DIGITALLY CAPTURING DESIGN SOLUTIONS

Kene Meniru[1], Hugues Rivard[2], and Claude Bédard[3]

ABSTRACT

Building designers rely on computers for many reasons. Some of these are accuracy in drawing measurements, accessibility to the design projects (past and present), consistency in standards and drawing symbols and ease in making changes and customization. These benefits cannot exist if design data is not in electronic format. Unfortunately the important early design stage, conceptual design, is presently accomplished using drawings that are not electronic.

This paper discusses issues concerning the production of electronic early-design drawings. It presents information about the decisions made by designers in creating design solutions and a way to digitally capture both the process and the solutions. It is based on the findings in a Ph.D. research project that culminated in the design and implementation of a prototype system (COBL DT) for assisting early building design process.

KEY WORDS

Early Building Design, Conceptual Design, Architectural Computing, Digital Design Studies, Computer-Aided Conceptual Design

INTRODUCTION

A building product is created through a process that is complex and requires the participation of numerous professionals who must work together as a team. It begins with the architect who formulates relationships between building spaces and the form of the overall building. The engineer then designs a structure that will withstand all forces acting on these spaces. Then as work continues, input from other professionals is considered until the building design is complete (Cross, 1998; Purcell and Gero, 1998; Atman et al., 1999)

The trouble with this early design process is that data is not yet interchangeable between these professionals. The architect's sketches do not exist in a way that is digitally usable by other designers or members of the building team. When the architect takes the time to formalize these sketches and re-draft them digitally, then the engineers can be invited to perform their part of the job. This process is error prone and can lock the engineer out of critical early decisions.

[1] Assistant Professor, Head of Program, Computer-Aided Design & Drafting (CADD) Department, Piedmont Virginia Community College (PVCC), VA 22902-7589, Phone +1 434/961-5256, FAX +1 434/971-8232, kmeniru@pvcc.edu

[2] Professor, Dept. of Construction Engr., ETS, 1100 Notre-Dame St. West, Montreal, Canada, H3C 1K3, Phone +1 514/396-8667, FAX +1 514/396-8584, hugues.rivard@etsmtl.ca

[3] Dean of Research and Technology Transfer, ETS, 1100 Notre-Dame St. West, Montreal, Canada, H3C 1K3, Phone +1 514/396-8829, FAX +1 514/396-8525, claude.bedard@etsmtl.ca

In the following text, references will be made to the prototype system COBL DT (Meniru, 2005) to illustrate or clarify the discussion. Figure 1 shows the interface to COBL DT.

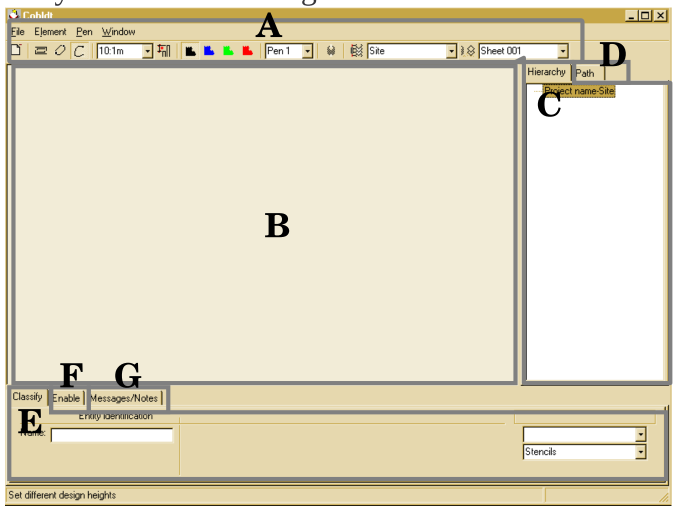


Figure 1: COBL DT Interface

There are 7 areas to note. The area at A provides a set of commands for creating and manipulating design items. B is used to draw or create design items. C & D are used to capture and organize the design space. C is specifically used to capture design items and the relationship between them while D is used to capture solutions and the relationship between two or more solutions. E, F and G are used to configure (customize) design items and to provide feedback to the designer.

DESIGN PROCESS

Building design begins when a designer is contracted and informed of a housing need. Work at this initial stage consists of determining requirements and understanding the exact use for which the building is needed. This process involves the creation and manipulation of data in the form of text and freehand drawings. The designer usually does this in an isolated environment with information that cannot be immediately shared with the other design team members. The goal is to establish a concept that best addresses the need for the building. It is of the best interest of the designer to explore more than one alternative for the concept in order to increase the possibility for success (Fricke, 1999, Casakin and Goldschmidt, 1999).

Ideally the integration efforts between the members of the design team should start early to make sure that the concept is optimal for all parts of the building. The sketch process of the designer defeats integration; even if sketches were shareable during this early stage it would be difficult to rely on them as they are continuously changing, which is one of the main reasons why designers isolate this process in the first place.

When the concept variations are ready, they are then reviewed by the designer and the owner. Parts of solutions may be mixed and matched (which is why it is best to provide variations) and finally a single solution will emerge or be chosen as the most viable to the owner's described needs (Atman et al., 1999; Cross, 1998). This solution is then transferred into a digital format using Computer-Aided Drafting and Design process. Thereafter digital plans are sent to other design members who use the information as a guide to completing their part of the project.

Often errors arise in the transfer from manual to digital data. Most of the errors come about as a result of misinterpretations of the designer's intent during sketching however the most important problem of this process is in early design data and how this data is made available to the designer.

The following section describes relevant characteristics of digital drawings, followed by descriptions for digitally capturing and relating building design solutions.

DIGITAL DESIGN ITEMS

When designers sketch or draw buildings there are two items that are dealt with; the first is in the form of a container which is usually an object with a size or more accurately an object that occupies space in the design. Typical example is a space e.g. Living Room. This item type is called a Corporeal item. The second item is any other data that the designer considers in the building design process. Such data do not occupy space in the design but they can influence Corporeal items. A typical example is a line. This item type is called an Incorporeal item (Meniru et al. 2003, Meniru 2005).

The highest level Corporeal item is the Space. It is made of sides (4) and ends (2) as shown in Figure 2. Spaces can be grouped together to form other more specialized objects such as a Floor Level or a Wall. Figure 1 shows a Floor Level that is made up of 3 Spaces and each space is made up of 4 sides and 2 ends. The highest level Incorporeal item is the line. It is made up of a direction (beginning point, end point and a width) and other specialized feature such as intensity for a force.

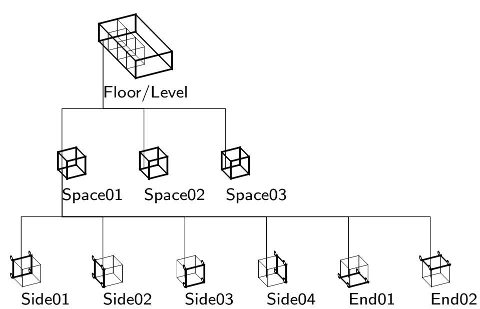


Figure 2: Space object

There are three stages in the use of design items during the design process:

FIRST STAGE (DRAWING PROCESS)

The first stage is the drawing or creation of a design item. This is a decision the designer makes and the basic properties of the item are recognized and recorded in the design automatically. For the Corporeal Item basic properties include boundary, location and orientation properties. Newly created Corporeal items are generic with default choices and parameters that have not been specialized for the design solution. This stage cannot be assumed to be a solution in the design process. For example a newly created Bedroom item does not have windows or doors and has not been shaped or specified to satisfy the client's needs.

SECOND STAGE (CLASSIFICATION PROCESS)

The second stage is made available so designers can designate the functions of the items being created in the first stage. A Corporeal item can be designated as a Kitchen. This action automatically recognizes the default characteristics of the Kitchen as a building space (a place for preparing meals that must include areas for cutting, washing and cooking) in addition to the properties described in Corporeal items (boundary, location and orientation). Many typical design items are provided in this stage as specializations from Corporeal items. Some examples are Living Room, Dinning Room, Storage, etc.

THIRD STAGE (CONFIGURATION PROCESS)

In this stage a typical building item is specialized to form a unique design solution. A typical Kitchen is a building space that has sides, a top (ceiling) and a bottom (floor). Detailing this kitchen may require the removal of one or more of the sides by adjusting the boundaries of the kitchen item. More configurations can take place which can include adding windows, doors or refining the proximity with other items in the design.

Much of the design process can be managed by assuming default values until the designer decides whether to change these decisions. The amount of flexibility is only limited by the designer's capabilities.

Figure 3 shows a newly drawn item in COBL DT at B1 and possible classifications at E1.

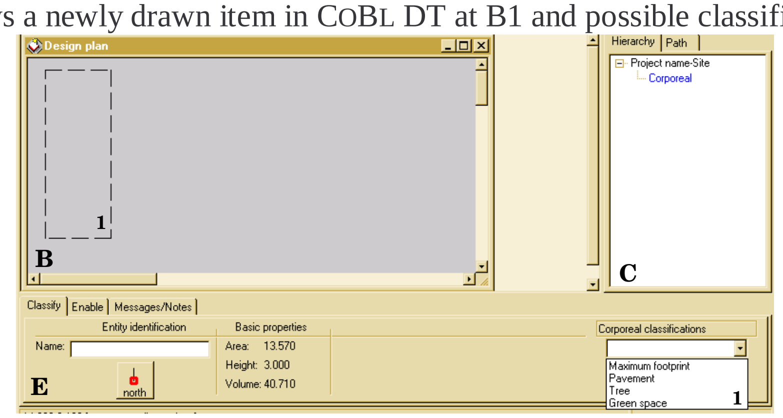


Figure 3: Preparing to classify a Corporeal design item

After classification, COBL DT automatically updates as shown in Figure 4 providing the designer access to configure the newly created design item at windows B1-5.

The discussion so far has been about how the designer can create typical design items and be able to detail them in a way that addresses a set of unique issues that will lead to a visual/aesthetic solution. What about the relationship between items and the functionality of the design solution?

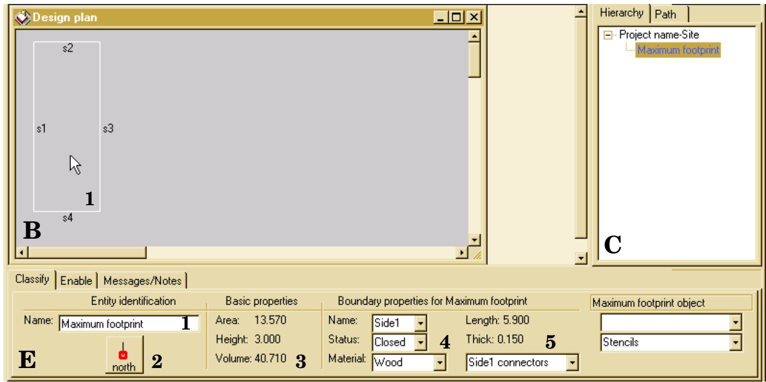


Figure 4: Classified design item

RELATIONS AND REASONING

The means for recognizing relationships is the use of hierarchies. Special Corporeal items called Connectors use these hierarchies to form the required relationships necessary in the design space. Utilizing relationships, it is then possible for design items to communicate and exchange information in a way that properly manages the design.

HIERARCHIES AND CONNECTIONS

As mentioned previously Corporeal items are like containers and they are able to determine their parent and any children. When a Kitchen is created, it comes with the knowledge of other spaces that can be contained within it such as cutting, washing and cooking areas. This makes it a parent space. In this way, parent spaces are able to manage the size and location of child-spaces throughout the design process.

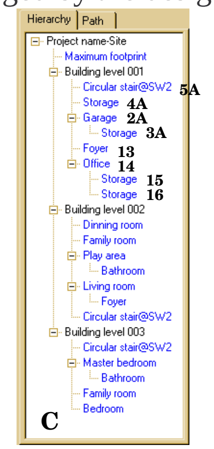
The parent and child relationship facilitates communications however communication between spaces outside of their hierarchies are managed through specialized Corporeal items called Connectors, which can be created only at the boundary of Corporeal items. There are 4 types: Door, Window, Hole and a Space Joiner. These connector items represent different relationships between spaces.

When a space is copied, it is possible to copy all of its child-spaces including the relevant relationships. When a space has been copied or moved into another, a check is performed to make sure that the relationship between the two spaces is feasible. It does not make sense to have a dinning area inside the garage for example. In the same way, entire hierarchies can be copied from an entirely different design solution and adapted to fit into a new one.

COBL DT allows the capture of relationships through the use of hierarchies as shown in Figure 5.

REASONING

During the early building design process there are certain requirements and properties that the designer establishes such as those for the owner, designer and the building codes/standards. Such data can be made available at the beginning of the design process so that during the design sessions, their implementation can be automatically checked and managed by the design tool.



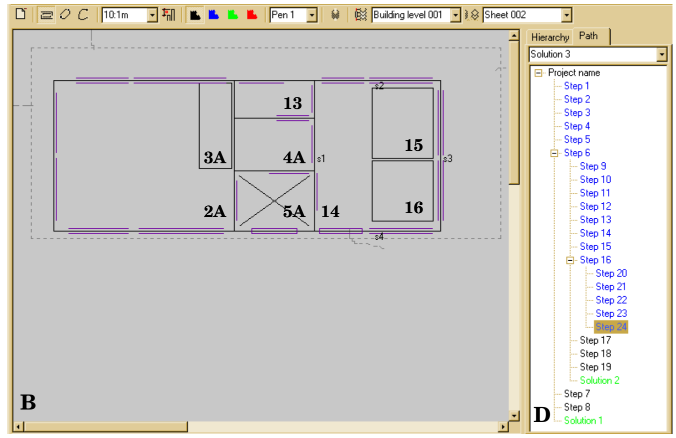


Figure 5: Relationships (hierarchies) Figure 6: Solution paths

Checking and managing requirements can be accomplished through the use of formatted files external to the design tool. For example in the case of designing a multi-car garage space, COBL DT provides assistance by checking to make sure that the required minimum space for the specified number of cars is satisfied. The following is an explanation of how this occurs.

An external file provides the tolerances and spatial requirements in a format shown in Table 1. All fields require single values except “tolerance” which require a list of six numbers in the form of: "length, number_0, value_0, width, number_1, value_1". Using these numbers, a tolerance of "value_0" will be applied to the total number ("number_0") of sides of type “length” and a tolerance of "value_1" will be applied to the total number ("number_1") of sides of type “width” for the design item.

Table 1: Standard design information

```
length    :  4750
width     :  1800
tolerance :  [ length, 2, 150, width, 1, 100 ]
height    :  450
```

To illustrate this, Figure 7 shows a space with sides "A" to "D". The tolerance required to accommodate the space is provided in the statement [length, 2, 150, width, 1, 100]. This states that along the length of the space, the two sides ("A" and "B") must have a minimum buffer of 150 while along the width at least one side ("C" or "D") must have a buffer of 100.

With the requirements set (earlier in the design process) for a 2 car garage space, the total spatial conformance will be calculated using data similar to Table 1 with the exception of the values in the “tolerance” which for a car is [length, 2, 500, width, 2, 500]. Any garage space that do not provide such minimum dimensions and tolerances are flagged for the designer's review.

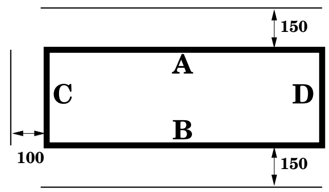


Figure 7: Illustrating tolerance determination

Checking and managing properties provides a utility for accomplishing complex computations that produce results applicable to the design. An example is the calculation of Solar Savings Factor (SSF). The SSF calculation compares the auxiliary heat needed by a solar-heated building to that needed by an energy neutral building that is otherwise similar. An energy neutral building is one with walls that have neither solar gain nor heat loss [Stein and Reynolds, 1992]. The data needed for this calculation is shown in Table 2.

The labels "ratio_low" and "ratio_high" identify values for the lower and higher ratios of solar glazing area to the floor area. "SSF_no_I_low" and "SSF_no_I_high" give approximate low and high SSF values when there is no night insulation whereas "SSF_I_low" and "SSF_I_low" correspond to the case with night insulation. These are applied in appropriate equations to obtain the required information needed by the designer for providing optimal indoor climates that take advantage of solar heating.

This minimizes the fear of errors and leaves the designer with the ability to apply required properties in a consistent manner.

Table 2: Values for calculating SSF

```
---area of solar glazing as ratio of floor area---
ratio_low      :  0.25
ratio_high     :  0.50
---approximate SSF values with no night insulation---
SSF_no_I_low   :  0
SSF_no_I_high  :  0
---approximate SSF values with night insulation---
SSF_I_low      :  54
SSF_I_high     :  72
```

Utilizing this ability to recognize relations and to provide support in applying the design requirements and properties, it is possible to capture design solutions in a way that is not disruptive to the designer during the design process but at the same time provides the necessary assistance and management of the design issues.

CAPTURING SOLUTIONS

At the beginning of the design process, it is not possible to specify all design parameters. Design solutions therefore consist of default (generic) and customized configurations. Capturing design solutions include the capture of both generic and custom items.

SOLUTIONS: GENERIC

A generic solution provides the same functionality regardless of its application or location. For example a Dinning room is a space where food is served and a Bedroom is a space where one lies down to rest. These are generic solutions for these spaces. Taking the Bedroom example, a generic bedroom must then be able to accommodate at the minimum a sleeping space and a changing space, including the circulation required to do both by an average sized individual.

This information can be predetermined so that the designer is assisted in creating such a space either by checking to make sure that minimum spatial requirements are met or by providing the designer with a sample bedroom that already satisfies all requirements.

This makes it easy to reuse prior knowledge that has been tried, tested. COBL DT allows the use and capture of generic design items as Corporeals as shown in Figure 3 at C.

SOLUTIONS: CUSTOM

Designers also may customize spaces to address unique parameters or requirements. For example a bedroom with an armchair and a split level is a custom solution. So customization in this sense could be a change in the position or configuration of a characteristic as well as the introduction of additional items that are not a requirement for the typical use of that space.

This information cannot be predetermined. The designer must make sure that the original function of that space remains applicable regardless of any customizations. Once a successful custom solution is created, it becomes possible to reuse them as complete solutions or a starting point for further customization. COBL DT allows the capture of custom design items as shown in Figure 4 at C.

SOLUTIONS: PATHS

Designers often create alternative solutions. These are most useful when they are accessible throughout the design process making it possible for the designer to view (assess), copy and move (modify) solutions or hierarchy trees as needed across solutions.

A path represents the collection of steps the designer goes through during the design process as shown in Figure 8. Path01 and Path02 are two solution paths. House, AP01 and AP02 are consecutive steps in the development of solution path Path01. Each step represents a snapshot of all design items that have been created up to that point in the design process. Solution path Path02 is derived from Path01 starting at step AP01. A complete description of the steps in Path02 therefore will include House, AP01, SP01, SP02 and SP03. Figure 8 also shows that parts of solution path Path01 have been reused in Path02 (data from AP02 to SP01). A simplified illustration of a solution path is shown in Figure 9. A step in Figure 9 is similar to any of the steps in Path01 or Path02.

Solution paths provide a fast and intuitive way to manage and manipulate design information regardless of the complexity of the design. The following is a description of the creation of a new solution path as illustrated in Figures 9, 10 and 11.

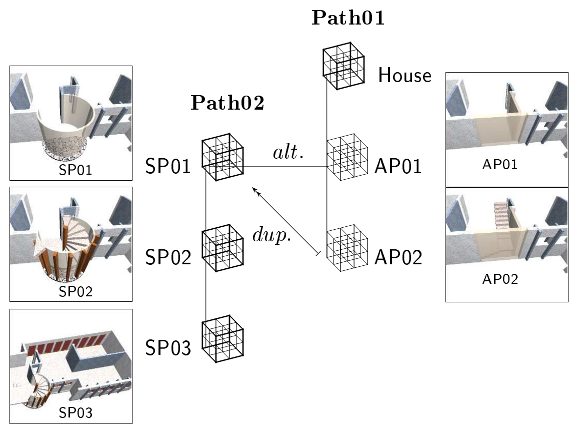
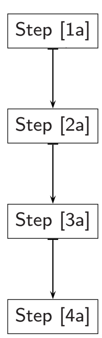
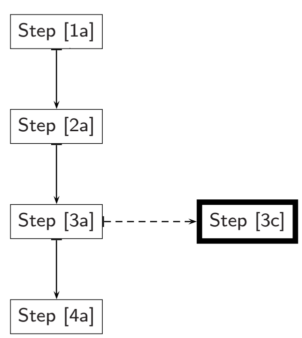


Figure 8: Solution paths

Assuming the designer selects Step [3a] in the current path (Figure 9) as the beginning point for a new path, the design items in this step are located and replicated in the design. These replicated items represent a branch-off step of a new solution path as shown in Figure 10 at (Step [3c]). All other design items in the steps that lead to this branch-off step, starting with the first entity in the current solution path (Step [1a] to Step [3a]), are all replicated and collected in a new list (Step [1c] to Step [3c]) as illustrated in Figure 11.





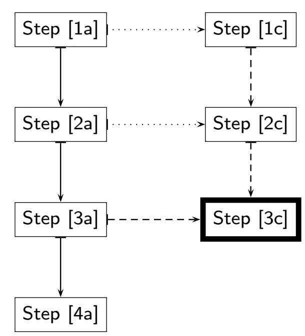


Figure 9: Solution Path Figure 10: Branch-off Figure 11: New solution path

The current solution path (Step [1a] to Step [4a]) is saved while the replicated items become the new solution path. Solution paths are independent from each other because design solutions are unique. Once a branch-off occurs the development of a new solution should not affect any previous ones. The aim of collecting solutions is to save or freeze previous efforts making it possible for the designer to investigate new ideas and still be able to return and continue development. Replication makes this possible as the solutions are represented by copies that are not connected. For example the designer starts a new solution path consisting of Step [1c] to Step [3c] as shown in Figure 11. If the object represented by Step [2c] is edited or removed and the designer decides to go back to the first solution path represented by Step [1a] to Step [4a], everything will be as it was because Step [2c] was a copy and all manipulations on it so far did not affect the state of Step [2a], its original version. COBL DT allows the capture of solution paths as shown in Figure 6.

SUMMARY

Design solutions involve complex considerations to satisfy housing design needs. The capture of these complex considerations and relationships is possible in a digital environment and necessary for collaboration in the building design process. To accomplish this however it is important to enable the designer to work in the digital environment from the early design stage which is when critical decisions are made that set the basis for all other stages of building design and construction.

The use of digital items (Corporeal and Incorporeal items) provides the means for the designer to consider building design issues and to create parts that have a basic knowledge of their purpose in the building.

Using such design items, the capture of relationships and knowledge in the design becomes possible. Relationships are captured through the use of hierarchies and knowledge is embodied in the digital items, which enable a rich interaction and feedback that enhances the design process.

Solution paths capture the solutions including any alternatives explored. The solutions can be browsed (reviewed) and parts of solutions can be developed, copied or moved around in an attempt to create the best possible solution.

Some issues for further research include; the need to provide an interface for transferring requirements provided in analog format into digital form. This includes an interface for on-the-fly applications of requirements. The design process is dynamic and therefore rules are created, removed or changed frequently. For example while designing a bathroom a designer might decide to keep an item or a part of the bathroom space out of the line of sight from another part of the design, for privacy issues. It should be possible to create, remove or adjust such rules during the design process.

Some implementation issues for future work include; the need to merge external solutions. COBL DT allows the designer to merge work from different solution paths however it is conceivable that a merge from different design sessions saved in different files would be desirable. A way of importing solution paths from other design sessions and selectively merging items should be implemented.

REFERENCES

Atman C J, Chimka J R, Bursic K M and Nachtmann H L (1999) A Comparison of Freshman and Senior Engineering Processes. In: *Design Studies*, vol. 20(2): pp. 131-152.

Casakin H and Goldschmidt G (1999) Expertise and Use of Visual Anology: Implications for design education. In: Design *Studies*, vol. 20(2): pp. 153-175.

Cross N (1998) Natural intelligence in design. In: *Design Studies*, vol. 20: pp. 25-39.

Fricke G (1999) Successful approaches in dealing with dierent precise design problems. In: *Design Studies*, vol. 20(5): pp. 417-429.

Meniru, Kene C.U. (2005). *Computer-Aided Conceptual Building Design*. Ph.D. Thesis, Centre for Building Studies, Concordia University, Montréal, Canada.

Meniru, K., Rivard, H., Bedard, C. (2003). "Specifications for Computer-Aided Conceptual Building Design." *Design Studies*, Elsevier Science Ltd., vol. 24 (1): pp. 51-71.

Purcell A T and Gero J S (1998) Drawings and the design process. In: Design Studies, vol. 19(4): pp. 389-430.