Design in a Hypertext Environment

John D. Gilleard, Ph.D and Y.C. Lee
Department of Building Services Engineering
Hong Kong Polytechnic.

In the paper it is suggested that hypertext is suitable for the development of design tools associated with the preliminary design phase of Building Services Enginnering. This argument is supported by the development of a prototype model for "sprinkler installation design". The Macintosh HyperCard has been chosen to demonstrate the effectiveness of the procedures.

INTRODUCTION

The use of computers in Building Services Engineering (BSE) design is mainly applied to problems that are well structured and precisely described by mathematical formulas and models. However, there are cases in BSE design work where the problem is not well structured and cannot be represented by simple mathematical modelling techniques. For example, conceptual or preliminary designs are often based on constraints determined by various design rules and codes. These rules and codes may predetermine the search pattern and eventual design solution. However, the design process may not be sequential in nature.

Sprinkler installation design in Hong Kong is mainly based on a set of safety codes, i.e. "The LPC Rules for Automatic Sprinkler Installations" [1]. These rules are a combination of British Standard BS 5306 : Part 2 : 1990 and the "Technical Bulletins" issued by the Loss Prevention Council (LPC) of the UK. The requirements stipulated in the LPC Rules range from the very general, e.g. the classification of occupancies and fire hazards, to the detailed description of particular component design of the sprinkler installation.

As a design engineer, his main concern is to achieve a sprinkler layout plan that satisfies both the LPC Rules and accommodates other design constraints imposed by the various disciplines and trades e.g. the aesthetic requirements set by the architect; or co-ordination with the structural system or other m&e services. In addition, with the rapid pace of building research, with its resultant increase in knowledge, coupled to the frequent updating of the rules and design practice codes, there is a growing need for design methodologies that can change both quickly and efficiently. Increasingly this is possible through the application of computer technology, particularly where CAD or engineering calculations are undertaken. However, these design activities form only a part of the entire design process, i.e. the routine presentation of an already well-thought design concept. The early stage of the design process, which is mainly a conceptual one, is still rarely aided by computers [2].

Construction Informatics Digital Library http://itc.scix.net/
paper w78-1993-37.content

In the case of sprinkler installation design experience tells us that although there are explicit requirements stated in the LPC Rules, the design cannot be represented by a specific formula that implicitly determines the most optimum design. For example, the number of factors to be considered is frequently too numerous. In addition, many of these factors cannot be dealt with in a sequential order. For instance, if the sprinkler array is drawn but the pipe size selected has exceeded the structural constraint in the opening size through the beams, then the sprinkler head spacing would need to be redesigned etc.

Currently there are several building services analysis software packages that attempt to provide facilities that carry out straight forward design calculation for sizing sprinkler installations such as the QUICKEST and SPRINKLE [3]. However, these generally lack an intelligent interface that attempts to advise the designer with respect to the overall design. Therefore, it may be suggested that an ideal methodology for the designer to invoke a computer-aided design tool may be through the adoption of non-conventional design tools. These could be provided to assist the design engineer to access design knowledge in a very fluid fashion instead of through the adoption of rigid rules stipulated in a traditional linear order.

HyperCard as Tool for Design and Decision Making

Hypertext can be defined as a database management system that provides connections between files of information, both textual and graphic [4]. Hypertext may also be described as a tool for building and using associating structures. Whereas traditional design documents are read in a linear fashion, i.e. from beginning to end, a hypertext document allows the reader to jump from idea to idea, depending on one's interest.

The arrival of both the Macintosh microcomputer and its accompanying hypertext programme, HyperCard, opened many new opportunities for design applications. For example, HyperCard is an authoring tool and an information organizer. It is an software engine and not an application program. It does not function like a word processing, spreadsheet, or database program [5]. HyperCard provides linking between different screens of information with "buttons" that the user can "click" with the mouse to go from screen (or card) to screen. These "linkages" may be readily customized by the user. To some extent the HyperCard programming environment can be considered to be an object oriented programming language which has definite tangible benefits for the programmer. For example,with a conventionally structured program the programmer typically considers "*What are my inputs and what are my outputs?*"; or "*How much of my data needs to be transformed ?*" However, with object oriented programming the programmer typically asks "*What are the things that I will be working with, and what are these things to do ?*" [6]. Some advocates of the object oriented approach claim that it is 25 times more productive when compared to conventional programming systems [7]. Other researchers are more modest in their claims, e.g. four to six times [8]. All indicate an improvement.

Applications of Hypercard/hypertext in the field of construction design and building engineering , including m&e, are limited. For example, the OASIS model was developed to provide ground-water modellers with a bio-degradation model, and database management tool [9]; Schuman [2] describes a daylight design tool; and Becker [10] describes a HyperCard program for a graphical interface to show fault propagation for Space Station Freedom; Williams [11] described a prototype application to provide a reference manual to help in the inspection and construction of asphalt concrete pavements; and finally, a Classification System to Extract Project-Specific Codes of the National Building Code of Canada with an attempts to model buildings with regard to their relation to building codes, standards and regulations [12].

The application of hypertext to BSE is still in its early stage. It is recognized that the BSE is just one member of the entire construction project team and that his design decisions are subject to the constraints of various disciplines. It may also be said that his problems are by no means well structured. Though most BSE design work is based on well-established rules, the optimum solution may require creative ideas and reasoning, which also has to be in full compliance with all of the relevant codes of practice and other building ordinance. In this connection, the choice of HyperCard would appear to be a logical option given the flexibility of the program.

As a demonstration to these arguments the design of sprinkler installation was chosen to test the hypothesis. The Macintosh HyperCard was used in preference to other hypertext software .

The LPC Rules design tool in a HyperCard environment

The intent of the prototype design tool for LPC Rules is to collect and organize the relevant clauses and constraints in such a manner that the design engineer is able to develop a preliminary sprinkler layout plan based on the architect's original floor plan. The design tool should then be able to provide the design engineer with a quick means to re-assess the design to determine whether the proposed design meets the LPC Rule design requirements. In addition, expert advice is to be incorporated where applicable in addition to the LPC Rules.

With the main objective defined, it is then necessary to determine the extent of the knowledge base and how this knowledge may be represented in a HyperCard mode. Although the sprinkler installation design is not a linear process and not well structured, it has to be organized and broken down into various subprocesses which HyperCard can handle. Therefore, the first step is to eliminate all the clauses which do not have immediate implication to the design, e.g testing and commissioning, maintenance, colour coding, specification of individual components, etc. The remaining parts may then be grouped into six individual design processes :

1. Classification of occupancies and fire hazards
2. Selection of installation type, size and design
3. Sprinkler spacing arrangement and location
4. Pipe sizing and sprinkler array design
5. Type of water supply
6. Storage requirement of water supply

These are shown in the flow chart, figure 1.

Classification of occupancies and fire hazard is the basis of the entire sprinkler installation design. It is the entrance door which the designer must go through before he can have meaningful manipulation of other design processes. The user can then follow the other processes in a sequential order. However, he is not strictly bound by the order as suggested by the system. For example, the designer may choose to browse through the other parts of the design process, seeking or confirming design parameters.

Each set of design rules are established on one or more "cards". Each card forms the basic elements of a HyperCard "stack", i.e. a collection of cards form a stack. Each stack may be thought to be a HyperCard program. Hypothetically,it is technically feasible to build the model of the LPC Rules within one stack. However, for ease of use it was decided to segregate each of the six major design elements into 'independent' stacks. Each of the six processes may be thought of as self-contained modules. This breakdown of the LPC Rules facilitated less programming effort. The user is able to start his navigation through any path that he prefers by clicking any one of the six boxes representing the various design processes. The card shown in figure 1 serves as the main menu for the entire program and becomes a critical reference point for the user when orientating himself during his navigation through the stacks.

Although this main menu card is called a flow chart, the user is not given the option to jump from one stack to another stack directly, i.e. it is necessary to return to the "design process card" , or main menu card,firstly before transfer to another stack. The main reason for such an arrangement is to encourage the user to explore different approaches of the design task that suit his job best. The linkage arrangement between the main menu card and other design process stacks is shown diagrammatically , figure 2.

The flow chart represented by figure 2 prompts the user to select the route best suited to his design requirements. However, where information is required by the user before a choice is taken advise may be requested to aid his decision making. To be more precise, the LPC Rules modeled in the HyperCard environment are organized to perform three types of functions, i.e. (1) to choose options; (2) to manipulate data; and finally, (3) to display geometrical and spatial relationships. Occasionally, a card may contain an imperative type of message. These explain to the user the meaning of a particular term. The imperative type of card is not favourable in terms of guidance for navigation because a stand-alone message does not convey explicitly its relationship with other cards. In addition, explanation of special terms is contained in a glossary accessed separately by "hot links".

While providing options for the user to choose is the major basic function of the model, there are different formats that represent the various choices available to the designer depending on two main factors, i.e. the complexity of specific rules; and the level of sprinkler design experience of user. In addition to the main menu card, the classification of occupancies and hazard group is one of the most straight forward stacks of cards in terms of making choices. The user is prompted automatically to the "light or ordinary hazard"- terms defined by the LPC Rules- once he has confirmed the occupancy type, i.e. industrial or non-industrial, and the maximum area of compartments and fire resistance of boundary wall. As "ordinary hazard" may be further subdivided into four hazard groups depending on the type of occupancies, the user is also given two options for his next design step. These subsequently determine the specific hazard group classification. The model allows for each "classification of occupancies" to vary with respect to different local authority requirements.

The above illustration has demonstrated the arrival of a conclusion based on selection criteria totally apparent to the user. Alternatively, in order to make the card look simpler, and to provide a quicker path to the answer, some criteria has been written using HyperTalk. This enables information that is not readily transparent to be interrogated by the user. For instance, the selection of installation type for wet or dry sprinkler systems is based on the ambient temperature of the installation. The designer may not know this information. Hence a "button" is provided to allow the user to determine this value if required. However, if this information is known upon input the program skips to the next design choice.

For a more complicated case, e.g. such as the pipe sizing and sprinkler array design, the technique of making the design criteria transparent to the user is applied more extensively. For example, in sizing the distribution pipe (D-pipe) for ordinary hazard, the user is required only to enter the pipe length and the program will determine automatically the pipe loss and subsequently advise him of the minimum D-pipe size allowed. In addition, the user may also be totally ignorant of the pressure loss. However, during the background calculations this data parameter will automatically form part of the calculations. Representation of the rules in such an intelligent manner therefore saves the designer from having to undertake repetitive work.

Graphic interface

The ability to handle graphics is one of the merits of HyperCard. However, the painting facilities of HyperCard is limited when compared with other draughting packages. Therefore, draughting inside HyperCard has been restricted to improving the aesthetics of the cards and to create simple sprinkler layout plans. However, in certain circumstances where the figures for illustration are too large and complicated to be placed inside one card, an external command may be used to capture drawings prepared by other programs. This facility is especially useful for explaining the requirements that stipulate spatial and geometrical requirements such as the relationship between the number of sprinkler heads and range pipe sizing, figure 3.

The advantage of graphics is also used to facilitate an user in decision making. For instance, there are several geometrical constraints in Sprinkler Spacing Arrangement. A graphical display of these choices will provide the user a clearer picture of respective design considerations as shown in figure 4.

Photographs may also be scanned and captured into the LPC model using the external command facility. One of the merits of external command is that the user is able to return to the original card after viewing pictures. Such a facility acts as a side reference slide library.

The second factor that affects the format of a card representing various options is how conversant is the user in dealing with the LPC Rules? For example, he may be a novice attempting to learn how to use the rules; a professional from another discipline, e.g. an architect, who simply wants to collect basic reference information; or an experienced sprinkler layout design engineer who would like to do quick search with respect to a certain design parameter. As a consequence, the capability of HyperCard for linking information in a non-linear manner has been fully exploited in building the LPC Rules model. The user is simply given the choice to determine the level of expertise and knowledge required for a particular design. That is to say, the user can bypass some of the cards if they wish to. Those cards that have the facilities to be bypassed can be treated as a junction or node card. While these junction cards can provide shortcuts for some people, too many of these junction cards may result in clumsy algorithms. Therefore, incorporation of a junction card has to be strategic and the advice from experienced designers is an advantage when making that choice.

In addition, it is good practice when writing hypertext to provide return buttons in each card This helps the user to find their way out when they are disoriented. In our case the return paths provided in the LPC Rules model are at three levels : 1 to return to the opening card of a design process; 2 to go back to the last card that he has just browsed; and 3 to return to the main menu card

Extending the power of the design tool

So far the LPC Rules model has been developed as a tool for representing the design intent for sprinkler installations which are to be used without any direct connection to a specific building plan. A designer will therefore treat the model as a handy dictionary for general application in the field of sprinkler layout design. In order to make the model more usable in the computer-aided environment, future development work is envisaged that takes the form of (a) a link model to computer-aided draughting (CADr) environment for generating the sprinkler layout interactively; (b) to establish an algorithm to represent the physical layout of a building plan matching the LPC model; and finally (c) to build additional HyperCard models for other installations in order to form an integrated design guide package where HyperCard acts as the design link with each other system.

For example, linking with CADr environment is based on the argument that "*current CAD systems are inherently limited in their ability to meet the needs of the more powerful and wide ranging design support systems required to more closely integrated design activities, and that it is these limitations that have motivated research into AI-based design support systems* " [13].

Generation of a sprinkler layout interactively is an attractive feature, yet it is by no means a full representation of the knowledge base in terms of compliance with the design constraints in relation to a particular building plan. Inasmuch as the typical objects in a building plan (eg. fire zones, rooms, doors, walls, etc.) are fairly abstract, a rule-based approach in and of itself would not provide an adequate representation. Thus a frame-based representation is suggested to represent the building plan and its attendant object [14]. The viability of frame-based representation in a hypertext environment has to be further investigated.

Beside satisfying the LPC Rules, a sprinkler installation has also to be well co-ordinated with other m&e systems. Thus a design model for sprinkler installation is in itself not a complete design tool. An integrated design model across various types of services should be a powerful tool for a building services systems designer.

Conclusion

The LPC Rules model is strictly in the development stage and not yet implemented in real production environment. However, features of this system have been demonstrated to several groups of practising building services engineers who did react positively to the potential development of such an approach to solve BSE design problems.

The current development work clearly indicates that it is feasible to represent the LPC Rules as a hypertext document. The design engineer is able to obtain data efficiently and to determine the design constraints that need to be considered at the preliminary design stage of the work. It is also possible to construct a non-linear pattern for the flow of information within the model. This non-linearity is possibly the most suitable method for storing and accessing information for typically semi-structured problems. However, it is also recognized that the development time for this type of programming is large. In addition, questions arise with regard to the nature of the model. For example, the initial design model was construed to be a detailed design tool. It was further envisaged that inexperienced "sprinkler layout design engineers" would be able to operate the model without outside assistance. However, adopting this approach would appear at this juncture in the development of the work to be wrong. HyperCard/Hypertext, although admirably suited to storing and disseminating information, is possibly too restricted to be a fully "intelligent" interface. It would appear from our initial experience that where the user is truly "naive" too much information needs to be explained before a solution can be determined. However, where the user is partially experienced with this type of work solutions may be generated more quickly.

It is therefore anticipated that the model may usefully be used either for "approximate" sizing and layout design for the novice. In this respect an architect may use the system to develop a conceptual design before handing the work over to a more experience m&e engineer. Or a detailed design may be determined by a practicing sprinkler designer who uses the model either for code compliance and verification, or for questioning design assumptions. In addition, it is envisaged that building ordinance officers in Hong Kong whose job it is to verify the adequacy of sprinkler layout designs may also use the model.

REFERENCES

[1] The Loss Prevention Council, Rules for automatic sprinkler installation, (1990).

[2] J.R. Schuman, S. Sullivan, M. Selkowitz, M. Wilde, A daylight design tool using HyperCard on the Macintosh', *Third National Conference on Microcomputer Application in Energy* , Tucscon, Arizona, (1988).

[3] S. Lloyd, *BSRIA Software for Building Services*, BSRIA, 94 - 96, (1992).

[4] K. Parsaye, M. Chignall, S. Khoshafian and H. Wong, *Intelligent data-bases*, John Wiley, (1989).

[5] D. Goodman, *The complete HyperCard Handbook*, Bantam Book, (1987).

[6] J.S. Linowes, It's an attitude, *Byte* ,13(8) 219-224, (1988).

[7] J.W. Verity, The OOPS revolution, *Datamation*, 73, (1987).

[8] M.M. Waldrop, Artificial intelligence moves into the mainstream., *Science*, 237, 484-485, (1987).

[9] C.J. Newell, J.F. Haasbeek and P.B. Bedient, OASIS: A graphical decision support system for ground-water contaminant modeling, *Ground-Water*, 28(2) 224-234, (1990).

[10] F.J Becker, Space Station Freedom Integrated Fault Model, *Graphics Technology in Space Application,* 155-163 (1989).

[11] T.P. Williams, Hypertext Database Applications in Construction, *Journal of Construction Engineering and Management,* 117(3) , 460-467, (1991).

[12] D.J. Vanier, A Classification System to Extract Project-Specific Building Codes, *CIB'92 Poster Book*, Montreal Canada, 524-525, (1992).

[13] T. Smithers, AI-based design versus geometry-based design or Why design cannot be supported by geometry alone., *Computer-aided Design* 21(3), 141-149, (1989).

[14] C.L. Dym, R.P. Henchey, E.A. Delis and S. Gonick, A knowledge-based system for automated architectural code checking, *Computer-aided Design*, 20(3), (1988).

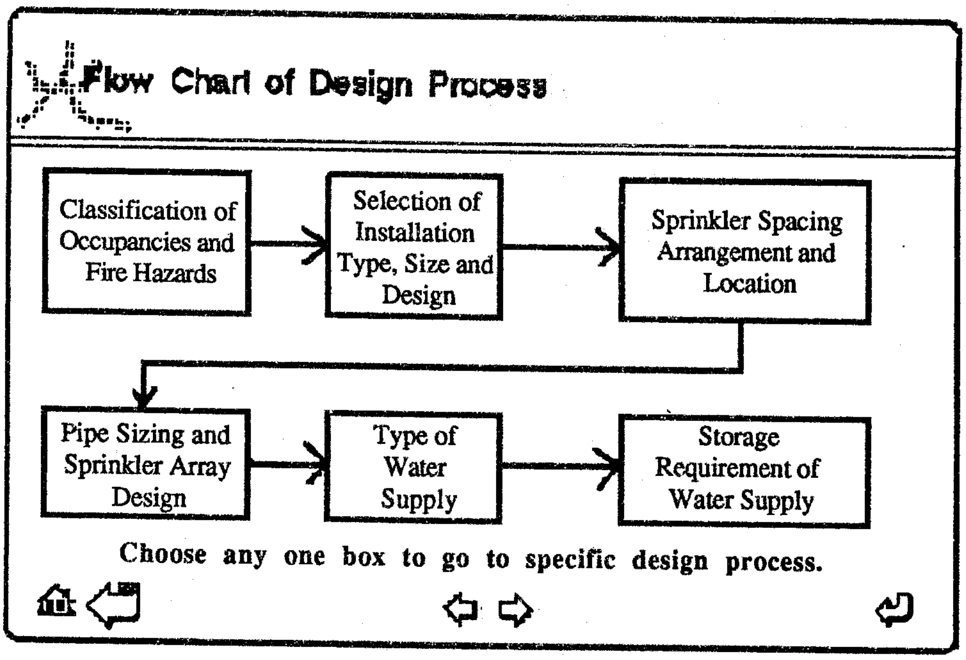

Figure 1. Representation of the six main design functions contained in the Sprinkler Layout Design Model.

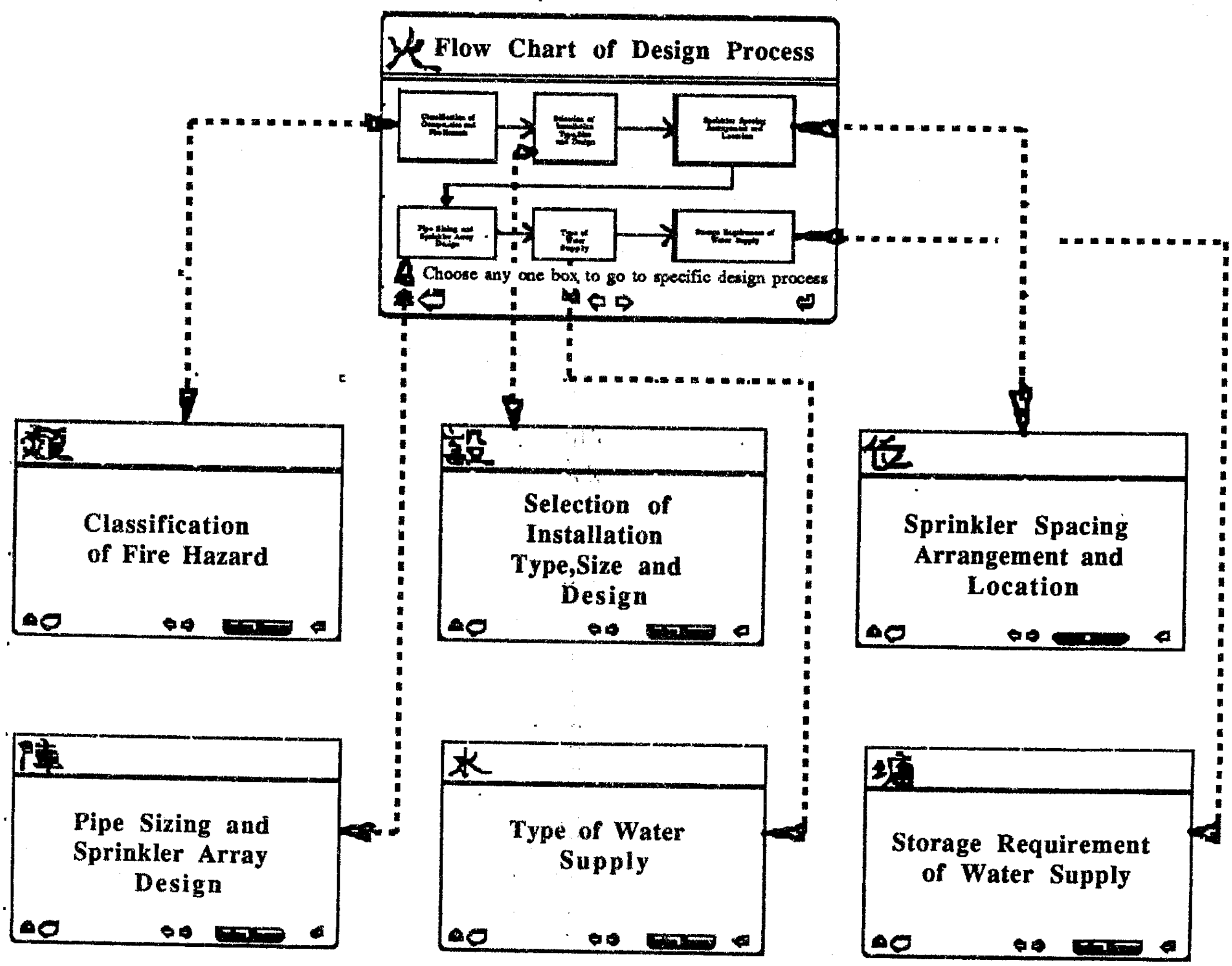

Figure 2. Possible routes through the six main design elements.

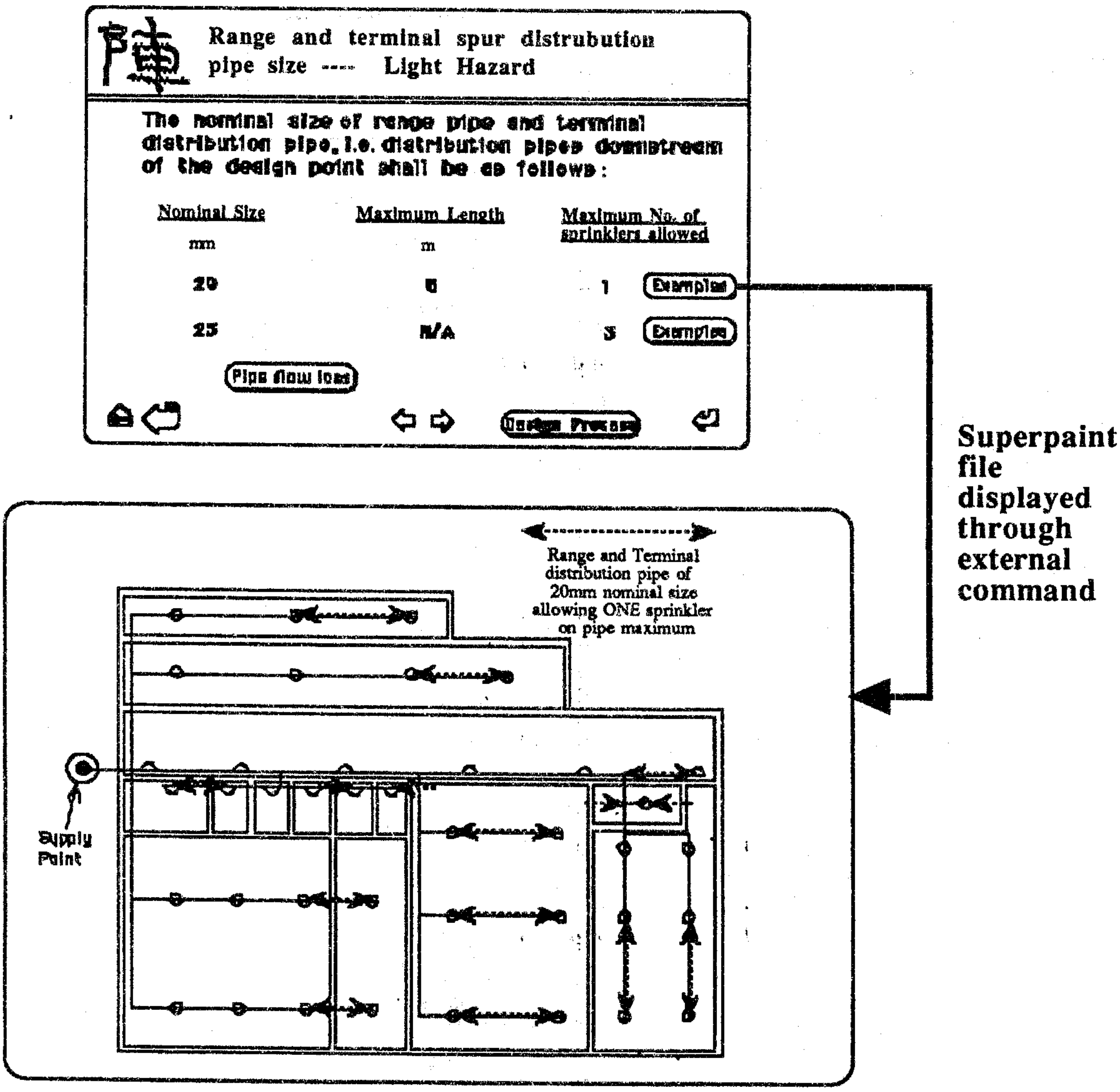

Figure 3. Use of the "external command" function.

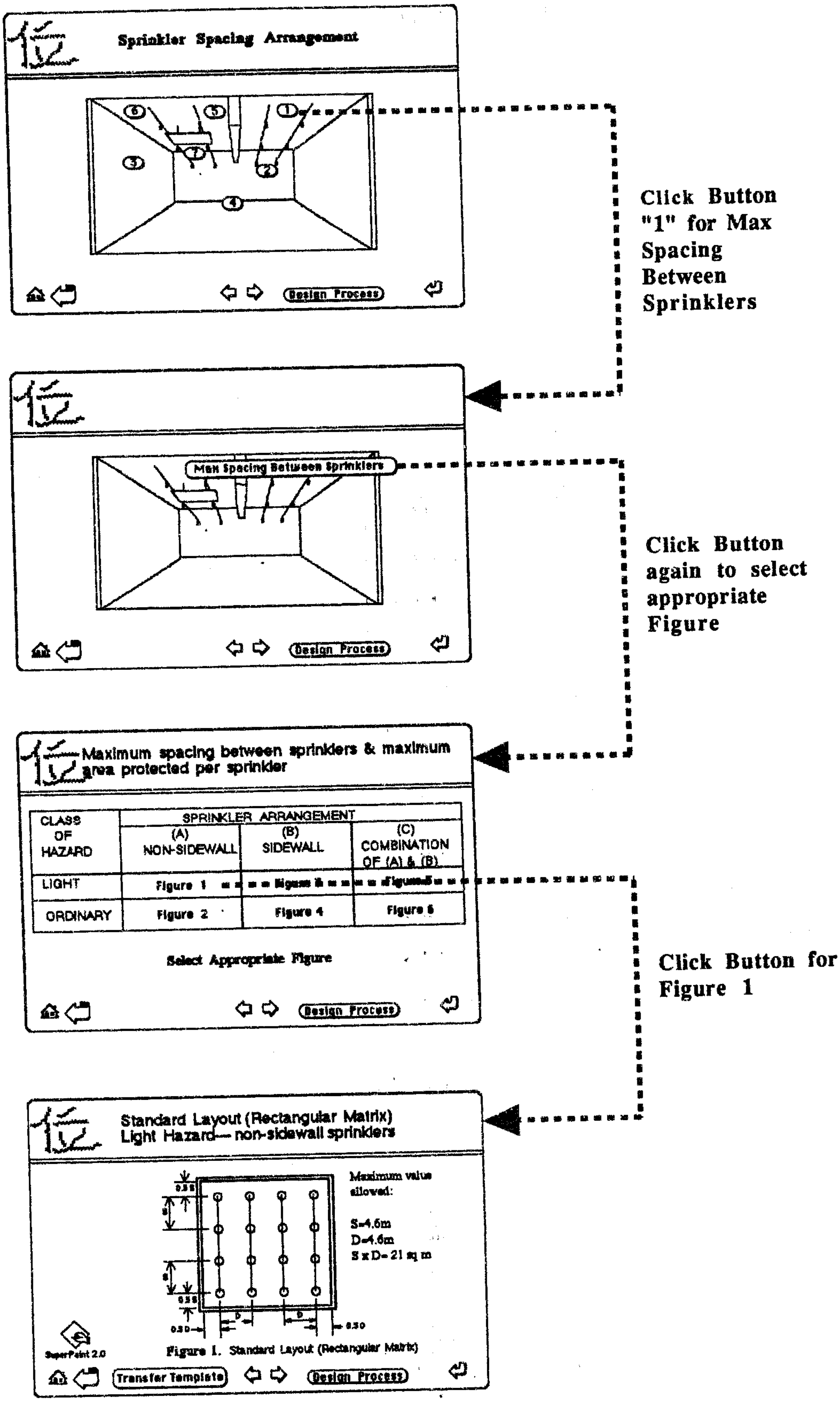

Figure 4 Series of graphical display to facilitate easy access to design constraint data