How to Present Designs

Marton E. Balazs, David C. Brown, Peter Bastien and Craig E. Wills

Worcester Polytechnic Institute, Department of Computer Science
100 Institute Rd., Worcester, MA 01609
Ph: (508) 831-5618; FAX: (508) 831-5776
Email: mebalazs@cs.wpi.edu, dcb@cs.wpi.edu, pbastien@ll.mit.edu, cew@cs.wpi.edu

Abstract

The basis of Knowledge Intensive CAD (KIC) is that "intensive life-cycle knowledge regarding products and design processes must be incorporated in the center of the CAD architecture." We would like to extend this concept slightly by concentrating on the presentation of designs and design information to the designer. In this we argue that knowledge about some aspects of the design process must be included in the presentation phase of the CAD system. In addition we propose that presentation itself is a design process and give a description of this process.

Keywords

Presentation, design process, function, style, configuration, display description language

1 INTRODUCTION

The basis of Knowledge Intensive CAD (KIC) is that "intensive life-cycle knowledge regarding products and design processes must be incorporated in the center of the CAD architecture." We would like to extend this concept slightly by concentrating on what might be seen as the 'edge' of the CAD architecture, the presentation of designs and design information to the user (i.e., to the designer). In this we argue that knowledge about some aspects of the design process must be included in the presentation phase of the CAD system. In addition we propose that presentation itself is a design process.

2 THE PROBLEM

The problem we address is that of automatically generating the presentation of some given information about a designed object. In particular we are interested in the presentation of information about computer networks. We assume that a presentation will be instantiated in the form of a display, on a given device. Such a presentation is very useful for tasks such as network design, network administration and network maintenance and service, and a working prototype computer system (TENNIS) has been implemented [Brown et al 1995].

While this paper uses computer networks as its sample domain, it is easy to see many strong correlations with other designed objects, such as mechanical, electrical or civil engineering designs. Hence, the issues discussed are quite general.

Different tasks may require different kinds and amounts of information. Obviously for different pieces of information different presentations will be appropriate. Also, while different tasks may require the same information, different presentations may be adequate for each of them. When generating a presentation the needs of the user, general and domain specific human-computer interaction (HCI) principles and the capabilities of the device will determine which displays will be appropriate for the presentation and which one of these will the best possible choice.

As an example let us consider the problem of presenting the down-time cost for all the computers in a network. If the user to whom this information is to be presented is interested in identifying the computers with the highest down-time costs (for instance a network manager), the display to be generated should make this identification easy. For this purpose a histogram or a sorted list of down-time costs would be good candidates. On the other hand, if instead the user (for instance a service engineer) is interested in topologically or geographically locating the computers with high down-time costs, it would be more appropriate to present the values in a graph or on a map.

Further, it is a general principle that a histogram is a better way than a list of values to present numeric information that needs to be compared. This principle gives a ranking to these alternative presentations. Finally, if the device for which the display will be generated only allows alphanumeric output, the histogram alternative has to be discarded.

2.1 Information about computer networks

One of the factors that determines the kinds of displays and display elements that can be used for a certain class of presentation problems is the nature of the information. In the following we give the set of computer networks information types we consider for presentation, together with their impact on the presentation needs.

Structural information

Structural information about a computer network consists of information about the network components (e.g. devices, network components) and about structural relations between components (such as subnetworks, logical and physical connections and so on). Network components are typically objects with several attributes, both qualitative and quantitative. When presenting information about network components we need means to identify the type and identity of the components as well as a way of attaching identifying information about values of their attributes. The presentation of structural relations in a computer network may require generic means for presenting relations (such as graphs or tables) as well as domain or task specific presentation elements (e.g. geographic maps).

Dependency information

Parts of a network may depend on other parts of the network while performing their tasks. This can be expressed by different types of dependency relations between network components. We have identified three types of dependency relations: communication dependency, service dependency, and task dependency. Since dependencies are relations their presentation also requires the availability of generic means for presenting relations, such as graphs and tables.

Inspect net_downtime_cost of computers produced by AVANTO

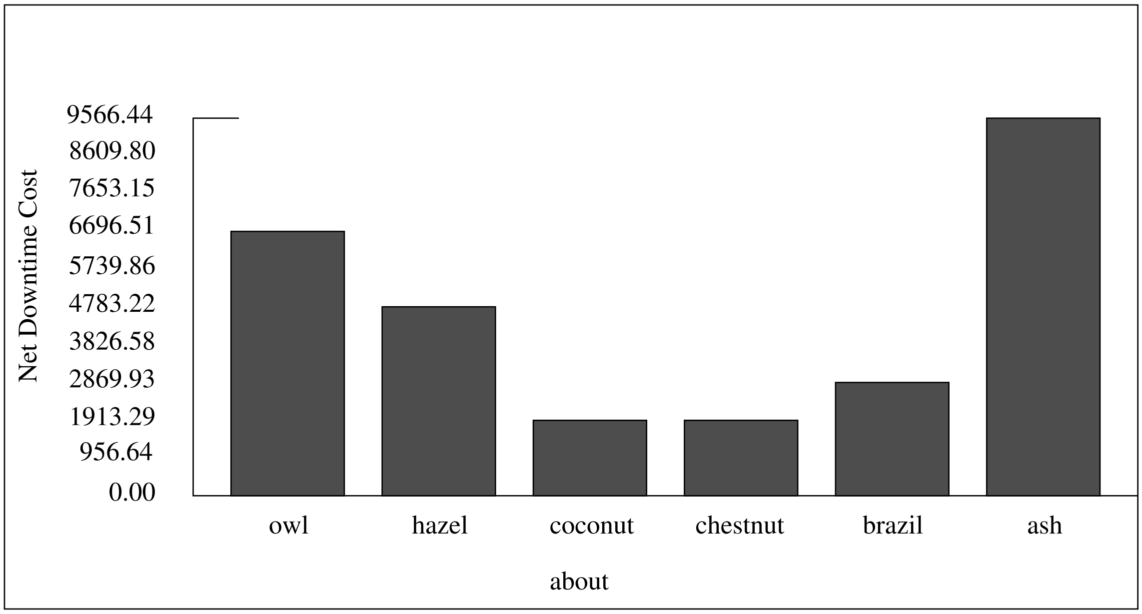

Figure 1(a). Presentation of downtime costs using a histogram

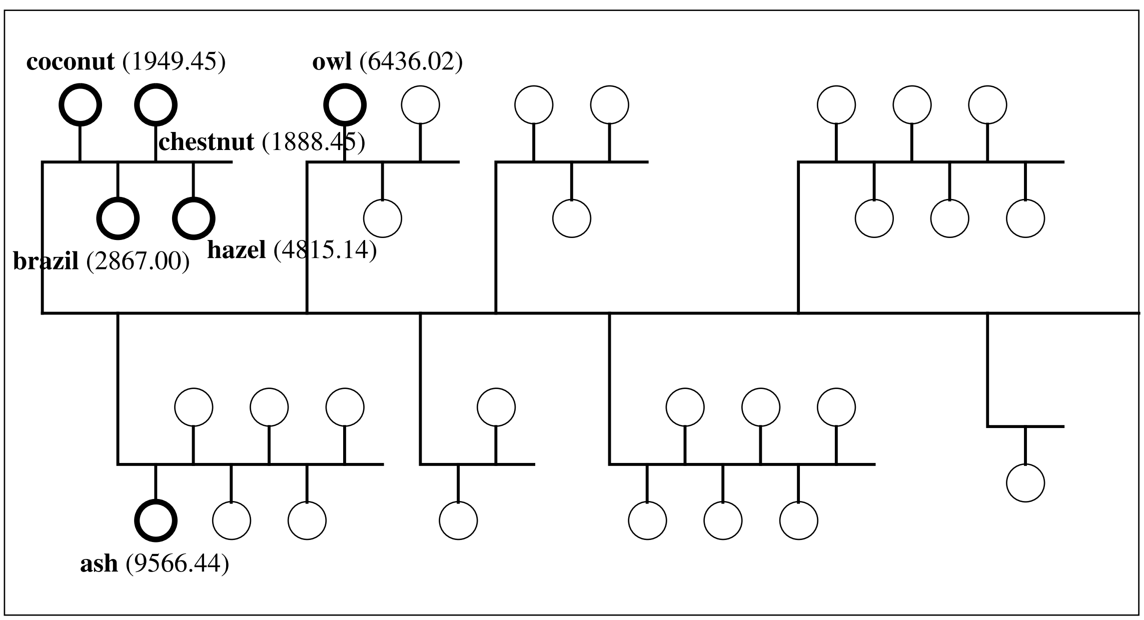

Figure 1(b). Presentation of downtime costs mapped on the topology of the network

Performance information

Performance information (e.g. speeds, thruputs) may refer to network components, to parts of the network (e.g. a subnetwork) or to the whole network. While generally performances are (quantitative or qualitative) measures and, as such, similar in nature to attributes the fact that they not only characterize network components, but also parts of the network, that may be formed dynamically (e.g. an arbitrary grouping of devices). Hence the presentation of information about them requires means to attach measures to the presentation of such entities. For instance, if the problem is to present the average CPU utilization for all the computers the names of which start with the letter "a", presentation needs a means for grouping these computers, presenting the required information and connecting the two together.

Maintenance and service information

We consider maintenance and service information as a separate category for two reasons: First its nature may be significantly different from the nature of the other information about computer networks (e.g. accessibility information, tool and work-force needs, etc.). Second, the tasks for which service and maintenance information is used may also be special (e.g. maintenance planning, human resources planning, etc.).

Some presentation examples

We conclude this section by giving some presentation examples produced by the system we implemented. Figure 1, a and b show the two presentations corresponding to our example at the beginning of this section.

Figure 2. presents a more complex example. The goal of the user was to "Inspect all the service dependencies of the computer 'raven'". From the service dependency types considered only two, file service and name service, implied 'raven'. To clearly separate the two types of dependencies two separate displays were produced. These were then combined into a page given the need to present them together, resulting from the goal. Since dependencies are relations they are represented using graphs. The domain preference, to see dependency information related to the network, suggested the mapping of the graph to the topology graph of the network.

3 THE PRESENTATION PROCESS

Using the example in the previous section we can discover some similarity between a presentation problem and design problems. Let us explore this similarity in more detail.

The user's needs (e.g. what the presentation is going to be used for) determine the appropriate display alternatives, exactly as design requirements determine the solutions to a design problem. Just as these designs are restricted by various constraints (for instance material constraints), the graphical elements and graphical properties of the device restrict the set of dis-

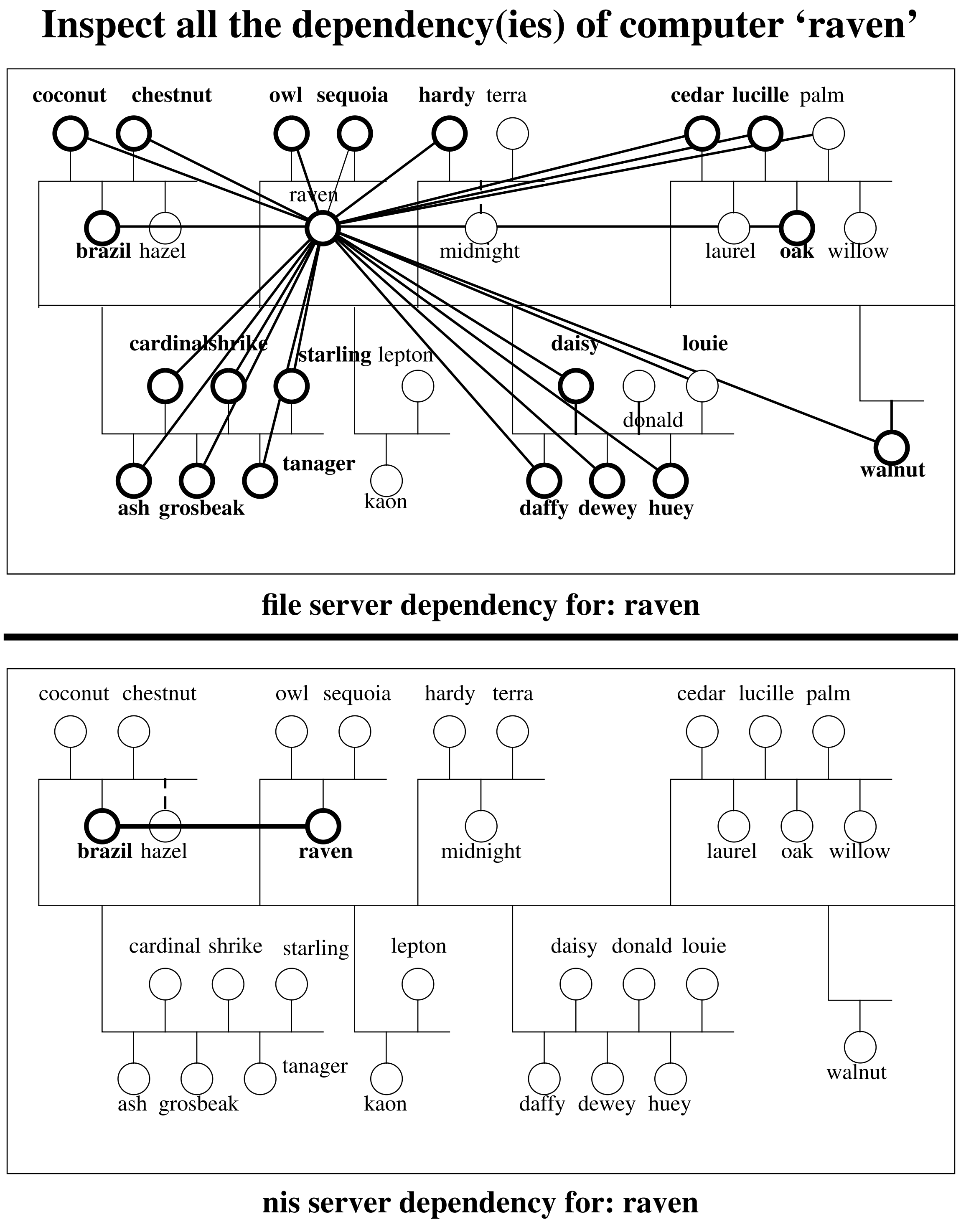

Figure 2. Presentation generated for the goal: "Inspect all the dependencies of 'raven'".

plays that can be generated.

Further, a presentation is intended to be used for some given purpose. This use assigns a certain role to the presentation, i.e., the corresponding display will fulfill a certain function when used. For instance in our previous example of identifying computers with high down-time costs, the display will have to relate some values to each other, corresponding to the user's goal of comparing these values.

Depending on the amount and structure of the underlying information and the goal of the user, the function of a display may be simple (such as "to relate" in our example) or complex. In the latter case no single display element may be suitable to provide the required functionality, and so the function has to be decomposed into sub-functions (possibly in a recursive manner). For instance, if the function of the presentation is to describe a given network, this could be decomposed into showing the topology of the network and relating its performance characteristics to average performance values of similar networks. Obviously the decomposition itself will also depend on the needs of the user.

Displays can be constructed by combining smaller displays. For instance, displays can be grouped together, linked, placed in different relative positions, and so on. This is similar to configuration as step of a design process.

Finally, in order to be generated, displays must have their parameters (e.g. colors, sizes, distances) set properly, exactly as parameter values must be set in parametric design.

The above considerations allow us to view presentation problems as design problems and as a consequence to model the presentation process as a design process.

The presentation process has as input a set of requirements. These may be either specified (e.g. user goals, current preferences) or implicit (e.g. readability, unambiguity). The output presentation will be a (detailed) description of the display. Since the goals of the user constitute one of the most important inputs to the presentation process we give a brief discussion about goals in the context of our problem.

The user's goals may be described in terms of goal types and goal parameters. Goal types specify what the user intends to use the presentation for, while goal parameters describe the objects of the presentation. Different goal types may have different goal parameters. Some examples of goal types together with their parameters are:

- to compare some objects or attribute of objects (e.g. the down-time costs of the computers in a network);
- to analyze a set of information about an object or class of objects (e.g. the importance of the devices in a network, based on their dependencies);
- to identify an object or set of objects satisfying certain conditions (e.g. the computer with the highest number of file service clients).

We have already seen that the same information may be presented in different ways. One of the most frequent reasons for this is that, even for the same information, different presentations may be more appropriate for different goals of the user than others. (It should be clear that different information may require different presentations even if the users goal is the same.)

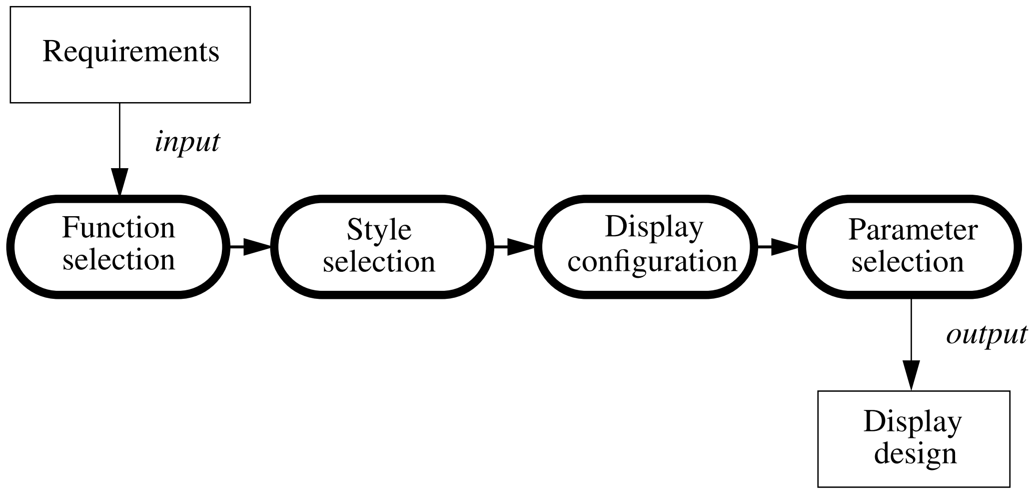

Figure 3. Phases of the presentation process

3.1 Phases of the presentation process

Based on the analogy we built between presentation and design we model the presentation process as having four phases (Figure 3): function selection and functional decomposition (corresponding to functional design), style selection (corresponding to conceptual design), display configuration (corresponding to configuration design) and parameter selection (corresponding to parametric design).

Function selection and functional decomposition

Function selection and functional decomposition is the first phase of the presentation. Its goal is to describe the presentation function(s) of the display to be generated, such as to show, to illustrate, to suggest, to highlight etc.

This phase receives as input a description of one or more presentation goals of the user. These goals may be of different types, for instance to inspect, to analyze, to synthesize, etc.

The output generated by this phase is a set of partial designs. Each of these is a description of what functionality the display to be generated will have. These functions may be either simple functions (such as “to show a piece of information”) or a functional decomposition of a more complex function (such as “relate the degrees of file service dependencies of the computers in the network”).

The processing performed is the mapping of each of the user goals in one or more presentation functions and, possibly recursively, decomposing these presentation functions. This mapping is based on knowledge about tasks & goals of the user and knowledge about the domain (e.g. what is the domain-specific for practice presenting certain kinds of information).

Style selection

The goal of style selection is to generate presentation style for the presentation functions generated in the previous phase.

This phase receives as input a set of functional descriptions of the presentation. For each of these it generates one or more high level style descriptions. These style descriptions may refer to general styles, such as text or graphics, as well as to more specific (preconfigured) styles, such as table, bar chart, etc.

The output of function to style mapping is a set of partial designs, each of which is a functional description of the presentation with styles attached to every function and subfunction.

To complete this phase two kinds of processing have to be performed. First, functions must be mapped into styles. This mapping may result in a large number of different function to style associations for each of the functions and subfunctions. Second, the style resulting must be composed in accordance with the presentation functions were decomposed into subfunctions (for instance texts and bar charts may have to be combined into tables). This step may lead to situations where some style alternatives have to be dropped. Dropping alternatives can happen when composing two styles makes no sense, when there is no known good way to combine certain styles, or when there is some knowledge about the display capabilities of the device for which the presentation is produced that makes certain styles and/or style combinations useless.

The processing in this phase uses presentation knowledge, based on general HCI principles and on domain specific presentation practice, as well as knowledge about the user's goals, expertise and preferences.

Display configuration

The display configuration has as its goal to build an overall style for the presentation.

It receives as input a set of partial designs, each of them describing a presentation function (decomposition) with a high level style attached. For each of these it produces as output of one or more partial designs which contain a detailed description of the presentation style. This detailed style description is obtained by configuring the styles in the input, that is building spatial relations among them.

Thus an element of the output of this phase (a partial design) will be a description of a presentation function, of the styles attached, and of the relative placement of each on the display to be generated (e.g. above, close to, grouped together with, inside, etc.).

The processing performed consists of building spatial relations from a set of predefined ones (above, below, close to, etc.) in a way consistent with the goals, functionality and style of the presentation. This building uses general HCI principles, domain specific presentation practice knowledge, and knowledge about user goals, expertise and preferences.

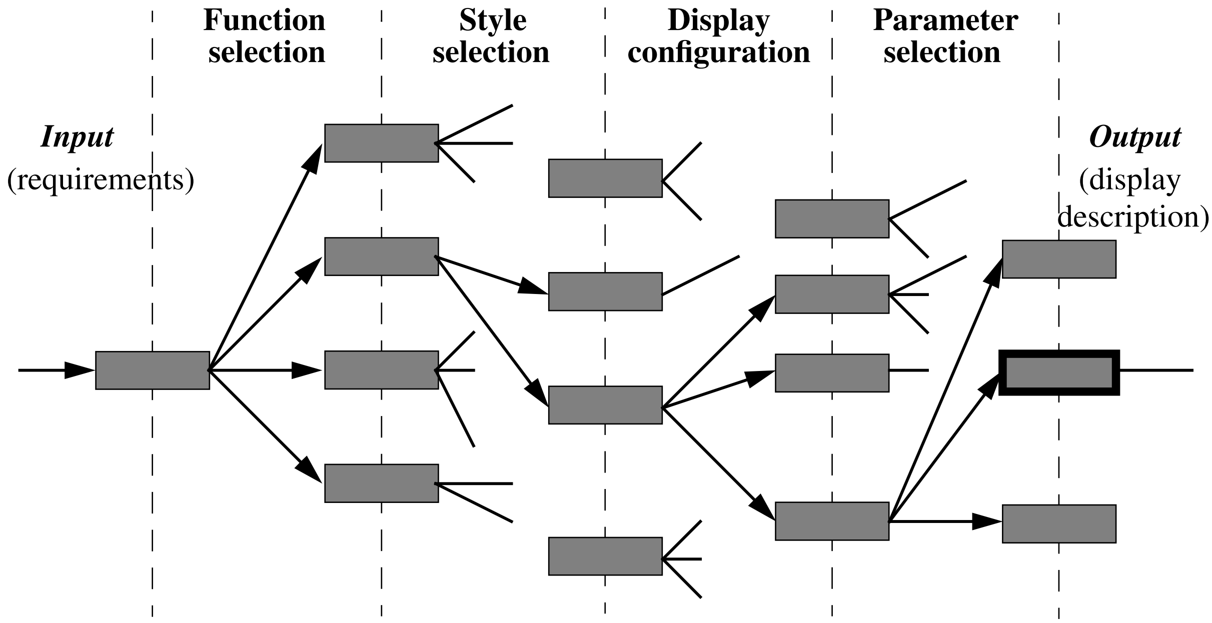

Figure 4. Multiple design alternative generation

Parameter selection

The last phase of presentation generates a detailed description of the display to be produced. It receives as input a set of partial designs containing functional descriptions of the presentation together with detailed style descriptions generated in the previous phase. For each of these several detailed display descriptions may be produced. These display descriptions include details such as color, size, distance etc.

The output is produced in a device- and presentation tool-independent display description language described in a later section. To produce the detailed display description both general and device/tool display knowledge is used.

3.2 Preferences

In each of the phases of the presentations several (partial) presentation designs are produced (Figure 4.). We call these design alternatives, or simply alternatives. The output of the whole presentation process is a single display design, the one which is considered to be the best. This process is accomplished by generating a ranking of all the display designs produced by the presentation process and selecting the one ranked the highest. Using a complete ranking of the display design has the advantage of providing an “also good alternatives” option to be inspected by the user.

The ranking of display designs is produced using “preferences”. Preferences are classified along two dimensions: scope and duration. We call these dimensions types and classes respectively.

Types of Preferences

Based on their scope we consider two types of preferences: phase preferences and overall preferences.

Phase preferences are given for each of the phases of the presentation process. For each phase the corresponding phase preference gives a set of possible choices together with an ordering. For example, graphs, tables and lists may be given as possible style choices for presenting binary relations and (table, list, graph) may be a corresponding ordering. This example of style preference expresses that when mapping a function implying the presentation of a binary relation to styles, tables will be preferred to lists and lists will be preferred to graphs.

Overall preferences are given for the whole process of presentation. They express the relative importance of the four phases. For instance if style selection (such as, deciding on graphics versus text) has to be important the overall preference for style has to be set high (relative to the other phases). Overall preferences are expressed by weights assigned to each of the phases.

Classes of Preferences

For duration, preferences may be of three classes: global preferences, user preferences and session preferences. Each of these classes may have either or both of the two preference types (phase and overall).

Global preferences express general HCI principles. They are always taken into account and are used by default, for situations where preferences of no other class are provided.

User Preferences express overall and/or phase preferences of a user. They are part of a user profile. If they are provided, they have priority over global preferences (if their use makes sense, that is they produce acceptable partial designs). Also, if given, they are used as default in a session performed for the corresponding user if no session preferences are provided.

Session preferences are used to express overall and/or phase preferences for the current session with the system. If provided, they override existing user and global preferences.

Using Preferences

Global preferences are always taken into account by the system. User preferences are only considered if the system maintains user profiles (Figure 5.). Neither of them changes during one session.Each presentation process uses a set of current preferences which are computed from the provided classes of preferences using the overriding and default rules described above. The current preferences contain a complete set of preferences of both types (overall and phase).

Applying Phase Preferences

Phase preferences are applied at the end of each phase, after all the (partial) design alternatives

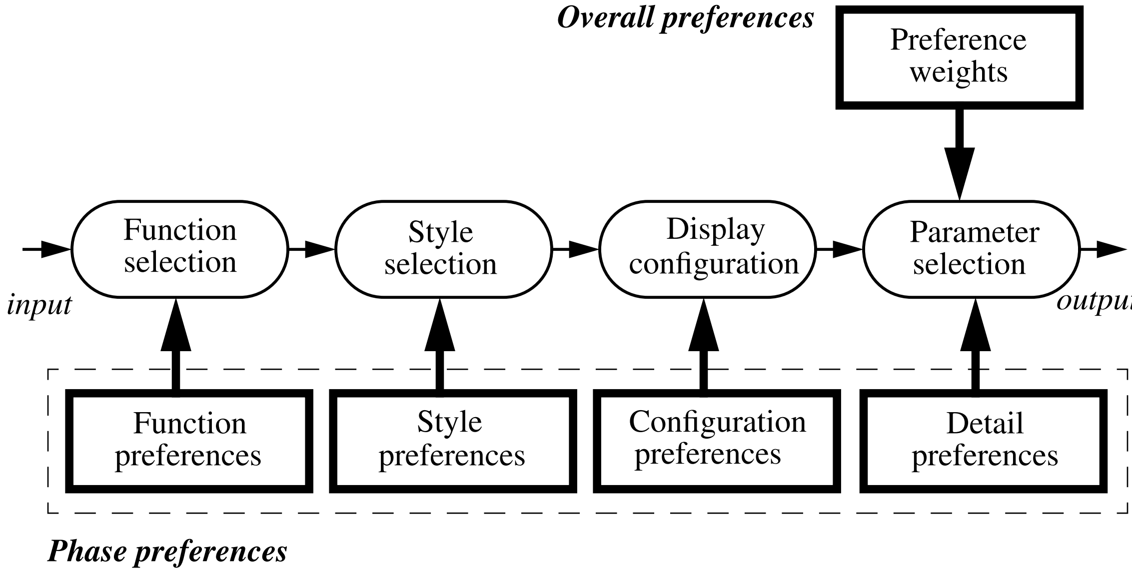

Figure 5. Using preferences

have been generated. They are used to compute a score for each of the generated alternatives obtained in the corresponding phase. These scores result from the ordering specified in the preference for the respective phase and are normalized in order to give a uniform basis for ranking.

After all the phases have been completed each of the resulting presentation design alternatives have a score assigned for every phase.

Applying Overall Preferences

Overall preferences are applied at the end of the presentation process. They are used to compute a weighted sum of the scores of phase preferences for every presentation design alternative produced. The final scores are used to order the alternatives and thus produce a ranking.

The design with the highest score is considered the best suited for the goal input to the presentation process and it is selected for displaying. The ranking can be further used to inspect other “relatively good” alternatives.

3.3 Display description

As we mentioned earlier the output of the presentation phases is a detailed design (i.e. description) of the display. For this purpose we chose to use a display language that is powerful enough to describe all the possible displays we generated for our problem domain and general enough to be instantiatable in formatting languages of general use (e.g. HTML). The decision to use a display language, rather than a direct presentation using HTML for instance, was because a display language would provide a large degree of display independence.

In designing the display language, the first major decision was what should be explicitly stated and what should be determined by the instantiating process (in our implementation the "HTMLizer"). The highest level structure of the display language is the display element. Each display element is an independent single display. Each display is composed of one or more "paragraphs". The language allows for some constraint between paragraphs in terms of spatial position. It can be specified that paragraphs are near other paragraph or a group of paragraphs should be displayed together. The idea behind these constraints is that in the rendering of the display it may be desirable to split a display into several separately displayed units based on implementation criteria.

Each paragraph is either text, a table, a graphic, or a menu. The text and table paragraph types are as their names suggest. The menu paragraph type allows the HTMLizer to be used as in an interactive mode rather than a strictly display-only mode. The graphic paragraph type is further divided into graphs, bar charts, pie charts, scatter plots, and network maps. These types were selected based on the possible graphics needed for the Tennis system.

The description of each paragraph has, in general, two major segments: display attributes and data. The display attributes are parameters such as size, specific formatting criteria, orientation, and order. The data segment contains the actual data to be displayed and any per element display criteria.

4 IMPLEMENTATION

We implemented our model of the presentation process as part of our Computer Network Ease of service Evaluation System. The presentation part of this system was implemented as two communication modules (Figure 6.): the presentation module and the post processor (translator/interpreter)module.

The presentation module is composed of four submodules corresponding to the four presentation phases. Each of the modules is built up from a set of rules which map the partial designs (or requirements) received as input to new (partial) designs produced as output. The rules encode the knowledge used by the respective modules. The only type of knowledge that is not represented in the phase modules is the global preferences. The output of the presentation will be a display description represented in the display language.

The post processor module takes as input a display description in the display language and translates it to a target language. In our implementation the target language is the HTML. The HTML output of the presentation subsystem can be interpreted by a WWW browser. Our choice of HTML as target language was motivated by the followings: hypertext links are fully supported, text formatting is easier, the resulting interface is well understood by users and the interface would be available to remote sites through the Internet.

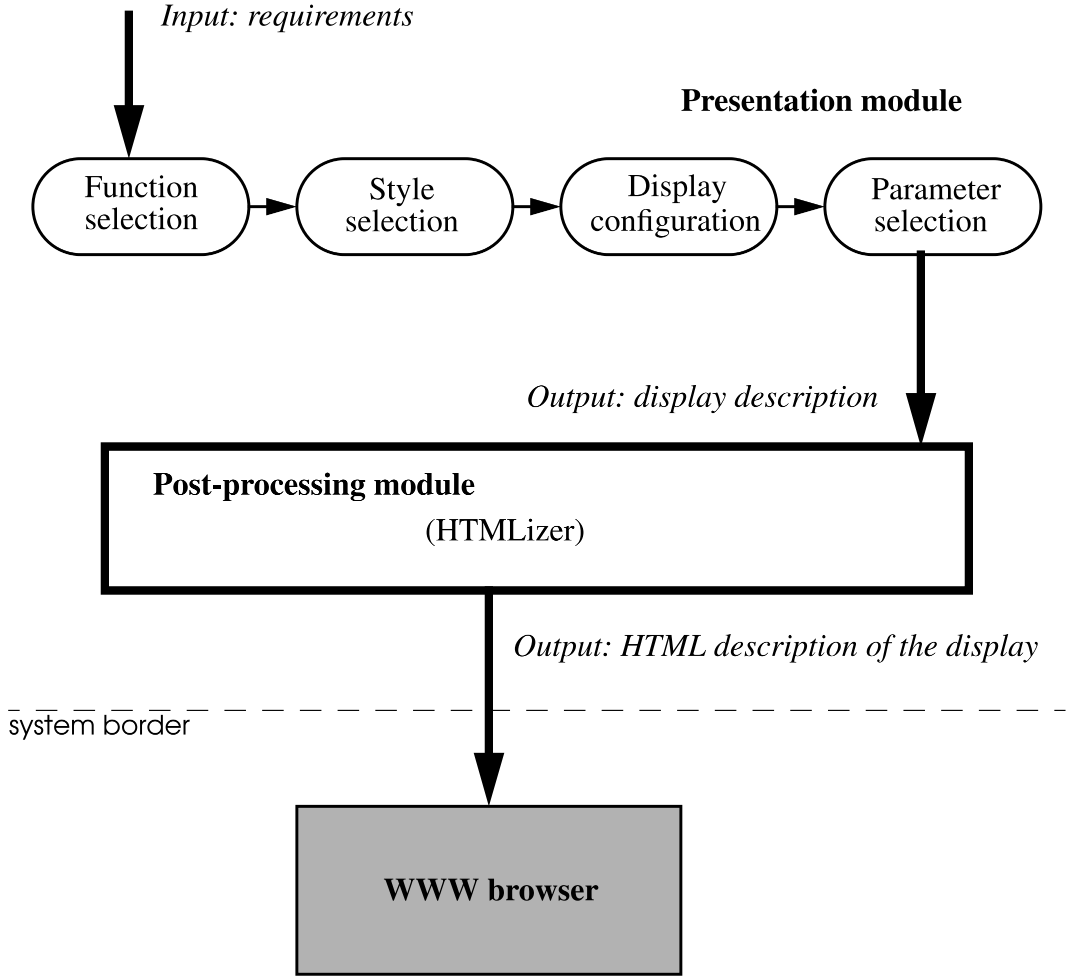

Figure 6. System structure.

We developed our system while cooperating with network assessment experts. For testing we developed a set of test cases for the aspects we were mostly interested in, such as: the generation of multiple presentation alternatives, the impact of preferences on the choice of the best alternative, and generation of unusual display combinations.

5 RELATED WORK

Related to our display of results is work associated with the previous phase of the TENNIS project [Kemble 1994]. In this work Kemble developed a system for the automatic generation of a display based upon the relations inherent in a set of multi-attribute data. He used this system to generate displays for both network-related information and also classic graphical displays, such as Napoleon's march.

Kumar et al [1994] describes work on a user interface for a prototype network configuration management system. More generally we found related work on general display techniques. [Weitzman 1992] discusses a knowledge-based graphic design assistant for the design of two-dimensional interfaces. Other work we examined was [Dewan & Choudhary 1995], [Dourish 1995], [Petre 1995], [Wehrend & Lewis 1990].

One of the most closely related pieces of work is the SAGE system by Roth at Carnegie Mellon University [Roth et al 1994], [Roth & Mattis 1990], [Roth 1995], [Chuah et al 1995]. He and fellow researchers are interested in automating the presentation of information. They confronted many issues we examined, although in a more general information domain than computer networks. Our approach is different from that of SAGE in the way it approaches the problem. We view and model the presentation problem as a design problem and propose a corresponding solution.

Another important piece of work was [Chappel & Wilson]. They provide general guidelines on appropriate displays for different types of data objects. These results were incorporated into the design of the Tennis'95 presentation process.

6 CONCLUSIONS

We believe that an important aspect of "KIC systems" will be the knowledge intensive design of displays of designs and design information. In this paper we have presented some general characteristics of such a system, and some details of how this can be applied to a particular kind of designed object, the computer network.

7 REFERENCES

Brown, D.C., Wills, C.E., Dunskus, B.V. & Kemble, J., TENNIS: A computer network ease of service evaluation system, *International Joint Conference on Artificial Intelligence, Workshop on AI in Distributed Information Networks,* August, 1995.

Chappel, H. & Wilson, M., Knowledge-based design of graphical responses, *Proceedings of the ACM International Workshop on Intelligent User Interfaces*, pp. 29-36, ACM Press, January 1993.

Chuah, M.C., Roth S.F., Kolojejchick, J., Mattis, J. & Juarez, O., Sage-Book: Searching data-graphics by content, *CHI'95 Mosaic of Creativity,* May 7-11, 1995.

Dewan, P. & Choudhary, R., Coupling the user interfaces of a multiuser program, *ACM Transactions on Computer-Human Interaction*, Vol. 2, No.1, March 1995.

Kemble, J., *Display of Multi-Attribute Data Using a Presentation Description Language,* Masters thesis, Worcester Polytechnic Institute, August, 1994.

Marks, J., A Formal Specification scheme for network diagrams that facilitate automated design, *Journal of Visual Languages and Computing,* December 1991.

Petre, M., Why looking isn't always seeing, *Communications of the ACM*, Vol. 38, No. 6, June 1995.

Roth, S.F., Kolojejchick, J., Mattis, J. & Goldstein, J., Interactive graphic design using automatic presentation knowledge, In: *Human Factors in Computing Systems, CHI'94,* Boston, Massachusetts, USA, April 24-28, 1994.

Roth, S.F. & Mattis, J., Data characterization for intelligent graphics presentation, *Proceedings CHI'90*, May 1990.

Roth, S.F., Kolojejchick, J., Mattis, J. & Chuah, M.C., SageTools: An intelligent environment for sketching, browsing, and customizing data-graphics, *Proceedings CHI'95 Mosaic of Creativity,* May 7-11, 1995.

Wehrend, S. & Lewis, C., A problem-oriented classification of visualization techniques, *IEEE Computer,* 1990.

Weitzman, L., DESIGNER: A knowledge-based graphic design assistant, *Artificial Intelligence in Engineering Design*, Volume I, Academic Press, (eds.) T. Tong & D. Sriram, page 433, 1992.

8 BIOGRAPHY

Marton E. Balazs is a Computer Science Ph.D. Candidate at WPI. His research interests include analogical reasoning, reasoning about function of physical systems, creativity and evolutionary computation. Balazs received a B.S. and M.S. in Computer Science and a Ph.D. in Mathematics from the "Babes-Bolyai" University in Cluj, Romania. He is a member of AAAI, the Romanian Society of Mathematical Sciences and the "John von Neumann" Computer Science Society in Hungary.

David C. Brown is a Professor of Computer Science at WPI, and is a member of the ACM, IEEE Computer Society, the AAAI, and IFIP WG5.2. He is on the Editorial Boards of the Journals "AI in Engineering, Design, Analysis and Manufacturing" and "Concurrent Engineering: Research and Applications". His research interests include computational models of Engineering Design, and the applications of Artificial Intelligence to Engineering. He is the author, with B. Chandrasekaran, of the book "Design Problem Solving: Knowledge Structures and Control Strategies", Pitman Publishing, Ltd., and a co-editor of "Intelligent Computer Aided Design", Elsevier Science Publishers B.V. (North-Holland).

Peter Bastien is currently a member of the technical staff at Massachusetts Institute of Technology Lincoln Laboratory. He holds a Bachelor of Science degree in Computer and System Engineering from Rensselaer Polytechnic Institute and a Master of Science Degree from Worcester Polytechnic Institute.

Craig E. Wills is an associate professor in the Computer Science Department at Worcester Polytechnic Institute. His research interests include networking, user interfaces, distributed computing and operating systems. Wills received his B.S. in Computer Science from the University of Nebraska in 1982 and his M.S. and Ph.D. in Computer Science from Purdue University in 1984 and 1988, respectively.