Intelligent Critiquing of Design Sketches

Yeonjoo Oh, Ellen Yi-Luen Do, Mark D Gross

Computational Design Group, Carnegie Mellon University
{yeonjoo; ellendo; mdgross}@cmu.edu

Abstract

Design Evaluator is a pen-based system that provides designers with critical feedback on their sketches in various visual forms. The goal of these system-generated critiques is to help end users who draw and then reason about their drawings to solve design problems. This paper outlines the implementation strategies of the Design Evaluator and shows example applications in two visual design domains: architectural floor plans and Web page layout.

Machine Annotation of Freehand Drawings

Designers draw diagrams and sketches to record design idea and to reason about design constraints or alternatives. Designers in many domains (e.g. engineering, architecture) sketch graphical elements, accompanied by shorthand notes that record design rationale and other design information (Davis 2002). Design sketching is considered a component of a reasoning process that uses both visual and textual representations to solve problems. In this process, the 'seeing' that follows sketching involves both interpreting and evaluating designs that are represented as drawing marks on paper (Verstijinen, Heylighen et al. 1991). Seeing initiates subsequent design moves and problem reframing (Schön 1985; Goldschmidt 1991)). In this seeing process, textual and visual annotation or other forms of critique on the design drawing may help designers to explore alternatives and to reframe design problems.

Critiquing is a familiar and important activity for designers. In a studio setting, a critic sees and reframes the design problem for another designer. Knowledge transfer occurs when the critic communicates what s/he sees and thinks to the designer through the combined media of sketching, talking, and showing examples. These acts help to restructure the designer's knowledge, which in turn helps the designer to discover and explore new design alternatives.

Critiquing systems were the research focus of Gerhard Fischer's group at the University of Colorado during the late 1980s and 1990s. This effort resulted in a series of projects such as JANUS, which linked critiquing of kitchen layouts with design rationale (Fischer and Mørch 1988), KID, which linked text critiques with a construction kit (Fischer, Nakakoji et al. 1993), and the visual critiquing system Petri-NED (Stolze 1994). Each of these systems employed critiquing to support design. However they all employed structured editors which, we argue, tend to inhibit early design exploration.

Sketch interaction has been explored in various domains. In UI design, for example, Landay and Myers's (Landay and Myers 1995) SILK allows a designer to sketch elements of a prototype user interface; the system then transforms the sketch into a working interface. Sketch systems have also been built that support simulation in mechanical engineering (Davis 2002). Forbus, Usher and Chapman (2003) have explored pen-based interaction for military course-of-action planning.

Our Electronic Cocktail Napkin project explored using sketch recognition as an interaction paradigm for a variety of tasks and domains, including accessing cases in a design database, simulation and knowledge based advising, and constraint based graphical editing (Gross 1996; Gross and Do 2000). In the Right Tool Right Time project (Do 1998) the system attempted to infer the user's task; then based on this inference it suggested an appropriate knowledge-based tool. Although there is a great deal of interest in applying sketch interfaces to other kinds of intelligent systems, we are unaware of any other effort to provide a sketch interface to knowledge based critiquing. We believe this is a powerful combination, and that is the motivation behind our current work on the Design Evaluator.

The Design Evaluator is an intelligent sketch system that attempts to reason about the design at hand and provide the designer with useful feedback in the form of criticism and advice. An intelligent sketch system requires two basic components: First, because a design problem-solving process inevitably involves specific domain knowledge, the system should have access to knowledge about the domain. Second, the system should present advice or critiques in appropriate form. For example, when a design process is heavily involved in sketching and diagramming, critiques should appear in similar and related form without distracting the designer from the tasks in which s/he is engaged.

This paper describes the current Design Evaluator system, a domain-independent framework for sketch based critiquing. The system is composed of three layers:

description, evaluation, and visualization. We have created specialized versions of the Design Evaluator for two design domains: architectural floor plans and Web page layout design. We describe each of these domain-specific versions, which have different rules and methods of displaying critiques.

Design Evaluator—the three layers

We have implemented the Design Evaluator, a pen-based critiquing system, in Macintosh Common Lisp. Design Evaluator consists of three layers: Description, Evaluation, and Visualization. Each layer supports a different sort of activities that designers carry out. After recording the designer's sketch, the system tests it against previously stored predicates that express domain-specific design rules. Then, the system generates critiques and displays them in textual and visual forms.

Description Layer

Design Evaluator first records and identifies graphical elements of lines, circles and boxes as object symbols. The capture and parsing of the designer's graphical input is the task of the description layer. The visual structure created by elements and spatial relationships is contingent on the domain (Ferguson and Forbus 2002).

In architectural floor plans the bubbles and lines represent zones (large functionally specified spaces), rooms (individual functional areas), and doors. Figure 1 shows an example of a simple hospital floor plan layout. It has three functional zones: support, clinical, and nursing zone. Each zone contains several rooms for medical functions. The lines connecting rooms indicate doors.

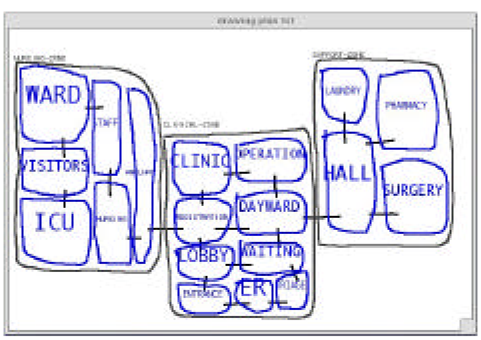

Figure 1. Sketch design for hospital floor plan

In a Web page layout (Figure 2), these same symbols represent elements such as the screen, panels, images, headlines, and blocks of text. The largest 'bubble' represents the outer screen boundary. It contains symbols of text (lines, zigzag lines) and smaller boxes that indicate the placement of images. Figure 2 shows an example of a Web Page layout design sketch.

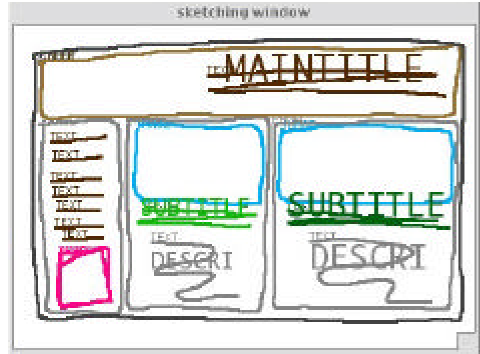

Figure 2. Sketch for Web Page Layout: A large heading image is placed at the screen top. Three panels have links, images, and descriptions.

After the initial capture, the Design Evaluator identifies spatial relations in the drawing to produce symbolic descriptions of the design. The design representation in the Design Evaluator system comprises object lists—a zone, room, and door list for architectural floor plans and panel, text, and image lists for Web page layouts. Whenever the designer sketches graphical elements such as bubbles and lines, after the system recognizes the element it creates an instance and adds it to the appropriate list. The sketched objects are related with one another in various ways; each object also stores a list of related objects.

For example, in the architectural design evaluator, a zone object stores a list of all rooms that it contains, and a room object stores a list of its doors that connect with other room. Conversely, each door object stores the two rooms it connects. In the Architectural Design Evaluator, the system also post-processes the data after it captures the sketch, for example, identifying circulation paths through the floor plan. This sets the stage for rule checking in the next step.

In a Web page design context, a subtitle text symbol is contained in a panel and the panel is contained in a screen (Figure 2). In a floor plan of a hospital, a patient ward room symbol is contained in the nursing zone and a door symbol overlaps two rooms, indicating that a path connects them (Figure 1).

Evaluation Layer

In this layer the system compares sketches against predicates that represent design rules and generates critiques when rules recognize a pattern in the design. The system compares the recognized spatial information with each rule. If it finds a rule violation, it generates a design critique to be displayed in the Visualization Layer.

Note that rules can also be structured to critique positively as well as negatively, for example to reinforce a designer's decision ("good location for the picture heading"). Critiques may also indicate various degrees of imperative ("the waiting room must be adjacent to the entrance;" whereas, "it's a good idea to put the elevator and the stairs in the same lobby").

Although the rules are different in each domain, the Design Evaluator provides a generic interface for users to specify design rules. Basic spatial relation predicates are built in to the system, which the Design Evaluator uses to diagnose designs (Table 1). The current rules for both domains depend on the spatial relationships of containment, adjacency, and relative size. For special cases the user can extend them by writing additional code.

Each rule is—at minimum—associated with a text critique message, but depending on the domain, a rule may also generate other forms of feedback. In critiquing an architectural floor plan, the system provides visual annotation and 3D texture-mapped VRML models as well as textual critiques. Three-dimensional visualization is essential in architecture — architects use 3D perspectives, mockup models, and 3D computer models to help see the spaces they design. In the Web page layout domain, the system offers visual annotation of the designer's sketch, good and bad examples related to the critique, as well as textual critiques.

Table 1. Elements & Relationships in Two Design Domains

	Floor Plan	Web Page Layout
Element	Minimum-Area	Number of Images
		Content Hierarchy
Spatial Relationships	Room Placement	Image-Text Ratio
	Adjacency	Color Scheme
	Room Sequence	Text-Background

Table 1 shows example graphical elements and spatial relationships in the rule-sets for two design domains. These rule-sets are related with individual graphical elements and spatial relationships like 'contain', 'overlap', and 'next to'. Each rule-set has several kinds of rules. In the case of architectural floor plans, the system operates with 4 kinds of rules (minimum area for room, appropriate room placement, adjacency requirement, and appropriate room sequence). The Web page evaluator has 5 kinds of rules (number of images, content hierarchy, image-text ratio, color scheme, and text-background).

Rule-Sets for Architectural Floor Plans The Architectural Design Evaluator supports four kinds of rules for architectural floor plans: Minimum area, Room placement, Adjacency, and Room sequence.

A Minimum Area Rule takes the form:

(<Minimum-area> <room> <minimum-size>)

For example, (MINIMUM-AREA WARD 10000) expresses a minimum area requirement about the specific room. It states that a ward be no smaller than 10,000 area units.

A Room Placement Rule takes the form:

(<Placement-rule>
<Zone>(<Room><Room><Room>…))

which indicates that all rooms in the list inside the inner parentheses should be in (or not in) the given zone. The <placement> can be either "must-be-in" or "may-not-be-in". The expression:

(MUST-BE-IN
CLINICAL-ZONE
(ER TRIAGE DAYWARD CLINICAL-FOR-OUTPATIENT))

represents a typical room placement requirement in hospital design that states ER, TRIAGE, CLINICAL-FOR-OUTPATIENT, and DAYWARD should be placed in the CLINICAL-ZONE.

An Adjacency Rule takes the form:

(<Adjacency-rule> <Room1> <Room2>)

indicating a desired adjacency relationship between two rooms. For example, the following expression represents a required adjacency of two rooms in a hospital design:

(SHOULD-BE-ADJACENT ER INPATIENT-SURGERY).

An adjacency rule can also express a prohibition, such as the following noise-related requirement:

(MAY-NOT-BE-ADJACENT CHAPEL BOILER-ROOM)

A Room Sequence Rule takes the form:

(<Sequence-rule> <room1> <room2> [<room3>])

which indicates that there should be a path that follows from room1 – room2 – room3. For example, the expression

(MUST-PASS-THROUGH ENTRANCE TRIAGE ER)

represents a required circulation sequence in a hospital design. In order to access the Emergency Room from the ENTRANCE, the circulation path must pass through the TRIAGE area.

Rule-Sets for Web Page Layout. To build the Web Page Design Evaluator we borrowed basic rules from Nielsen's web usability guidelines (Nielsen 2000). The system supports five kinds of rules for Web page layout. They concern Number of Images, Content Hierarchy, Image-Text Ratio, Color Scheme, and Text Background.

A Number of Images Rule takes the form:

(<Number_Image-rule> <Image List> <Number>)

The system counts the number of images. The number of images on a page can be a design criterion. A page with too few images may be considered dull; on the other hand, too many images can slow page loading. The expression (MAXIMUM_#_IMAGES 10) states that a Web page should have no more than ten images.

A Content Hierarchy Rule takes the form:
(<Hierarchy-rule> <Text-List>)
The system takes the text-list, counts the kinds of texts like MAIN_TITLE, SUB_TITLE, and DESCRIPTION and then reminds the designer of the hierarchy of text content.
In web page design is important that one spatial layout reflects the hierarchy of information. A hierarchy rule checks that the various title and text boxes are spatially organized in appropriate manner.
An Image-Text Ratio Rule takes the form:
(<Image/Text Ratio>
<Image/Text Area> <Screen-Area>)
The system calculates the total area covered by images and the total area covered by text and compares these with the screen area. It calculates the fractions of each area against the screen area. For example the following expression represents an area limitation of images in one Web page.
(MAX-RATIO IMAGES_AREA SCREEN_AREA 50)
This says that images may not occupy over 50% of a single Web page.

A Color Scheme Rule takes the form:
(<Color_Scheme><Container_Obj_Color>
<Contained_Text_Color>)
A Web page usually conveys textual information and the color scheme is important to allow users to easily read the text. A Color-Scheme rule compares the color of the container_object (panel or screen) and the color of text it contains. If the two colors are too similar, the system offers a critique. The expression
(DIFFERENT_COLOR PANEL2 TEXT1)
indicates that the color of TEXT1 (which is placed in PANEL2) cannot be similar to the color of PANEL2.

A Text-Background Rule takes the form:
(<Text-Background> <Text-List>)
It indicates that the specified text may not overlap any background images, because it may obscure readability.

Visualization Layer

Different design domains call for different methods of displaying critiques. At the most basic level Design Evaluator displays text critiques in a special message window. Each text critique may also be linked to other visual displays, for example in a sketch annotation, items retrieved from a database, or a 3D visualization. Below we briefly describe critique display methods for the Architectural and Web page layout Design Evaluator.

Visualizing Architectural Critiques The Architectural Design Evaluator displays critiques in three ways: as text messages, annotated drawings and texture-mapped 3D models. The system checks the sketch of Figure 1 and generates critiques: "ICU AND ER SHOULD BE ADJACENT, TOO FAR IN THE CURRENT DESIGN", and "BETWEEN HALLWAY AND WARD, YOU SHOULD PASS NURSING_ZONE" as shown in Figure 3.

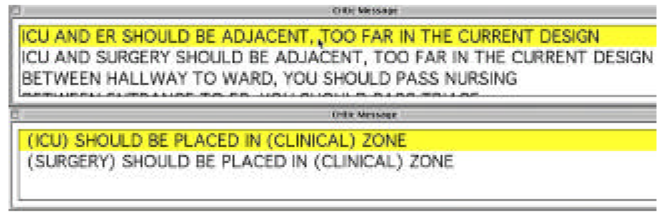

Figure 3. Text Critiques for Architectural Floor Plan

When the designer selects a message in the text critiques window (Figure 3), the system shows the critique in two other forms. For example, when the designer selects the message "ICU AND ER SHOULD BE ADJACENT, TOO FAR IN THE CURRENT DESIGN" the Design Evaluator annotates the floor plan, highlighting the path between the Emergency Room and the Intensive Care Unit. The Design Evaluator also builds a 3D texture-mapped VRML model that shows the path through the floor plan (Figure 4).

When the user selects the text message, the system visualizes the problematic path and annotates the incorrectly placed room in red (see Figure 4).

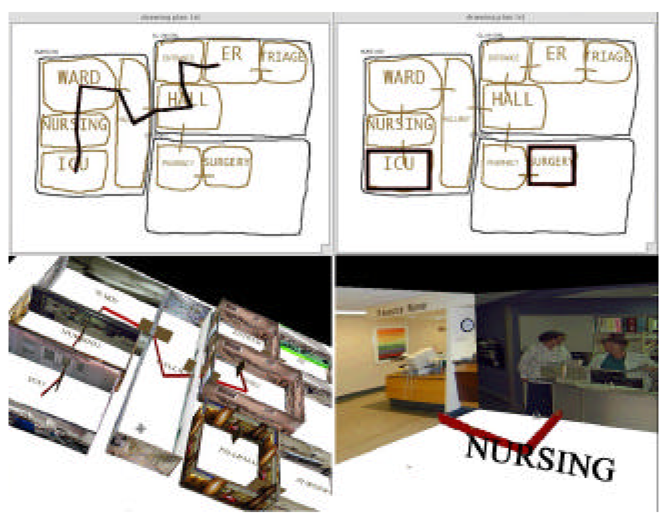

Figure 4. Critique visualizations in architectural floor plan. Sketch annotation (upper) and 3D model (below)

The third method for displaying floor plan critiques is a 3D visualization of the space in a VRML (Virtual Reality

Modeling Language) model. The system uses stored coordinates and heights of objects to generate the 3D models, including walls and doors, from the sketch. In addition, to lend realism to the model, each wall of the generated 3D models is texture-mapped with image pictures. The system retrieves a hospital photo appropriate for texture-mapping the walls of each room. For example, the system will texture map the walls of the ER model with photographs from an actual emergency room.

Visualizing Web Page Layout Critiques. The Web page Design Evaluator also generates text critiques. These are linked to visual critiques through sketch annotation and also to design examples, or cases. When the designer selects a text critique, the system illustrates the issue not only by annotating the designer's sketch; it also brings up an example Web page in the browser that demonstrates the issue or concern.

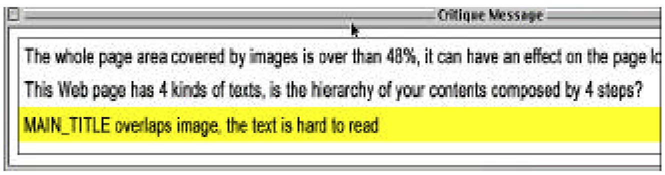

Figure 5. Web page layout text critiques

For example, when the designer selects the critique "MAIN_TITLE overlaps image, the text is hard to read" (Figure 5), the Design Evaluator annotates the sketch design with red warning marks (Figure 6-left). The system highlights the MAIN_TITLE in red. It also displays an exemplar Web page in which background images partially obscure the text (figure 6-right).

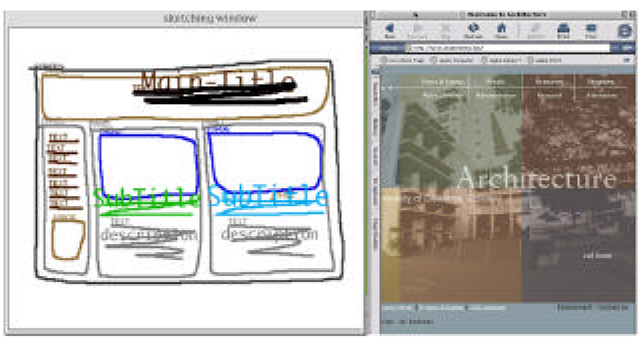

Figure 6. Visualizing Web page critiques: visual annotation (left) and exemplar Web page (right)

Implementation

The description layer of Design Evaluator is limited to capturing sketch data from the user and applying some pre-processing steps to prepare a design representation for the second, evaluation layer. The description layer labels the sketch marks, and creates and structures objects in the design database. The correspondence between sketch marks and object types is handled by domain-specific functions. Likewise the structuring of relationships among the design objects is handled by domain specific functions that link the objects according to spatial relationships that obtain among the sketch marks. For example, in the architectural Design Evaluator, a special "path finding" algorithm generates a list of all the circulation paths through the floor plan.

In the Evaluation layer, rules are coded as Lisp predicates that apply to the design objects. The rule expressions are stored in a list that the end user (designer) can inspect and edit. The rule expressions are transformed into function calls to the corresponding predicates. Each rule expression is also associated with a text critique, as well as code that specifies how to annotate the sketch when the critique is applied. A Rule may also carry additional information to be used by auxiliary visualization routines, such as the VRML model creator (for the Architectural Design Evaluator) or the URL of a representative example case (for the Web page layout Design Evaluator).

Finally, the visualization layer contains the basic routines for displaying text critiques in a message panel as well as linking the text critiques with the sketch annotations when the designer clicks on a critique. The visualization layer may be enhanced in a specific domain with the extended visualization methods — for example, linking with a Web browser to display example cases for the Web page layout Design Evaluator.

Discussion

Our work on the Design Evaluator is motivated by our conviction that critiquing can be most effective in the early stages of design, when designers' ideas are still fluid. Early design critiquing, integrated into a freehand sketch design system would be useful to stimulate designers' thinking, to help them reframe problems and explore alternatives. Of course, this requires parsing and recognizing the domain semantics that are embedded in sketches. We built the Design Evaluator system to demonstrate the potential of this approach to integrating knowledge-based tools into a design process. The approach applies to any domain where a sketch or diagram is a valid representation of a design. Mechanical designs, military course of action plans, digital logic and analog electronics are all domains in which diagrams are used to represent designs. We illustrate sketch based critiquing with two applications of the Design Evaluator: one for architectural floor plans; the other for web page layouts.

We would highlight one feature of sketch-based critiquing: the ability to provide annotation or critique directly on the work — the sketch itself. In both the example domains we demonstrate here, in addition to providing text critiques and other forms of feedback, the Design Evaluator annotates the designer's sketch. We believe that, in addition to other forms of feedback, it is important to offer a critical response that is directly linked to the sketch representation. This enables the designer to remain focused on the sketch representation s/he is making.

At the same time, we believe that it is an advantage to offer design critique in various formats. Design Evaluator delivers design critiques in several ways; text, graphical annotation, 3D annotated walk-through models (architectural floor plan) and case library (web page layout). Communicating design information in only one mode can be problematic, and we believe that the combination of displaying graphical critiques with text critiques is likely to be more effective than selecting a single method.

The domain knowledge in each of the applications of the Design Evaluator is expressed in the form of rules about the spatial arrangement of sketch elements. We have translated higher-level design constraints into spatially expressed rules, which the system can test to determine whether the design constraints are satisfied. When the sketched diagram does not satisfy the predicates, the system brings these concerns to the attention of the designer. This raises questions about what design constraints can (or cannot) be expressed in terms of the sketch, and how a system might translate higher level constraints to specific graphical representations and vice versa.

Future Work

Our plans for future work on this project fall in two categories: rules and domains.

The Design Evaluator uses rules expressed symbolically as knowledge for critiquing the proposed design. These text rules express design constraints, design conventions, or designer rules of thumb. However, the development of more sophisticated rules, constraints, and preferences remains an active area of research. The examples illustrated here are simple; a topic for future work is to develop expressions for more sophisticated design rules.

In addition to the expression of more sophisticated rules in symbolic (code) form, the visual critiquing in a sketch requires the development of graphical output conventions to annotate sketches with rule based critiques. This is likely to be domain specific, at least in part: what visual annotations are appropriate to express particular design critiques. Conversely, we would also like to be able to express and define rules graphically, indicating the desired or required positions, dimensions, and spatial relationships of design elements by sketching.

We are also interested to develop specific application domains for Design Evaluators. The two domains we have discussed here are both design domains. However, it is not difficult to think of other application areas beyond design where sketch-based critiquing would be useful. Education springs to mind, for example, supporting learning in domains such as geometry and elementary physics in which people use drawings to solve problems. A geometry or physics "coach" could support student learning with an intelligent sketch based critiquing system.

References

Davis, R., 2002. Sketch Understanding in design: Overview of Work at the MIT AI lab. *Sketch Understanding, Papers from the 2002 AAAI Symposium*. R. Davis, J. Landay and T. F. Stahovich. Menlo Park, CA, American Association for Artificial Intelligence (AAAI), pp: 24-31.

Do, E. Y.-L., 1998. The Right Tool at the Right Time: Investigation of Freehand Drawing as an Interface to Knowledge Based Design Tools, Georgia Institute of Technology.

Ferguson, R. and K. Forbus, 2002. A Cognitive Approach to Sketch Understanding,. *AAAI Spring Symposium Series, Sketch Understanding*, AAAI.

Fischer, G. and A. Mørch, 1988. A Critiquing Approach to Cooperative Kitchen Design. *International Conference on Intelligent Tutoring Systems*. Montreal, Canada, ACM Press, pp: 176-185.

Fischer, G., K. Nakakoji, J. Ostwald, G. Stahl and T. Sumner, 1993. Embedding Critics in Design Environments. *Knowledge Engineering Review* **4**(8): 285-307.

Forbus, K., J. Usher and V. Chapman, 2003. Sketching for military courses of action diagrams. *International Conference on Intelligent User Interfaces*. San Francisco, ACM Press, New York, pp: 61-68.

Goldschimdt, G,: 1991, The Dialectics of Sketching, Journal of Creativity Research 4(2), pp: 123-143

Gross, M. D., 1996. The Electronic Cocktail Napkin - working with diagrams. *Design Studies* **17**(1): 53-70.

Gross, M. D. and E. Y.-L. Do, 2000. Drawing on the Back of an Envelope: a framework for interacting with application programs by freehand drawing. *Computers and Graphics* **24**(6): 835-849.

Landay, J. A. and B. A. Myers, 1995. Interactive Sketching for the Early Stages of Interface Design. *CHI '95 - Human Factors in Computing Systems*. Denver, Colorado, ACM Press, pp: 43-50.

Nielsen, J., 2000. *Designing Web Usability : The Practice of Simplicity,*. Indianapolis, New Riders.

Schön, D., 1985. *the Design Studio*. London, RIBA.

Stolze, M., 1994. Visual Critiquing in Domain Oriented Design Environments: Showing the Right Thing at the Right Place. *AI in Design*. J. Gero. Lausanne, Kluwer, pp: 467-482.

Verstijinen, I. M., A. Heylighen, J. Wagemans and H. Neuckermans, 1991. Sketching, analogies, and creativity on the shared research interests of psychologists and designers. *Visual and Spatial Reasoning in Design II*. J. Gero, Tversky, B, and Purcell, T., pp: 299-310.