information and services, reacting to each other’s changes for an integrated system. Modules can be distributed over the network, collectively forming a distributed model for a collaborative, multidisciplinary and concurrent design evaluation.

In the distributed environment, each group of designers can define their own modules, loading them into their local work area and eventually connecting them to the other parts of the design problem through appropriate networked interfaces. Figure 2(a) shows a DOME model involving two designers and three modules. Designer 1 defines modules A and B while designer 2 defines module C. The two domains communicate through an Internet connection. Once the whole problem is loaded and interconnected, each group of designers typically has write access to local parts of the model (i.e., they can exert decisions within their local range of influence) and read access to relevant aspects of remote parts of the model. This allows them to see the remote effects of their local decisions. Figure 2(b) shows that both designers 1 and 2 might see the complete problem, but with different access privileges. Designer 1 can see modules A and B as local and module C as remote. Conversely, module C is local to Designer 2. The remote part seen by Designer 2 could show modules A and B or just a single distributed object (AB) if Designer 1 restricts their visibility/access.

The interaction between distributed modules is achieved by publishing and subscribing to services. The term publish refers to making the services of one’s local model visible to other designers. The term subscribe refers to making use of published services.

Such design problem models are mixed variable, where independent parameters within modules are set and catalog selections might be used to substitute entire modules. An optimization engine based on Genetic Algorithms has been developed and integrated with the framework to perform mixed variable search and optimization. Design solutions can be assessed and compared with each other using decision-making tool embedded in the framework.

ARCHITECTURE FOR PUBLISHING AND SUBSCRIBING TO MODELING SERVICES ON THE WORLD-WIDE-WEB

In DOME, modules are connected together so that they can exchange services to form larger integrated system models. The module structure of DOME lends itself to a client/server-oriented architecture using distributed object technology. The main system components of the proposed three-tiered client/server architecture [31] are shown in Figure 3. Each of these components interact with one another using a communication protocol (CORBA) so that it is not required to maintain the elements on a single machine [7].

In Figure 3, open circles are Interfaces. As a gateway for providing services, the interface of a system component invokes the necessary actions to provide requested services. To request a service, a system component must have an interface pointer to the desired interface (illustrated using solid circles). A solid arrow is used to illustrate a service request between an interface and interface pointer.

Figure 3  Main system components for the DOME architecture

(a) GUI           (b) DOME server       (c) Model repository
(server)        (application server)              (database server)

Distributed object technology refers to an interoperable computing environment for inter-process and cross-platform integration of software components.

Since the GUI becomes CORBA compliant at runtime, it does not require the user to install any additional system other than a Web browser. Not requiring a thick client is important to provide users with flexible accessibility.
Interaction between modules exchanging services

The architecture is designed to allow experts to publish and subscribe to modeling services on the World-Wide-Web. These services will operate when information is received from other clients or servers.

The distinct characteristic of the DOME architecture is that when modules services are connected, the resultant service exchange network forms an integrated concurrent system model, as shown in Figure 4. Any one service request in the module network can invoke a chain of service requests if needed to provide correct information.

![Figure 4 Service exchange in a module network](image)

When a design alternative is evaluated, the local model asks for the services of subscribed models. If the subscribed models themselves need services from other models in order to provide the request services, they will again request those services from their own network of remote models. Therefore, service requests are propagated through connected modules.

However, the complete system may not be visible to any given model. Since modules can only interact through services, it is possible for a module or local model to encapsulate its internal modules and hide intellectual property if desired. Before a designer publishes, they can assign access privileges for their services. Three levels of model access have been identified: Owner, Builder, and User. Owner is the original creator of the model and has access to all the services defined in the model and control over their publication. A Builder can see the internal details of a model which the owner chooses to make public and can add new modules. However, they cannot destroy modules created by the owner or other builders. Users can subscribe only to published services.

There are other architectures for network-centric collaborative design. One of common approaches is a centralized multi-user system architecture as shown in Figure 5.

![Figure 5 Architecture of a centralized multi-user system](image)

These systems pool data and models so that consistency can be maintained and provide means for version control. When multiple users change data or models stored in the central database, the system checks for conflicts between data modifications and either mitigates problems or notifies users to resolve the conflicts. The system may assign different levels of modification privilege for different parts of data or model. Some systems facilitate a common workspace called a blackboard, often managed and organized by the main system, where different users can put data or messages. The DICE [28] system and DIS system proposed by Bliznakov et al. are of this architecture. Although powerful, a central system is less suited for loose and flexible collaborations?it is not an open environment and does not allow for true knowledge encapsulation. However, such an architecture could be supported within a module of a larger DOME network.

Since global open networks such as the Internet have become a cost-effective infrastructure for exchanging information, less structured design information systems allowing users to exchange engineering data or standard product description files have emerged (e.g., the SHARE project by Cutkosky et al., EDN, MADEfast, RaDEO). We refer to this architecture as a data and model exchange system (Figure 6).

![Figure 6 Architecture of a data and model exchange system](image)

This architecture tends to provide an over-the-wall sequential interaction between designers and models. When a designer receives a model or data from another designer, he/she works on the design and send the result of design modification to others. Therefore, this architecture is not intended to provide concurrent system modeling functionality.

Agent-based distributed system architectures provide the most abstract level of system interaction and interoperability. An agent is generally referred to a program that behaves according to its own preference or logic and performs tasks
without direct human supervision [32]. There are several different types of architectures for agent-based systems: agent network, federation, and distributed blackboard architecture [33]. From a system architecture viewpoint, the distinction between DOME and other agent-based systems is the use of distributed-objects versus agents5. DOME maintains generalized communication objects (i.e., modules) and mediates the propagation of explicitly defined messages or services containing design information. This is referred as the communicating objects paradigm [34]. In the autonomous objects paradigm, a message has its own identity and behavior, deciding at run time what tasks to perform. Agents are more appropriate for loosely coupled environments where mutual interaction between objects are not well defined. In DOME, the interaction between subproblems are explicitly defined through design negotiation so a communicating objects paradigm is appropriate. However, within DOME, agents will be useful when designers are not certain about what modules can provide the service they require. Agents could locate appropriate modules.

IMPLEMENTATION OF THE WEB-BASED MODULE NETWORK ARCHITECTURE

The implementation of the module network architecture, (henceforth Web-based DOME) is based upon WindowsNT®-based environment with a Web-browser-based Graphical User Interface3. The underlying DOME framework and optimization engine is written in C++ by members of the MIT Computer-aided Design Laboratory.

The Web-based DOME implementation shown in Figure 7 uses the three-tier client/server architecture of Figure 3 to support collaborative design interactions. The components in gray have been implemented in the CADlab prototype. The underlying DOME framework and optimization engine is written in C++ by members of the MIT Computer-aided Design Laboratory.

The Web-based DOME implementation shown in Figure 7 uses the three-tier client/server architecture of Figure 3 to support collaborative design interactions. The components in gray have been implemented in the CADlab prototype. The CORBA standard is used to add distributed communications capabilities to modules (Orbix® 2.2 and OrbixWeb® 3.0 from IONA Technologies Ltd. [35]). CORBA serves as an information and service exchange infrastructure above the computer network layer and provides the capability to interact with existing CAD applications and database management systems through other Object Request Brokers (ORB). In turn, the DOME framework provides the methods and interfaces needed for the interaction with other modules in the networked environment. These interactions are graphically depicted in Figure 8.

---

4 It is true that many people use the term agents and distributed object interchangeably. However, the agent can be considered as a subset of distributed objects with abstract messaging and independent decision-making capabilities.

5 WindowsNT® is a registered trademark of Microsoft Inc.

---

6 OrbixWeb is a registered trademark of IONA Technologies Inc.
**DOME server**

The GUI interacts with designers and delegates user events and requests to a DOME server which provides the back-end implementation for the modeling of design problems. The core of the server is based upon DOME’s Object-oriented Modeling and Evaluation (OME) kernel [3] written in C++. The back-end implementation of DOME server is written both in C++ and Java while the front end interface to the GUI is written in Java.

The DOME server manages each design session in a Workspace (see Figure 3) and can simultaneously maintain several workspaces. Additionally, the workspace manages administrative aspects of a model (e.g., ownership, access privilege, links to other workspaces in different DOME servers, etc.). The DOME server itself is a CORBA-compliant distributed object and can communicate with other DOME servers.

**Model Repository Server**

The Model Repository Server maintains persistent storage for models created by the DOME servers. The repository stores a model in a Model Definition File (MDF). A MDF is comprised of two parts: meta definition and model definition. The meta definition contains the information such as model ID, ownership and access privilege information. The model definition is based upon a Model Definition Language (MDL) used by the DOME system.

**EXAMPLE: COLLABORATIVE DISTRICT HEATING PLANT DESIGN**

The example illustrates how designers from different teams, divisions, and companies may participate to create an integrated design model for a complex problem. The overall topology of the problem is illustrated in Figure 9. The models and simulations embedded in modules used in this example were developed at the Swiss Federal Institute of Technology in Lausanne under the supervision of Professor Daniel Favrat.

**Figure 9** Problem topology of the central district heating plant design model. The gray rectangles represents organization boundaries.

Figure 9 shows that two companies (gas turbine and heat pump manufacturers) are providing their design models to the plant design team, who in turn develop the technical model for the plant system. The plant operation team collaborates with the plant design team by providing models and data for plant operating conditions and requirements. The central heating plant design model is used by the management to develop cost evaluation models.

**Gas turbine and heat pump manufacturer**

The gas turbine and heat pump manufacturers develop models for their products so that customers can obtain performance predictions for different parametric configurations and operating conditions. These individual models are constructed, published and served by each company. If a single individual creates the model to provide these services they will work in an individual workspace, as illustrated in Figure 10.

**Figure 10** Individual workspace for a gas turbine designer.
Figure 11 Design problem workspaces for gas turbine and heat pump designs.

The Web-based DOME GUI shown in Figure 11 depicts the modules in the gas turbine and heat pump workspaces. The designers can use any commercial Web-browser to access and work on these modules. Since customers will connect to these models to assess the performance of their products, the designers will decide what simulation services the models will offer given appropriate input information. This is referred as publishing a model. When a model is published anyone can use its services if he/she has the appropriate access privileges. The owner of the model may want to conceal knowledge intensive engineering formulae or supply chain information embedded in the model. Figure 12 shows the Web-based DOME GUI for service publication. A designer sets access privilege levels for the services of each module in their workspace.

Figure 12 Publishing services. The designer working on the heat pump model is assigning access privileges to the services that modules can provide.

**Plant design and operation teams**

The plant design plant operations are tightly coupled so it would make sense for these groups to share a common model, as shown in Figure 13. Therefore, while designers from the different groups are in remote locations, they can access into the same workspace. This is referred as a shared workspace.

Figure 13 The Plant design team is connected to the gas turbine and heat pump manufacturers. The plant designer can test their plant design integrated with gas turbine and heat pump models. The designer from plant operation team is sharing their workspace with the plant designer.

Figure 13 shows the design workspace as viewed by the plant designers and operation designers. Modules in the upper left corner are created by the plant design team while the rest are made by the plant operation team designer. In this case the plant design team owns the session and the operation team has joined as a builder. Although builders cannot modify the modules created by other builders or owners, they can add new modules and utilize all services. For example, the plant operation team designer can use a service from a plant design group’s model to obtain the plant heating capacity and ca
build new modules in the workspace that utilize this information.

Figure 14 also indicates that the plant designer is using services from the models published by the gas turbine and heat pump manufacturers. Utilizing models provided by other designers is referred as *subscribing* to a model. Figure 15(a) shows the process for subscribing to the gas turbine model. The input parameters needed by the gas turbine model to provide these services appear as unfulfilled dependencies (see Figure 15(b)). It is the responsibility of the plant designer to provide these data or to locate other models that can provide these data as services.

Figure 14 Design workspace for the central heating plant.

![Figure 14 Design workspace for the central heating plant.](image)

(a) Subscribing the gas turbine model

(b) Unfulfilled gas turbine dependencies

Figure 15 Services from gas turbine model subscribed to by the plant designer

**Plant management division**

The management division wants to evaluate the plant design from a cost viewpoint. They link their models to the central plant design to obtain the information services needed by their model (Figure 16).

![Figure 16 Plant management model connected to the central plant design model.](image)

Figure 16 Plant management model connected to the central plant design model.

Figure 17 illustrates the design workspace of the plant managers. The services provided by the plant design group (the plant and operation model) are illustrated in Figure 17(b). The plant designers have only published cost related aspects of their model. Therefore, plant managers can only observe elements of the plant design model that were published as the plant designer wanted to protect their proprietary models (compare to Figure 14).
CONCLUSIONS

This paper presents a web-based design modeling and evaluation framework intended to provide geographically dispersed designers with a tool for collaboratively building integrated system models. Large problems are decomposed into subproblems called modules. Models or other software applications are encapsulated in modules. A module can provide information services through its interface. A network of modules exchanging services form a concurrent system design model.

The DOME framework (Distributed Object-based Modeling and Evaluation) is built upon to provide an architecture for integrating modeling services made available on the network. This architecture is referred as a module network. In a module network, design resources, models, data, and activities are not centralized nor concentrated in one location. They are distributed among many companies, designers, or design participants working together over computer networks.

The module network architecture is extended to a computer network environment. Design models are created by fully implementing locally defined modules and subscribing to the services of remote modules. The software implementation hides the details of the remote interaction mechanism from the user, allowing the designer to model interactions between local and remote modules in a transparent manner. In turn designers can selectively publish modeling services for use by others.

The three-tiered client/server architecture was adopted to allow experts and designers to publish and subscribe modeling services on the World-Wide-Web. In this architecture, when modules services are connected, the resultant service exchange network creates an integrated concurrent system model or module network that invoke a chain of service requests if needed to provide correct information. The district heating plant example illustrates the concept and different modes of collaboration supported by the prototype implementation.

Future Work

Although the preliminary implementation illustrates the potential for Web-based DOME, there are a number of fundamental issues yet to be addressed. Some of the key issues are as follows.

Interface Standards

The use and reuse of distributed modules is somewhat dependent upon the standardization of interfaces. Further, the exchange of standardized product model data structures such as those based upon STEP would be desirable. For instance, the employment of neutral representation for geometric information using STEP AP203 might be used to extend a design problem model based on DOME to incorporate complex geometric information. Modules might also be classified to offer services of different granularity depending upon the degree to which their input.

Computational Strategy for resolving circular dependencies in a DOME model
The current version of web-based DOME is limited to design problem models without circular dependencies. Several conceptual solutions to this problem are currently under investigation. To determine the existence and location of loops within a problem model, a loop detection algorithm is being implemented. Although the loop detection in a local problem scope can be performed using several techniques (i.e., using incidence and adjacency matrix), it is difficult to find a circular dependency when distributed modules are utilized. In this case, the global topology of the integrated problem model cannot be obtained due to proprietary restrictions.

**Parallel service request invocation**

In order to improve the computational performance of web-based DOME in evaluating design solution alternatives, the simultaneous invocation of services is under consideration. When a module needs a number of inputs for providing its service, it can simultaneously send requests to the modules providing the needed inputs. However, careful coordination of service requests may be needed. For example, it may be necessary to determine whether or not the modules providing inputs are independent of each other. In this respect, the global topology of the module interdependency in a design problem model could be used to determine the best computational strategy at run time.

**Collaborative Design**

Web-based DOME does not enforce a particular design paradigm or methodology. The framework should accommodate top-down and bottom-up approaches in the context of both traditional sequential design processes and concurrent design. In a collaborative design environment other aspects, such as human interaction, will require the integration of additional support tools with the framework (e.g., email, video conferencing, hypertext documentation, etc.). If a modeling and decision support framework such as DOME is integrated with other computer-based collaboration tools, it will provide designers with a powerful infrastructure for concurrent product design.

**REFERENCES**


