**Web-Based Knowledge Management for Distributed Design**

Nicholas H.M. Caldwell and P. John Clarkson, University of Cambridge
Paul A. Rodgers, Napier University
Avon P. Huxor, Middlesex University

**Contemporary Product Design** is a knowledge-intensive process undertaken by virtual teams dispersed across multiple organizations. Successful design requires contextual knowledge of the target market. Knowledge management and decision support can assist in this task. By considering AI as a form of writing, we can represent knowledge in a transparent and modifiable form, which better supports the multiplicity of contexts required in real design.

To assist designers in conceptual-design evaluation, we have implemented the WebCADET Web-based decision-support tool. Based on a knowledge-server architecture, WebCADET operates in design-guidance, knowledge-viewing, and knowledge-capture modes. This article discusses our efforts in developing WebCADET and deploying four WebCADET servers at different locations for various applications.

**Evolution of Design**

In recent years, responsibility for new product design and development has shifted from a single designer in a single organization, through group design work within a single organization, to multiple designers combined into virtual multidisciplinary teams dispersed across several organizations (see Figure 1).

There have been two key reasons for this shift:

- Humans have confined the scope of their knowledge (*specialization*) to go in depth in a particular subject.
- Humans rely on other specialists (*delegation*), computational agents, and reference materials to enhance their cognitive abilities.¹

At the same time, greater global competition is pressuring organizations to produce higher-quality product levels for specific markets in much shorter timescales.

Successful new product design, especially the development of consumer products, depends largely upon the target market’s context. This context can form a set of constraints that new products must meet. Such constraints might include a product’s aesthetics (the associations of various colors differ from culture to culture), its ergonomics (human physiognomy is not constant), its reliability (excessively hot, cold, or humid regions might demand different tolerances), and so forth. In new product design and development, no universal solution is likely for any specific design problem.

An effective concept solution that meets the needs, or attributes, of the target market is critical for a successful new product design and development process. For example, this solution might include attributes such as
attractive, comfortable to grip, durable, or hygienic. Time invested in determining these requirements and analyzing proposed designs against them during the early stage of the design process usually helps produce a better final product for the target market.

However, effective conceptual design evaluation requires large amounts of knowledge to identify desirable attributes and determine which values of design parameters—length, width, material, texture, color, or shape, for example—result in an attribute being achieved or not. We can view this knowledge as an organizational asset that generally resides within the memories of individual designers, textbooks, corporate libraries of past products, the scientific literature, and online resources. Thus, finding the right knowledge at the right time and in the right format is a time-consuming process for designers.

There is a need, therefore, to support designers during their concept evaluation tasks by supplying knowledge to better guide them through the design process. Effective knowledge-management support will alleviate much of the time-consuming searching designers face when trying to find the right information and free them to concentrate on more demanding tasks during the design process. Moreover, there is a further need to support the design organization through preserving and managing the knowledge gained in past and current designs so that such knowledge assets remain accessible to the organization. (See the “Related work” sidebar for a discussion of work that has gone on elsewhere in this arena.)

**Methodology**

User acceptance has been a major stumbling block in the deployment of traditional expert and knowledge-based systems in many organizations. In the context of design, failures in user acceptance of a number of systems have resulted from the emphasis on attempting to emulate or replace the human designer (such as by performing semi automatic design), rather than on supporting them in a particular design task.

Rather than viewing an AI system as an autonomous agent, we have adopted an “AI as text” paradigm. In this approach, we can view the knowledge in a knowledge-based system as a form of writing, with the knowledge-representation formalism and knowledge models forming a grammar and vocabulary to produce consistent “texts” (which might be rules or cases, for example). As a text, such knowledge is closer to an organization’s major sources of explicit knowledge such as standards, design rationale documents, or design process checklists, and places the AI system firmly within an organizational memory. As a text, the knowledge gains the attribute of transparency: users (“readers”) can see each text’s nature, relevance, and origin—the knowledge itself, the context wherein the knowledge is applicable, and who “authored” the knowledge. The knowledge also gains the attribute of modifiability: readers can amend extant texts such that the “edited” (revised) texts better suit the needs of the amending author’s target context. This inherent modifiability makes the “AI as text” approach effective in context-dependent design domains because new contexts will often require new editions or amendments of existing rules.

In earlier work, researchers implemented the CADET (Computer-Aided Design Evaluation Tool) system as a standalone system. This research provided proof-of-concept and validated the initial knowledge base for certain specific classes of consumer products. As a standalone system, users could not easily update the original CADET without incurring distribution overheads of transferring updated knowledge bases to and from user sites. Moreover, CADET did not possess any mechanisms to exploit network connectivity and as such could not be effectively deployed in modern distributed design teams.

Our solution has been to reimplement CADET in a Web-based form as WebCADET using the knowledge-server architecture. By locating the knowledge base and inference engine on a networked server, knowledge updates and system maintenance need only be applied to the server, significantly minimizing long-term overheads. As a knowledge server, WebCADET can fully exploit intranet and Internet technologies, rendering the system practical for distributed design teams. Furthermore, the Internet’s ubiquity has rendered Web-based documentation and forms a familiar medium to most professionals, avoiding the need for potential WebCADET users to comprehend a completely unknown interface to interact with the system.

**System overview**

The knowledge-server architecture adopted in WebCADET (Figure 2) locates the inference engine and knowledge base on a conventional Web server.

For efficiency, the knowledge base resides in a compiled format but is distinct from the inference engine to preserve modularity. The user interface consists of pure HTML for input forms and results presentation, with some minimal Javascript to provide immediate user assistance.

Its developers used Mac-Proslog and the Flex expert system toolkit (both from Logic Programming Associates) to implement the original CADET standalone system. We reimplemented the CADET standalone system using Win-Proslog to produce the inference engine and the ProWeb toolkit (also from Logic Programming Associates) to provide the Web-based user interface, so that WebCADET could be deployed on a Web server. The ProWeb toolkit currently only supports the Windows ‘95/’98/NT operating system environment, so WebCADET servers can only reside on PC platforms. However, WebCADET users need only possess a suitable graphical browser with forms support to access the server from the platform of their choice. At this end, WebCADET is truly client independent.
Related work

Over the last four to five years, the Internet, and in particular the World Wide Web, has let designers carry out a range of design tasks more effectively throughout the new product development process. A number of recent projects have proposed techniques that will support designers in their communication and collaboration activities. One project developed a framework for a collaborative product development system that offers an environment for visual presentation, interactive refinement of product specifications on VRML-based sharable CAD models, a Web-integrated product design-data management system for supporting CPD tasks, and a Web-based manufacturability analysis tool. The WebShaman project demonstrates how a distributed “smart” virtual prototyping system could support a geographically distant designer–customer in the product development process of small electronic devices.

A number of Web-based systems have been developed to support specific design tasks later in the product development process, such as during the product realization process. For instance, recent advances in client–server interaction, computer automated-process planning (CAPP), and rapid manufacturing services have made it possible to develop a Web-based, seamless CAD/CAPP/CAM system (CyberCut). CyberCut offers designers a design-for-manufacture CAD interface, written in Java, a choice between two CAPP systems, and access to an open-architecture machine tool for fabricating mechanical parts. CyberCut is the extension of an already proven Integrated Manufacturing and Design Environment into a distributed agent environment.

The WWW has also served to provide designers with various forms of knowledge-management support, including knowledge capture, adaptation, and presentation, and knowledge sharing and reuse. The Angelo system, for example, aids designers by capturing knowledge about the designers’ individual preferences and the problems they tackle. Angelo, through proactively observing user interactions with project knowledge, adapts the knowledge and information before presenting it to the designer. Angelo acts as a support tool for collaborative design meetings where designers require relevant knowledge to avoid or resolve design issues.

VisionManager is a prototype tool for design capture, visualization, and reuse to support multidisciplinary collaborative design work, providing both a model-based and content-based approach to these needs. The Personal Electronic Notebook with Sharing tool takes a different approach to a similar problem. The researchers propose the concept of an electronic or digital engineering design notebook that designers will use to capture information for reuse and sharing. PENS is easy enough to use to compete with paper notebooks in functionality. Every time informal design notes are entered in PENS, a project information web automatically grows. As the project develops, information can be shared with collaborators over the WWW.

Finally, a Conceptual Design Information Server (CDIS) has been developed, using emerging Internet standards, such as those associated with the WWW and Wide Area Information Service (WAIS), coupled to a robust Structured Query Language (SQL) database and traditional CAD packages. This gives designers case-based conceptual design information stored in the richly expressive medium of hypermedia (hypertext incorporating multimedia). The system contains design cases of industry best practice, indexed to let users browse, explore, and locate specific design case information.

Knowledge representation

For easier navigation through the collection of texts, WebCADET employs a design hierarchy that leads the user from the general to a specific rule text. The hierarchy’s top level comprises the set of design sectors, such as consumer products, aerospace, automotive, or biomedical. Design sectors contain individual products such as mobile phones, shavers, and toothbrushes in the consumer products design sector. Below the product level comes a layer of product-design-specification elements, such as aesthetics, ergonomics, reliability, maintenance, performance, and safety. The PDS elements provide a convenient mechanism for grouping desirable product attributes. They also act as an initial focus for designers cataloging a new product class’s requirements into WebCADET. The number of attributes found under any specification element in any given product varies according to that element’s relevance to the product. For instance, under the ergonomics PDS element of the mobile phone, we have identified several attributes: comfortable_to_hold, fits_face, easy_to_dial, operable_with_one_hand, and not_too_heavy. Attributes form the WebCADET design hierarchy’s penultimate level with the actual context-dependent rule texts being grouped under the attributes.

WebCADET supports designers in concept evaluation. It is not targeted at replacing the human designer in producing a complete design in any specific product class. As a consequence, WebCADET’s knowledge consists of heuristics to evaluate specific design parameters with reference to specific design attributes. WebCADET readily represents this knowledge in a variant rule-based formalism—that is, the phone__easy_to_dial__1 rule text shown in Figure 3.

The rule_id slot uniquely identifies the rule text with reference to product and design

References

attribute. This identifier is automatically generated when WebCADET captures the rule text. The name and precondition slots identify the rule as being relevant to a given attribute and product for faster matching during concept evaluation.

The conditions slot encapsulates the heuristics pertaining to germane design parameters as if-then-else expressions, with distinct pass and fail scores recorded during rule activation depending on whether the design meets the condition’s if component. (A pair-wise comparison method defines the scores for each parameter.) The mnemonics gt, gte, eq, lte, and lt represent the standard relational and equality operators (>, >=, =, <=, <). The operator iaiof abbreviates is_an_instance_of and links a specific condition’s validity to a set of facts asserted elsewhere in the WebCADET knowledge base. The operator has_aspect links a specific condition’s truth or falsehood to a specialized internal rule encoding the physical properties of satisfactory materials. Other internal rules encode color associations for particular product images.

The scale slot simply serves as a normalizing factor to reduce a design’s accumulated score during rule activation to a number between 0 and 1.

The history slot contains the provenance of the rule text in terms of author and creation date. It also indicates why a given rule text has been included in WebCADET. Additional amendment entries would be recorded in this slot as other authors successively edited the rule text.

The keywords slot indicates if the rule text has any particular context—“generic” in this instance signals an original rule text that has not been tuned for a particular target market. Elsewhere in WebCADET, the conditions slot holds distinct explanation facts for each clause. These provide canned text justifications and resource references to the scientific literature, technical reports, or online documents.

System examples

WebCADET operates in three distinct modes: design guidance, knowledge viewing, and knowledge capture. The design-guidance mode serves for interactively evaluating a design concept. The knowledge-viewing mode is principally an inspection mode to let designers and potential rule-text authors examine knowledge base contents directly without the need to supply a hypothetical design concept. The knowledge-capture mode enables rule text authors to edit the knowledge base’s contents in various fashions.

In the design-guidance mode, a designer first loads the WebCADET server homepage into a browser and enters his or her name, organization, and email address in the relevant portions of the form and then selects the design-guidance mode. WebCADET returns a series of Web pages containing further forms that help the designer navigate down through the hierarchy from design sector, through product and specification element, to a given attribute (Figure 4a). Once the designer has selected an attribute, WebCADET determines which rules are applicable for this product and attribute combination and lets the designer choose which rule to apply. To assist the user in this choice, WebCADET uses the contents of the history and keywords slots of the respective rule texts to identify the rule’s source, the reason for creating the rule, and any specific context markers, and presents this information explicitly to the designer (Figure 4b).

Once the user has selected a rule, WebCADET responds by producing a parameter entry form that lets the designer submit the values of design parameters (considered rel-
event by this particular rule text) using a combination of text-entry fields and drop-down menus to minimize transcription errors. Parameters whose values are as yet unknown or undecided can be omitted. As an aid to the designer, pictorial representations indicate the meaning of shapes, button configurations, and so forth for a specific product. These images are deliberately schematic to prevent them from biasing designers toward specific sorts of designs irrespective of a given rule text’s validity (Figure 5a).

WebCADET then activates the rule and processes the parameter data against the heuristic conditions. Where necessary, WebCADET will compare data against fact datasets, retrieve the intended material’s physical characteristics, and match this against an internal heuristic, or retrieve a specific color’s normal associations and match these against the product of interest’s desirable aesthetic image. WebCADET normalizes the cumulative rating and returns an evaluation indicating overall score and a breakdown by design parameter to the designer (Figure 5b).

The designer might opt to obtain an explanation of the evaluation result (Figure 6), prompting WebCADET to retrieve and present the justifications and associated references. The references format broadly resembles conventional bibliographies to maintain the designer’s sense of familiarity with ordinary paper-based documents in this implementation of “AI as text.” Unlike paper-based bibliographies, the designer can immediately follow hyperlinks to associated online resources.

The knowledge-viewing mode initially gives the designer a snapshot of the current instantiated design hierarchy (Figure 7), the designer can then select specific rule texts to inspect. In both knowledge-viewing and design-guidance modes, WebCADET updates the access count if a rule text is activated or inspected, and this measures a rule text’s interest or value. WebCADET does not record who inspected or activated a specific rule text.

The knowledge-capture mode is a password-protected feature of WebCADET. The rationale is twofold:

- to restrict authorship of rule texts to potential users who have received training in the operation of WebCADET and
- to restrict authorship to competent designers so that contributed rule texts have design value.
For ease of use, the knowledge-capture mode comprises four submodes, handling color and materials database modifications, design hierarchy manipulation, and rule-text creation. Access to the color database lets the designer add and remove colors and set the associations (such as alluring, conservative, healthy, pastoral, or stylish, for example) for chosen colors. The initial database uses a standard color scheme. Access to the materials database lets the designer add and remove metal and thermoplastic materials or modify the properties of existing materials (such changes should be rare in normal WebCADET usage).

In the construction submode, the designer can add or delete design sectors, products, specification elements, and attributes to specialize a WebCADET server for some design domain of interest. Before any rule texts can be created, the designer must first have added the product and at least one attribute to the server’s knowledge base. Deleting attributes or products is also the only way of removing rule texts from a WebCADET server. The justification here is that while an individual designer might contest a rule text or a rule text might be inapplicable for one context, it might still retain validity in other contexts.

In the rule construction submode, the designer first chooses a product and attribute of interest and might then create a wholly new rule text or amend an existing rule text. In each case, WebCADET generates a new rule_id based on the product, attribute, and number of existing rule texts for that combination and preloads a clause capture form with any previously supplied parameter names, dataset names, property names, and citations (Figure 8).

The aim here is to assist the designer in producing rule texts consistent in terminology with prior submissions without biasing the designer toward previous experiences. When amending an existing rule text, all data-entry fields are prefilled with the contents of the existing rule text, and the designer can choose to delete the entire clause. The designer provides WebCADET with not only an individual item for the conditions list but also the supporting justifications and associated bibliographic references. As rule-text capture proceeds, WebCADET displays acquired clauses as part of later capture forms.

Once all the primary clauses have been captured, WebCADET then switches over into capturing any new data sets and material or color rules indicated by the presence of previously unused dataset and property identifiers in clauses (Figure 9).

After WebCADET has captured all the associated rules, the designer supplies a reason for creating or amending the rule text, and the complete set of changes is then committed to the server and available to other WebCADET users.

### WebCADET applications

Several distinct WebCADET servers exist or are under construction. The first server, at the University of Cambridge (hibernia.eng.cam.ac.uk/proweb/proweb.exe?home=pa89100534), serves principally for testing new WebCADET functionality. The design-guidance and knowledge-viewing modes on this server are available for general access.

---

**WebCADET applications**

Several distinct WebCADET servers exist or are under construction. The first server, at the University of Cambridge (hibernia.eng.cam.ac.uk/proweb/proweb.exe?home=pa89100534), serves principally for testing new WebCADET functionality. The design-guidance and knowledge-viewing modes on this server are available for general access.
We are deploying a WebCADET server at the University of Middlesex, primarily for use as a repository for designer guidance in the domain of virtual environments and to determine whether heuristics from this disparate domain can be structured in the WebCADET formalism. Currently, we are still populating this server’s knowledge base with an initial collection of rule texts.

The Engineering Design Centre at the University of Cambridge and Hong Kong Polytechnic University are working on a joint research project to explore issues and evaluate computer-based support tools in distributed design. The objective is to design and prototype a novel mobile telephone product. This project uses Prosus—a process-based support system that acts both as a passive medium for capturing and structuring data produced from design activities and as an active workbench to create design data by activating embedded and linked design tools. A WebCADET server has been deployed at Hong Kong to serve as one of these linked tools to evaluate mobile telephone concepts.

Furthermore, a globally based materials-handling equipment manufacturer is currently deploying a WebCADET server within its corporate intranet. To date, this work has included using WebCADET in a small knowledge-management exercise to orient and train new employees in this design domain, as well as to obtain a preliminary formal user evaluation of WebCADET’s usability by this organization’s designers.

WebCADET user evaluation

We conducted a user evaluation exercise to determine the WebCADET user interface’s strengths and weaknesses for supporting designers. As time was limited, the testing concentrated upon the knowledge-capture modes, which we had identified as the most difficult aspects of the system for potential users and hence most likely to require modification.

Three employees with differing levels of experience and computer literacy participated in the exercise, with two of the authors assisting the test takers and videotaping their efforts. We encouraged the subjects to think aloud, describing what they were doing and what they were thinking about at the time.

Michael Levi and Frederick Conrad have identified seven broad categories of usability problems specific to Web-based software: management and maintenance, technology constraints, navigation issues (scrolling, Back and Forward buttons, and so forth), structure, content, mismatched goals, and page layout. These provided a framework for the evaluation that yielded a number of significant results.

The test subjects indicated that WebCADET could be a useful tool in supporting their conceptual design activities and made a number of constructive criticisms of the user interface. The key need identified was “interruptibility.” Currently, the WebCADET server will drop a network connection with a client browser after prolonged inactivity. Interruptions (sometimes of considerable duration) frequently occur in a designer’s working day, as designers usually work on multiple tasks concurrently. Hence it must be possible to save “work in progress” in future versions of WebCADET, so integrating WebCADET into the designer’s work patterns rather than forcing them to adapt to WebCADET. Other observations included a preference for more hyperlinks for faster transitions between WebCADET modes rather than using the Back button and requests for changes to form layout and additional interactive help facilities.

Work is underway to modify WebCADET as a result of the user evaluation.
The ongoing development of WebCADET involves adding knowledge client functionality to individual WebCADET servers. The term knowledge client denotes a knowledge-based system that accesses the network (whether intranet or Internet) to acquire knowledge (content that the system can directly manipulate—rules and cases, for example) for current or later needs. In the context of WebCADET, this will involve individual servers acting in the role of client to other servers. For example, in Figure 10 a designer at a client browser accesses a WebCADET server based in the UK for design guidance. The designer then decides to browse for alternative guidance and requests the UK WebCADET server to search for other WebCADET servers with more relevant design knowledge. The UK server does this by assuming the role of knowledge client. In this mode, the UK WebCADET server connects to other WebCADET servers and retrieves knowledge concerning their design guidance coverage. The UK WebCADET server informs the designer (client machine) of a number of servers with relevant knowledge. In the example in Figure 10, the designer directly connects to the Hong Kong server for more appropriate design guidance knowledge.

Initially the implementation will involve storing a list of known WebCADET server locations at each server. An individual server will launch a Netscape browser and connect to another server. Using DDE or other interfaces, the browser can be manipulated to access the other server and determine its particular locus of design expertise. At the same time, the server-location database can be updated at the first server.

Future directions for WebCADET include developing the keywords slot of rule texts for retrieval purposes, thus providing cross-product and cross-attribute searches for identifying design guidance biased towards particular markets. A further avenue is a standardization of terminology in the more active design domains to allow for future WebCADET servers to amalgamate distinct knowledge bases.

Acknowledgments

Nicholas Caldwell is grateful for the financial support of the Isaac Newton Trust. This work was also funded by the Engineering and Physical Sciences Research Council (Grant GR/M23236).

References


Nicholas H.M. Caldwell is a research associate in the Department of Engineering at the University of Cambridge, working jointly for the Scientific Imaging Group and the Engineering Design Centre. His research interests include electron microscopy, applied AI, knowledge-based systems, Internet technologies, and software engineering. He received a BA in computer science and a PhD in engineering from the University of Cambridge. He is a fellow of the Royal Microscopical Society and a graduate member of the British Computer Society. Contact him at Cambridge Univ. Engineering Dept., Trumpington St., Cambridge, Cambridgeshire, CB2 1PZ, UK; nhmc1@eng.cam.ac.uk.

Paul A. Rodgers is a lecturer in the Department of Design at Napier University, Edinburgh. His research interests include the development of AI Web-based tools to support conceptual design, the communication and collaboration of designers in multidisciplinary teams, sketching in design, and design work via the Internet. He received a BEd in design and technology and an MA in computing in design, both from Middlesex University, and his PhD from the University of Westminster, London. Contact him at the Dept. of Design, Napier Univ., 10 Colinton Rd., Edinburgh, EH10 5DT, UK; p.rogers@napier.ac.uk.

Avon P. Huxor is a senior research fellow at the Centre for Electronic Arts, Middlesex University. His research interests include computer-supported collaborative work, intelligent Web aids, and the relationship between the built environment and the virtual. He received a BSc in architecture, planning, building, and environmental science from University College, London; an MA from the Royal College of Art; and his PhD from Middlesex University for work on the “AI as text” paradigm. Contact him at the Centre for Electronic Arts, Middlesex Univ., Cat Hill, Barnet, Herts. EN4 8HT, UK; a.huxor@mdx.ac.uk.

P. John Clarkson is a university lecturer in the Department of Engineering at the University of Cambridge, where he is also director of the Engineering Design Centre. His research interests include general engineering design, particularly the development of design methodologies to address specific design issues such as design for reliability, design process capture and reuse through the “signposting” paradigm, and the use of knowledge-based systems in design for the aerospace and medical equipment domains. He earned a BA in electrical sciences and a PhD in engineering from the University of Cambridge. Contact him at the Eng. Design Centre, Cambridge Univ. Engineering Dept., Trumpington St., Cambridge, Cambridgeshire, CB2 1PZ, UK; pjc10@eng.cam.ac.uk.