David Ullman
Department of Mechanical Engineering
Oregon State University
Corvallis, Oregon
USA

The foundations of the modern design environment: An imaginary retrospective

This paper is written as if the author were in the year 2010 explaining how design is practiced and taught in comparison with the methods of 1990. In this way, the advances in design understanding and tools that are beginning to alter the way we teach and perform design can be best described. It is always easier to look back and describe what has occurred than it is to foretell the future. Thus the paper is written by an older, hopefully wiser, David Ullman looking back on the advances of twenty years. Note that no references are given in the text as they would detract from the format. However, a bibliography is given at the end for more details on the topics discussed.[1]

Design, as it is now practiced, is much different than I remember way back in 1990. As I recall, the design tools and the understanding of the product development process were very primitive in those times. Now our design tools reflect our understanding of the product development process and so I will begin by describing the tools that make the design of quality products so much faster and more efficient today.

I perform most of our design work on a computerized assistant called DUDA (Dave Ullman's Design Assistant). A notebook sized DUDA is shown in the Figure 1. They also are available in desk and wall sized units and in folding models. DUDA embodies much of what has been learned about design in the last twenty years and thus it will be used to demonstrate how far we have come. In fact most of the basic technologies in DUDA existed in 1990 either in the research labs or, to some limited degree, in the open

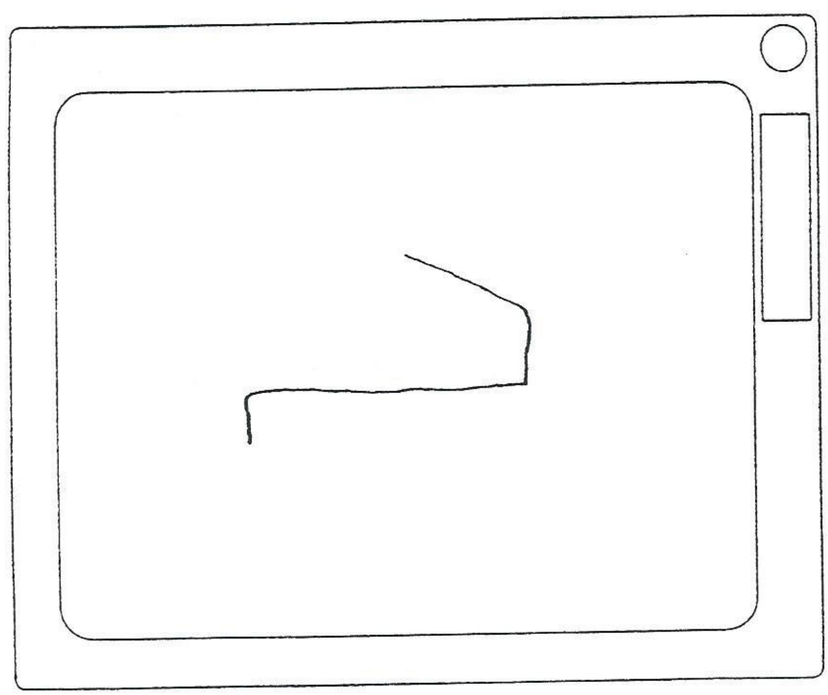

Figure 1: DUDA with a Partially Completed Sketch of a Clip

[1]This paper is dedicated to Mike Wozny of Rensselaer Polytechnic Institute. In 1982 he asked me to write a paper describing the ideal computer design assistant. Although I tried to comply with his request then, it took me nearly ten years to learn enough to do it.

market. However, it wasn't until about the year 2000 that the features that make DUDA so useful began to be packaged as one unit.

Basically, DUDA is a combination of a computer, a notebook, an information source, a communication link and, most importantly, a design process aide and history recording tool. Before describing the knowledge that went into the software that makes these features possible, I need to briefly describe DUDA's basic human interface features.

As seen in Figure 1, DUDA's entire front face is a flat screen. This screen is part of the human interface system as it both displays information and allows for data input. On the model shown, the screen is only 220 mm square keeping DUDA small and portable. However, even this small unit is very rich in its capabilities.

The flat screen is also a data input device. The screen is touch sensitive and thus input can be through a stylus or a finger. With the stylus, drawings, sketches, or written text can be easily input and selections made from lists. Also, objects can be pointed to and identified with a finger. These input methods will be used in the examples below.

While mentioning input devices there is also a microphone on the top of DUDA. This allows verbal commands and other design information to be captured by the device. This technology was available in 1990 to replace keyboard commands. For example, some voice systems were marketed to speed up CAD input. By saying a term that the computer had been trained to recognize, commands and routines were implemented.

Their introduction improved productivity by about a factor of two for routine design tasks. But, as I will explain, the importance of voice input has come far beyond replacing menu selections and macro recovery.

As an optional input device for DUDA, a keyboard can be used. Keyboards were the most popular input device in the 1980s, however their popularity has waned since the turn of the century as they were slow and solely text oriented. Many people still know how to type, but it is no longer taught in the school system the way it once was. Also, popular twenty years ago was a device called a "mouse". They were very useful for selecting from menus and for the input of graphic information. The demise of the "mouse" and its nonexistence on DUDA will become evident.

Also shown in the first figure are the slots for the memory cubes. These super high density storage media contain much of the information that was available in 1990 on CD ROMS. However, they have a density about 100 times greater than the ROMs and are erasable and re-programmable. They are used primarily for the storage and easy retrieval of information such as that formerly found in engineering handbooks, a patent library, journal articles and manufacturer's catalogs. Also, there is extensive information on material properties and availability, and on the costs of raw materials, components in the catalogs, and manufacturing processes. The memory cube is updated weekly, monthly or bi-annually, on a subscription basis. Optionally, DUDA can link to a global design database through cellular technology.

Although hardware capabilities have evolved significantly since 1990, the real differences in DUDA lies in the way information is handled and the manner in which it aides the designer. To make these differences clear a number of concepts that were beginning to be developed in the 1990s need to be explained.

In 1990, the design records for a completed mechanical product were a

This information, represented the two end points of the product development process and did a fairly poor job of supporting communication (with other designers, manufacturing, management, marketing, etc.); supporting the redesign or the design of similar products; and explaining the design sufficiently to enable product modification during the design process. Unfortunately the information about the two end points often failed to support these activities because there was not enough information recorded to answer all of the questions raised.

In the late 1980s, several researchers started studying the idea that redesign and design communication would be significantly improved if the design records included more information about the "history" of the product development process. This history would be a mapping of the evolution of the functionality, structure and behavior of the artifacts and of the decisions and constraints from the initial specifications to the final drawings of the product. Tracking these evolutions is one of the major strengths of DUDA.

With DUDA the following can be browsed:

- Proposals put forth and who posted them,
- Arguments about the proposals,
- Decisions and the order in which they were made,
- Constraints used in the resolution of any issue or the posting of an argument as part of a decision,
- The effect of the initial specifications,
- The effect of all constraints added by decisions,
- The evolution of any object or function,
- The relationships among the designed objects (e.g. construction, location, operation, etc).

Consider the component shown in Figure 2. This stamped brass part is a battery contact blank designed to connect two hearing aid type batteries together in a plastic housing. Let's say you want some information about this component. Standard manufacturing dimensions and tolerances are given on the drawing. However, what if you need to know more about this contact than is shown on the drawing? One way to get more information is to study a

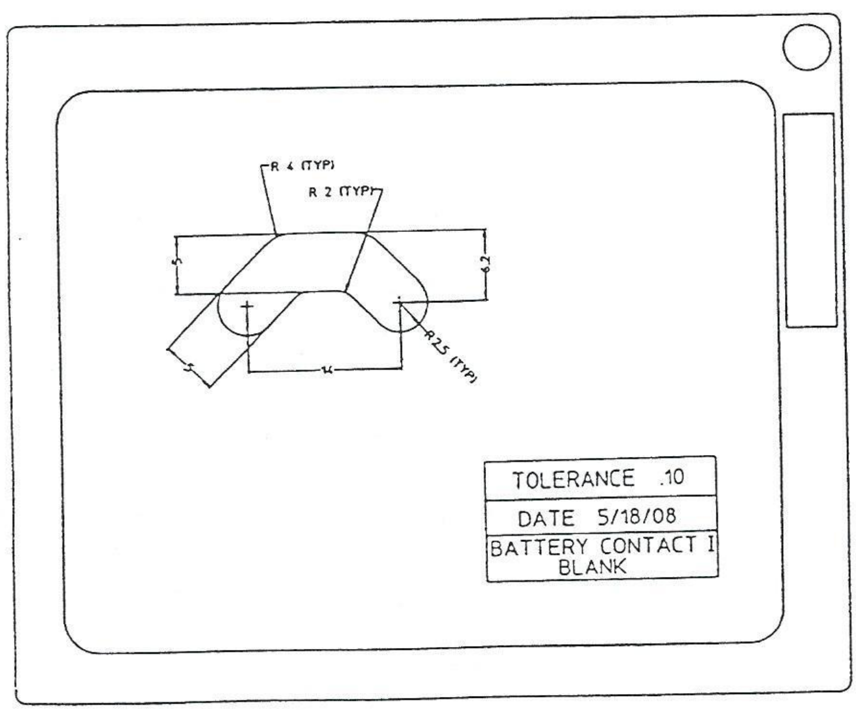

Figure 2: The Final Drawing for a Battery Contact

drawing of the entire assembly and use conjecture to infer additional information about this contact. Another way is to use DUDA to replay the history of the component to answer your questions. Lets say you wanted to know why the component was shaped like a boomerang and how the component might need to be changed if bigger batteries were to be used. DUDA can answer these questions.

In Figure 3, there are four drawings of the evolution of the component through the sketches and drawings made by the designer on DUDA. The design of the battery contact is one of continued refinement. Each figure in the series moves the design closer to the final form of the component. The initial sketch, in window 1, shows circles representing contacts to the battery and a curved line representing conduction of current. In window 2, the contacts have been refined into tabs connected with a wire. In window 3, the designer has patched or altered the design by tilting the connecting structure up. In window 4, the designer further patches the component by combining the wire and the two tabs, making the structure between contacts all one piece.

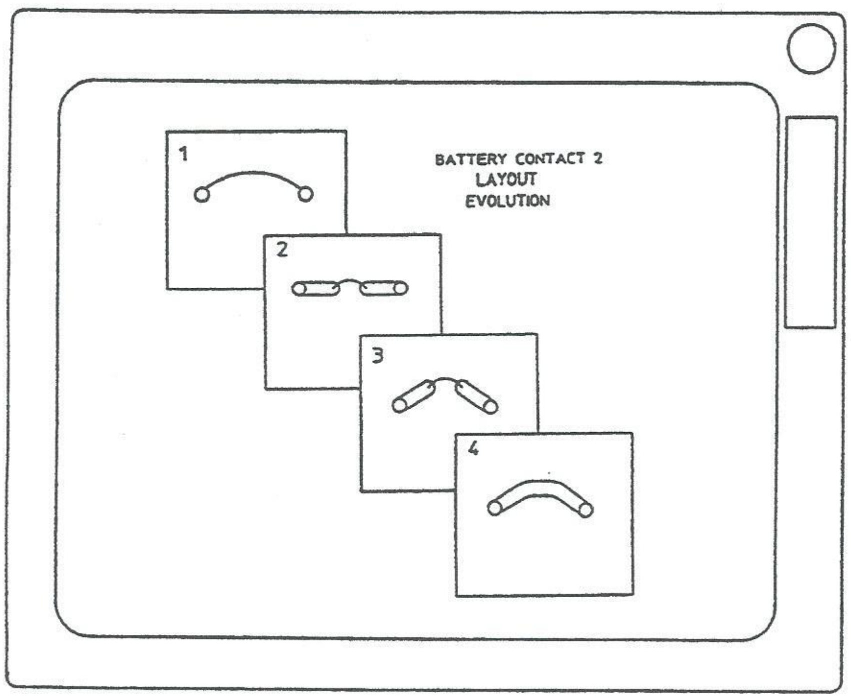

Figure 3: The Evolution of a Component

To explore the rationale for the change in shape from the configuration shown in window two to that shown in window three, I can touch the figure window three with my finger and say "rationale". DUDA's presentation changes to the top two windows in Figure 4. The top left figure is just a re-display of the third window and the right window shows more detail. Specifically it shows part of the structure from another component in the assembly (called a "wall") that must be avoided. Additionally, there is the message "clear wall". This information shows that the previous configuration (window 2 in Figure 3) interfered with another structure and had to be patched. The wall probably didn't even exist when the previous configuration evolved.

We can explore this stage of the evolution of the battery contact further.

determined at this stage of evolution are shown (5 mm diameter of the contact and 14 mm between contact centers). If I want to know why there are 14 mm between contacts, I can touch the dimension and ask "detail". The middle window appears showing the two batteries and the wall between them. I can, of course easily ask for more information if needed by touching and asking. DUDA makes every effort to give me only the amount of information needed to answer the question I put to it. How DUDA captured the information that was recovered in this example will become clear as I discuss more of DUDA's capabilities.

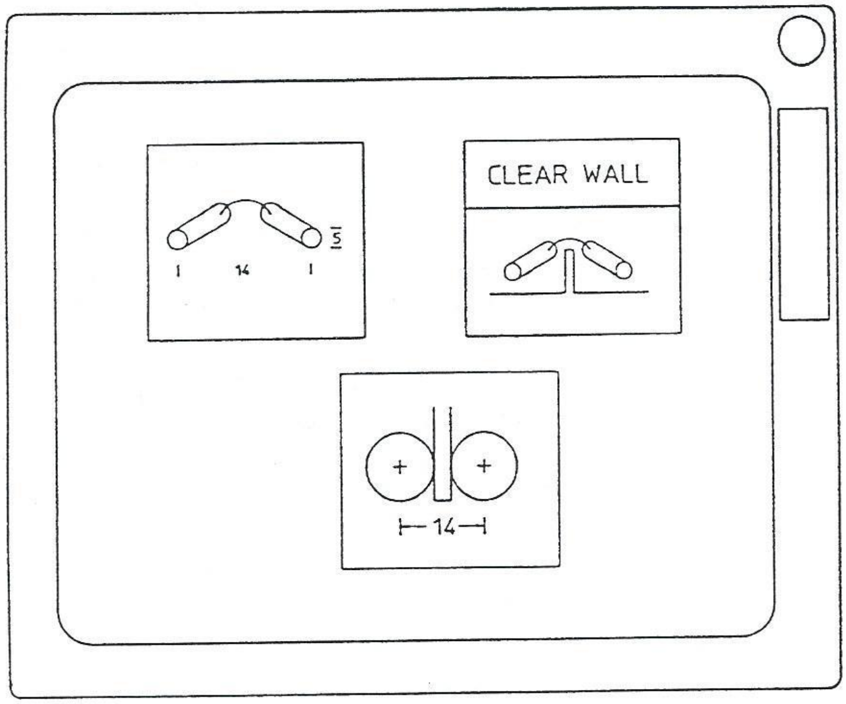

Figure 4: Details on the Evolution of the Component

Another feature of DUDA is its process aiding software, PASE. The term PASE was virtually unheard of in 1990, but has greatly aided in making DUDA the useful assistant it has become. To understand process aiding software we must look back at some of the software available in 1990.

The most common type of software in 1990 was called general purpose software (i.e. spreadsheets, word processors and CADrafting systems). This type of software had no inherent knowledge of the information being represented. In a CAD system the user could draw a line that represented the interface between two adjacent components, the edge of a single component, a thin wire, a road on a map or any number of other objects or features. The system had only knowledge of the geometry of the line without any knowledge of what the line meant. As we shall see, DUDA is much smarter than that.

A second type of software common in 1990 was called special purpose software. This type of software was usually designed for analysis where specific information could be input into the system and a unique result would then be output. In engineering, software packages that solved handbook equations or software specially tailored for a specific problem were common.

In process aiding software, the programs are general, but, at the same time, have knowledge about the order in which issues are addressed during the product development process. This type of software arose out of the parametric representation of geometry. The operation of a parametric system is best shown through a simple example on DUDA. The steps below show how I can design a simple spring clip on DUDA using its parametric features.

Step 1:
I say "New" (Note that any text in quotation marks represents spoken words). DUDA knows I want to start a new problem.

Step 2:
I say "clip". DUDA is told that this a design problem to be called clip. DUDA can recognize many terms and new terms are easily learned. With this command DUDA knows an object is to be created called "clip" and has put this name on the screen as shown in Figure 1.

Step 3:
As I start to sketch the clip I say "two-d, orthogonal" to tell DUDA how to interpret the information. With the stylus I sketch the outline of the clip as shown partially completed in the photograph of the screen in Figure 1. The sketch is made like a pencil on paper. DUDA infers that what I am sketching is a series of straight lines connected at their ends.

Step 4:
I add dimensions to the figure by touching each line segment with my finger or the stylus and saying the value and units. For the 3 mm dimension I indicate the top and bottom of the clip and tell DUDA "vertical distance three millimeters". To add dimensions to the angle I just touch the line and say "thirty degrees". Since my sketch was near to that value DUDA knew what I meant. If I had wanted to be more specific I could have added a qualifier to my statement and said "twenty degrees from horizontal" or provide even more information.

Step 5:
DUDA has enough information to redraw a scale figure of the clip, Figure 5. The parametric ability of DUDA knows which lines in the drawing are connected together and which are perpendicular to each other. A parametric system operates by keeping track of constraints on the geometry. Within

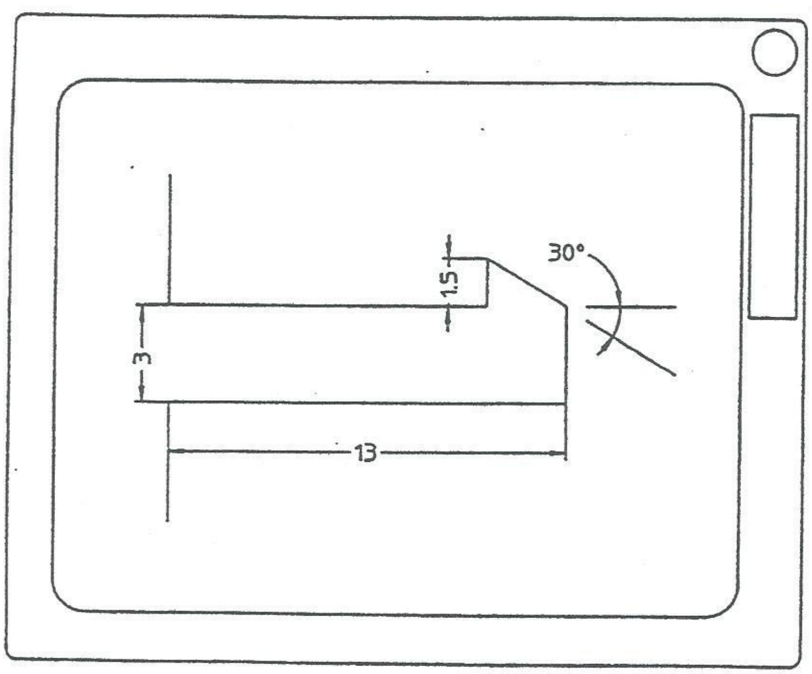

Figure 5: One Configuration of the Clip

such a system all the relationships between line segments are represented as constraints on each line's degrees of freedom.

Step 6:

To change some dimensions all I have to do is touch the dimension and verbalize a new value. To get Figure 6 from Figure 5, I touch the angle dimension and say "twenty degrees" and to change the thickness, I touch the 3 mm dimension and say "2 millimeters". The parametric nature of the constraint representation of the clip allows the new shape to be drawn.

Parametric tools are very useful during the design process as they allow changes in geometry to be made rapidly. However, the amount of information they can infer about the process is almost nil. A more sophisticated geometry inference system had begun to be developed in the late 1980s and is now an integral part of DUDA. It is called a sketch recognition system, and was partially utilized in the previous example. Basically, when I sketch on DUDA's screen, the sketch recognition system infers as much as it can to create a solid model of any device or a record of any gesture. The inference is based on an understanding of the order in which objects are designed. In other words, an understanding of the product development process.

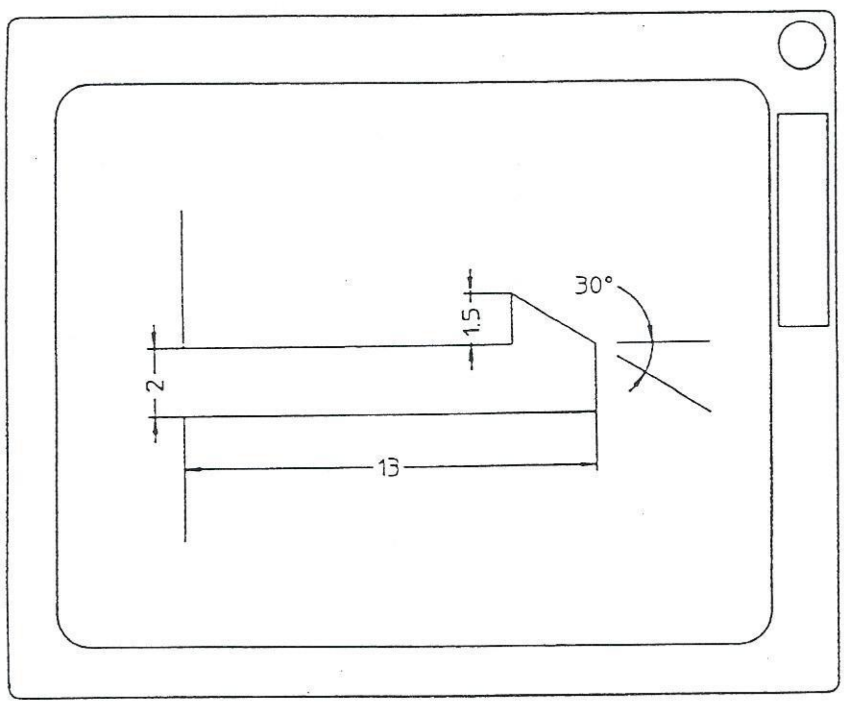

Figure 6: The Re-dimensioned Clip

There are three reasons why this type of system evolved. First, CADrafting tools, common before the turn of the century, had difficulty capturing abstract geometry. They were excellent tools for producing final drawings of components to be manufactured, but during design, when the form of the components were still evolving, they were difficult to use. Second, the inference capability of parametric tools was too limited. Parametric tools were intelligent only about the domain of geometric relationships between line segments. Third, the criticality of sketching and drawing during the product development process was not fully appreciated prior to the 90s. In fact, by 1990 training in how to sketch and draw had almost disappeared from many USA engineering education programs. It wasn't until realization that the USA was not competitive in the development of new products that the importance of basic skills like drawing and sketching were realized.

The operation of the sketch recognition system is shown in Figure 7. The user sketches on DUDA with the stylus. Input shown is in the form of isometric sketches, but any 2-D or 3-D sketch made can be accommodated. The sketched primitives are recognized as line segment primitives such as straight line, arc, circle, etc. These are stored as data for 2-D primitives displayed on the screen "cleaned-up" (i.e. the straight line, sketched not too

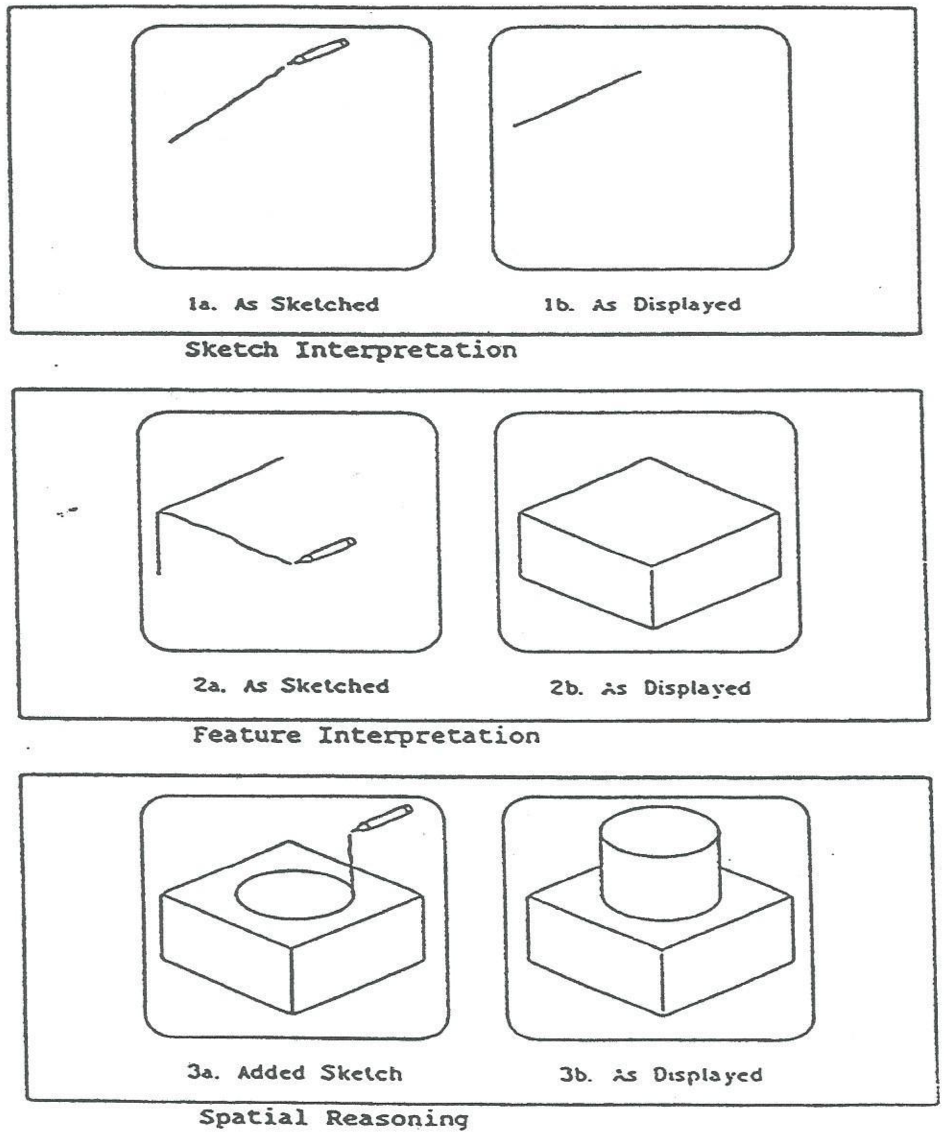

Figure 7: The Operation of The Sketch Recognition system.

straight is now straight). If sketching in 3-D, the system recognizes 3-D features from the 2-D primitives, the heart of the system. As the designer sketches 2-D primitives, the system interprets the 3-D features in the order the designer adds them to the sketch.

This capturing of the sequential order of the features aides in resolving ambiguities when making parametric changes later in the design process.

Next, based on the relationship of each new 3-D feature to the existing design, a feature based solid model is developed and displayed. Figure 8 shows a component sketched on DUDA that took me less than three minutes to sketch.

The importance of this system, and of all process aiding systems, is that it captures the object and information on the product development process (the sequential ordering of adding features to the component) while affording the designer a capability that has not been previously available. This system allows the sketch to be worked on from any angle and the development of the evolution of the component replayed and altered as shown in Figure 9.

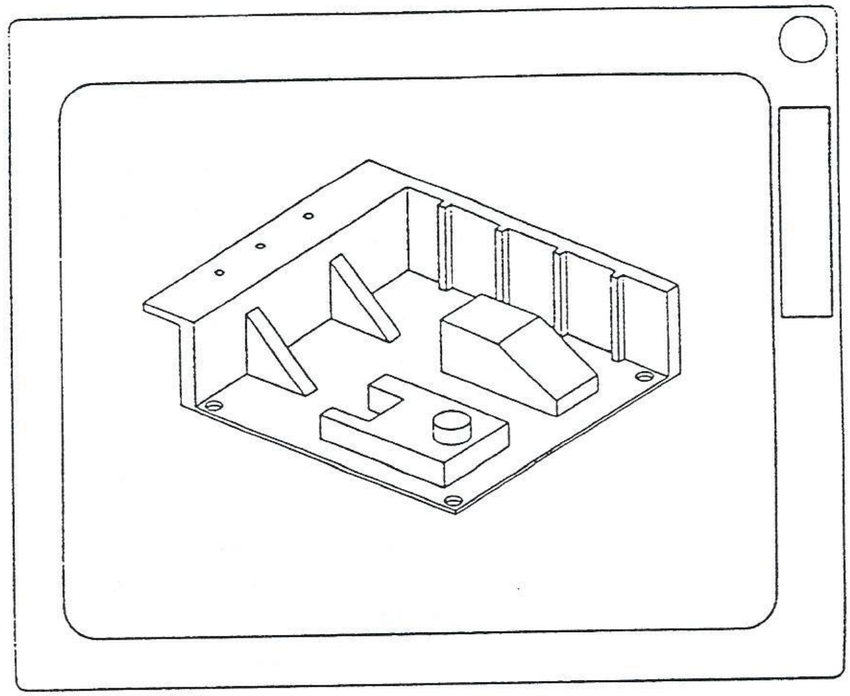

Figure 8: A Sample Component Sketched on DUDA.

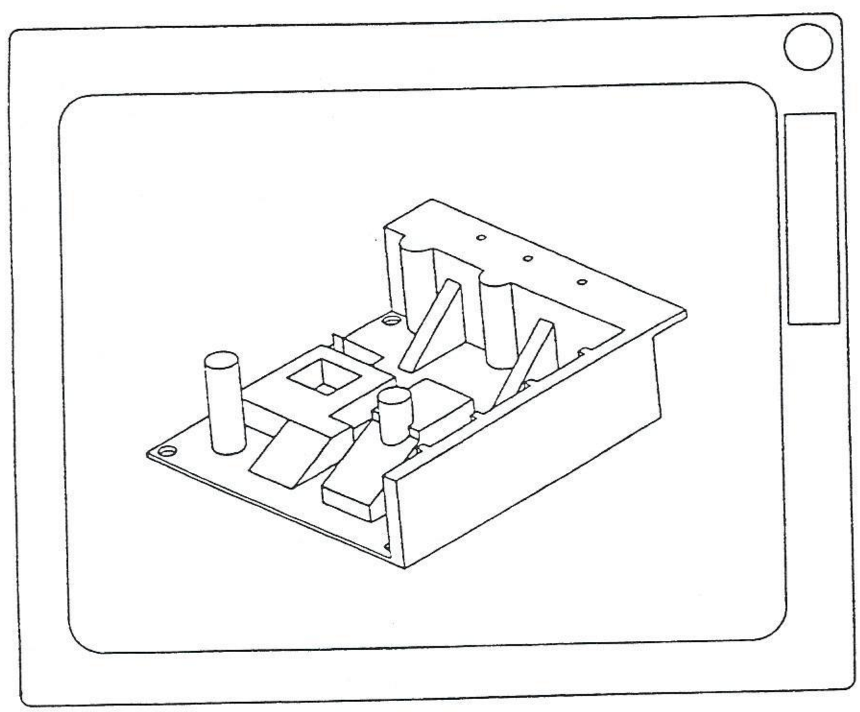

Figure 9: The Component Modified.

The above examples of PASE were oriented totally toward geometry. Another technology that gives DUDA the ability to capture data for later use is the use of design templates. A design template is much like a paper form with blanks to be filled in. However, unlike paper forms these templates are intelligent and flexible. The best way to explain this feature is to demonstrate one of DUDA's decision making templates. Specifically, Derald, a fellow designer working at another facility, and I am trying to decide on the location of a hole in a latch for a bolt to fasten the latch to a bracket. He is working on his DUDA and I am on mine. Communication is wireless and instantaneous. The dialogue is as follows with additional discussion after the dialogue:

Dave: "Where should we put the hole to fasten this to the that". Derald hears me and, when I touch the latch and bracket with my finger, these items highlight on Derald's DUDA so he knows "this" is the latch and "that is the bracket. Additionally, both DUDAs are now aware of the objects that are the focus of this issue.

Derald: Responds with a sketched hole and says "This is centered, o.k.?" The sketched hole appears on both DUDAs as a dashed line indicating that it is a proposed feature, not actually part of the latch. Also, I hear Derald's question.

Dave: "It looks too close to the edge", I touch the to specific edge and Derald can see this indicated on his DUDA, "it might tear out, let's move it back 10", the dimension is inferred to be 10mm based on size of component.

Derald: "Let me check that and get back to you". Derald recalls the size of the force from the original requirements and requests an analysis of the stresses and failure modes in the component. His DUDA performs an analysis and presents him with an evaluation. Part of the evaluation was the recall of material properties for the latch and an automatic estimate of a target factor of safety based on the level of knowledge about the problem. Derald says "Dave, it looks like the new placement is ok."

My DUDA inferred and recorded the rationale for the placement of the hole (as did Derald's). First it noted that I was working with Derald and that the issue we were addressing was the placement of a hole in a bracket. It recorded two proposed solutions for the hole placement and the arguments put forth for which to select. All the basic elements of a decision were recorded by DUDA: the issue, proposed solutions for the issue, arguments about the proposals, the objects that were referenced in the decision making process and the final decision. It makes no difference to DUDA whether the data is numerical or text, or whether a specific value or an abstract statement. DUDA has a template for decision making that allows the information to be captured for easy playback at a later time. For example, say that later on the bracket fails during loading, the bolt tears out of the latch material. I can easily recall the decision by verbally identifying the latch, selecting the hole by touching it and verbally requesting a history of the decisions that went into its placement. One of the decisions made was that in the conversation above. By reviewing this brief conversation between Derald and I concerning the placement of the hole, I will know why we decided to put it where we did. Since the material failed then at least one of the assumptions was incorrect. By reviewing how we arrived at the original decision, the error can be found and resolved.

Although the example above is a relatively simple one, DUDA can support much larger groups of participants on much more complex design concepts. Often many design proposals are generated to resolve an issue. The proposals may be too abstract to analyze and thus must be compared to each other in some relative manner. To aid in these types of decisions, there is a template for the decision matrix (Pugh's method), a method that gained popularity in the late twentieth century.

Another useful template for organizing design problems is for the development of engineering specifications. DUDA's method evolved from one that became popular in the 1980s, Quality Function Deployment. This method has three major goals: 1) identify the customers and their requirements, 2) translate these into engineering requirements, 3) compare existing solutions to both sets of requirements. DUDA's templates assist the engineer in developing the needed information.

DUDA has many other useful design templates, each with its own domain intelligence. Some are oriented toward mechanical engineering problems (e.g. functional modeling, function-to-form mapping and assembly assessment), some to electronic engineering (e.g. logic design and architecture layout) and some toward software design. In each, the system infers many details that the user does not have to specify. Some of the templates aid in controlling the flow of the product development process, some in performing analysis and some in information management.

What made the development of templates possible was an understanding of the cognitive nature of the product development process. Prior to the 1990s design was considered somewhere between an art form, a skill and an intellectual process. Calling it an art form was a way of saying that we did not understand how the designer progressed from problem to solution. Calling it a skill (implying hand-eye coordination) was left over from the days when engineers actually built what they designed. Calling it an intellectual process implied that the process was understood, which it wasn't. In the 1980s research began to fill in the missing knowledge on how designers solve problems making the development of templates possible.

Beyond design history recovery and the use of process aiding software, DUDA has other helpful capabilities. After the problem requirements or specifications are developed, DUDA has a number of features that help in developing concepts. The first of these is the extensive internal data base of catalogs and previous cases.

In the late twentieth century a design engineer spent a large percentage of time on the telephone or waiting for responses from letter enquiries for information about existing products. DUDA has greatly reduced this time with an internal catalog file in one of its memory cubes. This file contains over 100,000 catalogs that are updated on a subscription basis. Besides giving text and graphical information as they did in 1990, these catalogs have a number of features that aide the design effort. First, if more information is desired a picture from the catalog or a key word can be selected and a question asked of the supplier. This question can be either communicated by DUDA's cellular phone or the voice message can be translated to text and transmitted to the vendor via electronic mail. It is the users choice based on the urgency of the response. Secondly, what were just pictures on a printed catalog page in 1990, are now graphical solid models on the memory cube that can be made part of the evolving design.

Supplementing the catalog information DUDA has an extensive database of prior cases, captured from design histories, available on memory cubes. In the 1990s, the ability to perform case based design was developed. Case based design is dependent on a large collection of previous designs stored in memory. A new design problem is solved by retrieving cases that are similar to the new problem and then modifying and combining them to generate a solution. The chief advantage of case-based reasoning is that the system can function even when the database of cases is incomplete. Furthermore, the system can be improved easily and gradually, simply by adding new cases to the case base. The chief advantage of case-based reasoning is that a new design problem usually has a solution similar to a previous case, so old designs can be borrowed and slightly modified to satisfy the new requirements.

Historically, engineers have always used case based reasoning in their own thinking what worked on the last design will work on the next one. It is believed that in mechanical engineering design, engineers have past designs and partial designs stored in their memory, indexed by both their form and function. Thus, when a designer is presented with a new problem, the required function and spatial specifications are used to scan memory for a previous design with similar features. If a match is found, it is recovered and integrated into the design. If no match is found, some subset of the functional and spatial specifications is chosen and memory is scanned again. As partial designs are recovered, they are integrated into the design. This use of previous designs or partial designs (cases) indexed by function and spatial features is the heart of case based reasoning.

Additional features on DUDA that aide the designer are cost estimation, manufacturing process and tolerancing support. Since the designed product is continuously evolving on a single system and all the important features of it are in the memory of that system many manufacturing issues can be addressed during the product development process. Parallel to the evolution of the components and assemblies, potential manufacturing processes and their costs can be determined. Early versions of this technology were available in 1990 with expert systems. However, it was not until the mid 1990s that they were integrated into CAD systems and not until the development of DUDA that they were an integral part of the product development process even for unrefined, abstract concepts.

In Summary, DUDA has revolutionized the way design is performed. Its reliance on design histories and the use of design intelligent software based on a cognitive understanding, has afforded the design of higher quality products in less time. Using DUDA during the product development process, I am free to do what I do best - dream up new ideas and make decisions. DUDA does what it does best - manage information and present it to me. This is a strong design partnership.

References

Akin, Ö. (1990);
Voice Input Technology: Boosting CAD Productivity in Miles Per Hour; Dale Evans, CADENCE, Ariel Communications.

Akin, Ö. (1986);
Psychology of Architectural Design; Pion Limited, London, England.

Newell, A.; Simon, H.A. (1972);
Human Problem Solving; Prentice Hall, Englewood Cliffs, N.J.

Hwang, T.S.; Ullman, D.G. (1990);
The Design Capture System: Capturing Back-of-the-Envelope Sketches; accepted for publication by Journal of Engineering Design, September 1990; [illegible]

Conference on Engineering Design (ICED 90), Dubrovnik, Yugoslavia, pp. 893-903.

Ullman, D.G.; Dietterich, T.G., Stauffer, L. (1988); A Model of the Mechanical Design Process Based on Empirical Data; Artificial Intelligence in Engineering, Design, and Manufacturing, Vol. 2 (1) 33-52.

Nagy, R. D.G.; Ullman, D.G.; Dietterich, T.G. (1991); A Data Representation for Collaborative Mechanical Design; accepted for presentation at the 1991 ASME Design Theory and Methodology Conference, Miami Fl.

Ullman, D.G.; Wood, S.; Craig, C. (1989); The Importance of Drawing in the Mechanical Design Process; Computers and Graphics, Vol 14, No 2, 1990, also in The Proceeding of the 1989 NSF Engineering Design Research Conference, June 1989, pp 31-52.

Ullman, D.G. (1991); Design Histories: Archiving the Evolution of Products; Proceeding of the DARPA Workshop on Manufacturing, Feb 5-6, Salt Lake City, Utah.

Kuffner, T.; Ullman, D.G. (1990); The Information Requests of Mechanical Design Engineers; Design Studies, Vol 12, No 1, Jan 91, pp 42-50; also in Proceedings of the 1990 Design Theory and Methodology Conference, Chicago, pp. 167-174.

Yakemovic, K.C.B.; Conklin, J. (1989); The Capture of Design Rationale on an Industrial Development Project: Preliminary Report; MCC Technical Report Number STP-279-89.

Yakemovic, K.C.B.; Conklin, J. (1990); Report on a Development Project Use of an Issue Based Information System; MCC Technical Report Number STP-247-90.

Research in design thinking

Proceedings of a Workshop meeting held at the
Faculty of Industrial Design Engineering,
Delft University of Technology,
The Netherlands,
May 29-31, 1991.

Edited by N. Cross, K. Dorst and N. Roozenburg

Delft University Press