A METHODOLOGY TO BREAKDOWN BUILDING DESIGNS INTO A HIERARCHAL DECOMPOSITION ASSEMBLY OF DESIGN ENTITIES

Michael Cook[1] and Al Zeiny[2]

ABSTRACT

Using previous designs to generate new designs is desirable because it saves time and effort, and because the concept has proven effective in previous situations. In a previous effort, a knowledge based system to store building design information was developed for the purpose of using prior experiences to generate solutions for new ones. Before the system can be of any use, it is required to populate the knowledge base of the system with previous building design cases. This is achieved by breaking the building into design entities arranged in a containment hierarchical structure and entering every entity and its modular information in the knowledge base. This paper describes the methodology used to break down the building into design entities, and the underlying conceptual and data models.

INTRODUCTION

Engineers use various CADD design tools to aid them in generating their designs. These design tools may be described as islands of automation because they do not talk to each other. Each design tool is specialized in performing a detailed design task of a specific structural component of the building without any communication with either the tools responsible of designing other components or the central design representation. Furthermore, the drafting software captures the geometry only and does not capture any rationale or relationships among design entities. Not to mention that in many cases, different drafting files are used for every design drawings without linking entities of a certain view with entities of other views. For example, if drawing sheet S4 shows the foundation plan, a square footing is drafted as a square entity in the file that has the

[1] Undergraduate Student. Email: mc101@evansville.edu
[2] Assistant Professor, Mechanical and Civil Engineering Department, University of Evansville, Evansville, IN 47722. Email: az12@evansville.edu

drawings in sheet S4. This square entity will have absolutely no link to the same footing shown in a detailed cross section in drawing F7, which is also saved in a different file. The overall design effort depends mainly on the design engineers who extract output data from one island of automation and feed it to the next, make all professional decisions required to complete the design, and then pass the design sketches to the drafting technicians who draft different views of the design using drafting software such as Autocad without linking different drawing entities in different views that belongs to the same building component. One of the main disadvantages in such almost manual design procedure is that the engineers and draftsmen have to manually take the output from one island and feed to another, which increases the possibility of errors and repeated efforts and makes it more difficult to make changes and accomplish the trial-and-error design procedure.

METHODOLOGY

In order to overcome the aforementioned difficulty, a design support system, which is a computer software used to assist the engineer in performing design tasks, was built to store designs in a design repository, support automatic propagation of changes, support design iterations required for the trial-and-error design procedure, support adaptations of previous designs either fully or partially and facilitate the overall design process. In other words, the final objective of this research effort is to build a design support system that has efficient way of storing previously generated designs, assist the engineer to adapt prior designs into current design tasks, and assist him in performing design tasks as well. A prototype of the desired system was developed in the Civil Engineering Department at the University of Evansville and called A2ZCAD (Zeiny 2004). Before the system can be of any use, it is required to populate the knowledge base of the system with previous building design cases. This is achieved by breaking the building into design entities arranged in a hierarchical structure and entering every entity and its modular information in the knowledge base. These design entities are associated with parameter collections that store all material properties, cross section shape and dimensions, design rationale, etc. Alternatives can also be stored as branches in the hierarchy decomposition tree of design entities. In addition to the decomposition relationship among design entities, they are also related with groups, and qualitative relationships. Groups help assign the same parameter collection to similar design entities, while qualitative relationships relate entities with various relations such as “connect” and “support”. Classifications are used to index the design entities in the database for future retrievals using queries. The classification system was also arranged in a hierarchical decomposition fashion and linked to various types of design entities for possible use.

CONCEPTUAL MODEL

The breakdown of the project to design entities is based on the conceptual model used by (Rivard and Fenves 2000) as illustrated in Figure (1). The design entity could be as big as the entire project or as little as a small element in one of the connections. As shown in the figure, a slab element is part of the roof and floor, which is part of the horizontal subsystem, and so on all the way back to the general entity of the project. Figure (2) is a view from A2ZCAD where the Design Entity Type decomposition tree is partially expanded to show the arrangement of design entity types that may exist in a structure. The design entity type tree is a fixed tree that once created; it will appear every time the program is open. The purpose of the classification tree is to name every type of possible design entity within the structure, such as the highlighted column in Figure (2).

ILLUSTRATIVE EXAMPLE

To illustrate the methodology used to store design information, we will consider the simplified building project shown in Figure (3). The shown low rise steel building is intended to be portioned to three spaces. Each space was intended to satisfy certain functional requirements specified by the client. The structures is made of six rigid frames serving as the main load carrying system for both gravity and lateral loads in the east-west direction. Gravity loads are loads due to the weights of the building components, occupants, furniture, …etc, while lateral loads are generated due to wind or seismic events. Four x-bracings are provided to carry lateral loads in the north-south direction.

The building can be broken down into entities as shown in Figure (4). Each entity is entered into A2ZCAD in its hierarchical order as shown in Figure (5). The title that appears in parentheses next to the design entity name is the design entity type. Each design entity is entered into the hierarchical tree by a dialog box that is shown in Figure (6). The “Entity Name” is the name of design entity, and the “Entity Type” is its linked design entity type.

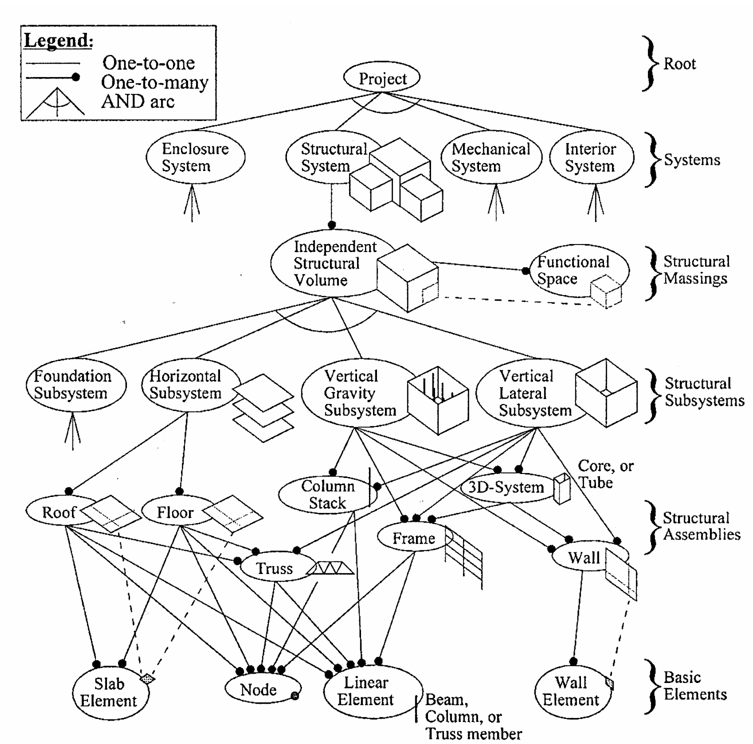

Figure (1): Evolution of Design in a Top-Bottom Refinement Fashion

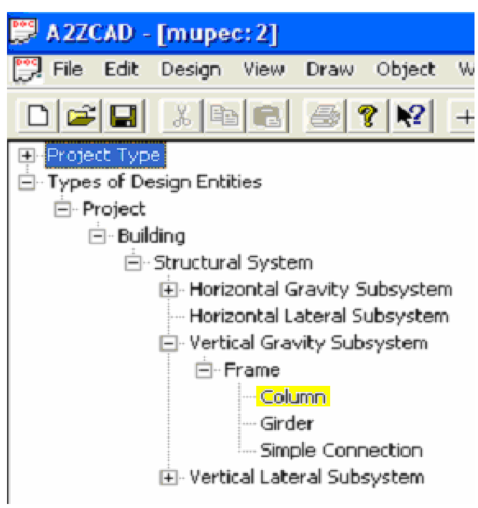

Figure (2): Classification Tree from A2ZCAD

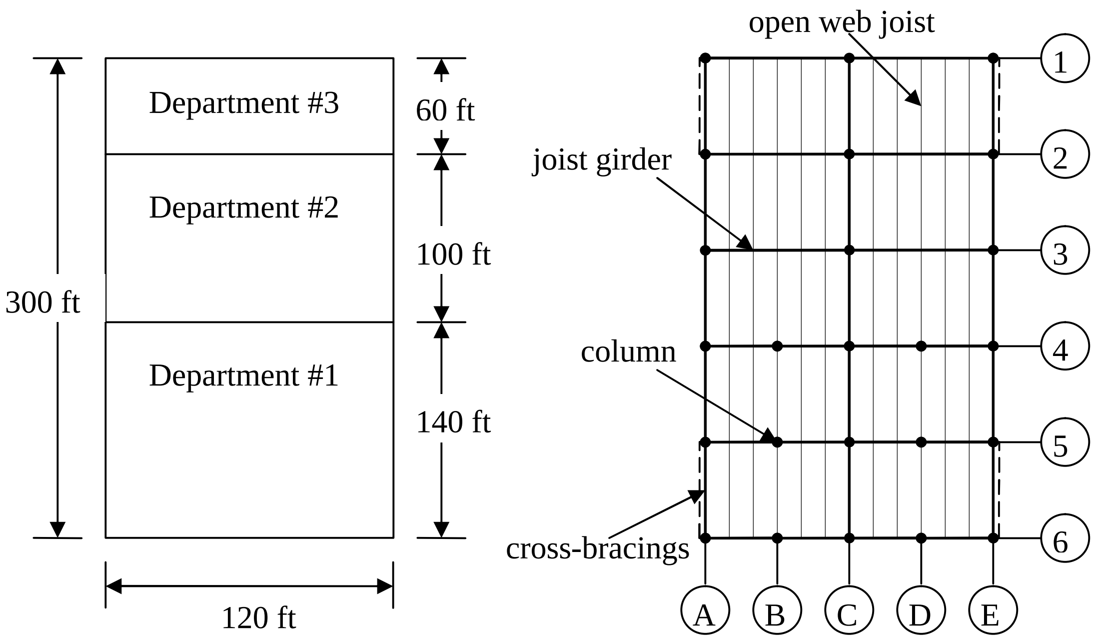

Figure (3): Low-rise Steel Building

Taking a further step into the example project, Frame 1 is a generic container for many more design entities. It can be broken down into more specialized design entities such as: columns, joist girders, moment connections, simple connections, and supports. These elements can be broke down into more specialized design entities as shown in Figure (7). Each entity is a specific element from frame one. For instance, columns A1, C1, and E1 are shown in the plan view of the structure in Figure (3) as well as the elevation view in Figure (8) as part of frame 1.

Each design entity is linked to a set of parameter collections that give information about that member's details such as its length, orientation, slope, cross section shape, material properties …etc. Some of these parameters are geometric parameters that determine the dimensions of Autocad entities and are linked directly to the related Autocad entity for the purpose of automatic propagation of changes. For example, changing the length of the beam causes the length of the corresponding Autocad entity to change automatically resulting in a faster trial-and-error design process. In addition, related design parameters are linked together with a set of quantitative relationships that capture the mathematical relationships between various design parameters as shown in Figure (9). For example, if the depth of the beam is equal to the span measured in the horizontal plane divided by thirty, a quantitative relationship between the depth of the beam and its span is generated as shown in the figure to represent this mathematical relationship between them. In case of changing the beam span, the beam depth will also automatically change by the re-evaluation of the previously established quantitative relationship. This causes automatic propagation of changes in a manner similar to what happens in spreadsheets.

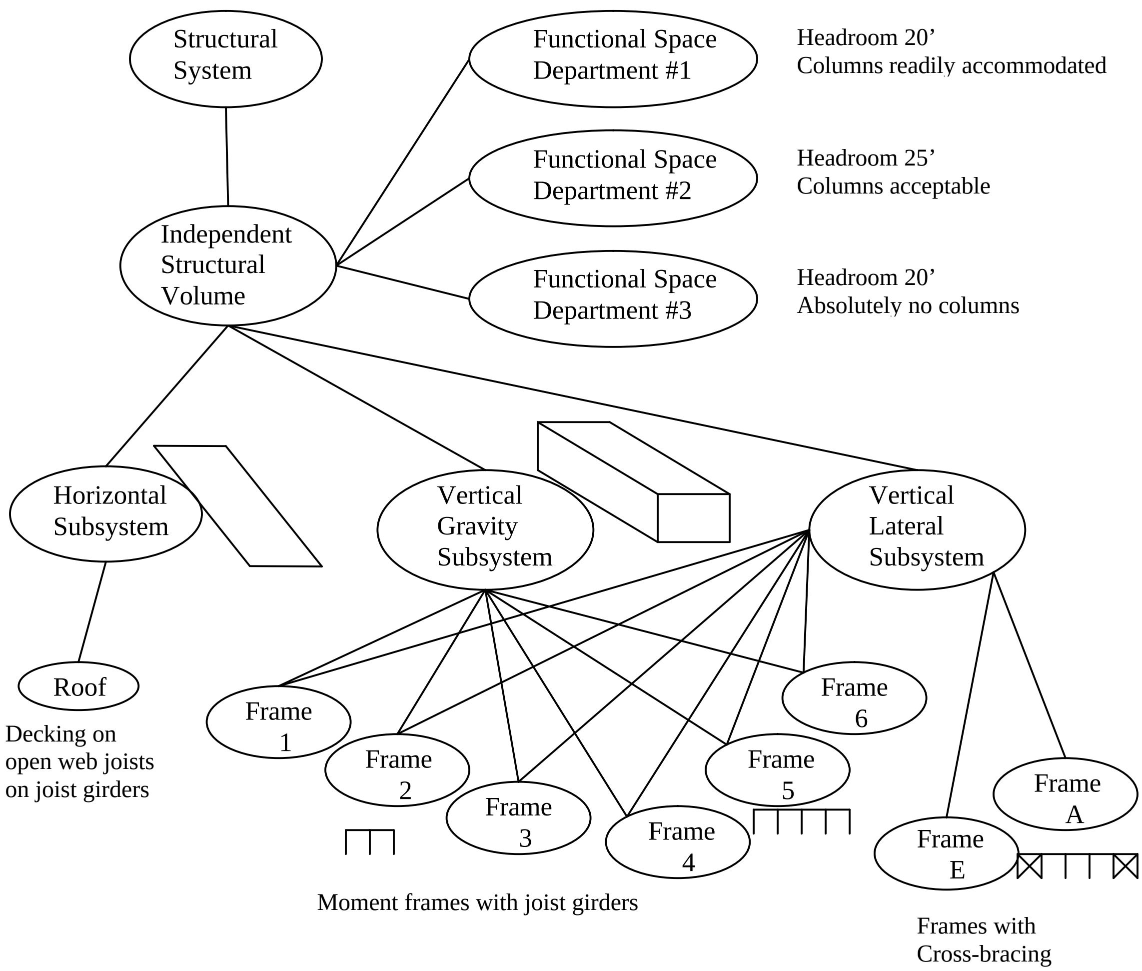

Figure (4): Hierarchical Decomposition of the Low-rise Steel Building

STORING PRODUCT DESIGN INFORMATION

Each Design Entity object is linked to Entity Parameter Collection objects that stores the design parameter data in a set of parameter-value pairs. These pairs may be assembled into small cohesive subsets organized at two hierarchical levels, the group level and the collection level. At the group level, data are grouped into three subsets, the Capacity Parameter Collection group, the Demand Parameter Collection group, and the Geometric Parameter Collection group.

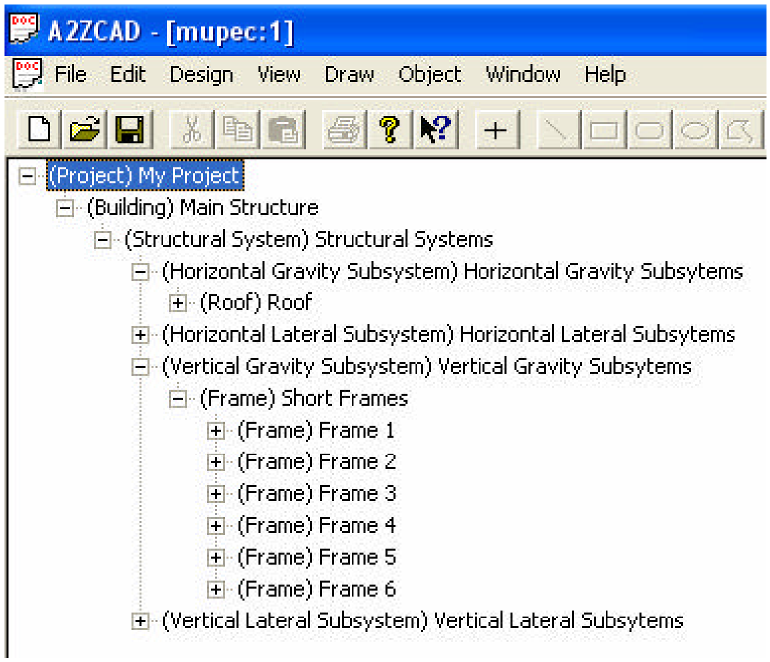

Figure 5: Hierarchical Decomposition of Design Entities in A2ZCAD

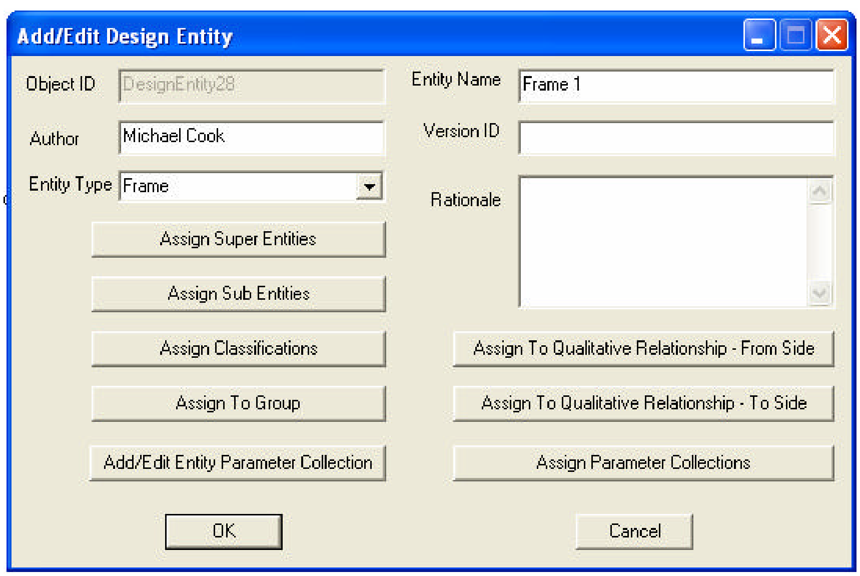

Figure (6): Adding Design Entities

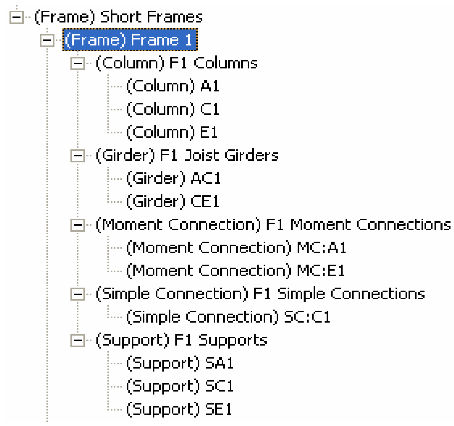

Figure (7): Frame 1 Design Entities

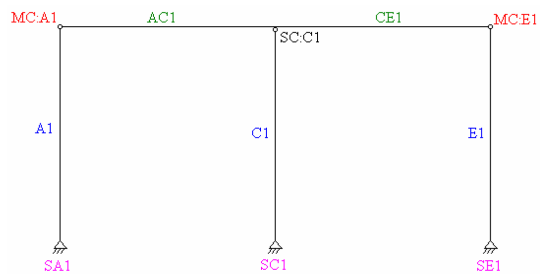

Figure (8): Elevation view showing frame 1 components

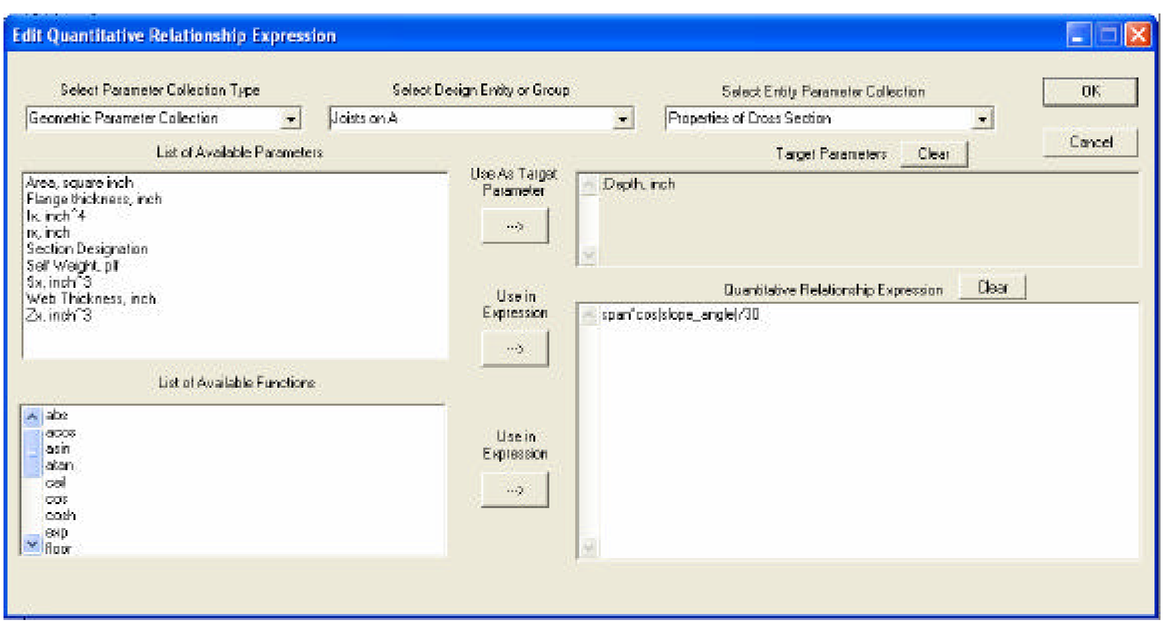

Figure (9): Dialog Box used to Add Quantitative Relationship Expressions

The Capacity Parameter Collection group defines intended purposes, requirements, and constraints on the entity that have to be satisfied to realize the intended purpose. The Demand Parameter Collection group includes all the physical and spatial characteristics that define the actual design of the entity as well as the behavior of this entity under various loading conditions. In order for the design to be successful, designed demands must not exceed capacities, i.e., D/C ratio is less than 1. Demands are stresses and deflections resulting from the applied loads on a building entity while capacities are the corresponding strength and allowable deflections. This requirement is checked by the D/C Checker objects. The Geometric Parameter Collection group has the geometric parameters linked to the drafting software to ensure that the product drawings change automatically as geometric parameters change. While Qualitative Relationship objects link design entities together with domain specific relationships, the Quantitative Relationship objects link various parameters together to ensure the propagation of any parameter changes accordingly in a fashion similar to what happen in spreadsheets. Such a feature makes the model highly computable and various design alternatives may be explored easily, not to mention the ability of generating similar new designs from old ones by creating templates from old designs.

At the second level of data aggregation, the parameter-value pairs of an entity are combined into small cohesive subsets, each of which is called a parameter collection. A collection is defined as a group of closely related parameters that are found together in a repository (access-cohesive), instantiated at the same time (time-cohesive), and that represent the same concept (concept-cohesive). Cohesion is the only criterion used in grouping entities. It is defined as a measure that shows how closely the parameters of an entity relate to one another. An example of a Parameter Collection is one that collects together the parameters used to describe the section properties of structural members such as depth, width, cross section area, and moment of inertia. Parameter collections allow entities to be refined in staged steps by adding sets of parameter-value pairs to the entity as they are generated in the design process. Hence, there is no need to predict all possible parameter-value pairs needed in a product entity at the outset. Parameter collections also allow the integration of multiple views by multiple design teams in one entity by including collections that are specific to each view and each design team as well as components that are shared among all views and all design teams (e.g., material properties. Figure (10) shows the dialog box used to enter a design parameter collection.

CONCLUSION

The presented methodology to store design information is based on the conceptual model of breaking the structure into a collection of design entities arranged in a hierarchical decomposition fashion. Such a conceptual model is similar to the strategy of divide and conquer because larger more generic design entities are divided into smaller more specialized (or less generic) ones and so forth up to the very atomic detail. This divide and conquer strategy is efficient in both storing design information and handling complicated design tasks. Collections of design parameters are linked to design entities for the purpose of storing all relevant design information required to design, fabricate and finally erect the design entity. Automatic propagation of changes is achieved using quantitative relationships between various design parameters as well as attaching geometric design parameters to Autocad drafting entities. The overall developed design supporting environment is found very helpful tool to reduce the design time and cost, as well as increase the efficiency of the trial-and-error design iterations.

REFERENCES

Rivard, H. and Fenves, S. (2000). “A Representation for Conceptual Design of Buildings”, Journal of Computing in Civil Engineering, ASCE, July, Vol. 14, No. 3, pp. 151-159.

Rosenman, M. A., and Gero, J. S. (1996). “Modelling Multiple Views of Design Objects in a Collaborative CAD Environment.” Comp.-Aided Des., Oxford, U.K., 28(3), pp. 193–205.

Zeiny, A. (2004) “Computable Dynamic Design Repository for Product Data Representation”. ASME International 24th Computers and Information in Engineering (CIE) Conference, September 28-October 3, 2004, Salt Lake City, Utah.

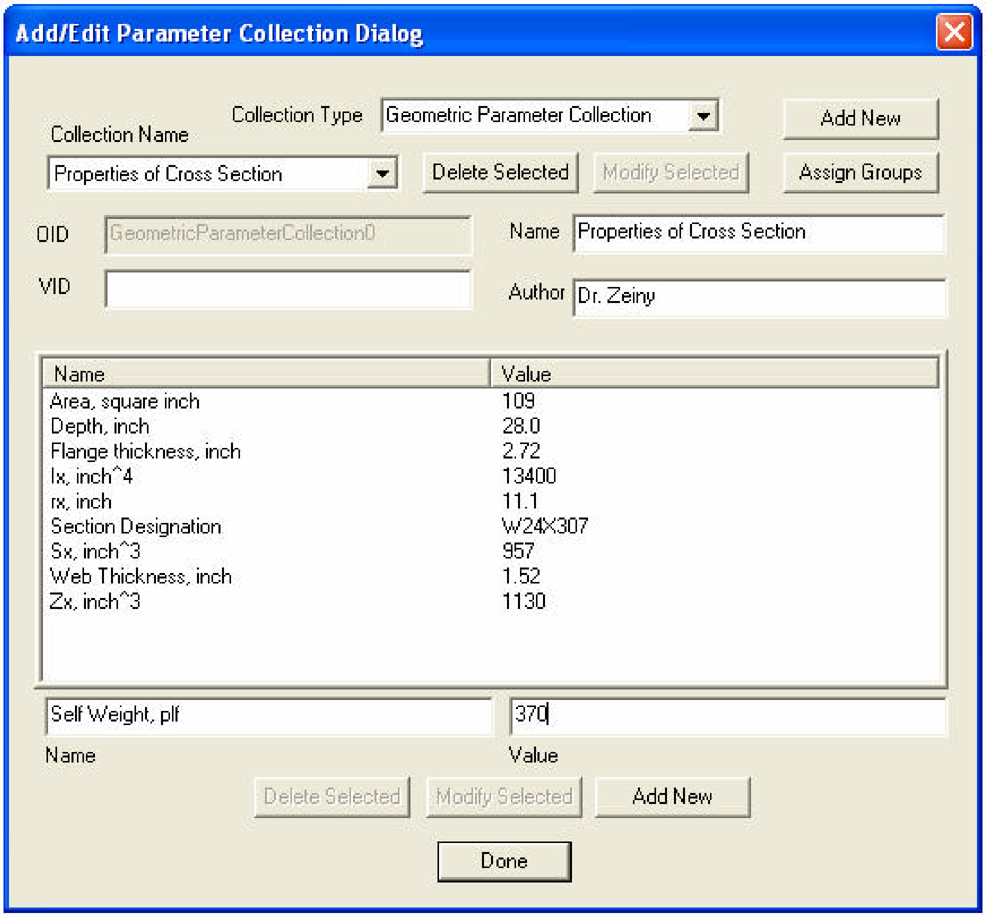

Figure (10): Dialog Box used to Add design Parameter Collections