Constraint-Based Three-Dimensional Modeling as a Design Tool

Kenneth L. Tobin
Ohio State University

Architectural design knowledge which may take the form of constraints and how it can be incorporated into the solid modeling process is explored and discussed. This theoretical exploration leads to an illustration of a functional general purpose three-dimensional solid modeler which utilizes design knowledge as constraints upon the interactive solid modeling process. From this illustration, it is shown that the incorporation of constraints into Computer-Aided Architectural Design can effectively assist in the early stages of the design process. A prototypical application is presented which provides an example in which the definition and implementation of specific design knowledge constrains or characterizes the generative and interactive behavior of user definable three-dimensional entities. The basis of this work lies in the ability to create a design space, or building envelope, and to allow the interactive modeling of conceptual elements within, and constrained to, that design space, including the realistic interaction between the entities themselves.

In 1972, Nicholas Negroponte wrote "Computer-aided design cannot occur without machine intelligence - and would be dangerous without it" (Negroponte, 1972). This early insight to the possible introduction of knowledge-based systems to architectural design has led the way for many theories and applications (Coyne, 1985, 1986, 1990; Mitchell, 1977, 1986, 1990; Yassios, 1983, 1987). However, until recently, the profession has greatly ignored the potential of such systems, relying on the utilization of computer technology to aid in the production phases of the design process - mainly drafting. More recently, advances in computer technology, predominantly in the micro-computer arena, has led to a greater understanding, representation, and manipulation of knowledge in a manner which has made it possible to construct knowledge-based design systems capable of aiding the designer in the early phases of the design process. According to R.D.Coyne, "The emphasis now is in finding methods of appropriating and rendering operable the knowledge available to designers" (Coyne, 1990).

The following discussion explores the use of design knowledge with the intent of establishing a more intuitive and effective method of utilizing computers to assist in the design activity. A prototypical application, CMod, is presented to support the fundamental applications of constraint-based solid modeling, and to provide a visual representation of the concepts presented.

Overview.

The goal of this discussion is to illustrate an avenue in which existing knowledge can be synthesized and utilized to constrain the interactive generation, manipulation and behavior of geometric solid and spatial elements. By utilizing available knowledge to constrain geometric entities, it is possible to bring a basic three-dimensional solid modeler closer to the early stages of the design process, and allow the CAAD system to be more of an aid to the designer.

Knowledge, for the purpose of this discussion, includes the process, declaration, and state of having information pertaining to a particular entity or group of entities. It becomes the representation of the ability to gain, apply, and state the behavioral attributes of an object or entity may possess. The interaction of an entity and the world, or between entities, is therefore limited or constrained to the knowledge representing that entity. A constraint, therefore, becomes the ability to limit the use, generation or manipulation of entities within the
established bounds of knowledge.

Constraining a geometric element can be illustrated by considering a design of a building on a site on which zoning regulations impose limitations to the design. Initially, there are limitations to the site itself: boundaries which limit the breadth of construction and descriptive symbols which establish relationships and adjacencies.

![Design space constrained and defined by the specification of site restrictions.](image)

Figure 1: Design space constrained and defined by the specification of site restrictions.

Typically, there may be setback requirements for the front, side, and back yards, establishing a minimum distance that a building must maintain with respect to the site boundaries. There may be incremental requirements such as height limitations and shadow/sunlight restrictions. Additionally, legal constraints, such as rights-of-way, may be imposed, further restricting the site.

By applying these limitations to a geometric entity, we create a design space which contains specific information, or design knowledge, about the allowable design volume. Figure 1 illustrates a visualization of the design space defined by site restrictions. This knowledge of the design space can then be imposed on the interactive solid modeling process, allowing only acceptable conceptual design solutions to be generated within the allowable design space. Figure 2 provides an example of conceptual design elements created within the limitations imposed by the design space.

![Illustration of conceptual modeling within the allowable design space.](image)

Figure 2: Illustration of conceptual modeling within the allowable design space.

The basis for this discussion is this ability to constrain, with the use of design knowledge, the generation, manipulation, and behavior of geometric entities. By allowing the user to interactively identify and specify design knowledge, the interaction of these geometrical entities reduces to constraint satisfaction in which the lower level decisions are required to be met during the solid modeling process. Therefore, the benefit of such constraint-based modeling is that the designer has introduced these lower level decisions at the beginning of the design process, and allows him the ability to use the computer and design knowledge earlier in the development of a design solution.

To facilitate this approach, the implementation of the constraint-based modeler, C-Mod, chooses to model the design process, not as a complete generator of solutions, but as an interactive cycle in which the designer can manipulate the proposed solution and to guide it to a satisfactory conclusion. This method establishes the design criteria, or design knowledge, prior to the introduction of a design solution and limits or constrains the interactive generative process to that of the design knowledge established.

The implication of this approach to the current CAAD theory centers around the interactive use of the design knowledge during the conceptual stage of design. The application of such an approach seeks to utilize this information interactively in a graphic and visual mode, introducing design knowledge, and therefore constraints, to a three-dimensional solid modeler.
Constraint-Based Modeling Operations.

The realistic generation and manipulation of solids and spaces in an interactive environment requires the use of specific information pertaining to the entities themselves and to the interactive behavioral attributes between entities. To facilitate this realistic interaction of entities, two types of information are required: physical and relational attributes. Physical attributes such as representation, dimensional data, area limitations, volumetric requirements, rotational attributes and spatial mobility, are applied to the entity itself. Relational attributes such as proximity criteria, containment characteristics, and associative attributes, are applied to entities which interact with one another. By applying this specific information, the interactive solid modeling process is constrained and limited to the bounds established. The results of this application of specific information or knowledge about an entity in an interactive solid modeling environment is a constraint-based solid modeler.

Allowing for the definition and implementation of specific design knowledge which constrains the behavior of user definable three-dimensional entities is critical to a successful implementation of constraint-based solid modeling. A constraint-based solid modeler should, therefore, provide four primary operational capabilities: 1) The ability to specify the design knowledge applicable to user definable three-dimensional entities such as solid and spatial entities. 2) The ability to modify, store, and retrieve the entity specifications provided through knowledge specification. 3) The ability to manipulate the three-dimensional entity in a manner which is consistent with the behavioral characteristics dictated by the entity specifications. 4) The ability to extract, or query, information from an entity which has been provided by the specifications of that entity. This section explores each of these primary operations in order to illustrate the role each must play in the interactive modeling process.

Knowledge Specification. The foremost important ability of a constraint-based solid modeler is to specify and distinguish specific information pertaining to an entity. Information, or design knowledge, which is typically represented graphically must be specified by the user, or read from a data base containing the required information, and made available to the modeler. Figure 3 graphically illustrates specific design knowledge pertaining to a site. An interactive method of communication between the user and the system allows the user to input specific design knowledge applicable to a specific entity, or to an environment which is to be modeled. The modeler, once the design knowledge is present, can utilize this information to aid in the process of interactive modeling by providing the lower level satisfaction checking to ensure that the requirements set forth are met.

Knowledge Manipulation. The specification of information, or design knowledge, is the initial step in conveying the scenario in which the modeling environment is to be constructed. Once the specific entity design knowledge has been provided, a constraint-based solid modeler should provide the ability to manipulate that information. This includes the ability to modify, store, and retrieve the entity the

Figure 3: Design knowledge of a space.

user to modify constraints, or information particular to a specific entity, the environment can remain dynamic in nature. This flexibility provides a method of constraint resolution in which the user can choose to apply or disregard a particular constraint during the interactive modeling process. Additionally, by allowing a set of constraints to be stored and retrieved, a preset description of entities, or entity types, can be applied to the modeling environment without having to specify the entire set of constraints.
again. Figure 4 illustrates an application of multiple entities with the same set of constraints.

![Diagram](image)

**Figure 4:** Preset constraints applied to subsequently generated solid entities.

To allow for this flexibility, two types of constraint knowledge are represented, entity specifications, and system specifications. Entity specification is the set of constraints/attributes which pertain to a specific entity. It is that set of specifications which control the operations involving that entity. The second form is system specifications, or system defaults. This set of constraints/attributes are what the system uses when a new entity is generated. Both sets should be accessible to the user at any given time. By allowing this manipulation of information, the user has the flexibility to control knowledge about a specific entity and the system specifications.

**Constraint Modeling.** Providing the ability to interactively manipulate the three-dimensional entity in a manner which is consistent with the specified design knowledge creates the largest and most important operation of a constraint-based solid modeler. To accomplish this, three levels of constraint modeling are provided: entity, relational, and identification.

The entity level applies to the physical characteristics of the entity and relates to the generation and editing of solid objects. Interactive relationships between entities make up the relational level, and apply to the geometrical transformations of translation, rotation, and scale. The final level of identification is required to distinguish the differences between entities, and provides the ability to store and retrieve the constraint set. The following discussion elaborates on each of these levels, and includes an illustration of the major attributes which provide the ability to interactively model design knowledge as constraints.

**Entity Level.** The characteristics at the entity level are the specific physical attributes which the entity is required to satisfy during the generation and manipulation of the entity in the three-dimensional solid modeling process. The physical attributes provide a realistic representation of a narrow spectrum of design knowledge upon a specific entity. For the purpose of illustrating the constraint-based solid modeling process, five major physical attributes have been chosen: spatial representation, dimensional attributes, spatial characteristics, rotational attributes, and mobility characteristics. Although an unlimited number of descriptive information can be obtained and applied to the generation and manipulation of geometric entities, these five physical attributes represent a selection of primary characteristics which govern the creation of physical objects. The following is a discussion of each of the five physical attributes which comprise the entity level constraints.

**Spatial Representation.** The fundamental object description/constraint is found in the spatial representation attribute. By denoting an object as a solid entity, the modeler must perform operations on that entity as a solid. Thus, a solid is expected to behave as a solid entity, and a spatial entity is expected to behave as a spatial container of other entities. The distinction between the two representations lies in the ability to contain mass. Solid elements are just that - solid - they contain mass, and cannot be occupied by another entity containing mass. Spatial elements do not contain mass, they are composed of space, or a spatial void, and have the ability to contain other entities such as spaces or solids. Figure 5 graphically illustrates the difference between the two representations in a solid modeling environment. As a
fundamental characteristic of constraint-based solid modeling, this ability to distinguish between spaces and solids interactively provides a strong foundation to utilizing design knowledge within an interactive environment which allows immediate feedback to entity interaction.

Figure 5: Three-dimensional representations of solids and spaces.

Dimensional Attributes. Dimensional attributes constrain the generation of entities, specifically extrusion, as well as the geometric editing operations of the modeler to minimum, maximum, and incremental values (figure 6.)

Figure 6: Visualization of the width, length, and height dimensional criteria.

The application of dimensional data supports the editing of all topological levels: point, segment, face, and object. To facilitate this characteristic in an interactive environment, a bounded box dimensional restriction upon the generation and manipulation of the entity is provided. The bounding box method maintains the dimensional information regardless of the effects of 3D transformations such as scale, rotation, or translation, regardless of the topological level on which the operation is being conducted (figure 7.)

Figure 7: Geometric editing within the limitations of the dimensional constraints.

Retaining the dimensional data in this manner allows the user to specify the minimum, maximum, and incremental dimensional constraints upon a geometric entity.

Simply restricting the operations to minimum and maximum dimensional limitations does not fully capitalize on the nature of dimensional data. Modular construction, such as kitchen cabinetry, utilizes dimensional data to indicate when another module should be generated. To support this type of design knowledge, the modeler must allow the user to specify a replication threshold, and replicate the entity when generation extends beyond this threshold. When the replication feature is enabled, the maximum value, in lieu of constraining the modeling process, becomes a threshold value used to invoke replication. Extending beyond this maximum value will result in the creation of an additional element with the same set of constraints and attributes. This results in an interactive replication of primitive during a single generation process (figure 8.)
Spatial Characteristics. The spatial characteristics impose similar constraints to the generation and editing features of the modeler as do dimensional attributes. In addition to, or in lieu of, dimensional attributes, spatial constraints of area and volume may be imposed on the entity. This allows the limitation of the generation and manipulation of the entity based upon minimum and maximum square and cubic units of measure. Figure 9 illustrates the concept of spatial area constraints.

Rotational Attributes. Rotational attributes effect the geometric editing feature of rotation. The entity can be constrained to the rotational level and degree of rotation which can be imposed during the geometric editing operations. The level of rotation allows the selection and indication of allowable rotation in three-space, and includes activation and limitation of each of the three major axis. The minimum and maximum values indicate the allowable range of rotation from 0 to 360 degrees in right-handed space. In addition, the rotational increment constrains the incremental rotation about the selected axis. Thus, an entity can be restricted to a axis rotation at 45 degree increments (figure 10).

Mobility Characteristic: Mobility characteristic’s establish the degree of interaction the element will exhibit when acted upon by another element and provides the ability of rigid placement within three-dimensional space. A primary use of the mobility characteristic is to fix the location of an entity so that the interaction between other elements will not cause the entity to translate from its set position. The use of this characteristic enables the modeler to prevent interferencexes caused by entities occupying the same location in three-dimensional space.

Once an object is placed in the modeling environment, it may become necessary to fix, or freeze, it’s position, as in the placement of a wall, stair, column, or any construction component (figure 11.) A fixed entity therefore becomes an obstacle which, when interacted upon by other entities, will restrict, or limit there movement (figure 12.)
An entity with a mobility characteristic of free, however, is not confined or restricted to a particular location in three-space. Free mobility enables the entity to respond to the interaction of other elements in accordance with the expected behavioral patterns established through the specification of other constraints (figure 13). The mobility characteristic does not affect any geometric or topological editing feature directly. It does, however, affect the resultant activity from associative relationships and manipulation of other entities discussed later.

Used in combination with the spatial representation characteristic, the mobility characteristic is the second most important feature the interactive constraint-based modeler possesses. These two characteristics alone can model, through direct response in real time, the interaction of solid objects in three-dimensional space, and can provide the means to interactively identify interferences between entities.

Relational Level. Characteristics at the relational level are attributes the entity possesses which indicate how that entity is to interact with, or relate to other spatial entities. Three relational attributes, proximity relationships, containment relationships, and associative relationships, used separately, or in combination, constrain the manipulation of the entity within the solid modeling environment. The following is a discussion of each of the three relational characteristics.

Proximity Relationships. Proximity relationship's establish a zone around, for solid entities, and within, for spatial entities, the entity which provides and acts as a buffer, or clear zone between the entity and any other entity. This zone provides the minimum distance in which other entities may encroach upon. A buffer zone is created by establishing an offset from the bounding box based on the width, length, and height distance from the entity, (figure 14) and is maintained regardless of the geometric transformations applied upon the entity. Therefore, a specification of an offset along a particular edge, such as the width edge, is maintained throughout the solid modeling process.
Figure 14: An example of the proximity zone specification for an entity.

The use of a buffer zone, or offset, allows the modeller to represent non-physical entities such as distances between floors, setbacks, and rights-of-way, or can act as physical representations such as mortar between bricks, walls, and ceiling systems. A solid modeller can, therefore, support abstract representations of design elements and can provide the ability to use design knowledge in an early stage of the design process by maintaining appropriate tolerances between conceptual elements.

Figure 15: The interaction between entities with and without proximity offsets.

To illustrate the effects on the modeling process, each case, solids and spaces, are graphically represented with and without a proximity zone established. Figure 15 above illustrates how a proximity zone effects the relationship between solid entities. Figure 16 below illustrates how the proximity zone effects the relationship between solids and spaces. The use of the proximity zone allows the user to establish a setback, such as a zoning restriction, or offset, such as a ceiling to floor distance criteria, thus constraining the interactive modeling process.

Figure 16: The interaction between a space entity and two solid entities.

Containment Relationships. Containment establishes the relationship between dissimilar spatial representations. A hierarchical relationship can establish constraints which effect the extrusion, scale, rotation, and translation of entities, and will restrict such operations on subordinate entities to the limits of the bounds of the spatial entity. This has the effect of enclosing entities within spaces, such as the design space previously illustrated, and effectively constraining the interaction and manipulation of entities within that space. Figure 17 illustrates the translation of a solid bound to the limits of a space. Depending upon the spatial representation given to an object, the modeller provides several options to the user. These options must include no containment, solid contained by a specified volume, or a spatial entity which contains user specified solid entities. No containment allows the entity to be non restricted in any space. If containment is desired, the entity must belong to a spatial entity, and thus contained by that entity, or contain solid entities, depending on the spatial representation. It is important to note that in the event that a solid entity has no containment relationship between a spatial entity, the solid entity should have complete freedom to penetrate or exceed the bounds of the spatial entity, and thus is not be restricted or contained by the spatial entity.
Figure 17: An illustration of the containment constraint on the modeling process.

Associative Relationships. Associative relationships establish the response an entity will exhibit during the interaction with another entity. Unlike the mobility characteristic which affects the physical interaction between entities, the associative relationship effects the spatial relationship between interacting entities. Utilizing an offset from the proximity zones for each entity, (figure 18), the interaction between two or more entities will be affected by their associative relationship, and respond in one of three actions: none, attract, or repel. None indicates that there is no inherent association required for that entity. Attract indicates that the object will attract, or try to attach itself, to another object which approaches the tolerance offset. Repel indicates that the object will oppose, or repel another entity which approaches the tolerance offset.

Figure 18: The offset from the proximity zone which determines activation of the associative relationship.

To illustrate the interaction between entities influenced by the associative relationship, several examples of attract and repel have been provided. The illustrations provide a point of reference which indicates the modeling scene prior to a translation and shows the results based on the association and mobility of the entities in the scene. Since this is still view of an interactive dynamic process, outlined representations of the object have been provided to indicate their position prior to translation.

Figure 19: The interaction between an entity with free mobility on an attract association.

To illustrate the effect of restricting mobility, figure 20 views the translation of object A within the offset tolerance of object B which is fixed in three-space. As the user interactively translates object A, and approaches object B, the association will cause object A to snap, or move in one continuous motion, to the face of the fixed entity, object B. Object A now can only translate along the face of the fixed entity and cannot extend beyond the bounds of this constraint.

Figure 19 illustrates the view of the modeling scene where an entity, object A, will be translated within the offset tolerance of another entity, object B. Both entities have an attract association with one another. In addition, object B has a mobility characteristic set to free. As the user interactively translates object A and approaches object B, the association will cause object B to snap, or move in one continuous motion, to the face of the translating entity. As object A continues to translate, object B will remain attached.
In either case, attract with free mobility, or attract with fixed mobility, the interacting objects will attempt to attach themselves to one another. This characteristic is useful when considering the placement of solids within a space, such as kitchen cabinetry along a static wall. The wall can be set to fixed, and the association between it and the cabinets can be set to attract, thus ensuring that the cabinets are placed against a wall.

The modeling effects of the repel association follow in similar fashion to that of the attract association. The difference is in the reaction of the entities. Entities with a repel association will attempt to reject the attachment or encroachment upon the space surrounding the entity. They will behave as magnets with similar charges attempting to repel one another.

An illustration of the repel association with both free and fixed mobility is presented to illustrate the visualization of the effects on the modeling process.

Figure 21 illustrates the view of the modeling scene where an entity, object A, will be translated within the offset tolerance of another entity, object B. Both entities have an association with one another of repel. In addition, object B has a mobility characteristic set to free. As the user interactively translates object A and approaches object B, the association will cause object B to react to the advance of object A by moving out of its path of travel. As object A continues to translate, within the tolerance zone of object B, object B will continuously move away from object A. This association will continue as long as object A encroaches upon this tolerance zone. A movement of object A in a direction away from object B does not effect object B.

To illustrate the effect of restricting the mobility of object B, figure 22 views the translation of object A within the offset tolerance of object B which is fixed in three-space. As the user interactively translates object A and approaches object B, the association will cause object A to be rejected, or repelled away from object B. As object A continues to translate, within the tolerance zone of object B, object B will continuously reject object A and force it back from its tolerance zone. This association will continue as long as object A encroaches upon this tolerance zone.
The use of the association relationship provides a method of realistically modeling the behavior of interacting entities. When used in combination with the other constraint characteristics and relationships, the modeling environment becomes responsive to the physical and relational quality of solid and space entities. In an interactive modeling environment, this translates to a dynamic and realistic modeling of simulated real world entities which will behave in accordance with anticipated and results.

Identification Level. The final level of constraint modeling is that of identification. The identification level provides the modeler with the ability to uniquely identify a set of constraints which have been established for a given entity. This description is provided for by the use of a type definition attribute. The following is a brief discussion of the role of the type definition in a constraint-based solid modeler.

Type Definition. This is a unique identifiable descriptive attribute associated to the set of constraints representing an entity. The specification of a type definition to a specific set of constraints provides the capability to store and retrieve the entire set of constraints with a single identifying macro. By allowing the set of constraints to be identified as a single type definition, the set of constraints can be created and stored once, and then retrieved and assigned to entities with a single reference indicating that type (figure 23.) This ability to assign a complete set of constraints with a single reference allows quick and efficient use of the design knowledge about entity types without specific knowledge about all the particular attributes required to represent that type.

Figure 23: An illustration of the type definitions.

Overview of the Prototypical Application.

The prototypical constraint-based solid modeler, C-Mod, was written and developed as an extension to the MacMod844 shell provided by the Department of Architecture, The Ohio State University, and currently runs on the Apple Macintosh platform. The basic user interaction with C-Mod is purely a graphic one. A user has the ability to generate and manipulate solid and spatial elements graphically, with the use of the mouse. The figures presented in this section were created solely with the use of this prototypical modeler.

The interactive simulation of solid and spatial entities, as previously mentioned, is critical to the success of a constraint-based solid modeler. To illustrate the orchestration and interaction between both spatial and solid entities, a brief example of the prototypical application C-Mod is presented. The illustrations represent the process of establishing a hypothetical design space in which solid entities representing spatial forms can be generated and manipulated within the constraints of the design space as well as the constraints imposed on the entities themselves.

Figure 24: Modeling the design space.

To begin the process of interactive constraint-based solid modeling, a design space is created. This allows a site to be modeled, with appropriate setbacks, height restrictions, and shadow/sunlight restrictions. Figure 24 illustrates the generation of a spatial entity which conceptually represents a site with restrictions or constraints. Although not visually represented, the design space has a proximity...
offset, conceptually representing a side yard setback, in the width direction, that is along the y axis in this example.

setback, geometric editing of the solid entity will not penetrate either this proximity zone or the established limits of the design space (figure 26.)

Figure 25: The generation of a solid entity within the design space.

Figure 27: Interactive modeling of the solid entity within and constrained to the design space.

Further geometric editing of the design space allows the entity to be modeled to allow for the shadow and sunlight restrictions which may be imposed upon the site. The results of this interactive topological and geometrical editing is a solid model representation of the site constraints, and thus the allowable design space. Figure 25 illustrates the completed design space and the introduction of a conceptual solid entity within that design space.

Additional geometric editing of the solid entity allows the form to be molded to the restrictions of the design space. Since CMod constrains this modeling process interactively, the user can perform editing functions up to the bounds of any established constraint. Figure 27 further illustrate this interactive geometric editing within the design space while figure 28 introduces an additional solid entity within the design environment.

Figure 26: Translation of the solid entity within the design space, and restricted to the constraints of the design space.

Figure 28: Creation of a second entity within the design space.

The interactive modeling process continues with the geometric editing of entity within the design space. Since the design space has a proximity zone established representing the side yard...

The interactive modeling process continues with the geometric editing of entity within the design space. Since the design space has a proximity zone established representing the side yard...

Once the entity has been generated, relationship criteria can be established between the entities. Figure 29 illustrates the vertical translation of the entity to the height restriction of the design space, followed by the specification of attract

204
association between it and the first solid entity created. By setting the mobility of the first entity to fixed, any transformation of the second

![Figure 29: Vertical translation of the solid entity constrained to height limitations.](image)

entity within an established tolerance from the first will cause it to snap up to the face and therefore connect the two entities (figure 30.) Providing this relationship enables the entities to act as a cohesive group, thus any transformation on either entity will effect both entities.

Continuing the process of generation and manipulation in a similar manner, additional conceptual entities can be introduced and modeled within the design space. Figure 31 illustrates the placements and editing of additional conceptual entities completing the conceptual design within the design space.

![Figure 30: Association attribute of attraction established between the two solid entities.](image)

![Figure 31: Completion of the conceptual design within the design space.](image)

The application presented in this section is a small representation of the prototypical application C-Mod. C-Mod illustrates that constraint-based solid modeling is a viable means of representing physical and relational design knowledge in an interactive solid modeling environment. The application demonstrates that the modeling of the physical interaction between solid entities, spaces, and constraints, can be effectively applied to the early stages of the design process, including schematic design. The demonstration provides a means of exploration of the design envelope that is consistent with expected behavioral patterns of the entities generated and modeled. A designer is therefore free to create solid and spatial entities and model not only the specific geometry or topology, but the interaction between entities in an intuitive and effective manner. The result is that the computer is capable of enforcing the lower level constraints, such as setbacks, clearances and interferences, and adjusting, or reacting to the user immediately.

**Extensions.**

During the course of its presentation, this modeler focused on a narrow spectrum of design knowledge which can be represented in the interactive three-dimensional modeling process. Its purpose was to provide a functional prototype which could be utilized to illustrate the practicality of implementation. This does not suggest that this is all that can be done on the subject. On the contrary, their is much room for extension and expansion of thought both in
terms of the prototypical application C-Mod, and the fundamental theory of constraint-based solid modeling. This section seeks to explore the frontiers of such direction by discussing several possible extensions to the presented constraint-based modeling operations and theory.

Extensions to Constraint-Based Operations. The entity characteristic operations presented as the basic set of constraints specifically apply to the physical description of the entity. These physical descriptions included representation, dimensional criteria, area and volumetric requirements, and mobility. This avenue of operations was selected as a fundamental set of operations which would illustrate the basic interactive nature of constraints upon the modeling process. In the event that the modeler was to assume a more realistic role, additional operations would be required. Operations such as mass, surface texture, rigidity, and selectable mobility. The following is a brief discussion of each of these possibilities.

The entity characteristic of mass would allow the modeler to introduce the natural laws of physics to the interactive process. The mass of an entity would determine potential energy which would be transferred during an interaction. Additionally, it would effect the momentum the entity gains and sustains during the interactive modeling process. The use of this characteristic would include modeling momentum of a structural member or component, and visualizing its reaction in a three-dimensional environment.

Surface texture would again allow the modeler to react to natural laws of physics. The surface qualities of an entity determine the friction generated when two objects are transformed along their faces, and would effect the relational characteristics of the modeling process. The use of this characteristic would include realistically modeling the interaction between elements of differing surface qualities such as a table leg being transformed along a carpeted floor.

Another physical characteristic which would benefit the constraint-based modeling approach is the use of rigidity. Rigidity would allow the modeling of materials with differing physical compositions. A wood column could be modeled
with its base fixed in three-space, and interactively forced to bend until failure. The use of this characteristic is evident in the interactive modeling of structural components during the design process.

A final extension to the entity characteristic operations is that of specifiable mobility. Currently the mobility of an entity is fixed as an object. To realistically model the interaction with other entities, and utilizing some of the additional features listed above, the modeler must be able to fix any topological level of the entity in three-space. With this extension it would be possible to model the column example presented above by specifying that the bottom face of the column was to be fixed in three-space. The interactive nature of a constraint-based modeler must allow, and provide for, relational characteristics between entities. This discussion proposed a few of the basic relationship which exist between entities such as containment, association, and proximity. In addition to these several other relationships could be modeled to enhance the realistic interactions between elements. Among those are ownership, grouping, and topological associations. Each of these additional operations are discussed below.

Ownership differs from containment in that an entity can own, or belong to another entity outside the bounds of a spatial representation. This relationship would establish a bond between entities which would permit it to perform as a contiguous collection of entities while still retaining the individual characteristics. Thus a roof system composed of structural beams, joists, decking, and insulation could behave as a contiguous unit when being transformed in three-space while behaving in accordance with the requirements of each individual component.

Grouping would allow a collection of entities to be grouped together into a cohesive unit. This differs from ownership in that the individual characteristic are not retained. The collection of entities behaves as a single entity with its own set of constraints to be applied to the modeling process. In this relationship the components of a chair, arms, legs, seat, and back, could be grouped together and given its own set of constraints, or design knowledge, which identifies it as a chair, and therefore controls its behavior as a single entity.

The final additional relational operation is an extension to the existing association operation. By allowing the specification of associations between topological levels, an interactive modeler could associate, attract or repel, differing parts of the entity. Certain faces of an entity may be required to retain an attract association, such as the ends of a wall, while others may be required to retain a repel association, such as an opening for a door. By allowing this diversity within the entity itself, realistic situations can be modeled.

To facilitate the realistic interaction between entities in an interactive modeler, the next logical extension would be to include the physical laws of nature to the modeling process. This is particularly important when determining the reactions from colliding entities. The resultant direction vector may not be in parallel with, and containing the same force of, a the moving entity. It is therefore important that when implementing any of the additional operations suggested, that the natural laws of physics be considered as a primary extension to the application.

Conclusions.

The potential for an interactive constraint-based solid modeling tool is clearly evident when considering the realistic interaction between entities, both in the physical and metaphysical sense. To bring solid modeling into the early stages of design, a modeler must be capable of supporting these interactive qualities to realistically model the environment. As the pretense for this discussion, interactive modeling within the guidelines established by the local zoning ordinances, building codes, and construction practices, provides a powerful foundation for future development of interactive solid modelers.

The future direction of constraint-based solid modeling stems from the ability to represent and model design knowledge to aid in the conceptual design process. This design knowledge, whether user specified, or read from a database containing the specific zoning ordinances, allows the modeler to perform as a powerful design tool.
in which the designer can call on and utilize that information in an interactive modeling platform. In this role, the constraint-based solid modeler provides an important tool which the designer can use to enhance the conceptual and schematic design process.

In 1973, Geoffrey Broadbent, discussed the use of computers to aid in the exploration and development of the design space interactively. By extending this early discussion to include the generative and manipulative capabilities of a solid modeler, and modifying the theories of the modeling process to include interactive constraint satisfaction, design knowledge, and therefore constraints, can intuitively and effectively enable a designer to bring the computer closer to the early stages of the design process. The ability to provide limited realism to both the solid and spatial entities, as well as their behavioral patterns, provides an additional benefit of interacting with the modeler in a manner more consistent with natural objects. This discussion has illustrated that the application of design knowledge in CMod, although limited in scope, can provide an interactive modeling environment which allows design knowledge to constraint or limit the generation and manipulation of solid and spatial entities, and that the use of such a design tool has a strong and desirable link to the conceptual stages of the design process.

Acknowledgements.

I would like to extend my appreciation to The Ohio State University Department of Architecture, Computer Aided Architectural Design program, for the use of their MacMod/44 educational shell. I wish to acknowledge Dr. Chris I. Yessios for his contributions and support to the design and development of CMod.

References


