IMPLEMENTATION OF SOLID MODELING
IN HIGH HIERARCHY ARCHITECTURAL LANGUAGE (HHAL)

Edna Shaviv, Professor of Architecture
Ofra Gavish and Uri Amir, Students of Architecture
Faculty of Architecture & Town Planning
Technion, Israel Institute of Technology, Haifa, 32000, Israel.

KEYWORDS
High Hierarchy Architectural Language (HHAL), solid modeling, shape grammar, formal language, design systems, basilican structures.

ABSTRACT
This work illustrates the use of solid modeling for the representation and analysis of complex architectural structures and details. While the approach is a general one it is demonstrated and implemented on the visual representation and evaluation of Mediaeval Age basilican structures. Basilican structures have the advantage that rules of composition for defining their form are well formulated. These rules are imposed on the solid modeller as high hierarchy architectural language (HHAL). It is within this language that the basilican structures are described and analyzed.

INTRODUCTION
Simplified and highly simplified methods for the representation of buildings by means of computers, so that they can be visually appreciated, are frequently in use today. These simplified methods include for example representation by 2.5D, or 3D representation with the limitation of applying only rectangular parallelepiped. The advantage of such comprising approaches is the very fast algorithms that are
required to eliminate hidden lines and their overall ease of use. Such methods can be very important for the fast visualization of any project during the schematic stage, as was described by Van Norman (1986). However such an approach is restricted to the use of a very unsophisticated architectural vocabulary and is therefore limited to routine and dull architecture, where for example, curved surfaces and other complex structures are not a part of the architectural glossary.

The use of solid modeling which contains curved primitives allows the representation of arbitrary shapes and the use of very rich forms as was discussed by Mitchell (1986a). Mitchell describes the vocabulary of primitive solids as the 'building block' out of which compositions are constructed. We found that the number of the primitive solids required to create practically any architectural form is quite small. However, we would like to extend the use of solid modeling beyond the simple application of primitive solids.

We demonstrate the above thoughts by a software called CHURCH that was developed in the Technion. CHURCH creates automatically the primitive solids and combined-objects which are elements of Mediaeval basilican structures and which can be read by a commercial solid modeler called ICEM Solid Modeler (ISM) developed by Control Data for mainframe and NUS workstations (see ICEM Solid Modeling reference manual, 1987). The main emphasis in this work is on the formulation of architectural formal language and the technical aspects of solid modeling applications in daily architectural work, and not on the research of a certain class of structures. The specific results for the basilican structures Investigation will be discussed elsewhere.

Our example of basilican structures has the advantage that its 'building blocks' and its rules of composition for synthesizing the form of the building (space, structure, and different elements) are well defined. These rules can be imposed on any solid modeler as a high-level high hierarchy architectural language (HHAL), through which the basilican structures can be visualized, synthesized, and analyzed. These rules of composition (relations, proportions, dimensions, etc.) form the syntax of the HHAL. By using these rules even when incomplete information about a given church is known, CHURCH has sufficient artificial intelligent to produce the complete structure and its details from partial data.

When compared with the simplified and easy to use methods for buildings visualization, the solid modeling methods are usually not applied effortlessly.
Nevertheless, CHURCH allows a particularly easy introduction of the basic elements of Mediaeval basilican structures through the use of a graphical user interface that was developed by implementing the HHAL approach. Thus, in a simple way and in a little time, a thorough analysis of the basilican space can be visualized with colors, shades and shadows. The user is free to describe the eye position, the aim point, and the direction of the light, so that the space can be envisioned and admired in a very realistic way and from different points of view.

Software like CHURCH, which is based on the HHAL approach, can be applied to other classes of well defined problems and are very easy to develop. This is because the HHAL is based on the implementation of an existing commercial solid modeling which need not be written from scratch. Only the high level architectural language that includes the rules of composition and the user interface should be elaborated.

HIGH HIERARCHY ARCHITECT'URAL LANGUAGE (HHAL)

The basilican structure has a well defined finite set of 'building blocks', finite set of allowable relations between the 'building blocks', and rules for composition. According to Mitchell "The concepts of a finite architectural vocabulary, a finite set of allowable types of relations between instances of vocabulary elements in compositions, and rules of composition expressed as productions, can be brought together within a unified framework by defining formal grammars which specify languages of architectural form." (Mitchell,1986b). We suggest in this work to define the language of the building form as a high hierarchy architectural language (HHAL). We shall demonstrate the implementation through the analysis, synthesis and the representation of the form and space of the basilican structures.

The language of the building form can be decomposed hierarchically to its components. First, as a higher level, are the vocabulary elements that define the general geometry and the schematic description of the building enclosed voids (the spaces). These enclosed voids are called by Guadet 'elements of composition' (Guadet, 1894). Second, are the vocabulary elements that describe the building structure, envelope and other standard elements as defined by the styles of the structures. Guadet called these the 'elements of construction'. Last, as a lower level, are the particular non standard elements. These last elements are not part of the vocabulary elements of the HHAL, but can be any primitive or combined-object of the solid modelling that is being used. In this way for the same general description
of the building geometry that defines the spaces, different standard and particular *vocabulary elements* can be implemented. Thus creating any building space and structure within the given style that was defined in the language or with arbitrary details that are exceptional to it, if the designer wish to add such elements.

**a. The 'Elements of Composition':**

*The General Schematic Description of the Building Geometry and its Enclosed Voids.*

In the example of the visual representation of Mediaeval basilican structures, the general schematic description of the building geometry and its enclosed voids may be driven from its definition:

"Basilica.- A church divided into a nave and two or more aisles, the former higher and wider than the latter, lit by the windows of a clerestory, and with or without a gallery. ... By the c4 the Christian basilica had acquired its essential characteristics: oblong plan; longitudinal axis; a timber roof, either open or concealed by a flat ceiling; and a termination, either rectangular or in the form of an apse." (Fleming, Honour & Pevsner, 1981).

According to the above definition the basilican space is characterized by having the nave and the aisles. The basilica essential characteristics is an oblong plan, longitudinal axis and a timber roof. The dimensions and the number of bays can change from one basilica to the other. Three Mediaeval basilican structure styles are defined in the present; the French Romanesque, the German Romanesque and the Gothic. The general schematic description of these styles can be driven from the definition:

"Romanesque architecture - 'Restyle current until the advent of Gothic... tunnel vaults in France; ... groin vaults in Germany; ... plans (mainly in Germany) with square bays in nave, transepts, and chancel, and square bays in the aisles one quarter the area . And in Gothic style: "The architecture of the pointed arch, the rib vault, the flying buttress, the walls reduced to a minimum by spacious arcades, by gallery or triforium, and by spacious clerestory windows. ..." (Fleming, Honour & Pevsner, 1981).

According to the above definition, the French Romanesque is characterized by having tunnel vaults in the nave, and groin vaults in the aisles with the same bay length as in the nave. (See Figures 1). The German Romanesque has square bays with groin vaults in both nave and aisles, but the bays in the aisles are one quarter the area of the nave bays. (See Figures 2). The Gothic has also groin vaults in both nave and aisles, however while in the German Romanesque the nave bay is twice as long as the aisles, in the Gothic, the bay length is the same (see Figure 3).
Fig. 1 The French Romanesque style.

Fig. 2 The German Romanesque style.

Fig. 3 The Gothic style.
The distinct styles and general schematic description of the building geometry and its enclosed voids constitute the higher level *vocabulary elements* of the *language*. These are the *vocabulary 'elements of composition'* of the desired style.

The HHAL provides first a dimensionless relational scheme that represents the rules of composition of the family of structures of the specific Mediaeval basilican style. This scheme is presented as a graphical user interface (see Figure 4 for an example). At this moment the HHAL asks for particular data required to complete the details of the presentation. In the case of French Romanesque style only the following data is required: the number of bays and its length and the height of the nave and aisles. The user is allowed to use the default dimensions that fits a certain French Romanesque church that was prepared for him as a demonstration case. The product of this stage is the schematic abstraction of the given particular church. However, since at this stage we have not yet defined the 3D vocabulary *'elements of construction'*; we are unable to visualize the building. We can only realize the general geometrical scheme of the building (see Figure 5).

![Fig. 4 The representation of dimensionless relational scheme of French Romanesque basilica and the introduction of selected dimensions.](image)

![Fig. 5 The realization of the general geometrical scheme of a particular French Romanesque basilica with its dimensions.](image)
b. The 'Elements of Construction':

In the particular case of the representation of the basilican structures, we shall farther decompose hierarchically the 'element of construction.' The stylistic structural elements are placed in the higher level. In a lower level the nonstructural stylistic elements like the elements describing the building envelope or other standard elements that are defined within the relevant styles are found. It seems to us that most buildings can follow this order of representation.

b-I. The Stylistic Structural Elements:

The first group of the 'element of construction.' contains the structural elements. The structural elements turn the stylistic schematic abstraction of the unapparent building as (cf Figure 5) into a real 3D object. The user should define the style of the pillar its base and capital (see Figure 6). At this stage the building can be visualized by its physical structural elements for the first time (See Figure 7). The last Figure was obtained by a substitution of the chosen stylistic pillar, base and capital with the grid of the schematic abstraction that was shown in Figure 5. This method allows us to substitute any stylistic structural element previously defined as part of the vocabulary element of the the HHAL. We can for example synthesize a Gothic schematic space, with German Romanesque pillars, if we wish to evaluate such structures. Default values are set to the structural elements of the particular chosen basilica style if no specific value is given. The inexpert user of the CHURCH software, needs not to know the details of these structural elements.

Fig. 6 The dimensionless representation of possible structural elements and determining their style and dimensions.
b - II. The Nonstructural Stylistic Standard Elements:

The second group of the 'elements of construction' contains the nonstructural elements which specify for example the building envelope like walls, roof, windows etc. At this stage additional nonstructural elements can be added in any order and seen in 3D immediately. Once the building has realized, new physical elements, can be appended to the composition and presented in the 3D model. Figure 8 shows the selection of a particular detailing of the church elevation, which includes; windows, triforium and clearstory. In Figure 9 the chosen detailing is seen along with the structural elements presented in Figure 7. Figure 10 shows a different perspective views of the church with and without the detailing.
c. The Particular Nonstandard Elements:

The third group of the vocabulary elements are the particular nonstandard elements. These lower level elements are not part of the vocabulary elements of the HHAL. In the example of the basilican structures, the fine ornaments of each window can be particular elements. No rules can be found for the nonstandard elements as these are arbitrary forms. Hence, there is no reason to predefine these elements as a stylistic vocabulary element, which constitute part of the HHAL. We can always represent these nonstandard elements by a primitive, or by a combined-object of the solid modelling. These terms will be explained in the next section.
d. The Hierarchy in the Representation of the Building Form

The present method can be generalized to any class of structures. A class of structures defined by a known style can be represented hierarchically according to the language of its building form and with different amount of realization and detailing. The diagram presented in Figure 11 shows the hierarchy in the architectural language and in the representation of the building.

Fig. 11 The hierarchy in the architectural language and in the building representation.
e. The Shape Grammar that Generate the Language of the Building Form

"The principles by which vocabulary elements can be put together at the syntactic level to form artifacts constitute a grammar" (Coyne and Gero, 1986). When rules of shape grammar are executed, the building form is created. Possibly, the best known shape grammar is the one by Stiny and Mitchell for the generation of Palladian villa floor plans (Stiny and Mitchell, 1978). Following Stiny and Mitchell, we find some other development of shape grammar, like the work by Koning and Eizenberg (1981), who demonstrated that a grammar for Frank Lloyd Wright's prairie houses could be written in less than one hundred production rules using the Stiny and Mitchell's shape grammar approach.

The analysis of the vocabulary elements of the basilican structures and the way they are put together, leads to the identification of generic rules that explain how these elements are put together to form the structure and the spaces. A column, for example, consists of a base, a pillar and a capital, each of which can be of a different style (see Figure 12). Different styles of basic vocabulary elements yield different columns. All elements are treated in the same way to constitute the whole column.

The many generic rules according to which the basilican structures were built, were put into the code CHURCH. Thus, if the user knows only few dimensions of a given basilican structure, CHURCH finds the missing ones according to the rules and the known (partial) data. CHURCH knows the composition rules of the three basilican styles mentioned above and can lead the user to choose the coherent stylistic vocabulary elements. However, CHURCH leaves the freedom to the user to try any combination between vocabulary elements that could be composed together even if historically such buildings do not exist. Thus, one can appreciate how such building would look like.
Fig. 12 Four alternate columns that can be obtained from two different capitals and two different pillars.
SOLID MODELLING FOR ARCHITECTURAL VISUALIZATION

a. The Elements of Solid Modelling.

a - 1. Primitives:

The fundamental elements of any solid modelling are the primitive solids, which are generalized polyhedra, i.e. a cube, a slab etc. The solid fundamental elements can have curved faces as well as planar ones, like a sphere, a terrain (that presents a topography), or any arbitrary 3D curved surfaces.

The use of solid modeling that contains curved faces primitives allows the representation of the very rich spaces like a Gothic one. We demonstrate this capability through the use of ICEM Solid Modeler (ISM). The ISM software contains fourteen basic primitives (see Figure 13), through which any capricious shape can be created. However, BOX, SLB, CYL, SPH, REV, and TER are the most commonly used primitives in architecture. (see the definitions in Figure 13).

BOX - Parallelpiped
CON - Truncated right circular cone
CYL - Right circular cylinder
ELC - Truncated elliptic cone
ELL - Ellipsoid
GEN - General solid
POL - Polyhedron with four, five, or six faces
REV - Revolution solid
SLB - Slab
SPH - Sphere
STO - Solid torus
TER - Terrain solid
TOR - Torus
WEG - Wedge

Fig. 13 The set of the fourteen basic primitives that are defined within ISM, and the most commonly used primitives in architecture.
a-II. Combined-objects.

The ISM software allows the definition of combined-objects, which are complex forms created by Boolean operations on the basic primitives. The ISM software contains three Boolean operations, which are; union, difference, and intersection operators (In Figure 14 one can see for example four different combined-objects that can be obtained from the primitives BOX and SPH). Elements of the language of the basilican structures were built by means of the Boolean operations (See for example Figure 12 where the column is the union of a slab and a revolution solid). Every element gets a name and can be placed in space to realize the entire architectural composition.

Fig. 14 The set of the three Boolean operations that are defined within ISM
a - III. Transformation of Primitives and Combined-objects.

All primitives and combined-objects can be manipulated in space through the following transformation operations which are defined within ISM; MOVE, TURN, MIRROR, SCALE, PURGE. In these way the designer may place and orient the final object at will. He can also preserve the topology of the object and change its dimension or proportion.

The frequent way to use the primitives and combined-object objects for the creation of the basilican spaces is to create a library containing all the elements of the formal language of the basilican structures. These vocabulary elements may be located in space by the expert user of the ISM. This way of operation reminds a child playing with LEGO cubes and building whatever structure he wishes. However, we face two problems. First, even an expert user of ISM (or any other solid modeling software) requires a long time to master all the transformation command needed to place the vocabulary elements correctly in space. Second, one should remember all the rules which dictate the form of the style to achieve the desired space to analyze visually. Here a different approach is taken namely: to impose the rules of the basilican architecture on the combined-objects of the solid modeler. The rules are treated as shape grammar rules which create the high hierarchy architectural language (HHAL).

b. Implementation of Solid Modeling in High Hierarchy Architectural Language

Two extreme user interface approaches for introducing the building data can be found. The first one is the command language. In this approach the user types commands via the keyboard. The commands may describe the primitive, the operation, the combined-object, or the transformation command. For example one can define a building called HOTEL by using the following commands:

A=BOX(0,50,6,20)
B=BOX((0,6,0),12,60,20)
HOTEL=A+B
MOVE, HOTEL (60,0,80)
VIEW, HOTEL
In this short example we find the various elements of the solid modeling vocabulary. A and B are one of the fourteen basic primitives defined in the ISM. A is a primitive called box, defined at the origin (marked by 0) with length 50 units along the X axis, height of 6 units along the Y axis, and width of 20 units along the Z axis. B is another box defined on top of the box A with the geometrical data: origin=(0,6,0) and dimensions 12,60,20 along X, Y, and Z axis respectively. ISM allows at this stage to synthesize HOTEL as a combined-object defined by the union of the boxes A and B. We can locate the new entity HOTEL anywhere in space by using the command MOVE for example. The command VIEW permits us to see our design.

An alternative way to introduce the geometrical data of HOTEL is through a menu driven interface. The menu driven interface is easier to use, as it shows all the commands that can be invoked at a particular moment, so that one does not have to memorize the available commands. However, as these menu driven interfaces are hierarchically organized, they are often a source of irritation to the sophisticated users (Van Norman, 1986). Moreover, each command should be executed separately and an in interactive mode, which can take a long time. The command language user interface, on the other hand, can be more compact, as these commands may be executed together as macros either interactively or in a batch mode of operation. The ISM permits both ways of operations.

The existence of command macros lets us write the high hierarchy architectural language (HHAL) which creates automatically the command macros whenever a certain conditions are satisfied. The code CHURCH is an example of such HHAL. Thus, when the user defined 'Gothic' as the style of the basilican structure, the knowledge base in the computer software CHURCH generates all the commands required to present such a style. The user is free to define the number of bays and their lengths, the width and height of the nave and aisles, and the height of the side and central vaults. (See Figure 15). Moreover, as the HHAL contains the knowledge of the formal rules of the building under consideration, it has the ability to complete missing data from partially given one, so that the complete building can be realized.

The code CHURCH is used as an intermediate interface between the user and the solid modeler. On one hand, CHURCH creates the command language interface required by the solid modeler to present a basilica structure. On the other hand, it includes a graphical user interface (as was seen in Figures 4, 6, 8, and 15), that allows an easy introduction of the vocabulary elements and rules of compositions of the Mediaeval basilican structures. This approach has two merits. First, the user is not
required to memorize or write all the rules of compositions. Second, the HHAL creates a graphical user interface which is very easy to use.

Enter
Schematic Description:
(in meters)

<table>
<thead>
<tr>
<th>Number of Modules</th>
<th>?</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Size (MS)</td>
<td>?</td>
<td>10</td>
</tr>
<tr>
<td>Side Aisles (SA)</td>
<td>?</td>
<td>6.5</td>
</tr>
<tr>
<td>Center Nave (CN)</td>
<td>?</td>
<td>17</td>
</tr>
<tr>
<td>Aisles Height (AH)</td>
<td>?</td>
<td>16</td>
</tr>
<tr>
<td>Nave Height (NB)</td>
<td>?</td>
<td>27</td>
</tr>
<tr>
<td>Side Vault (SV)</td>
<td>?</td>
<td>4</td>
</tr>
<tr>
<td>Center Vault (CV)</td>
<td>?</td>
<td>7</td>
</tr>
</tbody>
</table>

IS INPUT O.K (Y/N) ? Y

Elements of Envelope:
(in meters)

<table>
<thead>
<tr>
<th>Side Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Parts</td>
</tr>
<tr>
<td>Width (WW)</td>
</tr>
<tr>
<td>Height (WH)</td>
</tr>
<tr>
<td>Edge (WE)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Triforium</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Parts</td>
</tr>
<tr>
<td>Height (TH)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clerestory</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Parts</td>
</tr>
<tr>
<td>Width (CW)</td>
</tr>
<tr>
<td>Height (CE)</td>
</tr>
<tr>
<td>Edge (CE)</td>
</tr>
</tbody>
</table>

IS INPUT O.K (Y/N) ? Y

Fig.14 A Graphical user interfaces that allows an easy introduction of the vocabulary elements and rules of compositions of Gothic basilican structures. This figure differs from those shown in Figure 4 and 8 for the French Romanesque. The introduction of the structural elements is according to Figure 6.
CHURCH includes an interpretation algorithm for perspective rendering from predetermined views and under specified lighting conditions that were found to be of interest. This is achieved by using command macros that were prepared for the inexpert user. On the other hand, the expert user who is familiar with the ISM is free to describe the eye position, the aim point, and the direction of the light so that the space can be envisioned and admired in a very realistic way from different points of view. ISM creates the perspective rendering by means of a 'ray-tracing' technique. In this way the HHAL creates in a very simple way and in little time a thorough analysis of the basilican space to be visualized with colors, shades and shadows (see Figures 16 and 17).

Fig.16 A perspective rendering. The views, and lighting; conditions were predetermined according to previous experience that found these particular views most interesting.
Fig. 17 A perspective rendering. (remark: at this stage the groin vaults are presented by the arches only).
SUMMARY AND CONCLUSIONS

In this work we presented an example of a high-level design system, which can be characterized as a high hierarchy architectural language (HHAL). A system can be characterized, according to Winograd, as a language, if it consists of a set of elements, a set of operations, a vocabulary and grammar (Winograd, 1983). The high-level, high hierarchy architectural language design system, that we have introduced, is a very efficient system for the representation, analysis, and synthesis of structures within a particular class of design problems.

According to Mitchell: "A high-level design system can be built upon a foundation of the type of basic design system, and is more specialized and powerful than a basic design system. It supports work within a particular architectural style." (Mitchell, 1986b). The high hierarchy architectural language 'CHURCH' is such a high-level design system, while the solid modeler 'ISM' is the basic design system. CHURCH is a powerful system, but supports work within a particular architectural style only. ISM, on the other hand, is a general design system. The combination of a high hierarchy architectural language and a general design system is the source of strength of this method.

The approach can be expanded to other types of Architecture, or architectural problems, whenever an artificial intelligent methodology can be applied. For example, the method is not limited to standard elements. The fundamental prerequisite is a vocabulary of elements and rules of composition.

We would like to summarize some of the conclusion that can be drawn from this work:

1. It is important to use a versatile solid modeling design system, if complex architectural spaces are to be visualized, analyzed, or synthesized.

2. It is possible to develop high-level design systems, which are characterized as a high hierarchy architectural language (HHAL), for specific design problems like housing, schools, or in our particular case, the basilica structures. These high-level design systems allow a very efficient work within a particular class of design problems, and/or design styles.
3. The high-level design system can include graphical interface, that presents the vocabulary elements which are defined within the HHAL. Thus, the representation, analysis, or synthesis of structures within a particular class of design problems is very simple.

4. Much of the strength and appeal of the HHAL is the capability to manipulate in an easy way a complex structure and yield attractive results. This is achieved by the implementation and the integration of solid modeling in HHAL.

ACKNOWLEDGEMENT

We would like to acknowledge Architect Liora Shahal for the helpful discussions.

REFERENCES


Order a complete set of eCAADe Proceedings (1983 - 2000) on CD-Rom!

Further information: http://www.ecaade.org