Integrated Design in a Service Marketplace

Shaun Abrahamson
David Wallace,
Esther and Harold E. Edgerton Associate Professor of Mechanical Engineering
MIT CADlab
Room 3-455b, 77 Massachusetts Avenue,
Cambridge MA 02139
drwallac@mit.edu

Nicola Senin
Dipartimento di Ingegneria Industriale,
Universita' degli Studi di Perugia
Via G.Duranti,
06125 Perugia, ITALY
nsenin@unipg.it

Peter Sferro
Ford Motor Company
Advanced Manufacturing Technology Development
24500 Glendale Ave
Detroit, Mi 48239
psferro@ford.com

Abstract

This paper presents a service marketplace vision for enterprise-wide integrated design modeling. In this environment, expert participants and product development organizations are empowered to publish their geometric design, CAE, manufacturing, or marketing capabilities as live services that are operable over the Internet. These services are made available through a service marketplace. Product developers, small or large, can subscribe to and flexibly inter-relate these services to embody a distributed product development organization, while simultaneously creating system models that allow the prediction and analysis of integrated product performance. It is hypothesized that product development services will become commodities, much like many component-level products are today. It will be possible to rapidly interchange equivalent design service providers so that the development of the product and the definition of the product development organization become part of the same process. Computer-aided design tools will evolve to facilitate the publishing of live design services. A research prototype system called DOME is used to illustrate the concept and a pilot study with Ford Motor Company is used in a preliminary assessment of the vision.

Keywords: integrated modeling, system modeling, design service marketplace

Introduction

Predicting overall product and enterprise performance during development phases is becoming increasingly important as companies seek to produce better products with fewer resources. Interaction with our industrial partners (CIPD, 1999) suggests primary objectives include reducing time-to-market and costly design, build, test and refine cycles, while simultaneously increasing the breadth of knowledge that can be applied to solving a problem and considering more design alternatives. In recent years CAD has made the transition beyond geometry to include tightly linked CAE, revision management and BOM (Bill of Materials) capabilities, and more recently parametric optimization.[1] However, even this scope is limited in comparison to an enterprise-wide modeling viewpoint.

This paper's vision is a product development environment where digital representations of products and associated organizational infrastructure such as design, manufacturing, marketing, distribution, and support can be readily evaluated against corporate, customer, market, and policy objectives. Expert participants and organizations publish their geometric design, CAE, manufacturing, or marketing capabilities as live services that are operable over the Internet. These product development services are made available through a service marketplace. Product developers, small or large, can flexibly subscribe to and inter-relate services through appropriate computer-mediated interactions, building service network models that embody the capabilities of a product development organization.

Such models will allow the mathematical prediction, simulation, and visualization of product performance. Additionally, it will be possible to dynamically analyze the structure and performance of the service network (or product development organization) in real time. Tools will be introduced to facilitate the rapid elucidation and evaluation of tradeoffs across the product development organization. Product development services will become commodities, much like many component-level products are today. As a result, it will be possible to rapidly interchange equivalent design service providers so that the design of the product and the organization become part of the same process. In short, the design service marketplace provides a virtual environment for the integrated development of new products. Product development firms will compete on their ability to identify new product opportunities, to develop product system models and organizations leveraging appropriate service providers, and to embody and publish new product development services.

This is consistent with views such as the *once and future craftsperson* described by Lester Alberthal (Albethal, 1998) and in work by Wellman (Wellman, 1994). In this paradigm, highly responsive modeling and simulation systems move customers and suppliers closer together, allowing the rapid creation, validation, and delivery of products that closely match individual preferences. The modeling systems facilitate the exploration of how various objectives can be achieved.

---

[1] Tools such as Isight from Engineous Software Inc. and Parametric Technologies Behavioral Modeling which allow optimization and connection of models.

This paper illustrates three key elements of this vision: providing design participants with the ability to publish live services; creating a service marketplace that allows for the synthesis and seamless inter-relation of services to create integrated product models; and the introduction of tools that facilitate the rapid elucidation and evaluation of tradeoffs.

An overview of integrated modeling challenges and approaches is first provided, and then the design service marketplace is demonstrated using a prototype implementation of the concept called DOME (Distributed Object-based Modeling Environment). Architectural elements of the DOME prototype are provided in overview, and then a research and development deployment of the concept, in application to a door glass system at Ford Motor Company, is described.

Approaches to integrated product modeling environments

There are a number of challenges associated with creating integrated product modeling and simulation environments. Comprehensive product system modeling traditionally has been deemed infeasible (Ulrich 1995). This is partially because of the difficulty in modeling some elements of the design, but it is also because of the difficulties associated with the inter-connection of different sub-models. The task is complicated further by the scope and size of most product development projects. Cutkosky (Cutkosky *et al.*, 1994) identifies specific challenges as: (1) difficulty in defining a complete product model because of its size, complexity and evolving nature, (2) the need for different tools, data management, and representations, (3) all required data are not globally available for logistic or proprietary reasons.

Although not simulation based, enterprise systems attempt to provide an integrated management environment and face similar issues. In this arena, integration is represented by the extremes of complete, integrated solutions provided by a single vendor, to solutions synthesized from best of bread components. The integrated single vendor solution attempts to use standardization to ensure seamless operation, while the argument for best of bread is to ensure that the best of each available application is brought together. Examples of commercial systems based upon integrated solutions include SAP enterprise systems. Examples of best of bread systems include those making use of products offered by companies such as Oberon, Vitria or Crossworlds Software[2].

Corporations have found, upon embarking on large enterprise projects, that integration has its perils (Cooper and Kaplan, 1998). Individuals and organizations might be fit into a standardized solution, not recognizing the value of existing processes or methodologies. However, customized integration also has an associated cost. Not only is there additional overhead in setting up the system, but the system also requires customized maintenance. Integration companies that allow best of bread solutions attempt to simplify this process by providing development environments and management tools.

[2] The software companies and descriptions of their products can be found respectively at www.oberon.com, www.vitria.com, www.crossworlds.com.

In all cases, however, the interactions between participants and tools are explicitly and globally codified in a top-down manner by outside experts. At best, this slows the organization's ability to change rapidly or, at worst, freezes its structure and tools because the cost and complexity of redefining the global interaction model is prohibitive. Thus, it is critical to consider how the interaction model is created. There are a number of powerful integrated simulation environments available (examples include commercial rule-based systems such as ICAD or blackboard systems such as GBB (KTI, 1999), and academic systems such as Design Sheet (Reddy and Fertig, 1996) but they also face this problem when applied to larger systems that need to evolve rapidly. Ideally engineers, designers, or other users could flexibly create and reconfigure system interactions as needed within their own scope of responsibility.

The need to rapidly adapt systems to a changing marketplace is widely recognized. If system models cannot be updated in response to a changing environment, they cannot be relied upon and will therefore go unused. A primary difficulty in larger systems is the ability to respond quickly to structural changes. In the product development environment, this is particularly critical in the early design stages. The intention of the design service marketplace vision is to produce integrated systems with the management qualities of a centralized system, but the responsiveness of a locally autonomous system where each participant defines behavior and their relationships with others. Somewhat analogously, distributed database architectures are increasingly referred to as "federated"–they inter-operate at a global level, while maintaining control at a local level.

With this key goal and observations from interactions with industrial partners and other researchers in mind, the requirements for the design service marketplace concept were identified as follows (Abrahamson et al, 1999a).

1. Information system architectures that allow distributed users in different environments to participate, without investing significant training time into systems, tools, or modeling languages outside of their own expertise
2. Systems that allow each design participant to use the tools, representations, simulations, heuristics, or models which are most suitable within their domain
3. Flexibility to allow for the spontaneous and robust growth, extension, change, revision and reuse of integrated models, tools, or resources to solve evolving or new problems
4. Incorporation of a seamless mix of detailed models and incomplete or approximate models to support both top down and bottom up design
5. Accommodation of both tight and loose collaboration, ranging from close colleagues to customers or supplier, while respecting a diverse set of intellectual property and synchronization needs
6. Provide the ability to explore a design solution space, elicit trade-offs between participants and goals, and monitor design evolution in both a manual and an automated fashion
7. Methods to extract and analyze both explicit and implicit model or organizational structure, information flow, and resource behavior

Vision: an integrated design service marketplace

The integrated design service marketplace concept is embodied by three main elements. In this section each of the elements are described and illustrated through example.

The first component is providing the ability to *publish* interactive services so that they are widely accessible, addressing requirements 1 and 2 identified in the previous section. Individuals with particular product development expertise need to be given the capability to create models or model components whose services are readily accessible by others, without requiring them to have additional knowledge outside of their traditional domain specific tools.

The publishing mechanism should be similar to web publishing. If one is creating a document in a word processor, one can save it as HTML and then place it on a server for access by others with appropriate permissions. Likewise, model service publishers will enable users to define interfaces to their modeling capabilities. This concept can be demonstrated using the MIT CADlab's DOME software prototype in application to a Ford door moveable glass system (MGS). The MGS, shown within a door in figure 1, includes the glass, seals and the door mechanisms to raise and lower the glass.

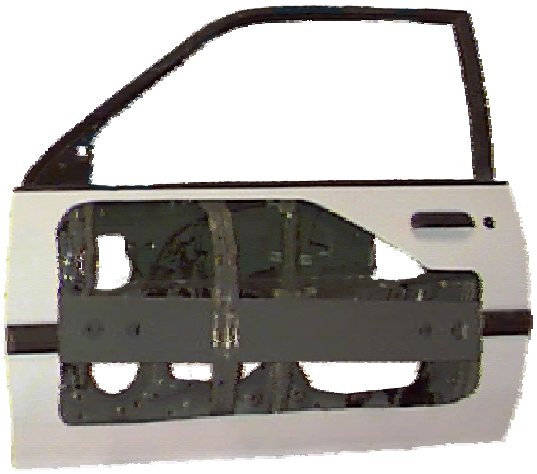
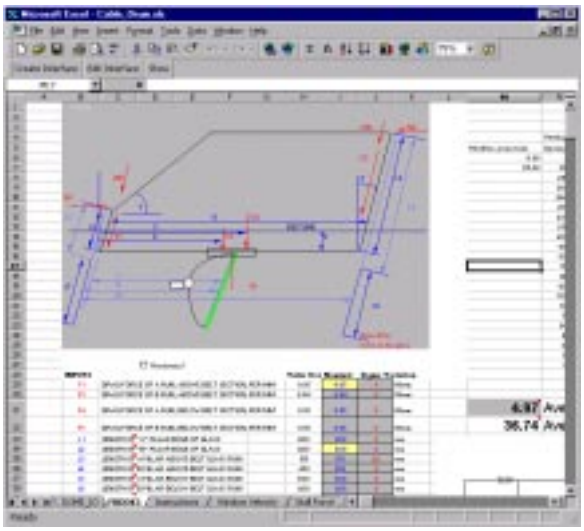
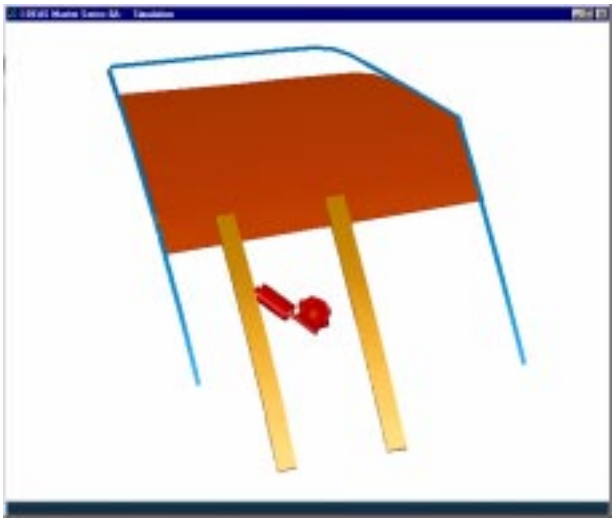


Figure 1. An example showing components of a Ford Escort moveable glass system (MGS).

Figures 2a and 2b provide screen images of two different models created by engineers and CAD designers at Ford. The Excel spreadsheet model requires door parameters to predict the glass velocity and stall force while the SDRC I-deas geometric model provides the overall physical configuration.



(a) Excel model to predict glass velocity and stall force.



(b) Geometric model of an MGS configuration

Figure 2: Models created by engineers and CAD designers for subsets of the MGS.

Model owners use simple DOME publishing programs to define interfaces that will mediate how other users will interact with their models. A publisher is a standalone program or macro specific to a third party application. It allows users to transparently create metadata defining desired service interfaces for their models and the types of DOME objects that will embody these services. For example, in Excel a wizard-like publishing macro is used to select cells and define model inputs and simulation outputs, while in I-deas special suffixes are added to dimensions, part names, or analysis capabilities. Then, model owners use a web browser to log into a DOME model server and use special wrapper objects to make their published services available over the Internet. A wrapper is an object written as a software plug-in to DOME for third party applications. It interprets the metadata generated by a corresponding publisher to create a front-end service interface constructed using standardized DOME objects (e.g., objects providing: real number services; matrix services; function services; probability distribution services; ASCII services; video services; VRML services; file services; or higher level object services). It also manages the back-end communication between these DOME objects and the third party application. This is illustrated in figures 3 and 4.

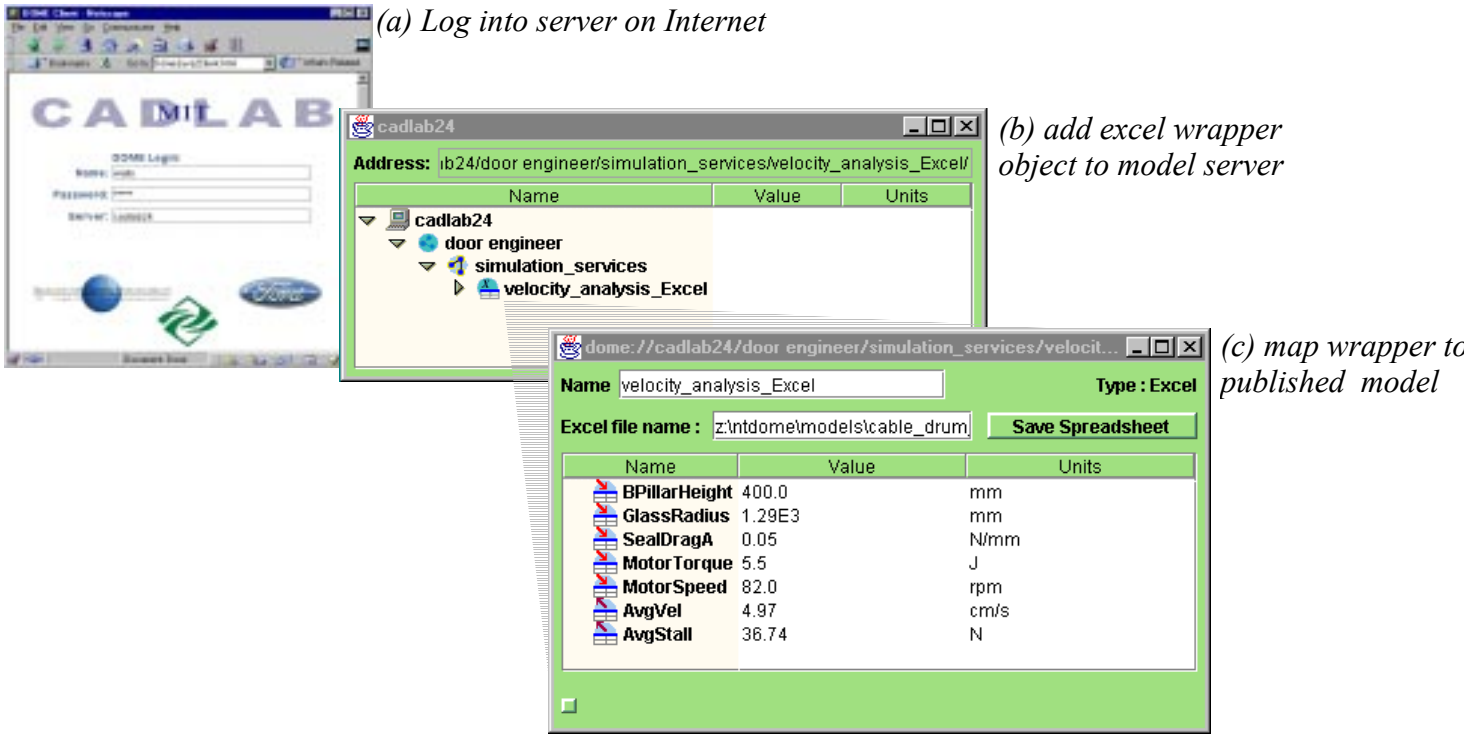
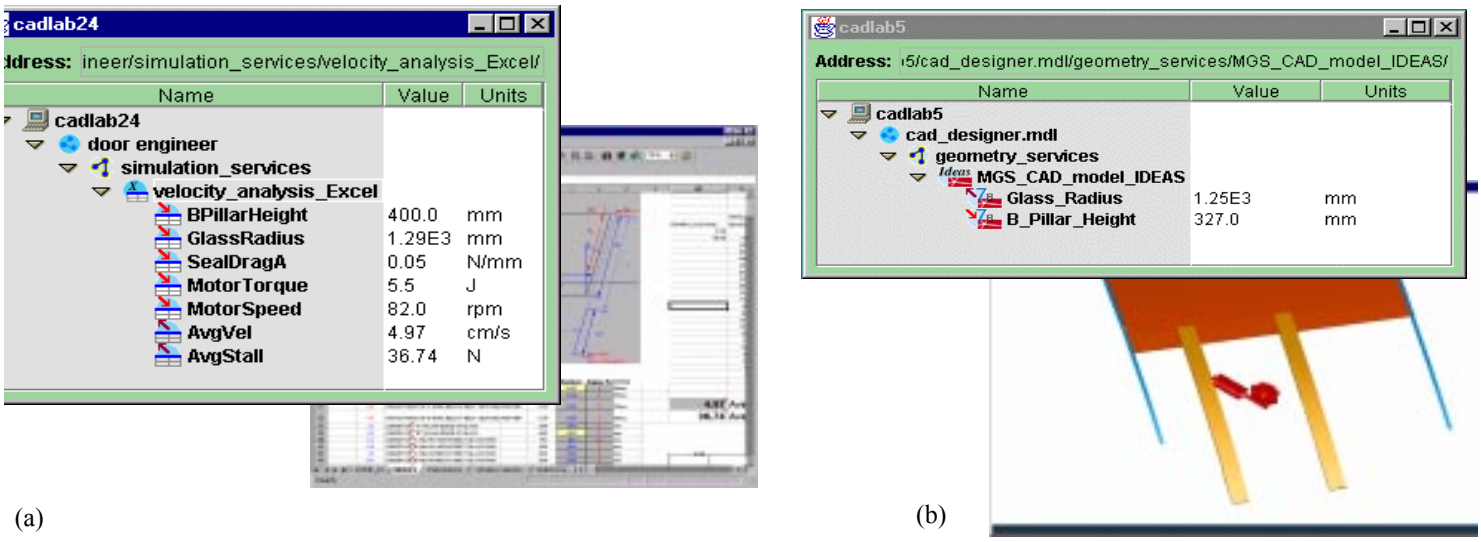


Figure 3. Adding an Excel wrapper object within a DOME model.

In figure 3 the engineer has logged into DOME server on host machine (cadlab24) and has added an Excel wrapper object. The interface for the Excel wrapper is open and it is being mapped to the published Excel model.



(a) (b)

Figure 4. Wrapper objects for Excel (a) mapped to the engineer's published spreadsheet and for a geometric modeling system (b) mapped to the published CAD designer's MGS model. The live modeling services are being offered through different DOME servers (CADlab24 and CADlab5).

Figure 4 shows two different DOME servers, one providing the door engineer's velocity analysis services (a), and the other providing the CAD designer's geometric modeling services (b). Within the *velocity analysis Excel* object there are other real number service objects that provide a web-accessible interface to the engineer's published model. For example, the BPillar height and glass radius are inputs that can be changed and the AveVel and AveStall are outputs based upon these inputs. Any design participant with permission to access this DOME server can now use this interface to change inputs to the engineer's model, which will drive the underlying Excel model and return corresponding predictive services for velocity and stall force. In figure 4b, a simple interface to the published geometric model is shown. When a user changes the pillar height the underlying MGS geometric model will rebuild automatically and then provide a new glass radius. The open interface of the CAD software allows the DOME wrapper to control the MGS model in this manner. In both cases, it is key that the process of using publishers to define the service interfaces and make the services available on a DOME server required about two minutes of the engineer's and CAD designer's time—with the ultimate goal of requiring no special programming expertise. The authors believe that in the future product development tools will support this type of functionality.

Once a model is published, other users can interact with the model through its service interface, just as most object-oriented programming languages selectively expose certain attributes and methods while protecting inner workings. DOME objects (also called modules) facilitate the object-oriented structuring of design modeling services. In this way each design participant can use their modeling tool of choice without requiring others to do the same. Any tool can be used to create simulation or model content. Publishing applications then allow interfaces to be created so that model services can be

accessed and operated through a web-browser. Design participants use publishers to transparently construct service interfaces for their models using a collection of standardized DOME objects. Participants are free to include any services they feel are appropriate for their model. While software programmers write the publishers, wrappers and standardized DOME objects, design participants compose service interfaces for their models without any software programming.

The second component is to create a mechanism for subscribing to published services and integrating them to build system models, addressing requirements 3, 4, and 5 from the previous section. In figure 5, a system integrator has logged into a DOME server (Cadlab14) and is beginning to define an overall MGS simulation model. The system integrator has subscribed to the engineer's and CAD designer's services by placing shortcut or alias-like service references (subscriptions) within their own model (on the Cadlab14 DOME server).

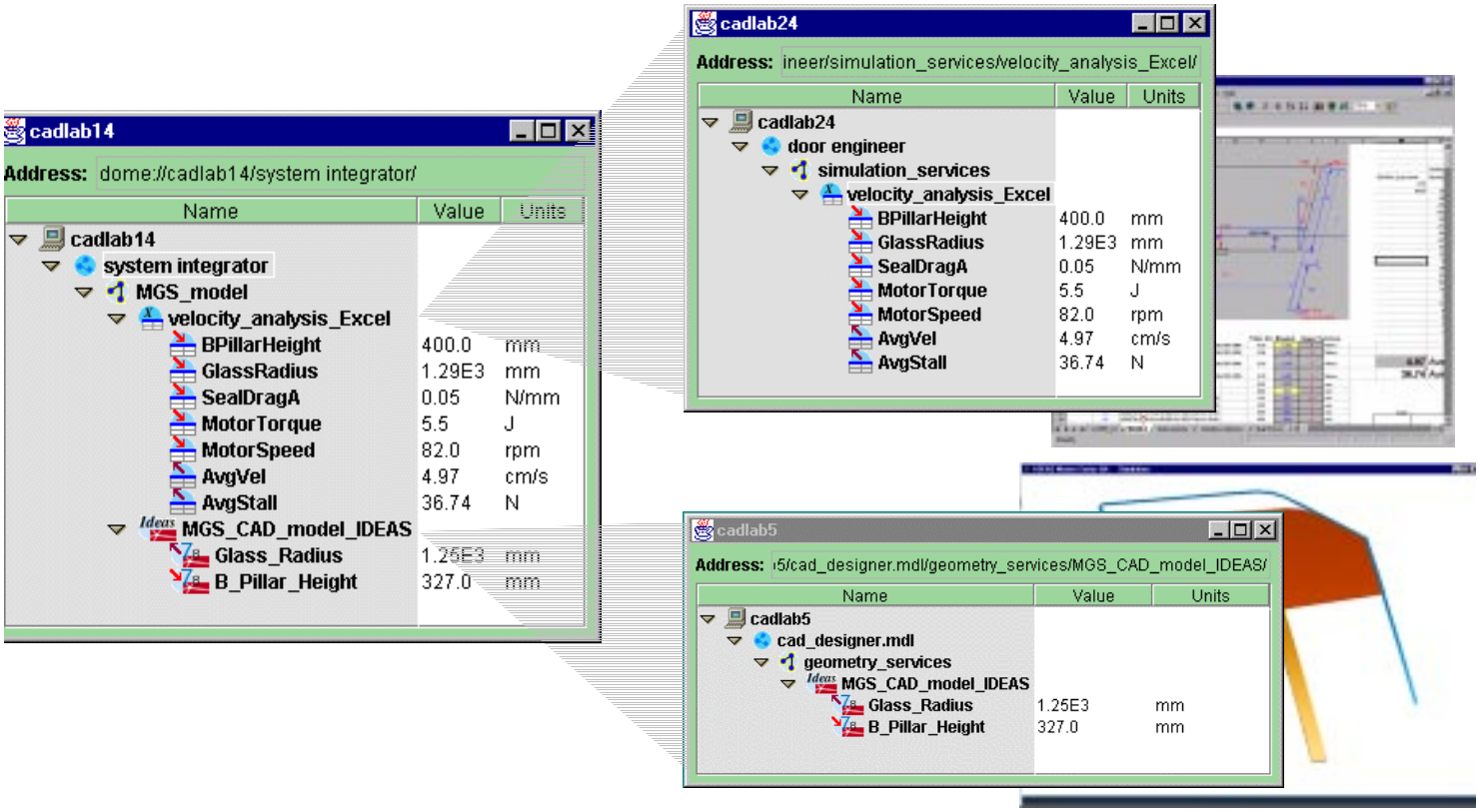


Figure 5. A system integrator subscribes to velocity analysis and MGS CAD model services by creating alias or shortcut-like references within their own DOME model.

Now, the system integrator wishes to integrate the modeling services of the engineer and CAD designer. An integration manager object is added to the MGS model that provides a synthesis environment for relating services and creating new services. Figure 6 illustrates this process.

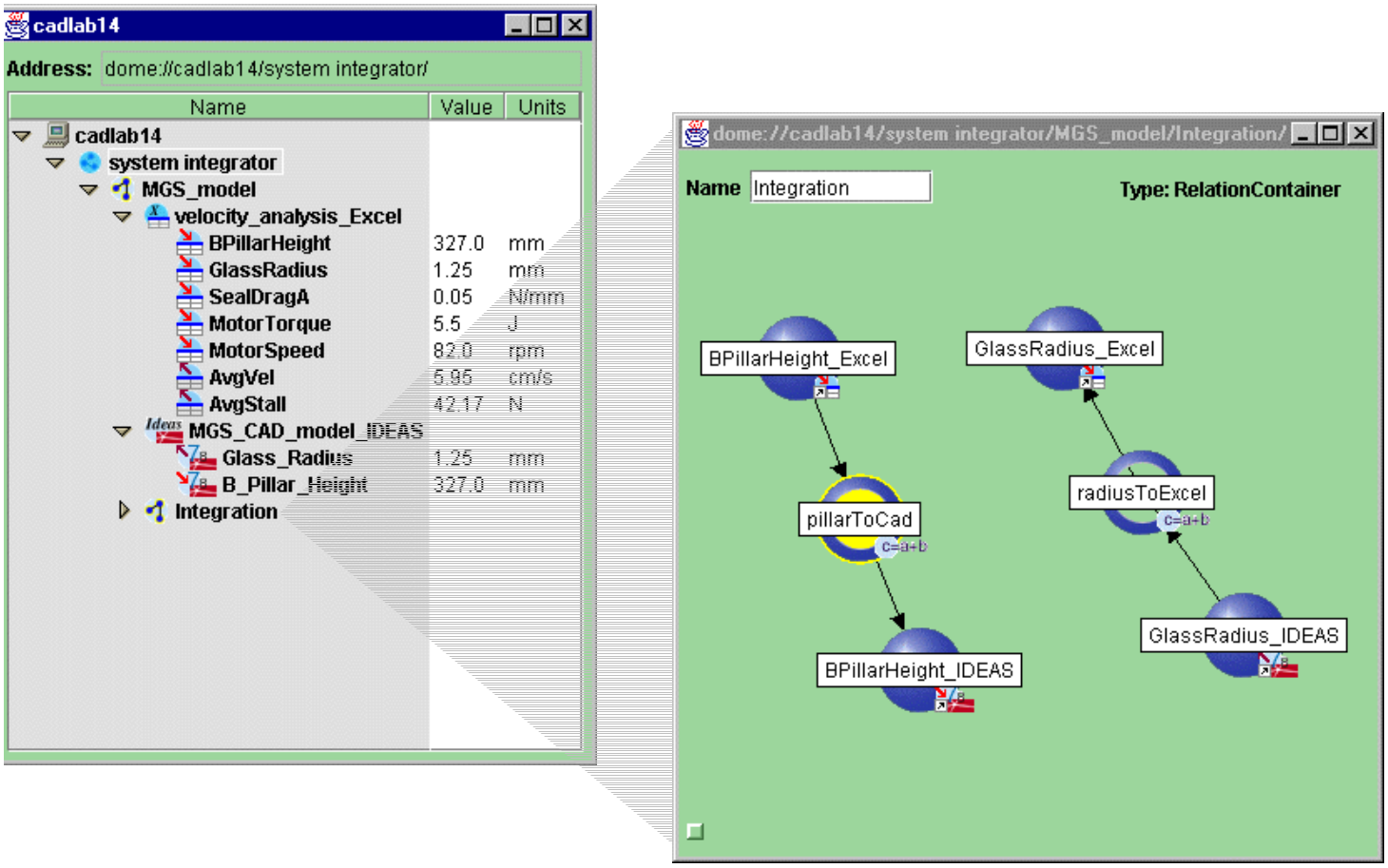


Figure 6. Adding a synthesis object (integration) and defining relationships between the engineer's and CAD designer's model to create an integrated system.

Within the integration object scope, simple equality relationships have been defined using a visual editor so that the Excel Bpillar height drives the MGS CAD model Bpillar height, and the CAD model glass radius drives the Excel glass radius. This particular integration object resolves relations according to directed graph rules. When the engineer changes the Bpillar height, the CAD designer's model will rebuild automatically and return a new glass radius, which again drives the engineer's velocity model. Relations between services can be arbitrarily complex and different integration objects will offer different types of relation coordination behaviors. A key point is that the system integrator is not required to have any special programming knowledge to coordinate the services of these distributed multiple-platform product development models.

Further, one can imagine that the system integrator could then encapsulate the integrated model (in objects called containers) and offer these services to yet other participants working on yet larger systems. Users can subscribe to these interfaces and use them in their own environment, similar to the idea of channels on the Internet that allow users to observe certain content and receive feedback when it changes. However, the fundamental difference in DOME is that the mechanism is bi-directional, and an array of actions can be taken in response to changes, from email notification to triggering a sequence of simulations.

The mechanisms for publishing, subscribing, and synthesizing relationships provide the underpinnings of a *service marketplace* for the producers and consumers of product development models and data. The marketplace should facilitate the matching of producers and consumers, and the linking and augmentation of services to create new system models. Each participant in the marketplace brings expertise and formal representations in the form of data and models, ranging from an individual offering finite element analysis to an application engineer offering a catalog of electric motor simulations. System integrators are able to flexibly define and alter relationships between different services derived from different software applications and residing on different DOME servers on the fly, without hard coding software connections.

The third and final component of the concept, addressing requirements 6 and 7 in the previous section, is the introduction of tools to support the interrogation and management of system models. Although the mechanisms for connecting and integrating models have been demonstrated, a number of issues remain. For example, how does one understand the structure or behavior of the resulting models? How does one select solutions once a large number of options become available? What are the mechanisms that drive publishing and subscribing in a marketplace?

For illustration purposes, a few examples from a multitude of possible tools to answer such questions are provided here. Figure 7 shows the DOME MGS model expanded to include data services coming from different seal suppliers. The integrator defines service relationships between the interface of the catalog and other model services. The catalog interface manages the back-side mappings to the chosen supplier so that the system integrator can substitute seals without redefining model relationships. The catalog interface takes on the service values of the selected seal supplier.

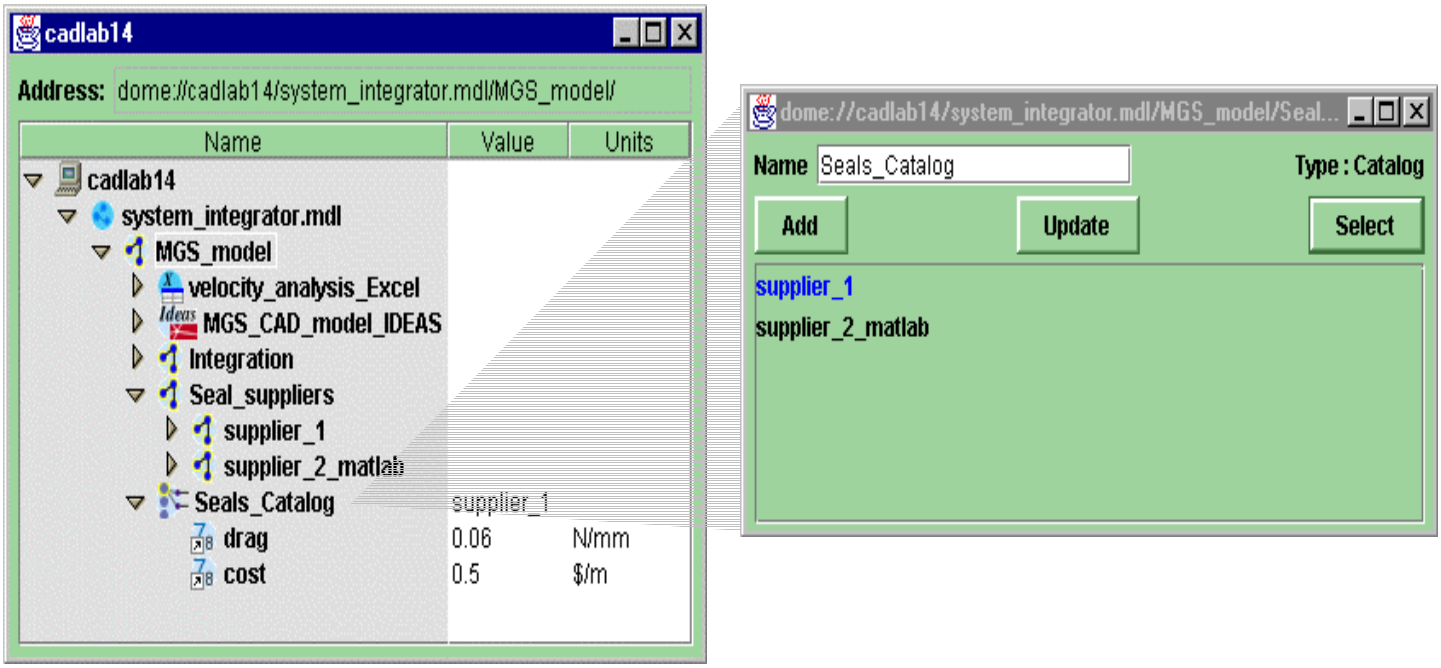


Figure 7: Catalog objects have been added to provide the service of switching between different models (in this case at different seal suppliers).

In figure 8, the services of a decision support object have been added (Kim, 1999; Kim and Wallace, 1997). This provides the integrator with a real-time view of system-wide tradeoffs as different participants make local design changes. A spider diagram shows performance assessments on four axes, while the expanded detail window shows the glass velocity prediction relative to its design specification. Additionally, an object has been developed using the Design Structure Matrix (Smith and Eppinger, 1997; Steward, 1981) to visualize service interactions as they evolve (Abrahamson *et al*., 1999b). A genetic optimization object has also been implemented to automatically search for models states that best meet design goals (Senin *et al*., 1999a).

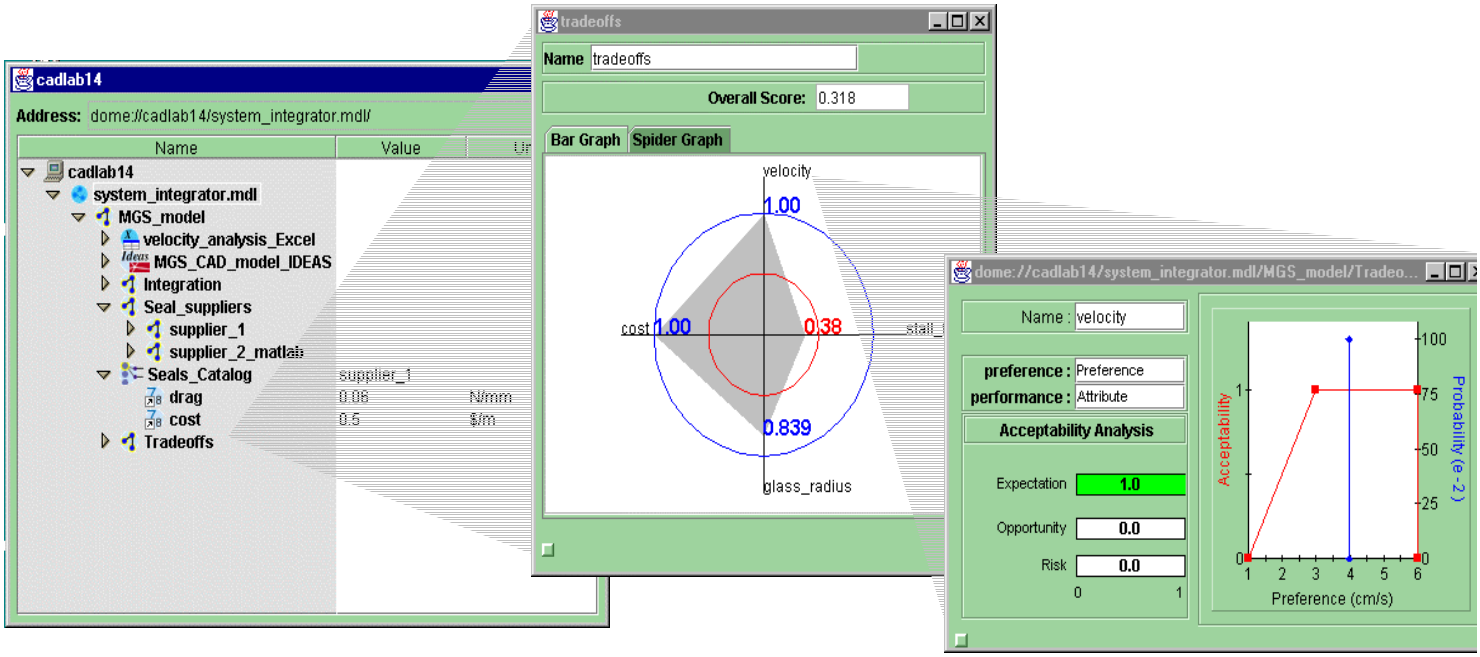


Figure 8. The services of a decision support object are added to visual system-wide design tradeoffs.

Software prototype

A software prototype called DOME has been developed in the MIT CADlab to test the design service marketplace. Early work on the concept is described in work by Pahng (Pahng *et al*, 1998) while the underlying object formalism is described in work by Senin and Wallace (Senin *et al.*, 1999b). There are objects that provide standardized DOME services corresponding to data types ranging from engineering quantities to files. Relation objects define service relationships between other objects, and manager objects coordinate the firing of relation objects within their scope. An application program interface (API) is provided so that third party software wrapper objects plug into the system. Wrapper objects provide a mapping between the interface of proprietary software and standardized DOME service objects. There are numerous objects (written as plug-ins) to support design exploration: selection of services from catalogs; decision or tradeoff analysis; model structure analysis; and model optimization. Finally, the publishing concept and mechanisms are detailed in work by Borland (Borland *et al.*, 1999. Borland *et al.,* 1998). The structure of the prototype system used for concept testing is illustrated in figure 9. This software prototype was used to produce the examples provided in the previous section.

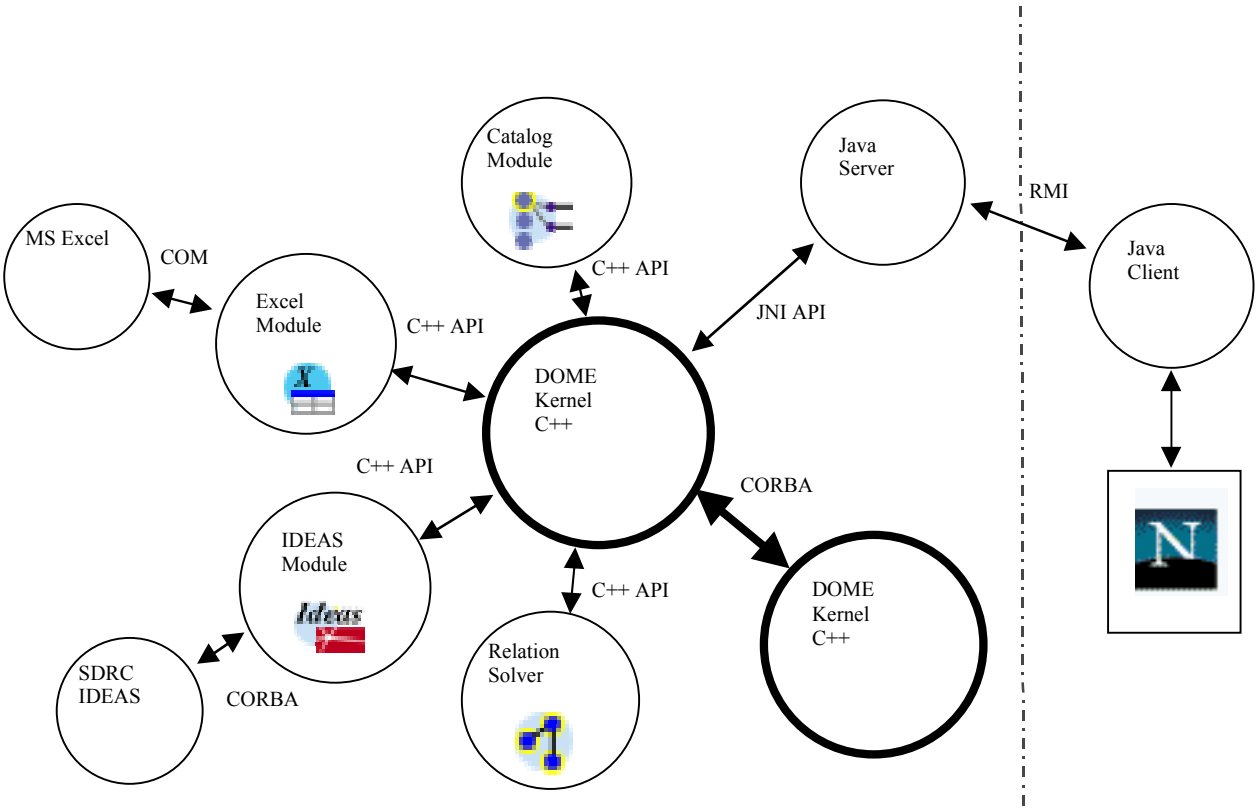


Figure 9: Architecture of the DOME prototype system.

The main elements are: the DOME server kernel that provides the underlying mechanisms for creating and interconnecting DOME models; and a variety of objects that plug into this kernel, providing links to external applications such as Excel and I-deas or particular functions such as catalog selection or model integration capabilities. Finally a browser-based client provides a graphical building and evaluation environment.

Concept Testing

The MIT CADlab DOME prototype has been tested for a number of feasibility pilot studies, including an in-lab study conducted with Polaroid using a preliminary version of the software (Abrahamson *et al*., 1999a). A more ambitious research and development pilot study is being completed[3] at Ford Motor Company and selected suppliers for a subset of the MGS system (the same system that was described in a highly simplified form in the vision section). Six different organizations participated in the pilot study with their own DOME servers, as illustrated in Figure 10. The Research Vehicle Technology (RVT) engineering scope included door channel sheet metal design, overall MGS configuration parameters, integrated performance simulation, and overall technical design assessment relative to system specifications. The RVT Purchasing scope included the definition of supplier target cost agreements and overall MGS system costing, while Commodities Management has the strategic responsibility of recommending suppliers to RVT Engineering. The two competing seal suppliers are responsible for seal geometry,

[3] Project is scheduled for final deployment in the fall of 1999

seal performance analysis (2D FEA and high speed approximate models) and seal costing. The engineering analysis firm provides 3D FEA services for the door seals.

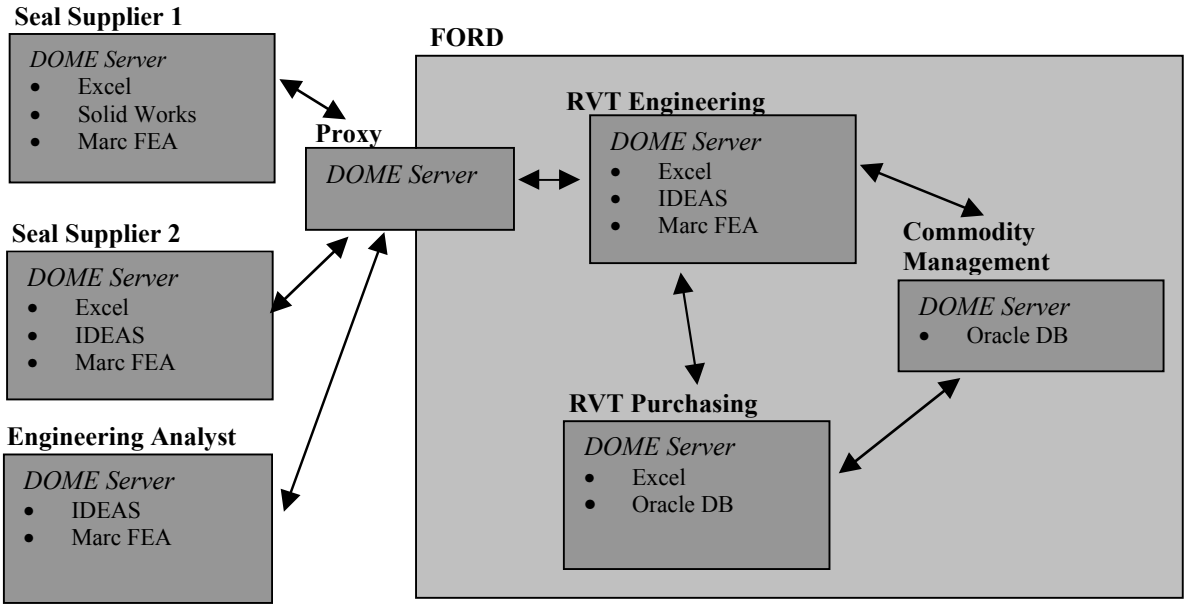


Figure 10 Organizations siting DOME servers in the Ford Pilot Study and applications used for the pilot study within each organization.

Two members of the MIT CADlab spent two months with Ford and its suppliers working to define interfaces, improve models, and publish DOME interfaces for 21 different third party models sited at the different organizations. These applications included Microsoft Excel, SDRC I-deas, Solidworks, MSC MARC, and Oracle. A total of one person month was required to build local DOME models integrating and coordinating model services between different applications within each organization, and then to define service interfaces and DOME relationships between external organizations. A total of roughly 1500 wrapper objects, real number service objects, relation objects, and decision support objects were distributed between the six organizations.

The pilot project is being used to validate and benchmark several elements of the design service marketplace concept. First, by working directly with engineers and suppliers it was possible to assess that they are receptive to the idea of publishing, yet still controlling the models underlying their design services. The pilot project also demonstrated the use of publication and subscription concepts to integrate model services from several different applications, including parameter transfer between databases, CAD systems, and spreadsheets.

The relatively small amount of time required for system integration is interpreted as an indication that it will be possible to rapidly create product development models and organizations if the design service marketplace is fully implemented.

The integrated system model illustrates how each participant in the MGS system can independently operate within its own local modeling scope and rapidly see the implications of changes and the changes of others on overall system performance. For example, door and channel dimensions at RVT Engineering are automatically propagated to the seal supplier currently chosen by Commodities Management (through a catalog selection). The seal geometry design is changed by the supplier which in turn alters its technical performance FEA simulations (of interest to RVT engineering) and the seal cost (of interest to the seal supplier and RVT Purchasing). Each organization is provided with decision support in its DOME model that allows a simultaneous visualization of how all participants view the quality of a given design model state. A fully integrated simultaneous design cycle requires roughly 20 seconds to 1 day (depending on the type of changes and quality of analysis), in comparison to a week to three months as is required in the traditional design cycle. Such cycles typically occur nearly 20 times, but this new product development concept presents opportunities for improved systemic design by allowing many more improvement cycles while still reducing product development time.

At the same time, security and firewall requirements can still be satisfied. Service exchanges over the Internet can be encrypted but, as this particular study involved only the exchange of parameters without model descriptions, this was not a major issue. Additionally, user access control is needed at the DOME model server level and for the individual services provided by each published model. The scope of the pilot did not require the strict management of model structure and states, but clearly this is necessary. In the future, interactions with PDM (Product Data Management) and PIM (Product Information Management) systems are envisioned for this purpose.

Conclusions and future research

An integrated modeling service marketplace concept for product development and a prototype implementation called DOME have been introduced. In this environment, digital representations of products and associated organizational infrastructure such as design, manufacturing, marketing, distribution, and support can be readily evaluated against corporate, customer and market, and policy objectives. By providing means for integrating software models, DOME allows the prediction of overall product and enterprise performance during development phases, thus reducing the number of design-build-test cycles, and allowing a higher number of alternatives to be explored at a minor cost—while still offering the potential to significantly reduce the time-to-market.

The service marketplace concept involves three primary elements: providing design participants with the ability to publish live services as part of their natural workflow; creating a service market that allows system integrators (rather than software programmers) to seamlessly inter-relate product development services across many different software applications and model servers; and the introduction of tools that facilitate the rapid elucidation and evaluation of tradeoffs. In this vision, design software applications and tools will become focused on allowing their users to publish live design services.

A moveable glass pilot project with Ford Motor Company indicates that the service marketplace concept offers the potential to reduce the time needed to build integrated system models and subsequently consider many alternatives to identify suitable tradeoffs.

Several areas of research related to the service marketplace concept remain. Research is needed to develop an understanding of templates for typical organizational structures and service acquisition strategies. Interface standards for different types of product development services are needed if they are to become highly interchangeable commodities. Versioning and causality analysis is complex in a distributed environment where global system knowledge usually will not exist. New voice-of-customer techniques may be possible through the observation of how people explore and configure model services. Agents could be employed to facilitate the location of suitable modeling services, or perhaps eventually automate portions of the system integration process. This would speed integration, reuse, and access to model content. There is a growing body of knowledge on ontologies and search techniques that could be applied to this problem. Surrogate modeling techniques might be applied to speed the response time of computationally heavy services to facilitate quick tradeoff exploration at the cost of lower accuracy. Finally, there is a need to develop representations of service quality that can be propagated along service chains.

Acknowledgments

The authors are grateful for the support provided by the National Science Foundation Center for Innovation in Product Development (CIPD), grant number EEC-9529140, the Ford Motor Company, and the Alliance for Global Sustainability. Additionally, several researchers in the MIT CADlab, including Nick Borland and Francis Pahng, have contributed to the development of the integrated design service marketplace concept. Jeff Lyons, Priscilla Wang, and Bill Liteplo of the MIT CADlab, and Bob Humphrey and Darrel Klienke of Ford were valuable contributors to the MGS study.

References

Alberthal, L (1998). The Once and Future Craftsman Culture. *The future of the Electronic Marketplace*. Leebaert, E Editor , MIT Press,. p37-62.

Shaun Abrahamson, David Wallace, Nicholas Borland and Nicola Senin. (1999a) "Integrated Engineering, Geometric and Customer Modeling: LCD Projector Design Case Study", *Proceedings of the ASME DT Conferences*, DETC/CIE-8780, Las Vegas, NV.

Shaun Abrahamson, David Wallace, Nicholas Borland. (1999b) "Design Process Elicitation Through the Evaluation of Integrated Model Structures", *Proceedings of the ASME DT Conferences*, DETC/DFM-8780, Las Vegas, NV.

Nicholas Borland, Heiner Kauffman, David Wallace. (1998) "Integrating Environmental Impact Assessment into Product Design: A collaborative modeling approach", *Proceedings of the ASME DT Conferences*, DETC/DFM-5730, Atlanta, GA.

Nicholas Borland, David Wallace. (1999) Environmentally-Conscious Product Design: a Collaborative Internet-Based Modeling Approach, to appear in *the Journal of Industrial Ecology*.

CIPD (1999). Center for Innovation In Product Development http://web.mit.edu/cipd/

Robin Cooper and Robert S. Kaplan. (1998) The promise—and peril—of integrated cost systems, *Harvard Business Review*, July-August, pp. 109-119.

Cutkosky M, Olsen G.R, Tenenbaum J.M, Gruber T,R (1994)"Collaborative Engineering Based on Knowledge Sharing Agreements*" Proceedings of the 1994 Database Symposium*.

Jun-beom Kim, David Wallace. (1997) "A Goal-oriented Design Evaluation Model", *Proceedings of the ASME DT Conferences*, 97-DETC/DTM-3878, Sacramento, CA.

Jun Beom Kim. (1999) *A Goal-oriented Design Evaluation Framework for Decision Making Under Uncertainty*, PhD Thesis, Department of Mechanical Engineering, MIT.

KTI (1999). Knowledge Technologies International http://www.ktiworld.com and http://www.bbtech.com

Francis Pahng, Nicola Senin, David Wallace. (1998) Distributed object-based modeling and evaluation of design problems, *Computer-aided Design*, volume 30, number 6, pp. 411-423.

Reddy, S. Y. and K. W. Fertig. (1996) Design Sheet: A System for Exploring Design Space, Application to Automotive Drive Train Life Analysis. *Fourth International Conference on Artificial Intelligence in Design* (AID'96), Stanford, CA.

Nicola Senin, David Wallace, Nicholas Borland. (1999a) "Object-based Design Modeling and Optimization with Genetic Algorithms", GECCO-99*: Proceedings of the Genetic and Evolutionary Computation Conference*, July 13-17, Orlando, FL.

Nicola Senin, David Wallace, Nicholas Borland. (1999b) Distributed Object-based Modeling of Design Problems, under review by the *ASME Journal of Mechanical Design*.

Smith, R. P. and S. Eppinger (1997). "Identifying Controlling Features of Engineering Design Iteration." *Management Science,* 43(3).

Steward, D. V. (1981). "The design structure system: a method for managing the design of complex systems." *IEEE Transactions on Engineering Management* EM-28(3): 71-74.

Ulrich, E. (1995). *Product Design and Development*, McGraw-Hill, Inc.

Wellman, M. (1994). A Computational Market Model for Distributed Configuration Design. *12th National Conference on Artificial Intelligence*, Menlo Park, CA, AAAI Press.