Integrating Creative Steps in CAD Process

Pedro Company-Calleja

Departamento de Tecnología, Universitat Jaume I
Campus Penyeta Roja - E12071 Castellón, Spain
e-mail: pcompany@tec.uji.es

Abstract

Despite very important advances in CAD, engineers and designers still prefer pen and pencil, especially in the more conceptual steps of design process, in which only an incomplete set of requirements and abstract ideas about the design are known. The reason is that CAD systems are too rigid to allow a fast scan of incomplete and non-formalized ideas and models. Moreover, engineers and designers are trained on *Engineering Drawings*, that has proved to be a powerful and flexible tool for the whole design process, and constitutes the, almost universal, engineering language. That is, Engineering Drawings is the "natural" language engineers and designers use in the design process. The consequent proposal is using Engineering Drawings as the interface language in the whole CAD process.

But for the Engineering Drawings to become the link between users and CAD systems some present limitations must be overcome. First of all, Engineering Drawings convey implicit 3D information, while CAD systems need an explicit representation of 3D objects. *Geometrical Reconstruction* will allow "explicitation" of 3D geometrical information contained in standard 2D representations. The part of the information that is contained in Technical Drawings but is not represented by means of geometrical projections (dimensions, tolerances...), must also be "understood" by computers. Finally, mismatches, errors and "complicities" present in all Technical Drawings ought to be filtered by computers. In the other hand, non-well formalised aspects of the language must be complete and univocally defined.

In other words, Engineering Drawings is a language that allows engineers, designers and other people implied in technological processes to communicate almost all kind of information related to design. So the goal is to use the *full language* to communicate with Design Systems. This is the key for a real easy-to-use and user-oriented interface between CAD system and users.

1. INTRODUCTION

Very powerful and specialized tools carrying out a large variety of analysis, to determine the goodness of new designs, are now market commonplace. Engineers and designers are also becoming increasingly trained in such tools, and, therefore, many people begin to feel more comfortable using those tools than with oldest and more heuristic ones. In addition, in the earliest steps of design the analysis-oriented tools are useless. The reason is that all those tools need complete and consistent models, and this is a too restrictive condition when only an incomplete set of requirements and abstract ideas about the design are known. This is especially true in conceptual steps of design process, in which an ill-defined problem that has no single "right" answer is faced. If computer tools are to be used to explore tentative ideas in a design process, a formal and complete representation cannot be required as a previous condition.

Furthermore, to test whether a concept is viable, or to get a feel for the general performance of a concept, we can use computer tools for symbolic calculations (like Mathematica) or even spreadsheets [1]. The idea is to explore the design space taking away the directional nature and complexity of predefined analysis tools. Typical analysis tools can get the output of a particular performance related to a set of design variables only after design equations and design parameters have been fixed. In the contrary, a symbolic calculator can allow you to change some of the output performances directly, or to redefine some of the design parameters as variables, and directly observe how design variables have been affected. This alternative has proved to be efficient in parametric models with a relative small number of governing equations.

Nevertheless, symbolic manipulation is feasible only when the concept been studied can be expressed in mathematical terms, and this is not the common case in the very early stages of design processes. As pointed out by Ferguson: "Pyramids, cathedrals and rockets exist not because of geometry, theory of structures, or thermodynamics, but because they were first pictures -literally visions- in the minds of those who conceived them" [2]. Consequently, and in spite of the dominance of mathematical formalism in the curricula of modern engineering schools, engineers and designers still tend to think visually. Following Bertoline we can say, "once you know the language of graphics communications, it will influence the way you think, the way you approach problems" [3].

We can conclude that graphical language is the best alternative for the designer to communicate with CAD system. Nevertheless, two main problems need to be solved in order to adapt the "communication channel" from man-to-man to man-machine communication:

a) Up to date, CAD systems cannot "read" all information contained in technical drawings.
b) Technical drawings have evolved to become a highly standardized language, quite complex and with a low redundancy level, but some conventions are still needed to get univocal interpretation of technical drawings.

The reason for the first problem is that Engineering Drawings convey implicit 3D information, while CAD systems need an explicit representation of 3D objects. Information contained in Technical Drawings but not represented by means of geometrical projections (dimensions, tolerances...) must also be "understood" by computers.

It is very important to remember that already existing designs suppose an important "know-how", and are specified in Engineering Drawings. This means that automatic solid-model generation from standardized drawings may be the "bridge" to recover the information related in the thousands of old designs filed in drafting rooms.

The second problem resides in the fact that apprenticeship of Engineering Graphics includes learning some non-formalized rules. For instance, the "simplicity criterion", sometimes expressed in the following terms: "the geometrical form represented is the simplest one that matches with current views". The problem is that those rules must be incorporated to the man-machine communication, or made unnecessary by an improvement of language specification. Furthermore, non-geometrical and a priori conventions (like graphical semantics and visual stimuli described in Gesthalt rules) are implicitly incorporated in technical drawings, as they are in all graphical communication [4, 5].

To summarize, in this paper we will briefly discuss the role played by Technical Drawings in the design process up to date. Later on we will present the state-of-the-art in both, Technical Drawings and Computer Aided Design interfaces. We shall put the emphasis in the main aspects we need to solve to convert Technical Drawings in a comprehensive and powerful communication language between designers and CAD systems

Before, we must remember that CAD means Computer Aided *Design*. So, to determine which kind of "Aid" computers can bring, the meaning of "Design" must be fixed in advance. More precisely, we shall talk about "Technical Design", meaning that we include not only aesthetics, but also functional specifications.

2. DESIGN

According with Suh [6], we can define Design ... "as the epitome of the goal of engineering [that] facilitates the creation of new products, processes, software, systems, and organizations through which engineering contributes to society by satisfying its needs and aspirations".

To facilitate the creation of new products and processes, Technical Design uses scientific principles, technical information and imagination, in the search of the "optimal" solution. Where optimal can mean tuning for one single criterion (like maximization of economy, efficiency, etc.), or a combination of differently weighted criteria. It is especially remarkable the main role imagination (or intuition, or experience) plays in design. Designing means choosing against uncertainty. It is an ideation or creation process. However, design is also founded on technological and scientific knowledge, because an optimal solution is always the objective. In fact, we cannot make good design decisions in the absence of a criterion for selecting a good design.

2.1. Design process

Only rarely, we can immediately select a completely satisfactory solution to a perceived environment need. Therefore, the activity carried to obtain a new "*design*" can be modeled as a *process*.

Notice the above use of "design" to refer to the set of requirements, constraints and variables that fully describe the "thing" that still doesn't exist. We deal with information, not with the product or the process itself. In other words, we define a "model" rather than the object.

Returning to the "process" idea, there is no general agreement about the nature of design process. Sometimes it is considered iterative, and sometimes it is considered concurrent. In summary, there have been many unsuccessful attempts to establish a generic and systematic description of the process followed to generate a new design. The reason, as pointed by Encarnaçao [7], is that..."design process is ... complex, and that neither a chain nor a tree is sufficient to represent its essential characteristics, even though it may sometimes look like a chain or a tree in certain respects".

Nevertheless for our study, a crude model reflecting the four basic tasks that are carried out in a design process (specification, synthesis, analysis and evaluation) may be enough (figure 1). In the proposed model, the tasks are accomplished in a well-defined order and in a cyclic process. The model is the same for "complete" design (to solve an environment need) and "detailed" design (to solve one single step in a hierarchical design process).

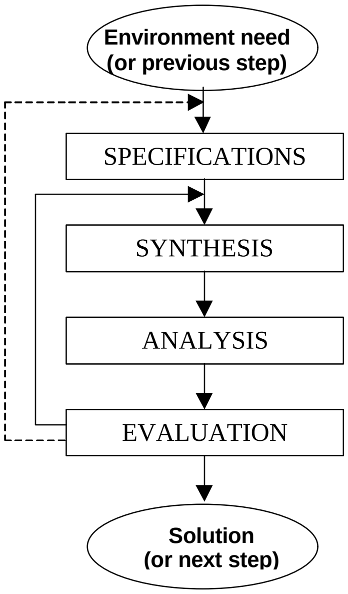


Figure 1. Crude Design process.

The specification is the problem definition in terms of requirements and constraints. Redefinition of specifications may sometimes become necessary, because it is difficult to judge when a set of requirements and constraints do correctly represent the perceived needs, until the proposed design is evaluated. (See dot line in fig. 1).

In the synthesis part of the design process, certain characteristics (or "design variables") are chosen. Again, decisions are made under incomplete, or even null, knowledge about their consequences with respect to the design goal. When requirements and constraints are completely defined and mathematically formalized, optimal synthesis, in the sense of "automatic search for the best alternative" can be done. On the contrary, when an incomplete or non-formalized problem is faced, synthesis becomes a fully creative or "ideation" process. Because it is highly subjective and heavily depends on the specific knowledge possessed by the designer, and his or her ability to integrate knowledge.

As a result of synthesis process, a design solution is obtained, whose behavior is analyzed and evaluated against requirements to determine ("validate" or "verify") its goodness. It is also important to notice that if we cannot analyze a design solution then we cannot generate the "best" design, since we cannot distinguish a good design from a bad one.

If the design deviates from specifications, the selection done in the synthesis part must be appropriately corrected, and new values for design variables must be obtained to improve the result (See backward continuous arrow in figure 1).

To conclude the design process, the design information must be documented and passed to the next step in the life cycle.

2.2. Differences Between Conventional Design and Computer Aided Design

Two are the main differences between computer-based and man-based design processes:

a) Information processing by human does not require a formal representation, while computers can process information only if it is represented in some formal way.
b) Computers can "crunch" a large amount of numerical data, while humans only can cope with a small set of information.

In the other hand, the first goal to achieve for using computer in the design process is to reflect the complex structure of the design process in the structure of CAD systems. Afterwards, those phases that have got completely defined can be automated using all CAD technologies. Nevertheless, integration needs special care if such systems are to support the design process as a whole and not only isolated parts of it.

In figure 2, all CAD technologies (all computer technologies) added to assist in the automation of different design phases are summarized:

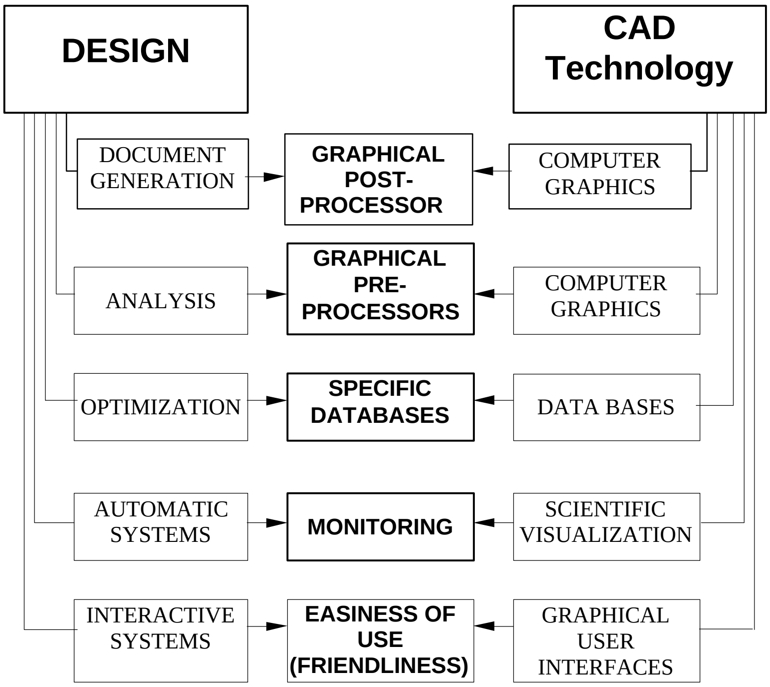


Figure 2. Computer technologies used in Design process.

As we can see in the previous figure, different steps have been needed in the very challenging goal of CAD technologies integration in design process. Mainly because the advent of those technologies, the need for them to be used, and the arrangement of appropriate computers and peripherals all have required a progressive incorporation. Most remarkable milestones are:

- Graphical post-processors to generate Technical Drawings associated with different kinds of designs are still improved, but appeared in the early 1970s and configure a mature field. This phase is now becoming an automatic rather than an interactive task.
- Graphical pre-processors, to generate tree-dimensional geometric models, were developed to support analysis tools like Finite Elements, to describe and handle mechanical parts for the manufacturing processes simulation and even for the definition of synthetic images of 3D scenes. (In fact, Computer Graphics rendering techniques have been incorporated in graphical pre and post-processors). Now, pre-processors are commonplace in the description of geometrical topology in CAD systems, and the convergence to a unique model and/or automatic translations between different models is the objective.
- Specific databases first appeared when numerical analysis techniques required it. They were improved when optimization algorithms were agreed to design systems.

Nevertheless, the definition of a global database encompassing the whole design process is still an ambitious field with many economical interests at play.

- Some monitoring capabilities were added as a complement to complex numerical analysis techniques (for instance, visualization of deformation, stress, thermal or other kind of analysis). Yet, only geometrical information, or information clearly associated with geometry is commonly displayed in CAD systems up to date.
- Finally, Graphical User Interfaces (GUI's) are present in almost all CAD systems since their advent. In fact, non-GUI user interfaces have been present in CAD systems since the beginning, and the enormous improvements after done in GUI's have completely been incorporated in CAD systems.

We must conclude emphasizing neither databases nor scientific visualization techniques have expanded to assist in the whole design processes. Moreover, transference of information between different models, necessary carried out during design processes, is a very time-consuming and "prone-to-error" task. As Encarnaçao states in [8], "it was much simpler to build CAD systems whose sole objective was to ease the task of drawing, rather than to worry at the same time about how the design results could be interfaced with manufacturing or assembly process". Therefore, "price must today be paid whenever one tries to assemble these isolated solutions into larger, integrated systems".

We also must point out that the paradigm for the whole integration is absent. Implementors have no pattern to follow. We strongly believe that enhancing and simplifying the communication between designer and CAD system is the key. For this purpose, Engineering Drawings is the best choice. To justify this assertion we are going to study the historical and present role of drawings in design.

3. THE ROLE OF DRAWINGS IN DESIGN

When faced to the most challenging problems, designers use prototypes. Because prototypes are a useful way to test alternatives to a large variety of specifications the designer has to fix in the process known as design.

The prototype can be "reduced" to a mental one only when the designer deals with a "know-how" design problem. That is, when the designer has some familiarity with the problem addressed, due to some previous experience. Such prototype which only exist in the designers' mind is what Ferguson calls de "Mind's eye" [2]. When the problem becomes more complex, the mind's eye cannot cope with all details. Mind's eye can still be useful for the overall design, but a formalized model is needed to complete the design.

Fixing the geometry that accomplishes with all the specifications involved in the design is one of the main problems in many design situations. Accordingly, definition of geometry and study of geometrical compatibility are many times the "core" in design processes of mechanical parts, assemblies, and even small systems. This is the reason that led to the development of so-called "design-by-drawing" method. In this method, geometrical study is done by using one formalized body of knowledge known as "descriptive geometry", where the physical prototype is advantageously substituted by well formalized Engineering Drawings.

In one sense, Computer Aided Design means an evolution of design-by-drawing method. Because fixing the geometry that accomplishes with all the specifications can be done, using the "geometric modelers" embodied in much CAD systems. However, we will see that a conceptual improvement is involved in the change: a virtual 3D prototype is directly generated and manipulated by 3D CAD systems.

There is a historical confusion between design and drawing; enhanced in the recent past by the ambivalent use of CAD as both Computer Aided Design, and Computer Aided Drafting. Emphasizing the difference between design and drawing is of great importance. As Booker states [9]: “Engineering drawing is not, however, the same as engineering design; neither are the two inseparable as some persons suppose, for a medium of expression can generally be isolated from what is expressed through it”.

3.1. The "traditional" Design-by-Drawing method

The Artisan’s way of evolution can be defined as a gradual change based on trial and error. This evolution is carried out on the successive units of the same product and drastic changes are out of scope. (The reason is that artisans evolution assumes that modifications affect only locally to the product). Furthermore, the evolution is based on the learning memories of the apprenticeship period of the artisan. Hence, historical moment and local geography where artisan lives greatly influence his evolution process.

It was the introduction of drawings in this “artisans’ process” that made the difference between craftsmanship and design [10]. In fact, design appeared when global modifications (affecting the whole product) were considered. In the other hand, the most basic design process was based on experience and experimentation. When experience is short or the proposed modification is quite radical, the process tends to “diverge”. Resulting in false solutions (not meeting all requirements) or non-acceptable increments in the cost of design process. Hence, the jump from artisans’ process to design was based in three qualitative changes that Technical Drawings made possible:

a) Technical Drawings permitted useful devices and ideas to be recorded. The bottleneck of artisans memory was broken. In other words, geographical and time barriers disappeared. Many picture books (notebooks and “theaters” of machines), and dictionaries of engineering were the seed for numerous new design ideas.
b) Drawings allowed cheap and fast exploration of new ideas. Neither the cost nor the time prevented from drastic changes to be explored. This was especially true with the use of non-formal (and consequently less restrictive) drawings like sketches.
c) Symbolic transmission of information made possible the division of labor. This division allows as much the increasing in the size/complexity of products, as the increase of productivity.

However, for Technical Drawings to let those changes, it was necessary for it to previously accomplish two conditions:

a) Geometrical coherence, in the information contained in Technical Drawings. This condition was obtained with Descriptive Geometry.
b) Univocal definition, of information contained in Technical Drawings. Standardization of Technical Drawings ensured the need for univocal definition.

Graphical tools traditionally employed in design are summarized in figure 3. Three are the graphical tools employed in the design-by-drawing method process (sketches, descriptive geometry and standard technical drawings):

- In the ideation phase, rough sketches, called ideation drawings, are used. Sketches improve creativity because allow a rapid record and communication of new ideas. What is more, polishing drawing sketches encourages new ideas to evolve from existing concepts.
- The Descriptive Geometry (or *Constructive* Geometry as defined by Hohemberg [11]) is used to synthesize a three-dimensional geometric model, and to simulate 3D geometrical compatibility and behavior in a 2D support.
- When design process finishes, Standards ensure the effectiveness of graphic language employed in the final documentation.

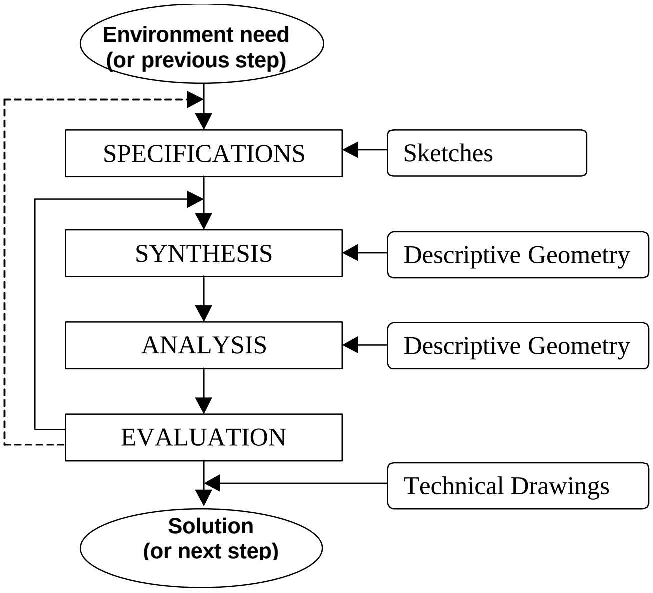


Figure 3. Graphics in the "Traditional" Design-by-Drawing Process

It must be emphasized that Descriptive Geometry (a discipline born in nineteen century and founded on Euclidean and Projective Geometry's) was an innovative tool in two different aspects. In one hand, it was the analysis tool used the first in design process. We must remember that the obvious importance of analysis tools is that they replace experimentation by simulation. Therefore, more alternatives can be explored in a reasonable time. In the other hand, synthesis and analysis are *simultaneously* done in the design-by-drawing method. Descriptive Geometry forces three-dimensional geometrical coherence to be accomplished every time a modification is done in the drawing. The *same* drawing that serves to analyze the functionality and behavior associated with the change.

3.2. Computer Aided Design

The Computer Aided *Drawing* tools (sometimes referred as CADD) were the first generation of CAD systems. They supposed an increase of productivity. Mainly because a lot of elementary, and tedious, geometrical constructions were automated.

Euclidean geometry declares "legal" all constructions done with the only use of rule and compass. Under that condition, one straight line passing through two points can be directly constructed; but a second line, parallel to the first one and passing through a third point requires some auxiliary constructions. Nevertheless, if we accept drawing triangles as valid instruments to carry out some legal constructions (including the parallel line construction), then we can draw the parallel line with a single operation. Using CAD terminology, we can say that the straight line passing through one point and parallel to a predefined direction becomes a "primitive" for a draftsman using drawing triangles. While the line passing through two points is "always", a primitive (with the only condition we have a rule). It was the increasing number of primitives introduced in CADD systems that made the difference between classic draftsmanship and Computer Aided *Drafting*.

In the most "depressing" situation, an elementary CADD system would only emulate rule and compass. Even in this situation, the possibilities of storage, modification and copying of technical drawings would be an impressive advantage in terms of productivity and improvement of precision. In fact, this "revolution" can be compared to the one promoted by the advent of printing (See "The hazards of Copying Technical Drawings" in Ferguson [2] for a detailed description of printing implications in Engineering).

We can conclude that, with the advent of CADD systems and besides the economical importance of the improvement of productivity, no new conceptual scenario was created: design-by-drawings continued to be supported by the Descriptive Geometry.

On the contrary, real three-dimensional Computer Aided Design systems (CAD 3D) have the capability to create and manage geometrical forms in three dimensions. Then the need to simulate three-dimensional models trough plane representations disappears. Also does the need for univocal plane representation of the model (which is the objective of the so called "representation systems" in Descriptive Geometry). Then, the real improvement is that the user creates what Cugini [12] calls a "digital prototype".

In figure 4, the CAD process is presented, and the role of graphics in that process is described. First (trough appropriate pre-processors) the digital prototype is defined, and compatibility is ensured. Then, the system can be asked to use the information contained on it to simulate different behavior aspects. Later on, graphical post-processors help the user to interpret results and evaluate the solution. Finally, CAD/CAM processors help in the generation of detailed "outputs" of information in the standardized language used in the technological community.

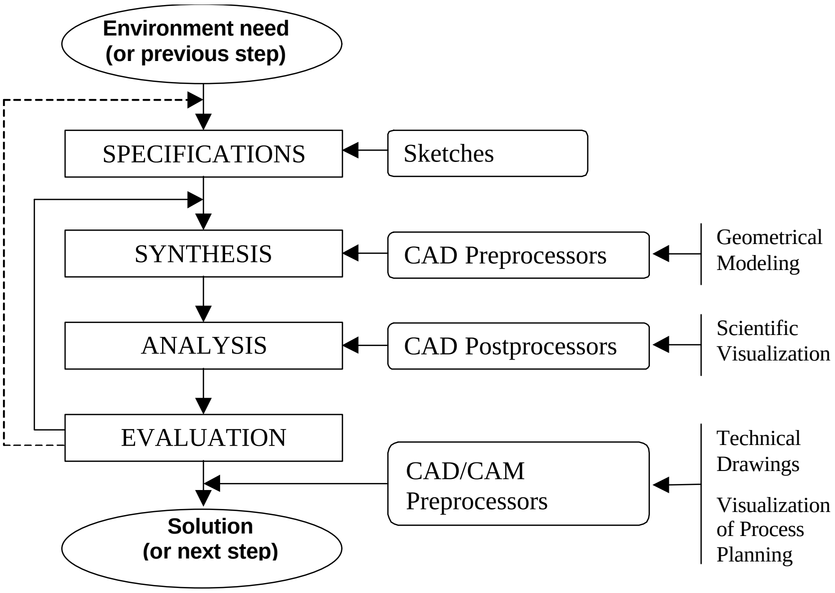


Figure 4. Graphics in CAD Process

Geometrical modeling substitutes Descriptive Geometry, and Scientific Visualization enhances the simulation capabilities of analysis tools. Yet, up to date, one important problem remains in the definition of geometric models: full topology is to be known *in advance*. What is more, non-parametric systems require *complete geometry* (form and dimensions) instead of “only” needing topology to begin the prototype construction.

Another main problem, related with the previous one, is that sketching capability (that has not been increased) continues “unplugged” to CAD systems. With CAD systems, a scanner can be used to scan an existing image into the computer. Nevertheless, scanned images enter the computer as “raster” images (that is, a tight grid of dots of varying colors), not as geometric models. Nevertheless, now, some “sketch modelers” are being introduced. Sketch modeling is a technique in which rough 3D computer models are created by the designer and easily modified. The technique provides the visual feedback necessary for schematic design. Its finality is that, early in the design process (during the idea generation phase), models can be constructed quickly so that the design ideas can be tested. In sketch modelers speed is more important than geometrical accuracy. However, a new tool is required to help designer in the phase of fixing ideas. Such a tool would be able to “capture” the ideas generated by the designer and automatically generate the solid model. In other words, the solid model must be made “transparent” to the designer (it must be an *internal* model for the CAD system). For this purpose, one language oriented to creativity enhancement must be defined for designer-CAD system communication.

The third problem is that no new paradigm for design documentation has been defined and standardized. Of course, generation of a large variety of Standardized Technical Drawings has been assisted or automated. In fact, most CAD systems can create standard technical drawings for design, manufacturing and other purposes with little help from the user. Even, some CAD/CAM systems can automatically convert from design models to manufacturing models; but no *standards* have been defined to convert CAD models and CAM models into single comprehensive CAD/CAM models. Moreover, a paradoxical situation can happen when CAD and CAM phases are not carried out in the same place: design information contained in a digital prototype, can be converted to traditional manufacturing planes (the “legal” way to specify “what “ is to be constructed), and later reintroduced as a 3D model in a CAM system.

To summarize, the graphic implications of CAD process start with the ability to visualize, to see the problem and possible solutions. There, sketches are made to synthesize initial ideas. Next, geometric models are synthesized and are used for analysis purposes. Finally, detailed drawings are automatically obtained; to record and transmit the precise data needed for the production process. In all this process, it must be noticed that:

a) The need to construct geometric models remains. In other words, the user creates a digital prototype, but the creation task is not assisted by the computer, in the sense of been “linked” to previous conceptual synthesis task.

b) The need to visualize the behavior increases in parallel with the growth of number and complexity of analysis tools. Even the mere existence of a digital prototype augments the simulation capabilities of analysis tools.

c) No new paradigm for design documentation has been created. Consequently, generation of a large variety of Standardized Technical Drawings has been assisted or automated, but not replaced or simplified.

3.3. Scientific visualization

Visualization is a stepwise transformation from information into images [13]. For historical reasons, in the computer world, visualization of non-geometrical information is associated with scientific research, while geometrical modeling and technical drawings are exclusively associated with design. Nevertheless, the general concept of “visualization” is clearly a general one, and encompasses graphical representation of *all* kind of information. Consequently, we do believe that a general-purpose visualization system must serve for both, scientific and technical processes [14]. A visualization system for scientific or technical purposes allows engineers and scientists a quantitative and global observation of numerical calculations.

We can trace back the origins of non-geometric data visualization in the next sentence by Tufte [15]:

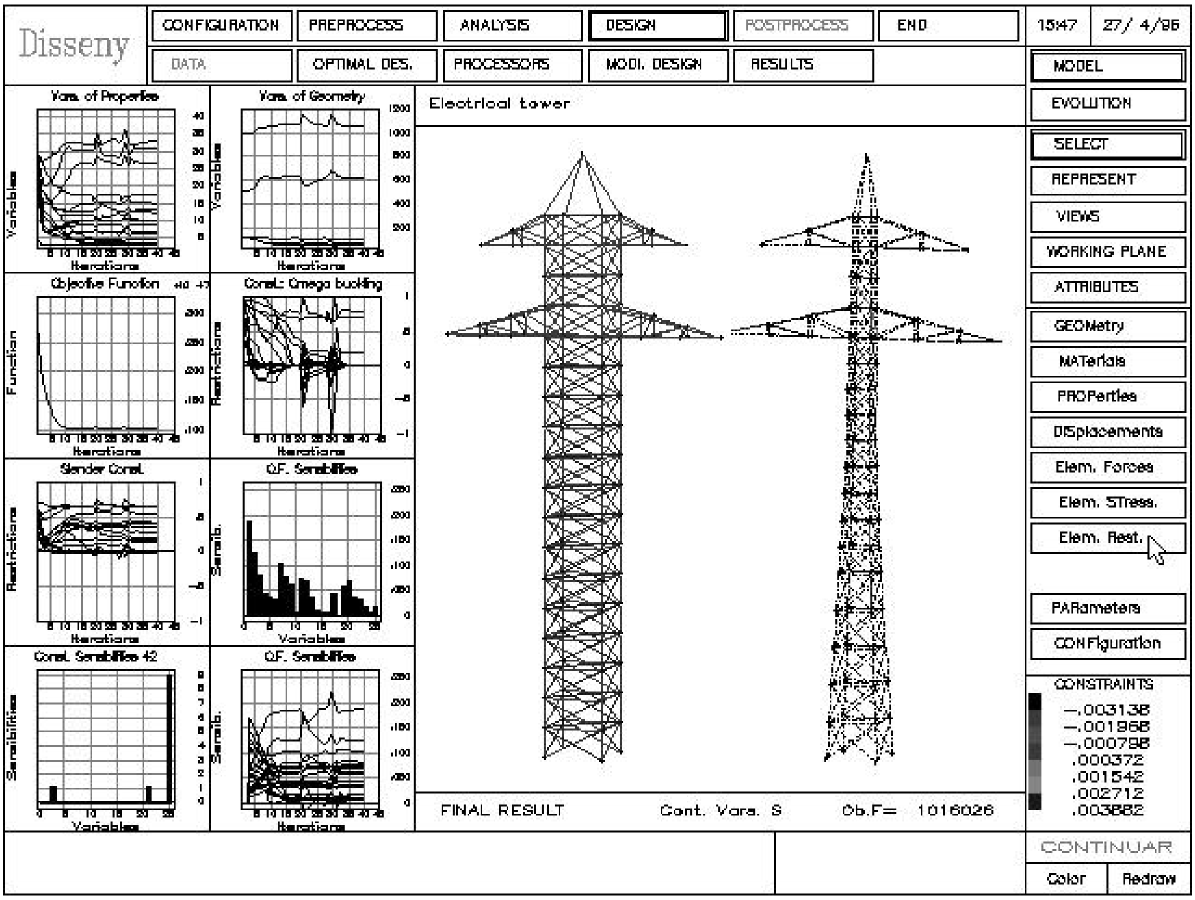
“The use of abstract, non-representational pictures to show numbers is a surprisingly recent invention, perhaps because of the diversity of skills required -the visual-artistic, empirical-statistical, and mathematical. It was not until 1750-1800 that statistical graphics – length and area to show quantity, time-series, scatterplots, and multivariate displays- were invented”.

The conversion of “data visualization” in “*scientific* visualization” is quite more recent. The change is due to the introduction of computers in the generation of images to represent sets of data. Computer community renamed the discipline. Also new is the “explosion” of the discipline, due to the need to manage the growing amount of information that automation of different phases in research and design processes produces. Those processes can generate a lot of information from a relative small quantity of departure data. Moreover, this information is generated at a faster speed than the maximum allowed for the assimilation process to take place.

We can conclude that scientific visualization is a well-defined discipline (See [4] and [15] for foundations). Nevertheless, computer graphics community still has not completely adopted theoretical foundations, and, consequently, still works in an empirical context. In addition, some resources need further development (See [16] and [17] for example). Yet, the main problem is the integration and use of that part of graphic language in the design community. Majority of designers still tends to believe that “advertising” graphics are superfluous in the design process. In addition, only a small group of designers is trained enough to avoid “lies” when they convert non-geometric information into images. (We lie when we convert numbers into figures without maintaining in the graphics the resemblance, order, proportion and neighborhood relations that were present in the original numbers).

Up to date situation can be summarized stating that scientific visualization (especially when computer is the drawing tool) is being slowly incorporated in analysis and evaluation processes for “monitoring” tasks. A single image can serve as an example of actual integration of graphical resources. The image shown in figure 5, is a copy of the screen in DISSENY, obtained when a session of optimal design for an electrical transmission tower has concluded. DISSENY is a general-purpose structures and structural elements optimal design system [18]. It is based in the finite elements structural analysis program ADEF [19], in two non-linear mathematical programming algorithms, and in some processors carrying the finite elements-optimization techniques coupling [20]. It has, in addition, some different modules to solve specific problems, such as basement optimization for electrical transmission towers [21], thin-wall profiles optimisation [22], optimal design under non-linear behaviour [23], etc. Some graphics can be highlighted in the figure:

- The design model appears represented in the great window in the right side. The initial tower appears in the left side of the window, and final design in the right one.
- Some characteristics of the GUI can be highlighted: The colour map in the small window in the lower right corner is associated with final design, and shows the violation level of constraint stresses in all the bars (the colour coding appears in the figure like a Grays scale).
- Five of the windows in the left side contain representation of different evolution values associated to the process (objective function, variables of geometry, variables of properties, buckling constraints and slender constraints). The other three windows (the lower ones), display sensitivities of objective function (both the evolution and the present values) and the sensitivities of the most violated constraints, with respect to every variable.
- At last, and surrounding all previous windows, menu bars (up), utility menus (right), and active submenu (selection menu), are placed.



FINAL RESULT Cont. Vars. 5 Ob.F= 1016026

Figure 5. Screen image, during an optimal design session in DISSENY.

We can conclude enhancing that monitoring of the most routine phases in design process is becoming a common practice in design process. Now, the new challenge is the use of appropriate scientific visualizations in the synthesis phase of design process. This use requires knowledge of what information is to be presented and how best to present it. It also requires breaking the ignorance and resistance of majority of designers and engineers to use what they pejorative name “advertising” graphics.

3.4. CAD languages

Communication between designers and CAD systems is unbalanced in favour of programming needs. In today’s state, CAD systems force designer to control a sequential flow, directed from specifications to detailed design.

Really, implementations of design process in CAD systems do not follow a strict flow, because the two main loops shown in figure 2 are generally considered. But it is not enough, because we have stated that only in a very simplest way can design be considered as a sequential process or even a loop process. In other words, to reflect the simple flow shown in figure 2, do not mean to reflect the complex structure of the design process in the structure of CAD systems.

Sequential nature of algorithmic languages is the reason, because those languages are in the back end of today’s computer tools (and Graphical User Interfaces are not an exception to this rule). The need for programmers to define an implementation model of the process to be executed reinforces the sequential tendency. Because, for programmers, defining a process “conceptual” model (“what” the system can do) as close as possible to the “implementation” model (“how” it does it) is always the simplest solution. As a result, designer is continuously asked for *actions* (well-defined and sequential actions), to be done by CAD system. And this is not a good strategy when the designers is trying to fix “visions”, that is, ill defined and non sequential ideas.

Transparent commands can give the wrong impression that user can do almost everything in almost every moment. And, in fact, CADD systems (*drawing* systems) are highly “interactive”, because they impose few limitations to “wanderer” users. But we must remember that the reason is that they are based on Descriptive Geometry and Technical Drawings, both non-sequential disciplines. Or better said, disciplines based on non-sequential languages. Unfortunately, this is not the case of real 3D CAD systems (*design* systems). CAD systems can create virtual three-dimensional models that, in turn, can be shown in pretty rendered (and, of course, graphic) images. But the construction of those models is strictly sequential. One single action follows every command, and the system turns back to the “neutral” state waiting for the next explicit command.

Moreover, sequential flow and a complete and explicit model are needed in some design phases. Design flow in a CAD system needs a formal representation of all information been handled. Of course this is an unavoidable condition for the most routinely analysis parts. It would be absurd trying to find an optimal option by analysing different options and comparing them in the base of incomplete and ambiguous models. But this requirement to obtain the optimal solution when the problem is completely defined, becomes a barrier to obtain a good or simply a valid solution (based on experience) when the problem is ill defined.

To sum up, our problem is that “non-verbal” thought cannot be expressed in a “verbal” language. Verbal is defined as synonymous of sequential. That is, verbal languages are based on variations of a set of signs along the time, never mind when the signs can be sounds or graphical forms. On the contrary, non-verbal (or “graphic”) languages are those in which transmission of information is based on the meaning of a predefined set of signs, but also in the spatial relations among all signs. That is, the resemblance, order, proportion and neighbourhood relations present in every written communication (and necessary absent in oral communications). It must also be noticed that in non-sequential communication, the time needed and order followed to write and read the message does not affect the information.

If computers could run processes to explore Engineering Graphics, they ought to be able to extract both explicit and implicit information. In addition, not only analogic (geometric) but also symbolic (non-geometric) information ought to be read from Engineering Graphics. Finally, mismatches, errors and “complicity” present in all Technical Drawings ought to be filtered by computers.

The utopian objective would be a Design System able to integrate all information contained in a sketch interactively during the sketch creation and refinement phases. Able to formalize the non-formalized ideas contained in the sketch. And able to analyse and evaluate the provisional model, and give the designer a feedback on the performance of the intended idea.

4. STATE OF THE ART IN ENGINEERING GRAPHICS

We have traced the state-of-the-art in the use of Engineering Graphics in the design process, and we concluded defending that the next step must be the definition of a really easy-to-use, user-friendly and comprehensive interface between designers and CAD systems. We also pointed out that the paradigm for that interface is absent. We strongly believe that Engineering Graphics is the best choice for this purpose. To justify this assertion we are first going to study the present state of the art in Engineering Graphics, and future improvements to be attempted in this discipline.

4.1. Classification of Engineering Graphics

The comprehensive term “Engineering Graphics” can encompass all kind of graphic representations related to design process, but some more specific denominations are usually employed:

- “Sketches” is the generic denomination for those drawings where geometric rules are not strictly followed.
- “Descriptive geometry”, used for those drawings done in accordance with all geometric rules, and employed to synthesize a three-dimensional geometric model, and to simulate 3D geometrical compatibility and behavior, in a 2D support.
- “Technical Drawings” is the name employed in Standards (like ISO, ANSI, DIN, etc.) to refer to the graphics applied in Engineering to present products, or processes, in a conventional form.
- “Data Graphics” (or “Statistical Graphics”) is a common way to reference non-representational pictures used to show numbers.
- “Scientific Visualization” is a new identification, which encompasses statistical graphics and the representational pictures used to show scientific processes.

Previous definitions are mainly referred to the nature of the information been represented (or the *contents*). Very different classifications can be obtained attending to other aspects. For instance, during the design process, and depending on the nature of the intended communication (the *purpose*), two kinds of graphics can be used: *descriptive* and *predictive*. A descriptive drawing presents products, or processes, in a recognizable form (A manufacturing drawing of a mechanical part is a descriptive model of a product and a process). A predictive drawing is used to understand and predict the behavior and/or performance of products, or processes (A finite element model is a predictive one).

In a different approach, we can consider Engineering Graphics as a language used for communication and in this sense is related with standardized conventions. However, languages are not only useful for communication; they play an inherent part in our thinking processes (it can be said that we use languages to "dialog" with ourselves), and, there, psychology and perception rules play the most important role.

Nevertheless, all previous classifications are clearly dependent each other. We can study some other classification, and we shall conclude that there is no one complete and independent Engineering Graphics classification. Yet, using some widely accepted classification (like the one contained in DIN 199), we can determine four "quasi" independent aspects that are to be considered in order to classify any Engineering Drawing:

- The class of the representation.
- The confection procedure.
- The contents.
- The purpose.

Referred to the class of the representation two kinds of Engineering Graphics are considered. When Engineering Graphics contains only incomplete information and the signs and figures used are to be interpreted only in an approximate sense, the representation is said to be a "sketch". In the opposite case, when a complete, exact and exhaustive information is represented, the Engineering Graphics is said to be a "plan". The distinction is important because sketches are not used as contractual documents as plans are. In addition sketches use to have a short life period while plans are filed and form part of the industries history. On the contrary, a plan must be "auto-contents" (it must require no complementary explanations) while a sketch is usually complemented with verbal explanations.

In the confection procedure, the main distinction must be done between "freehand drawings" and "line drawings". Where the former are all drawings done without instruments and the latter are those done using geometrical instruments. The main difference is, of course, the geometric information implicitly contained in line drawings. In other words, it is "legal" to measure (respecting geometrical procedures) in a line drawing to extract dimensional information; while only proportions and some other geometric characteristics (like symmetry, parallelism, and so on) can be derived from a freehand drawing.

Attending to the contents, we can distinguish among general (or assembly) drawings, group (or sub-assembly) drawings, and detail (or part) drawings. The distinction is very important because many specifications are needed to completely define a design. Therefore, a good classification process is the base to store and retrieve information as needed. In this sense, the hierarchical structure is seemed as a good solution, because it matches very close to the design structure, and allows a full integration and maintains a clear and fast access to every part. In addition, the hierarchical is a good structure to hide low level details when they are not needed.

Finally, Engineering Graphics can take different forms depending on the "audience". More precisely we can say that the dependence is on the amount of information (required clarity, precision and level of detail) the receiver requires and/or can process. Therefore, depending on the purpose of the Engineering Graphics (i.e. the audience), three forms can be distinguished:

- Engineering Graphics made for personal use, that are not meant to be understood by anyone but the individual who produced it.
- Engineering Graphics intended to communicate to someone who understands technical drawings.
- Engineering Graphics used to further clarify design ideas and to communicate those ideas to non-technical individuals.

4.2. Processes for representation of information

As long as we are concerned *only* with the representation of *geometric* information (as is the case in Descriptive Geometry and also in non-schematic representations in Technical Drawings), a crude model of the representation process can be defined by simply distinguishing two different steps: modeling and projection (see figure 6).

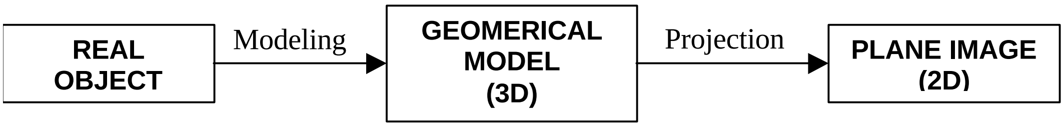


Figure 6. Steps in geometric information representation.

Modeling is the step in which the infinite complexity of a real object is reduced, in an arbitrary way, to consider only a finite set of aspects. This finite set must include all, or as much as possible, of the object characteristics that influence in the study we intend, or in the information we try to transmit. In particular, when geometry is considered, the generated model is said to be a 3D model.

The second part of geometrical representation process is one transformation to convert three dimensional models into two-dimensional geometric figures (2D figures). Among all possible transformations, projection was chosen because the resulting figure is an “image” of the 3D model. Saying “image” we mean that the resulting figure evokes the 3D model. That is, from the mere observation of a projection, topology, form and some general ideas about proportion can be obtained. Even if the observer is not “trained to read” geometric projections.

Representation of non-geometric information, in the other hand, requires the use of a more sophisticated representation process. Information must be presented in a readily understandable form, but none analogic process (like modeling and projection) is present. Consequently, a generic transformation is needed. In figure 5 one presentation of the more general process, or “pipeline”, to convert information into images is presented [13]. It can be easily concluded that the process of representation of geometric information presented in figure 4, is a sub-process of this pipeline.

It is important to realize that this model of representation process comes from computer science. In particular, computer science is concerned with the need to define a “flow” of information. , where data exchange between modules is controlled by import/export parameters, defined in the modules. In addition, modules are required to posses a single input and a single output for the information flow. The advantage to have all modules responding to the same exchange format, resides in the fact that they can be replaced by any other module in the same category, without any change in the architecture of the visualization system. The visible effect of change is to get an alternative visualization. For instance, a vector fields instead a colours map to visualize a wind speed field.

For our purpose, the diagram in figure 7 is useful to observe that information passes through three different *semantic levels* (information, geometry and image). For example, the “mapping transformation” from information to geometry is a generalization of modeling process, and projection is simply a particular case of rendering. It must be noticed that, if the reverse flow (that is, the extraction of information from images) is to be automated, the transformation between semantic levels must be biunivocally defined.

Another important aspect to be observed is the different natures of the two kinds of modules presents in the process: filters and mappers. While filters convert information within the same semantic level, mappers convert information to a different semantic level. Obviously, mappers are needed to convert information into images. Yet, it is often forgiven that filters are always needed too. We always need to filter information in order to get the image that best enhances the aspect under study. That is, non-relevant, redundant, incomplete or simply erroneous data must be removed.

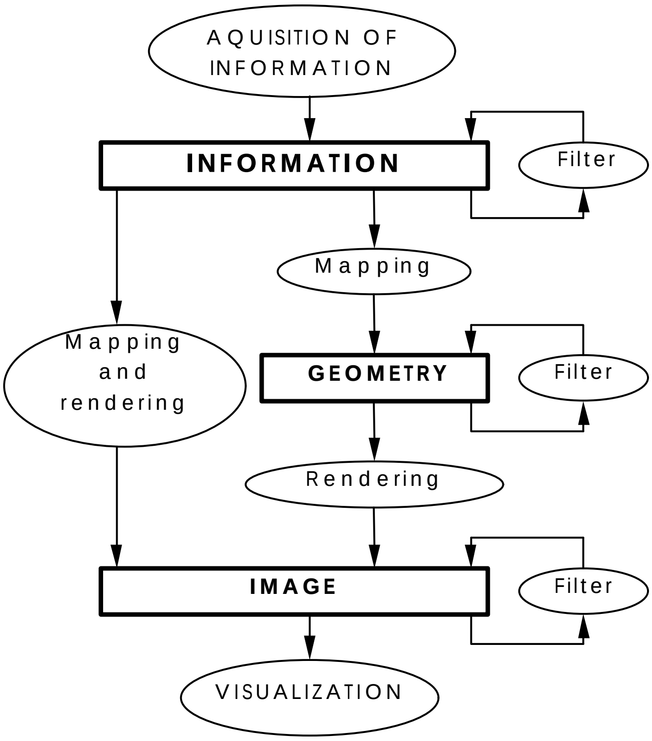


Figure 7. Steps in non-geometric information representation.

In the light of previously described representation process, we can notice that Technical Drawings are mainly analogic. The reason is that its mapping process is based on geometric projections. However, Technical Drawings also have symbolic aspects, because geometric projections are combined and/or modified with other symbolic representations. In addition, the object been projected is not the original one, but a geometric simplified model. The same can be applied to sketches and, in a low degree, to Descriptive Geometry. On the contrary, Data Graphics and Scientific Visualization are clearly symbolic transformations, because mapping process converts information into non-representative images.

Finally, the ultimate intention implicitly present in the name “pipeline” is that the unification of exchange formats is the key for the process to become a “graph”. In that graph, the information flows through different modules until it is converted into images (or vice versa). From our point of view, maintaining the integrity of information and finding fully expressive presentations are the objectives.

4.3. Standardization

Standardized representations of objets and processes in engineering have the clear purpose to support a “correct” interchange of information among all people involved in a design process. In fact, it was the economic importance of such an interchange the reason that forced the establishment of Standards that univocally fixed the meaning of all kinds of Technical Drawings. We cannot forget or minimize the *contractual* value of Technical Drawings. The result is that actually, communication using Technical Drawings is a language, a clear, precise language with definite rules that must be mastered to be successful in engineering design [3].

Nevertheless, drawing standards only deal with that part of engineering drawing practice that can be considered common to all industry. Superimposed upon this are innumerable practices used by specific industries or by individual firms to fit particular circumstances. As pointed out by Booker, “probably no single person has really complete picture of the varied and many complex drawing practices in existence throughout industry, and it is the complexity of this situation which makes it difficult to get national standards adopted throughout industry” [9].

In spite of this, we can conclude that Technical Drawings is a mature language to specify and/or describe almost all kind of information present in a design process. Nevertheless, the language is only intended for man to man communication. It is not appropriate for man-machine communication. In addition, the language is only prepared to directly convey 2D information. 3D information is only implicitly present, and Descriptive Geometry is needed to extract it. Two examples can emphasize this situation:

- In Technical Drawings, the first objective is to transmit information about geometry. Indeed, the model to be defined must contain a complete and non-ambiguous set of information describing the object geometry (topology, form and dimensions). Whatever other aspect of real object (like color, texture, etc.) must be avoided. The argued reasons are to save effort and, hypothetically, enhance the information been transmitted. Yet, in the present moment, very “realistic” representations can be obtained at a low extra-cost using the “rendering” capabilities of modern CAD systems. Nevertheless, Standards in the Technical Drawing field continue saying thinks as “the use of colors on technical drawings is not recommended”, or “ all objects made of transparent material should be drawn as non-transparent” (ISO 128-1982).
- Conventional representation of a threated part if fully standardized in orthographic views (see ISO 6410-81 in [24]). The same is not true in perspective views. Indeed you can convince yourself that we can "extend" the orthograpics' view convention to the perspective representations in a "natural" way. But now imagine yourself "telling" a 3D geometrical *modeler* that "... the crests of threads should be defined by a continuous thick line (type A of ISO 128), and the roots of threads by a continuous thin line (type B of ISO 128)...". Furthermore, you must demand the system that previously defined convention must always remain visible in the contour; never mind how many times the user changes the visualization point of view.

  The described task is the one been done (according with the Standard) when creating one orthographic or perspective representation. What makes the situation very different is that in both cases, the action done is the insertion of some 2D symbols in a 2D drawing. You cannot add 2D symbols to a 3D model and expect the *appearance* of such symbols will stay invariant when changing the visualization.

The examples show the need to enhance the models to include those characteristics (materials properties, like texture, and others) that can be visualized at no extra cost, and extend the 2D drawing standards in order to include the 2D representations of 3D models.

We can conclude that remaining lacks and inconsistencies in Technical Drawings ought to be eliminated, and explicit consideration of 3D must be added. Furthermore, assuming this objective is utopian, we must concentrate in reducing specification errors, and lacks of consistence, as much as possible, and prepare the computers to detect and solve the remaining non-consistent communications in an “intelligent” way. Just as humans do now.

In the other hand, pure non-geometric information representations (like diagrams, flow charts and so on), is still a more challenging problem. Nowadays, those representations are completely out of scope of the Standards. Nevertheless, we have argued their increasingly presence in design process.

In fact, many computers “Visualization Systems” have been developed. Some of them are devoted to specific fields (were well know graphic representations are used), but, some other systems declare to be “general-purpose”. Those general-purpose systems are faced with the fact that, many times, the optimum visual representation of non-geometric information is not known in advance. Therefore, visualization systems do provide highly interactive interfaces. However, this is none real solution: the problem is simply translated to the user. What is more, human psychology and perception rules cannot be changed to better adapt to our communication needs. They must be studied and indirectly used in our own benefit to get best visualizations.

To sum up, tentative representations can be useful when *visualization* is employed to analyze information (and, even in this case, a good user preparation in graphical semantics is needed). Yet, we cannot establish a safe and fluid *communication* with non-standardized graphics. Consequently, some kind of standardization, based on previous assumptions, is already needed.

5. STATE OF THE ART IN DRAWING RECONSTRUCTION

Description of three-dimensional objects geometry in a two-dimensional surface has been done for more than two thousand years. The reverse problem is concerned on how to interpret one or more two-dimensional representations, to recover the structure of a three-dimensional object (both its geometrical and topological structure). Of course, *implicit* recovery actions are carried out by humans to “read” drawings, since ever. Yet, *explicit* formalization of this problem began to attract some attention only in the end of 60’s, when the computers development made possible some kind of automatic approaches. This problem, named “reconstruction” (or more precisely “geometrical reconstruction”) implies the determination of geometrical and topological relations of all atomic parts of one object. It must not be confound with “restitution” and “recognition”; two well-defined fields concerned with some kind of identification of objects, and not with a detailed description of its geometry.

Most of the known approaches are now in experimental stages, and they are able to interpret (with no much errors), all kind of polyhedral objects. Interpretation of the most usual surface elements (like cylinders, spheres, etc.) is also considered by some of the approaches. Anyway, when complexity of objects increases, automatic processes usually give pass to different semiautomatic approaches.

A detailed state of the art can be traced with a reduced set of basic references. The book by Sugihara [25] is the most comprehensive reference to the early history of line drawings interpretation, dated back to the 1960s. Nagenda and Gujar [26] published a comment on eleven papers published between 1973 and 1984 on this topic, including a categorization tree. Wang and Grinstein [27] updated the categorization, and obtained one taxonomy of 3D objects reconstruction from line drawings in two-dimensional projection.

The classification was based on different but not-independent aspects. Distinctions were made based on the nature of objects been reconstructed; the model generated (the “internal” representation); the number of 2D views needed; the required premises, and the degree of interaction from the part of the user.

5.1. Nature of objects and models

First attempts of 2D line drawing interpretation were limited to *prototype objects*. Identification of shapes whose projections had been previously recorded was the objective. In other words, given an image of an object, the system identifies the object by first extracting a line drawing from the image and next searching for prototype whose projection coincides with the line drawing. The approach was more closer to recognition than to reconstruction.

A general solution was later obtained for reconstruction of *polyhedral objects*. Nevertheless, distinction between Eulerian and non-Eulerian polyhedral objects was sometimes necessary. In addition, the complexity of polyhedral objects was measured in terms of nodes “degree” (the number of edges ending in a node), and it posed a limit to some reconstruction processes. To illustrate how easily polyhedral objects can get a high degree, a polyhedral object with a 6th degree node is presented in figure 8.

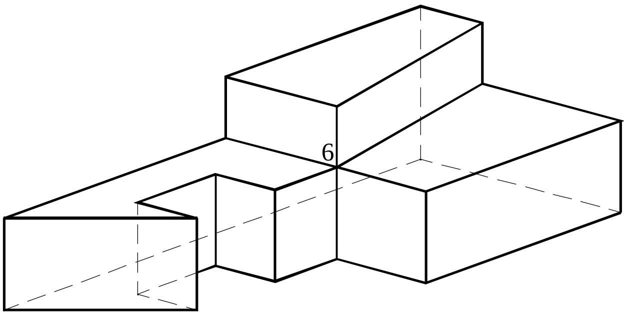


Figure 8. Polyhedral object with a 6th degree node.

Some other attempts were particularly concerned with reconstruction of *revolution objects* (like cylinders and cones) and *extruded objects*; two special cases of “sweep” geometry. Initially very important restrictions were necessary in the orientation of those objects. Finally, some improvements were done, and the orientation of curved objects was softened or even disappeared.

At the present moment, objects from a wide object domain can be reconstructed. This includes manifold and non-manifold objects containing flat and cylindrical faces. However, reconstruction processes tend to become more prone to error when the objects involve curved surfaces.

In the other hand, the model generated by the computer after reconstruction may be of different natures. Nevertheless, most commonly used 3D objects representation in reconstruction problems is BRep (boundary representation). Yet, some attempts have been done to reconstruct CSG models (Constructive Solid Geometry), from 2D representations of *extruded objects*.

5.2. Premises and degree of interaction

To simplify the reconstruction problem, it is generally assumed that in any projection of an object, only edges and contours are represented. Consequently, we can say that only "standardized" views are used to reconstruct. Sometimes it is said that only "pure" line drawings are considered. By standardized or pure we mean that texture, range, shadowing and other additional parameters are not considered. It is important to notice that these other parameters are currently used in object recognition.

Another limitation is usually added on the direction of parallel projections, and on point of view of perspective projections. In parallel projections the direction of projection cannot be parallel to any face, nor parallel to any pair of collinear edges. In perspective projections, the center of projection cannot be coplanar to any face, nor to any pair of collinear edges. This limit is named "general point of view convention", and usually eliminates potential degeneration cases (in which, for instance, one face can project in one line, or two distinct edges can project on the same line). Figure 9 shows two different projections of the same polyhedral. It is clearly observed that the one in the left is quite more difficult to "read", because it do not respect the general point of view convention (in spite of been a fully standardized "isometrical" perspective).

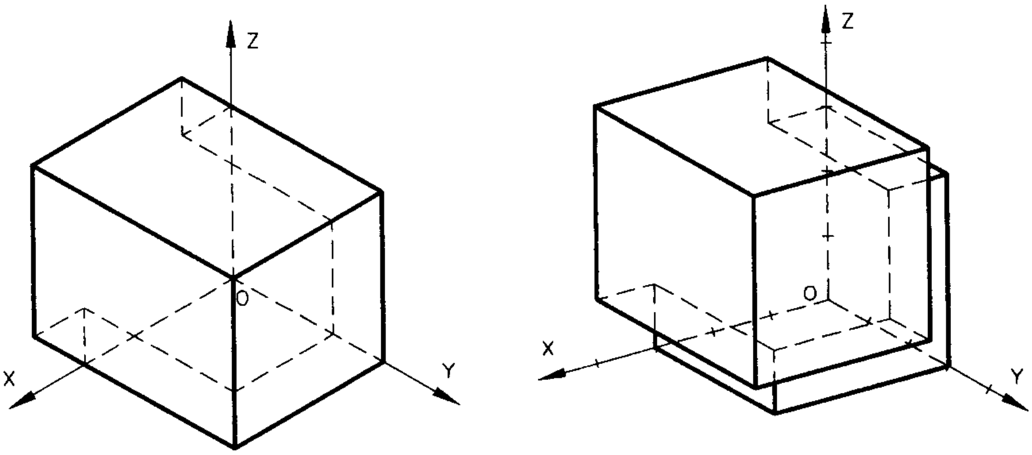


Figure 9. Non-general and general point of view projections of a polyhedral object.

Reconstruction systems can also be classified in terms of the participation they require from the user. We can distinguish between automatic and guided systems. Yet some guided systems require as much participation from the user that they can be classified as "intelligent" modeling systems, rather than reconstruction systems.

5.3. Classification of reconstruction approaches

In drawing reconstruction, and depending on the number of departure images, two main approaches can be found. The first one is based on some (usually three) orthographic views, and the second is the one based on one single axonometric or perspective view.

As said before, Wang and Grinstein [27] obtained one taxonomy of 3D objects reconstruction from line drawings. An updated version of this taxonomy is shown in table 1.

Table 1. Taxonomy of reconstruction approaches.

Multiple-view	Single view
BRep approaches	**Labelling approaches**
Idesawa 1973	Huffman 1971 and Clowes 1971
Wesley-Markowsky 1980, 1981	Lees et al 1985
Sakurai 1983 and Gu et al. 1985	Malik 1987
Preiss 1984	**Gradient space approaches**
Yan et al 1994	Mackworth 1973
Masuda et al 1996	Wei 1987
CSG approaches	**Linear programming approaches**
Aldefeld 1983, 1984	Sugihara 1984, 1986
Ho 1986	**Perceptual approaches**
Primitive identification approaches	Lamb et al. 1990
Meeran and Pratt 1993	**Primitive identification approaches**
	Wang et al. 1989, 1991, 1992
	Optimization approaches
	Lipson and Shpitalni 1996

When problem is restricted to the interpretation of multi-view drawings (such as engineering drawings composed of top, front, and side views), it can be subdivided in two main steps:

1. Establish a correspondence between different views.
2. Find a 3D realization (using previous correspondence), by assembling the individual pieces.

Approaches founded on multiple views are quite more advanced. Obviously, it is easier to reconstruct 3D objects from multiple views than from one single view. Nevertheless, those methods are usually limited to consider “main” views. The do not accept standardized conventions like particular views and sections.

Reconstruction from one single view presents more ambiguities en leads to more than one single solution. In some cases, the same view, or views, can be generated by different objects when they are projected. Therefore, one approach is called “multiple solution” when used to find all 3D objects that match with given view(s), and “single solution” when it stops after finding the first object matching the view(s).

Finally, reconstruction approaches can also be classified in terms of the consideration they have to measure. Most of the systems use only a sketch generated by user as input data, consequently they are limited to generate only a proportional model, and let for a latter phase the exact dimensions. Systems requiring a scaled drawing as input are usually the ones using standardized views as input. Usually they obtain final models as output.

5.4. Multiple-view reconstruction

As summarized before, some attempts to get a CSG model from multiple views have been done. The approaches in this category all assume that each 3D solid object can be built from certain primitives in a hierarchical manner. Extracting the primitives and combining them are the two tasks to be carried out by the system. Yet, up-to-date, existing systems require a great user interaction.

BRep approaches have been more successful. The paper by Yan et al [28] is a comprehensive and detailed description of polyhedral 3D solid model reconstruction from orthographic views. Their approach follows the major steps in many approaches to BRep reconstruction:

1. Generation of “candidate” vertices from 2D nodes.
2. Generation of “candidate” 3D edges from 3D vertices and 2D line segments.
3. Construction of faces from 3D line segments (First, “face-loops” are constructed like “candidate” faces, and in a second phase false faces are filtered).
4. Formation of 3D objects from faces.

Nowadays, it seems to be a general agreement on those major steps, and efforts concentrate in the study of pathological cases and the development of efficient techniques to implement every particular task.

Our own contribution to that field is referenced in [29]. The main innovation in the proposed approach is the use of an automatic oblique axonometry generation as an efficient alternative to guide and validate the vertices and edges reconstruction process (see figure 10). Construction of that axonometry, from three main orthographic views (front, top and side views), is a previous step, automatically carried out by the system, and based on Pohlke’s Theorem and Eckhart method described in [11].

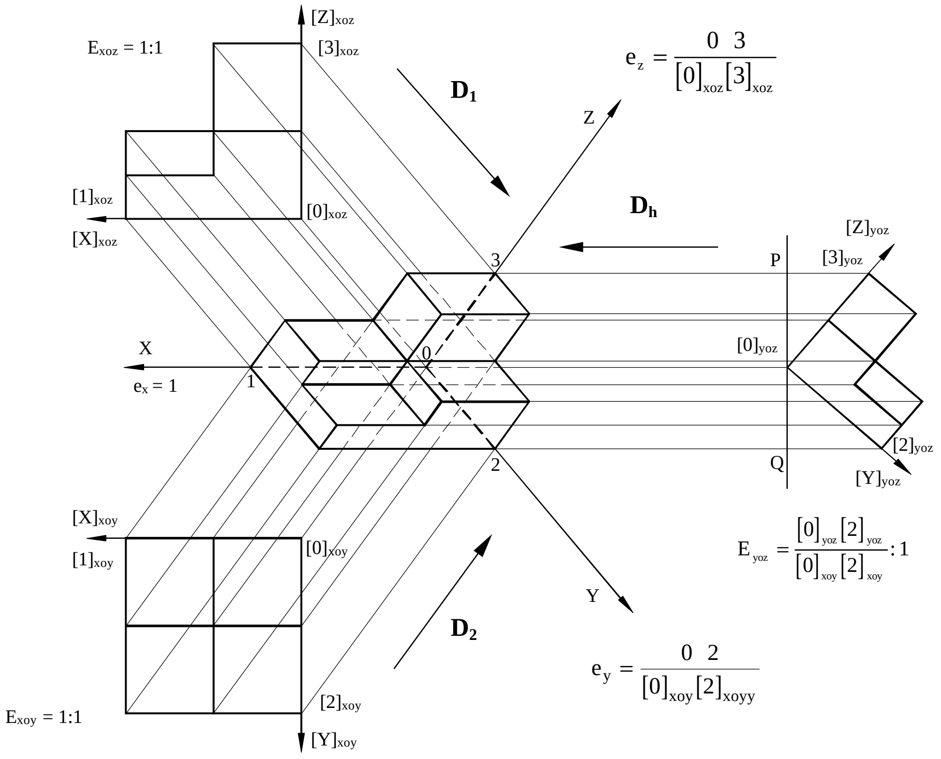


Figure 10. Automatic oblique axonometry generation.

5.5. Single-view reconstruction

If only a single-view line drawing is given and interaction between man and machine is limited, the problem of interpretation becomes more difficult. Nevertheless, some approaches for reconstruction of polyhedral objects (both, Eulerian and non-Eulerian ones) have proved successful.

Both, the labeling approach and the gradient space approach, represent interpretation rather than reconstruction approaches. They provide for the interpretation of the 2D line drawings and yield 3D information that can then be used for reconstruction.

In the linear programming approach, linear equations describing the conditions a polyhedral object must satisfy are defined. A system of equations is constructed and solved. The remaining problems are the existence of redundant equations, and the high mathematical precision required. Some drawings are considered to represent "incorrect" or even "unreconstructible" images because some coordinates deviate from "exact" positions. The approach was provided with some extensions to handle inaccuracies.

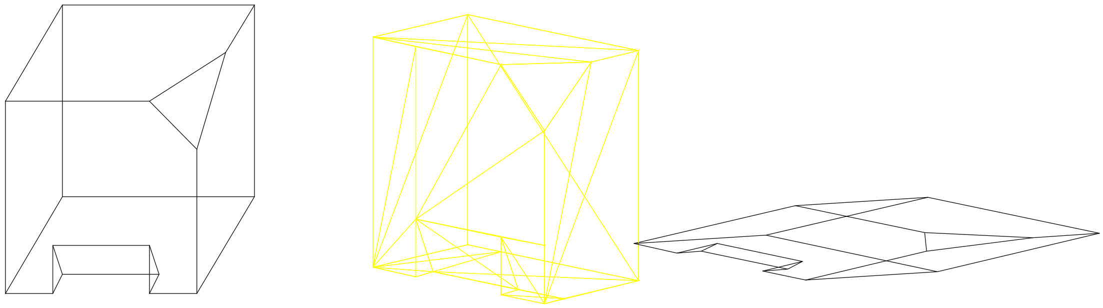
In the perceptual approach, the general idea is to generate an interpretation from rough drawings using different heuristic rules. This approach differs from the previous in that it does not use numerical methods, and, consequently, it is less susceptible to inaccuracies in the input. The approach is limited because heuristic rules are correct many times, but not always. The approach tends, for instance, to interpret as horizontal those lines intentionally made with a small slope.

Primitive Identification approaches tries to extract primitive blocks from polyhedron. Those algorithms suffer from their strong assumption of the nature of the polyhedral objects they can handle. With the introduction of curved primitives, ambiguities arise.

The approach by Lipson and Sphitalni [30] is the last contribution to reconstruction from a single (perspective) view. They propose an optimization-based algorithm for reconstructing a 3D model from a single, and inaccurate, sketch. The reconstruction process is an "inflation" of a plane image. The given 2D vertices maintain their plane coordinates (X,Y), while a set of Z coordinates is computed to obtain a 3D configuration that matches implicit spatial information contained in the drawing.

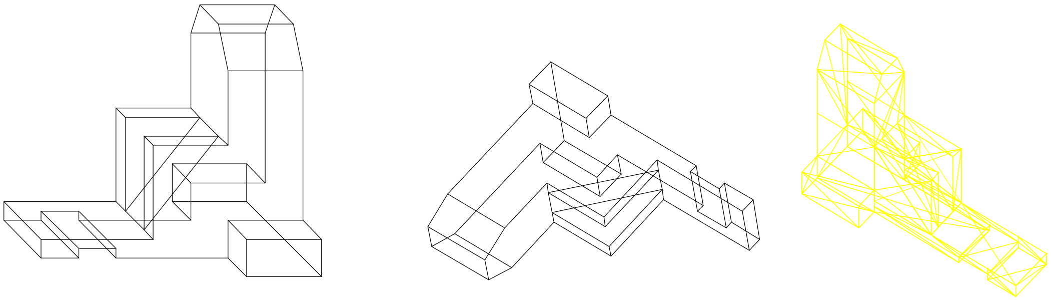
The proposed methodology is based on the mathematical formalization of this implicit spatial information that enables a human observer to have a 'feel' for the 3D object depicted by the graph. They consider the implicit 3D information coming from three sources: image regularities, face tolopogy and statistical configuration of entities. They define "image regularities" as a special geometrical relationships between individual entities (like parallelism, planarity, line orthogonality, etc.) or within groups of entities. The approach is more tolerant to faults and inaccuracies than previous approaches, and supports a wide scope of general (manifold and non-manifold) objects containing flat and cylindrical faces. It seems to be a very promising approach, because all constraints related with the increase of geometrical complexity are avoided and all implicit spatial information contained in technical drawings is exploited.

Finally, our particular approach is described in [31]. It is a semi-automatic algorithm to reconstruct Eulerian polyhedral objects. The input data is an axonometrical representation of the object, and the general point of view convention is not a requisite. Non visible edges must be present, but no distinction between visible and non-visible edges is required. During reconstruction, user is asked to confirm some "tentative" polygonal faces. He or she is also forced to identify the axonometric axes. Figures 11 to 13 show different phases in the reconstruction process of two different polyhedral objects. Input axonometries are shown in figures 11(a) and 12(a). In figures 11(b) and 12(b) wire-frame reconstructed models are shown.



(a) (b) (c)

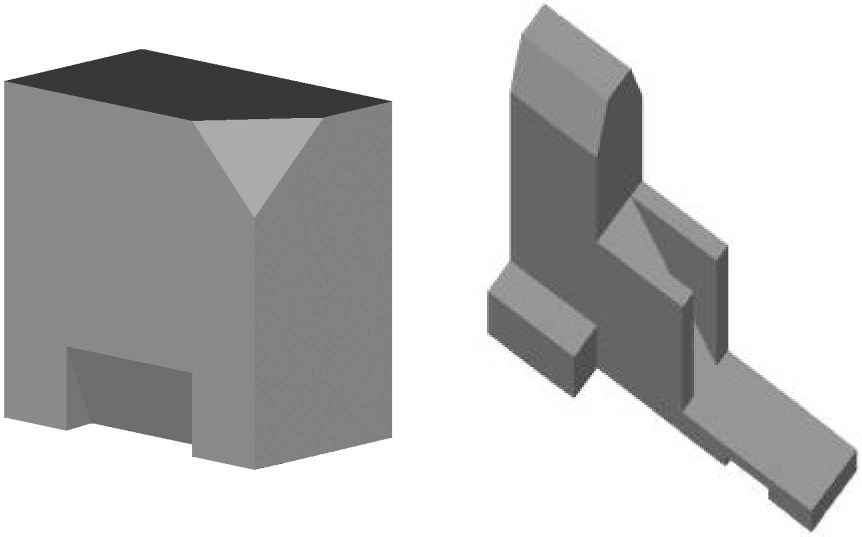
Figure 11. Simple, Eulerian, non-convex polyhedral reconstruction.



(a) (b) (c)

Figure 12. Simple, Eulerian, non-convex polyhedral reconstruction.

It can be observed how wire-frame models “raise” from projection plane (XY plane), where input axonometries (figures 11(c) and 12(c)) are contained. Finally, to emphasise that final result is a full 3D geometric model (a BRep model), two images rendered with shadows and lights are reproduced in figure 13.



(a) (b)

Figure 13. Render of two reconstructed eulerian polyhedron.

6. CONCLUSIONS

In this paper, we have discussed the role played up to date by Engineering Drawings used in the design process. We concluded that during centuries, humans have communicated design information using graphics. Because, thinking in the language of Engineering Drawings, engineers and designers can visualize problems more clearly and can find solutions to design problems with greater ease.

Latter on, we presented the state-of-the-art in both, Engineering Drawings and Computer Aided Design. We stated that the first "revolution" of graphical capabilities of computers in the design process was to assist drafting, and almost automate it. The second has been introducing interactive creation and manipulation of 3D virtual models to reduce (and almost eliminate) the need for Descriptive Geometry. The next "revolution" will be to convert Engineering Drawings in a "transparent" language for the whole design process, to reduce (and virtually eliminate) the need of data transfer among different phases in the process.

In other words, CAD systems have non-sequential (graphic) outputs, but accept only sequential (verbal) inputs. This is a direct consequence of current state of evolution in Computer Graphics, constrained by the sequential nature of algorithmic languages used for programming tasks. In the contrary, design process, and in particular ideation process, need non-sequential thought. Consequently, one graphical language must be defined (or adopted) to improve the present communication between designers and CAD systems. Yet it is important to notice that we do not claim for the physical implementation to become non-sequential; we only say that the conceptual model (and the interaction front end) must be graphical.

Finally, we putted the emphasis in the main aspects we need to solve to convert Engineering Drawings in a comprehensive and powerful communication language between designers and CAD systems:

a) Design systems must be able to automatically convert design information among the different formats used in the different phases of design.

b) Standardized language must be "refined", to reduce lacks and inconsistencies.

To move in this direction, geometrical reconstruction is one essential problem to solve. This means that automatic solid-model generation from standardized drawings is the most efficient way to establish a fluid communication between designers and CAD systems. This is the challenge of 3D *reconstruction* of design models from engineering drawings.

A general-purpose and automatic (or, at least, easy-to-use) system to reconstruct objects is our present objective. Yet this is only a necessary step in a more ambitious objective: convert standardized technical drawings in an *input* language for design systems.

To achieve this goal some "pure computer world" and some "pure Engineering Graphics" aspects must be achieved and integrated. In the former, text and other signs and symbols present in technical drawings must be identified and interpreted. In the latter, a better definition of 2D drawing standards and a comprehensive 3D-model definition must be done.

REFERENCES

[1] Ramaswamy R. and Ulrich K. "A Designer's Spreadsheet". *Journal of Mechanical Design. Transactions of the ASME.* Vol 119, march 1997, pp. 48-56.

[2] Ferguson E.S. *Engineerign and the Mind's Eye,* MIT Press, 1992.

[3] Bertoline G.; Wiebe E.; Miller C. and Nasman L. *Fundamentals of Graphics Communication.* Irwin, Graphics Series, 1996.

[4] Bertin J. *La Graphique et le Traitement Graphique de L'Information*. .Ed. Flammarion, Paris., 1977.

[5] Foley J.D., Van Dam A., Feiner S.K. and Hughes J.F. *Computer Graphics. Principles and Practice.* Ed. Addison Wesley, 1990.

[6] Suh N.P. *The Principles of Design.* Oxford University Press, New York, 1990.

[7] Encarnaçao J.L.; Lindner R. and Schlechtehdahl E.G. *Computer Aided Design*, Ed. Springer Verlag, Berlin 1990 (second edition).

[8] Encarnaçao J.L. and Lockemann P.C. (Editors). *Engineering Databases.* Ed. Springer Verlag, Berlin 1990.

[9] Booker P.J. *A History of Engineering Drawing.* Borthgate Publishing Co. Ltd., London, 1979 (First Printed in 1963).

[10] Jones C. *Métodos de diseño.* Gustavo Gili, Barcelona, 1982. (Jones C. *Design Methods*. John Wiley & Sons, 1980)

[11] Hohenberg F. *Geometría constructiva aplicada a la técnica.* Ed. Labor, Barcelona, 1965.

[12] Cugini U., The problem of user interface in geometric modelling, *Computers in Industry 17*, 335-339, 1991.

[13] Göbel M. and Frühauf M. *Interactive Scientific Visualization: Algorithms and Systems.* Eurographics Technical Report Series, EG92 TN10, 1992.

[14] Company P., and Gomis J.M. Gráficos en investigación y diseño**.** *Anales de Ingeniería Gráfica*. Vol. 3, num. 2. 1994. pp. 27-33.

[15] Tufte E.R. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, 1983 (Fourteenth printing, March 1995).

[16] Post F.H. and Hin A.J.S. (editors). *Advances in Scientific Visualization*, Springer–Verlag, 1992.

[17] Grave M., LeLous Y. and Hewitt W.T. (editors). *Visualization in Scientific Computing*, Springer–Verlag, 1994.

[18] Martí P. and Company P., "DISSENY: An Integrated System for the Structures and Structural Elements Optimal Design". *Advances in Optimization for Structural Engineering.* B.H.V. Topping (Ed.). Civil-Comp Press. Edinburgh, 1996, pp. 23-29. (ISBN 0-948749-42-3).

[19] Martí, P. y Company, P. (1993). *ADEF. Análisis interactivo de estructuras por elementos finitos.* Ed. Universidad de Murcia, 1993.

[20] Martí, P., Company, P., y Sanchis, M. (1986). Acoplamiento elementos finitos-técnicas de optimización en el sistema DISSENY. *II Simposium sobre Aplicaciones del Método de los Elementos Finitos en Ingeniería,* Barcelona,. 1986. p. A-133 a A-147.

[21] Martí P., Ortueta C. y Company P. (1990). Optimización conjunta de torres y cimentaciones de líneas de conducción eléctrica. *Memorias del I congreso de Métodos Numéricos en Ingeniería.* Ed. G.Winter y M.Galante, Barcelona, 1990, p 725-732.

[22] García-Estañ, J. y Martí P. (1994). Optimización de estructuras metálicas con perfiles de pared delgada. *Anales de Ingeniería Mecánica, Año 10, nºI.* 1994. p. 393-400.

[23] López-Alanís, C., Sánchez, G. y Martí P. (1994). Diseño óptimo de torres de transmisión de energía eléctrica teniendo en cuenta el comportamiento no lineal

(grandes desplazamientos). *Anales de Ingeniería Mecánica, Año 10, nºI*. 1994. p. 409-416.

[24] ISO. *ISO Standards Handbook 12, Technical Drawings*, 1991.

[25] Sugihara K. *Machine interpretation of Line Drawings.* The MIT Press1986.

[26] Nagendra I.V. and Gujar U.G.. "3-D Objects From 2-D Orthographic Views – A Survey". *Computers & Graphics.* Vol 12, No. 1, 1988. pp. 111-114.

[27] Wang W. and Grinstein G. "A Survey of 3D Solid Reconstruction from 2D Projection Line Drawings". *Computer Graphics Forum.* Vol. 12, No 2, 1993, pp. 137-158.

[28] Yan Q.W., Philip Chen C.L. and Tang Z. "Efficient algoritm for the reconstruction of 3D objects from orthographic projections". *Computer Aided Design.* Vol. 26, No 9, 1994, pp. 699-717.

[29] Gomis J.M.; Company P., and Contero M. "Reconstrucción de modelos poliédricos a partir de sus vistas normalizadas". *Anales de Ingeniería Mecánica, Año 11, vol. 1,* 1.997, pp. 383-391.

[30] Lipson H. and Shpitalni M. "Optimization-Based Reconstruction of a 3D Object from a Single Freehand Line Drawing". *Computer Aided Design.* Vol. 28, No. 8, 1996, pp. 651-663.

[31] Gomis J.M., Company P. and García J., "Preprocesador para modelado geométrico tridimensional a partir de la delineación 2D de axonometrías". *Actas del IX Congreso internacional de expresión gráfica en la ingeniería. Volumen 2,* 1997, pp. 345-354.