Representing Spatial Abstractions of Constructed Facilities

M. K. Zamanian, SJ. Fenves, E.L. Gursoz

EDRC 12-40-91

Representing Spatial Abstractions of Constructed Facilities

M. Kiumarse Zamanian

Steven J. Fenves

E. Levent Gursoz

Engineering Design Research Center

Carnegie Mellon University

Pittsburgh, PA 15213-3890 USA

EDRC 12-40-91

This paper is to be published in Summer 1991 edition of ***Building and Environment***

This work has been supported by the Engineering Design Research Center, a NSF Engineering Research Center at CMU

Abstract

This work presents a general technique to represent and identify various abstractions of spatial information in a uniform and consistent manner regardless of the dimensionality of the geometric elements used to model these abstractions. The principal motivation of this study is to provide a general spatial representation scheme that can be used by all disciplines involved throughout the lifecycle of a constructed facility to specify and reason about the facility information. However, this scheme is sufficiently general to be used in other domains in which spatial decomposition and configuration of objects are of great interest. In our implementation, the spatial information is modeled using a vertex-based, non-manifold geometric modeling system. The non-spatial attributes are handled by a separate information management system, such as a relational database, and linked to their corresponding spatial attributes in the geometric modeler.

1 Introduction

Constructed facilities are complex entities. They contain many components, each of which exhibits different behaviors for different disciplines. For example, a wall may be viewed as a partitioning element by the architect, as a load bearing component by the structural engineer, and as a routing agent by the mechanical or electrical designer. Each component possesses several attributes that determine its functional behavior of interest to different disciplines. These attributes arc of two general types: spatial and non-spatial. The spatial attributes pertain to the component's geometry and topology (e.g, a room is a cuboid of certain dimensions adjacent to a set of walls, floors, and ceilings), while the non-spatial attributes arc all the other properties of the component (e.g., the thermal conductivity of a wall, the color of a face of the wall, etc). Furthermore, the different disciplines may use different abstractions, aggregations, and spatial subdivisions of the components of interest.

Formal representation and management of spatial and non-spatial attributes of facility components are important, and often difficult, tasks for Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) systems. While the majority of the non-spatial attributes can be represented using relational databases or other traditional data modeling techniques, other, more specialized representation schemes are needed for dealing with the spatial abstractions. Presently, many CAD/CAE systems provide a geometric modeling system along with a database management system (DBMS) to respectively deal with the spatial and the non-spatial attributes of a facility. Although some of these geometric modeling systems offer a variety of representation schemes, such as wire-frame, surface, or solid modeling, these schemes are generally disjoint and have entirely different internal data structures and algorithms. This limitation prevents users from modeling and reasoning about various spatial abstractions of a facility in a single, uniform framework. Some existing CAD/CAE systems manage the spatial and non-spatial attributes of the facility components in the same DBMS by ad-hoc implementations using numeric or symbolic attribute-value pairs to represent the names and limited spatial and non-spatial relationships of those components. As a result, complex, but important, information queries that involve spatial reasoning as well as database interaction become extremely difficult if not impossible.

This paper presents a portion of a larger effort to provide a general framework for modeling and reasoning about the components of a constructed facility at any desired level of abstraction, and communicating the information across disciplines at any one stage in the lifecycle of the facility, as well as across stages, e.g., design, construction, and operation. Our research is motivated by an objective similar to that of STEP [1], which intends to establish an international protocol for the exchange of CAD data.

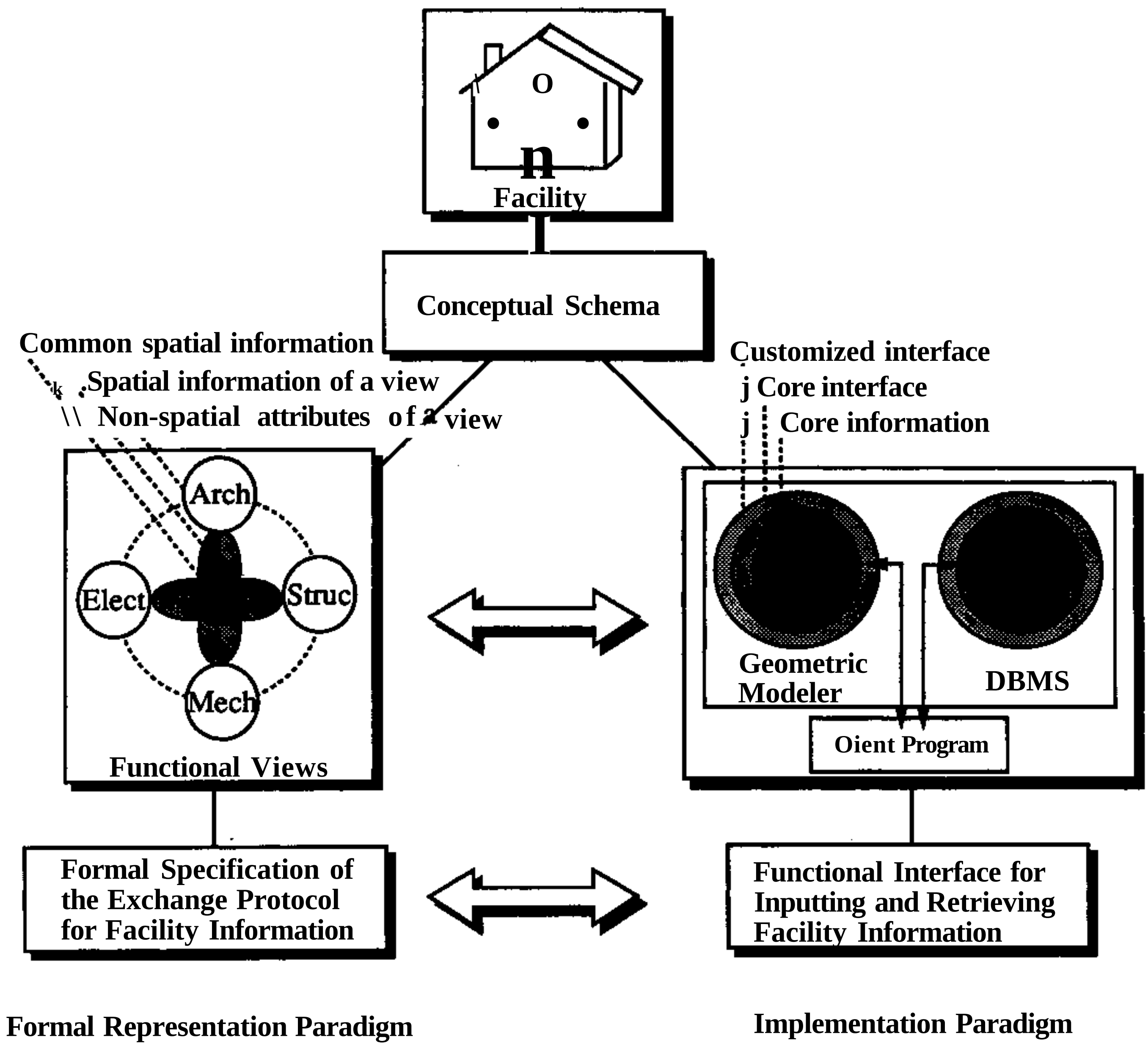


Figure 1: A Frameworic for representation and communication of facility data

In our approach, illustrated in Figure 1, we define several functional views for a constructed facility (e.g., architectural, structural, and HVAC for the design stage), where each view consists of a set of common and discipline-specific elements, their spatial attributes, and the discipline-specific non-spatial attributes of the elements used in that view. The spatial information is mapped onto a conceptual schema representing the union of all spatial information. Thus the system can, in principle, respond to queries such as: "find all architectural rooms supported by at least one structural beam whose grade of steel is the same as that of the structural column located at the north-west corner of the HVAC zone K3." The information exchange protocol, in turn, is an object-based functional interface to an encapsulated environment that consists of a non-manifold geometric modeling system, NOODLES[2], and a relational DBMS to deal with the spatial and non-spatial information, respectively. This interface consists of high-level data definition and retrieval functions specifically designed for dealing with all functional views of a facility throughout various phases of its lifecycle. The concept of an object-based data structure, shown in Figure 2, is similar to the concept of an object used in object-oriented languages and databases [3] in terms of providing data abstraction and encapsulation, specific interface methods for manipulating the attributes of an object and polymorphism, however the object-based model does not provide (or impose) the hierarchical taxonomy and inheritance mechanism associated with object-oriented paradigms.

Because all disciplines involved in various stages of a facility lifecycle are concerned with various forms of spatial information, a large part of the present effort is devoted to devising general spatial representation and modeling techniques that address the specific needs of the domain of constructed facilities. This paper discusses some of the results of this investigation, in particular a general representation and identification scheme for spatial abstractions of facilities. Work on combining the non-spatial attributes of facility components with their corresponding spatial data is underway and will only be briefly sketched.

2 Spatial Information of Constructed Facilities

A constructed facility and each of its components, including non-tangible components such and fire zones, are rigid objects that occupy certain volumes in the physical world. Here we define a "space" as a point set, i.e., a subset of the three-dimensional Euclidean spaces ($£^3$), that is an abstract geometric model of a facility component. In this work, we restrict the representation of these spaces to be valid but *not* unambiguous (complete) or unique; in other words, a space need *notbc* a/?-set as defined in [4], This is because depending on the level of detail available or required to describe an object, the representation of the space corresponding to that object can be of any dimensionality. For example, a pipe can be represented as a three-dimensional

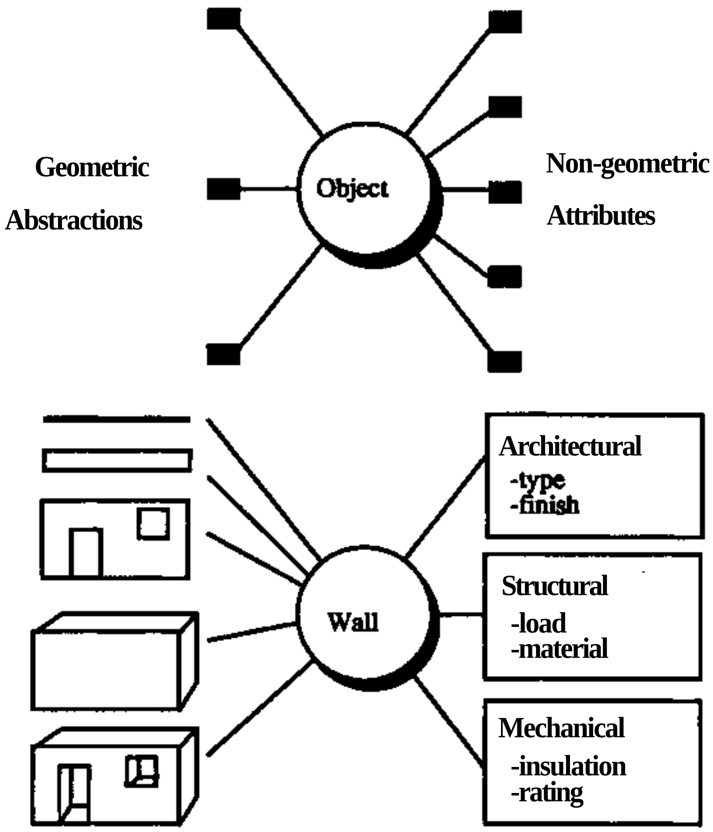


Figure 2: Object-based data structure

cylindrical solid, a two-dimensional circle representing the cross section of the pipe at a specific location, a one-dimensional line describing the pipe's center-line, or simply a point indicating the center-point of the pipe on a particular surface. Note that the notion of space is adopted here to refer to an object when only the spatial attributes of that object are of interest, and therefore, space and object are interchangeable in the general sense.

The organization of spaces (or objects) in a constructed facility can be conceptually viewed as a directed graph structure whose nodes correspond to the spaces and whose links describe the spatial relationship between related spaces[5]. As an extreme example, an HVAC zone can be linked to a structural column via the "contains" relationship. This graph structure can potentially contain all the spatial information about all the facility components; however, it is too general and contains too much data (spaces and relationships) for any one particular discipline to deal with. Therefore, it is necessary to provide each discipline with its own organizational scheme of the spaces while maintaining consistency between these discipline-specific models and the primary model. Consistency is achieved by making each specific spatial model a discipline-specific (or functional) "view" of the primary (or conceptual) model, in the same sense as used in database management systems. For example, architects may wish to deal with spaces such as floors which contain suites which in turn consist of rooms that are separated by walls. This organization of spaces is simply a specialized view of the primary model containing the spatial information about all the spaces in the facility, which may include rooms, pipes, walls, beams, etc.

Although different disciplines have their own special arrangements of spaces in a constructed facility, in most cases they deal with the spatial decomposition of the facility in a hierarchical fashion, i.e., given a particular space, they subdivide that space into smaller subspaces by defining either partitions or individual subspaces, or both. This process is in fact initiated by modeling the constructed facility itself, either by defining its bounding envelope or directly via its topology and geometry, where the three-dimensional universe of physical space is partitioned into the spaces corresponding to the "inside" and the "outside" of the facility. Once this "inside" space is defined, it can be recursively decomposed into disjoint subspaces until the desired level of granularity has been reached by a particular discipline. During this process, it is also possible to decompose a union of several spaces instead of limiting the decomposition technique to the leaf nodes of the hierarchical tree. It is important to note that although the decomposition methodology described here is the same for all disciplines, it does *not* imply that the level of decomposition or the arrangement of spaces must also the same. In other words, each discipline is free to arrange its own spatial organization; meanwhile, it can also use the spaces generated by other disciplines for defining its own spaces.

From the above discussion one can observe that while the space corresponding to a particular facility represents a unique subset of E^3, the other spaces that are contained within that space, e.g., spaces representing rooms, pipes, columns, etc., are not orthogonal, i.e., they may overlap, intersect, or coincide. This "non-orthogonality" of spatial information is of great importance in the formal representation of constructed facilities, and the model proposed in this work specifically addresses this issue.

2.1 Formal Representation of Spatial Information

Rigid solid objects are represented by various geometric modeling techniques[4]. The representation of a spatial abstraction of an object (referred to as a space here) is merely a mathematical approximation of the actual geometry of that object. The degree of approximation used for representing a particular object is generally determined by the level of detail needed for dealing with that object. For example, a structural beam can be modeled with a one-dimensional line for most structural analyses, because the end-point coordinates of the beam (and some cross-sectional properties that are stored separately from the geometric model) are sufficient for idealizing the beam for purposes of analysis. However, if one needs to check for the spatial interference of a beam with another object, the one-dimensional representation of the beam does not provide sufficient information. Therefore, it is necessary to be able to represent and identify components of a facility uniformly at various levels of abstractions in order to provide the desired information for spatial reasoning by different disciplines.

This work uses a vertex-based, non-manifold, boundary representation [2] for geometric modeling of all spatial abstractions of constructed facilities. This representation technique uses a set of disjoint, atomic geometric primitives[1], i.e, vertex, edge, face, and solid, and additional topological primitives, such as shell, loop, and bond, to represent any geometric model in a non-manifold paradigm. This paradigm allows for non-homogeneous models, such as models with "dangling" edges or vertices, and closure under all Boolean operations [4]. However, the primitives and functions provided by NOODLES to create and manipulate a geometric model are much too general and low-level to be used directly for representing the spatial information of facilities. As a result, higher-level primitives and functions, suitable for dealing with spatial attributes of constructed facilities, must be developed on top of this geometric modeler, as shown in Figure 1, to provide a more specialized interface and to encapsulate the low-level geometric data and operations from users unfamiliar with NOODLES. Regardless of the implementation of these high-level primitives and functions, the first step is to formalize the appropriate representation and identification schemes used as the basis of our implementation using NOODLES.

2.1.1 A General Representation and Identification Scheme

The spatial attributes of a component of a constructed facility are typically defined either with respect to some reference geometric entities, such as grids or boundaries, or relative to the spatial attributes of another component of that facility. The reference entities can be linear or curved, arranged in orthogonal or arbitrary directions, or be represented by zero- or higher-dimensional geometric entities. The key idea is that these entities provide a mechanism for defining and/or identifying the bounding envelope of a component. Specialized versions of this approach have been used in some spatial representation schemes, particularly in the Tartan Grid developed for rectangular, orthogonal buildings [6]. The scheme developed in this work is based on using special geometric entities, referred to as "superior elements", to represent and identify any space in a given spatial configuration. In this paper we present this scheme in the context of constructed facilities; however, this scheme is general enough to be used in any domain in which the spatial configuration of. objects, particularly in a non^manifold paradigm, is of interest.

The concept of superior elements can be described a general case of half-spaces (or half-planes for two-dimensional geometric entities) used in the CSG representation scheme [4] because a superior element

^OODLES refers to these primitives as "fundamental elements." [2]

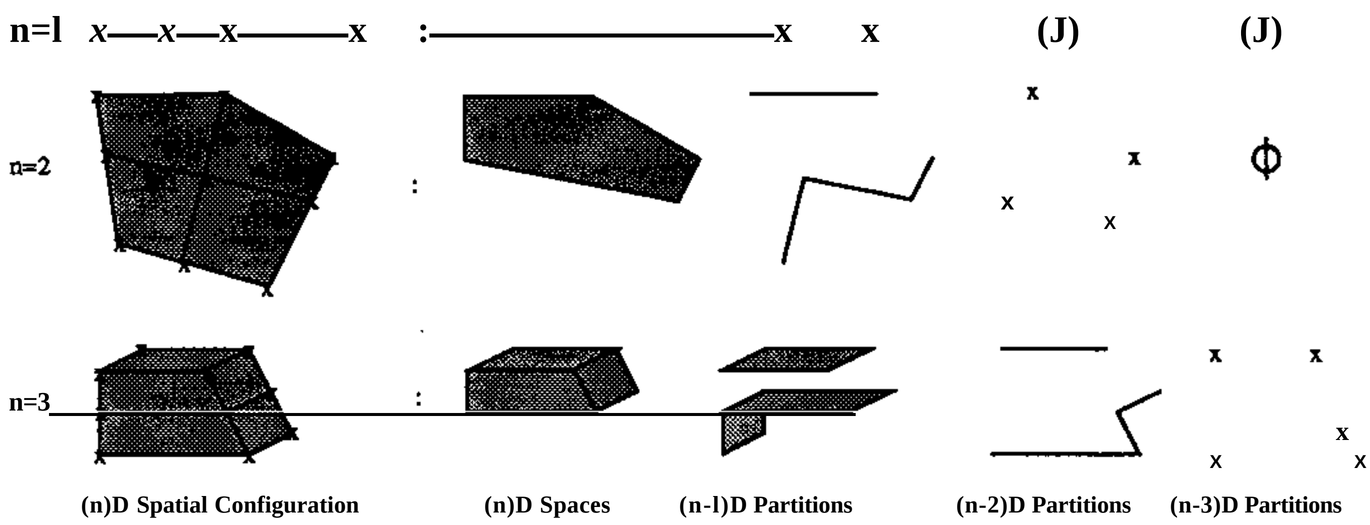


Figure 3: Spaces and partitions in any spatial configuration

does not have an orientation and is used primarily to identify the boundaries of geometric entities. The representation scheme described here can be theoretically used for non-linear configurations as well; however, due to limitations of the existing geometric and solid modeling techniques for dealing with arbitrary curved-boundary solid models[2][7], the discussion in this paper is limited to linear geometry, i.e., lines, planes, and polyhedrons. This limitation is by no means a severe.one, because the majority of spatial configurations, especially in the domain of constructed facilities, consist of linear partitioning and boundary elements; furthermore, any of curved geometry can be modeled as a piece-wise linear element with an acceptable level of accuracy.

The following axioms formalize the spatial characteristics of the proposed scheme:

I. A /{-dimensional ($1 \le n \le 3$) spatial configuration S is a subset of E^n and consists of two distinct types of geometric elements:/i-dimensional spaces, and/-dimensional ($0 \le i < n$) partitions, where neighboring spaces are separated by their adjacent partitions (see Figure 3). It is important to note that a space may be used as a partition in a higher-dimension abstraction of a configuration, and a partition may be used as a space in a lower-dimension abstraction of a configuration.

II. For any /z-dimensional spatial configuration S, there exists at least one ^-dimensional "carrier"[3] model C such that S CC. Note that in some cases $C = E^n$.

III. Two consistent [4] representation schemes, each defined by a set of geometric elements, can be used

[2]with the exception of some quadric curves and surfaces

[3]The notion of "carrier" is borrowed from [7].

for describing the spatial configuration *S*:

(i) the set *A* of all disjoint, atomic geometric primitives of dimension *n* or lower that comprise the configuration *S* [4]; and,

(ii) the set *B* of all superior elements, defined as (n - 1)-dimensional geometric entities that partition the configuration's carrier model *C* into a superset of *A* via the binary operator ©, i.e., $A \subseteq C © B$ [5] (The © operator is described in detail in Section 2.2).

IV. *B* can be derived from A, given the criteria for deriving the superior elements in *B* from the $(n - 1)$-dimensional geometric entities in *A* (For linear geometries this criteria is to derive the equations of lines or planes corresponding to the (n — 1)-dimensional entities in *A*.).

V. Any spatial subset of configuration *S* can be uniquely represented and identified by a subset of A.

VI. Any convex space in *S* can be uniquely represented and identified by *S*'s carrier model *C* and a subset of *B* whose elements spatially contain *exactly all* the boundaries of that space.

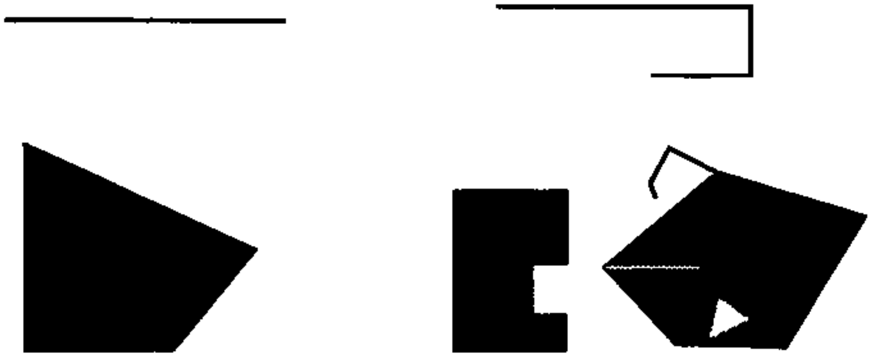
VII. Any m-dimensional $(0 \leq m \leq n)$ convex partition in *S* can be uniquely represented and identified by a subset of *B* in which either an element (if $m = n - 1$) or the intersection of two elements is a m-dimensional geometric entity that either is spatially equal to (if $m = 0$) or is a carrier model of the desired partition, and the remaining elements (if $m > 0$) spatially contain *exactly all* the boundaries of that partition.

VIII. Any concave space or partition in *S* can be represented and identified via Boolean operations on several mixed-dimensional convex spaces or partitions in *S*.

To clarify the above axioms, it is helpful to first describe the difference between the convex and concave geometric entities and then provide a concise syntax for defining these entities based on the superior element scheme. A convex geometric entity does not have any internal angle greater or equal to 180 degrees, thus it may not have any cavities or dangling parts. A concave geometric entity, on the other hand, may have internal angles greater than 180 degrees and possibly cavities and dangling parts of the same or lower dimensions. Figure 4 presents several examples of convex and concave geometric entities. From this definition and the above axioms, any geometric entity *a* of a spatial configuration *S*, i.e., a convex or concave space or

[4]This is analogous to a *minimal* representation scheme.

[5]This is analogous to a *maximal* representation scheme.



Convex **Concave**

Figure 4: Examples of convex and concave geometric entities

partition element, can be defined symbolically via the following syntax (a list of entities are enclosed in a pair of parenthesis, and a dotted pair indicates an intersection of two entities in that pair):

$$S = A$$

$$S ::= \backslash e\backslash,\ldots,e_n)$$

$$ti ::= (C(b_h\ldots,b_m))\backslash(BOOL(eje_k))$$

$$C ::= NULL \ \backslash E\backslash bi\backslash \ (bi.bj)$$

$$BOOL ::= \text{UNION} \mid \text{INTERSECTION} \mid \text{DIFFERENCE}$$

$$bi \ e \ B$$

To summarize the above discussion and explain the above formalism, we define *S* as a n-dimensional ($1 \leq n \leq 3$) spatial configuration that is comprised of various possible m-dimensional ($0 \leq m \leq n$) spatial subsets e_i and its minimal representation is the set *A*. Each convex *;* is defined by a m-dimensional "carrier" model *C* that spatially contains i< and a list of superior elements that uniquely contain all the boundaries of *a*. *C*, depending on the dimensionality and type of *eu* can be *NULL* (when $m = 0$), the Euclidean space E^m, a superior element, or the intersection of two superior elements. Each concave spatial subset is in turn defined by a Boolean operation and two other spatial subsets (concave or convex). Finally, the $n — 1$-dimensional superior elements *bi* corresponding to *S* arc members of a set *B* specified by the user.

The superior-element scheme may seem rather complex at first;* however, it simply formalizes a very intuitive and rather common technique where an application uses some reference geometric elements, such as grid lines, to specify and identify the spatial attributes of a group of components. The following example illustrates the above axioms by using a simple, yet general, two-dimensional spatial configuration.

An Example Configuration. As a simple example, consider the star-shape spatial configuration of Figure 5a. Figure 5b is a visualization of the set A which is comprised of disjoint zero-, one-, and two-dimensional atomic elements; Figure 5c shows the superior elements constituting set *B;* and Figure 5d shows a convex and a concave space, where each space is represented and identified by both the superior-element and the atomic-element schemes[6].

Partition elements, i.e., line segments and points, although not shown in Figure 5, can also be readily dealt with by using either scheme. For example, the left vertical partition of the convex space in Figure 5d is represented either by five atomic edges and their corresponding six vertices, or by *(s2(sl0$_y$sl5))* using the above formalism. Also note that once a particular space or partition has been identified, the lower-dimension atomic elements that are spatially contained in that space are also identifiable.

2.2 A Recursive Spatial Decomposition Technique

Based on the above discussion, a recursive spatial decomposition technique is developed. In this technique, a spatial configuration *Si* (with *SQ* representing the starting configuration) is created by partitioning its carrier geometric model *d* with a set of superior elements in *Bi* via the © binary operator. This decomposition technique can then be repeated for any one or any union of the atomic elements created in the previous decompositions, i.e., from A;_n(i $\leq n \leq$ i), using a new set of superior elements *Bi*+\. This recursive process is continued until the desired level of spatial granularity is reached. Note that the superior elements in *Bi-mW* $\leq$ *m*_< o, provide sufficient means for identifying any one or any group of atomic elements in A*.

The decomposition technique is formally described as follows:

$$S_i \subset C_i$$

$$C_i \equiv E \mid a_{(i-m)k} \mid \{a_{(i-m)k} \cdots a_{(i-n)l}\}_{(1 \leq m,n \leq i)}$$

$$S_i \equiv A_i$$

$$a_{ij} \in A_i$$

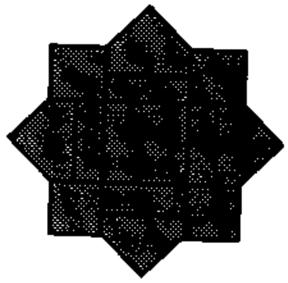
$$a_{ij} ::= \text{vertex} \mid \text{edge} \mid \text{face} \mid \text{solid}$$

$$b_{ij} \in B_i$$

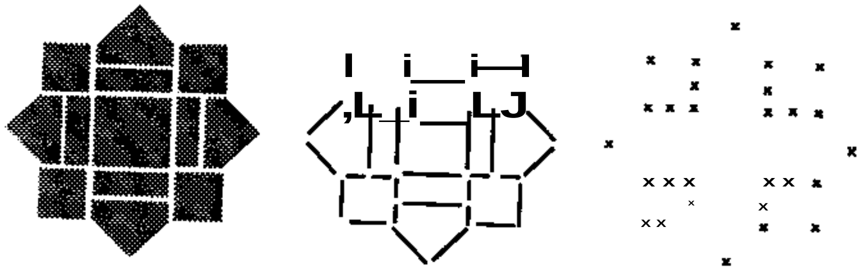
$$C_i \circledR B_i \Rightarrow A_i$$

The above formalism is intended to describe the proposed decomposition process and the relations between the various entities that are used in this process; therefore, it should not be confused with generative

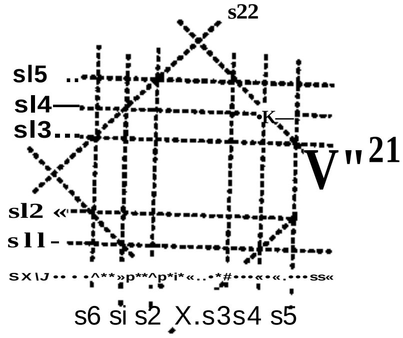
[6]For a similar three-dimensional configuration, the atomic elements in Figure 5b will be augmented with polyhedrons, while the superior elements in Figure 5c will be replaced with plane elements.



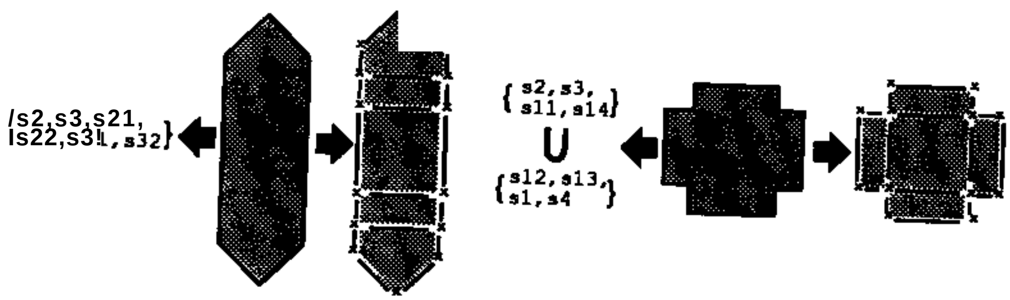
a) Spatial Configuration S



b) Disjoint, Atomic Elements Constituting *A*



c) Superior Elements Constituting *B*



d) Representation of Convex and Concave Spaces

Figure 5: An Example of a Spatial Configuration

grammars that arc often used in applications such as design synthesis [8]. In this formalism a spatial configuration *Si* (with *So* representing the starting configuration) is created by partitioning its carrier model *d* with a set of superior elements *by*·in B,·via the © operator. The carrier model is equivalent to either the Euclidean space (often used to initiate the decomposition process) or one or more of the atomic elements created in the previous decompositions, i.e., from $Ai\text{-}_n\{\backslash \leq « \leq i)$. This recursive process is continued with a new set of superior elements, which may have a different dimensionality than the dimensionalities of sets used in the previous steps, until the desired level of spatial granularity is reached.

Decomposition of the Example Configuration. The example configuration presented above is used to demonstrate the proposed spatial decomposition technique. Figure 6 illustrates a series of spatial decompositions, starting with E^2 as *CQ.*

2.3 Creation of Functional Views

During the spatial decomposition process described above, a spatial configuration *Si* may be viewed differently by various disciplines. These discipline-specific (or functional) views F* are created via the binary function 0 as subsets of A; using a given set of selection criteria Da, i.e.,

$$F_{ik} = A_i \ominus D_{ik}$$

The 0 function is similar to the "select" operation used in relational databases, with *Ai* being analogous to a relational table and £># representing the constraints imposed on the attributes of selected tuples.

The elements in D# can be of three types:

1. geometric class of elements in A/,
2. superior elements in *But* defining the spatial extent of F#, and
3. a list of labels of labeled elements in A_i.

The selection criteria in *D&* are combined through the logical connective "AND" in order to create a functional view. In other words, the 0 operator first selects groups of elements in A_i that satisfy individual criteria in £># and then returns only those selected elements that satisfy all the given criteria.

Using the above decomposition example in the context of constructed facilities, the last three decompositions in Figure 6 can be interpreted as various ways of arranging the interior space of the facility, created via the first decomposition, for different functional views. For example, the edges and vertices in the decomposition A i can be abstractions of the structural frame, while the faces of A3 can depict a possible

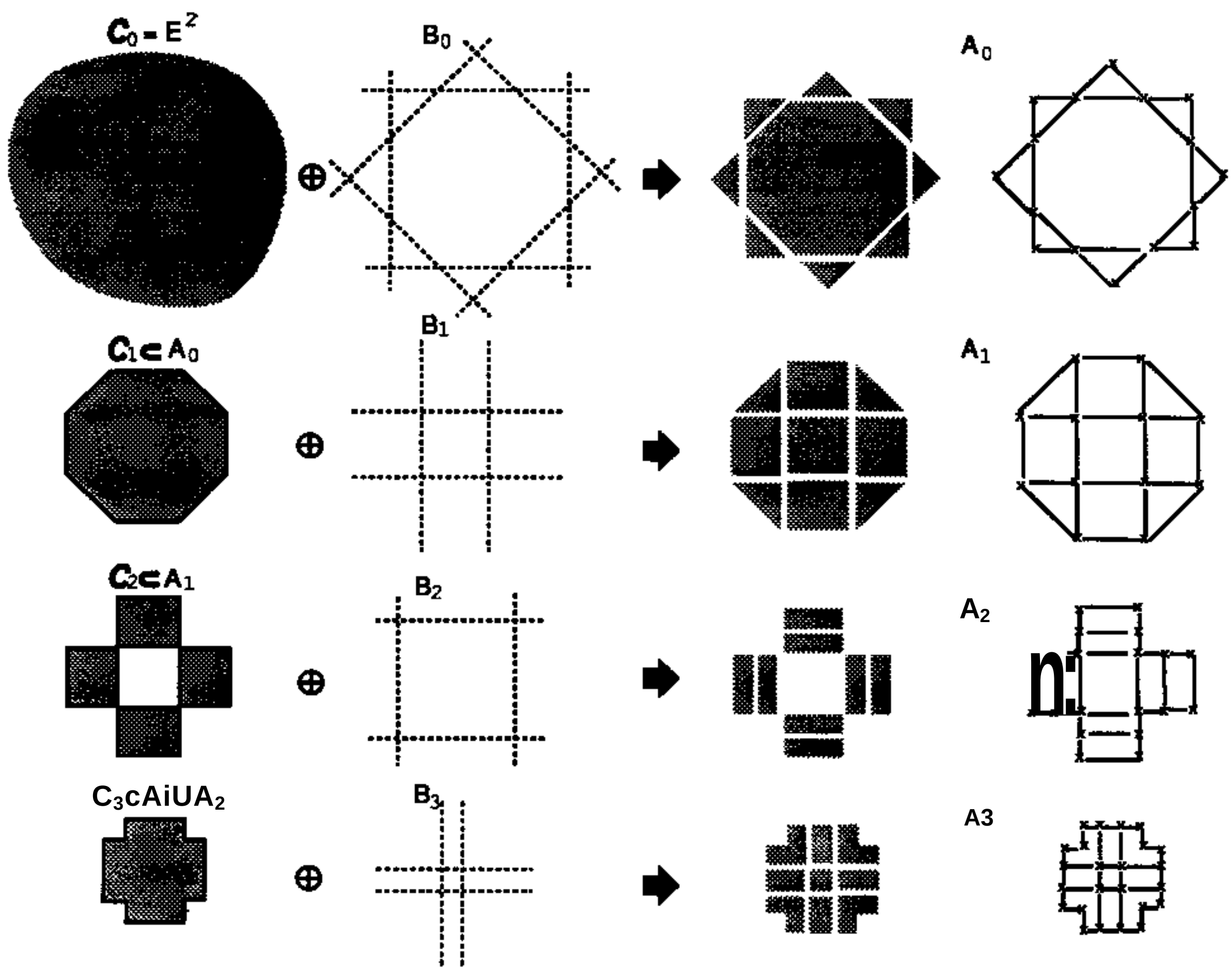


Figure 6: An Example of the Proposed Spatial Decomposition Technique

architectural layout of the main corridors. If any elements of *Ai* were labeled, as described in the following section, the labeling information could also be used either by itself or in conjunction with other types of constraints to create various functional views of the configuration.

2.4 Labeling of Spatial Elements

The superior-element representation scheme can be used for identifying the space and partition elements of a spatial configuration. This identification technique may appear cumbersome and not suitable to those who wish to access these elements in a more customized manner. Therefore, a mechanism for labeling spatial configurations, their elements, and their functional views are provided. This mechanism allows user-defined labels to be attached to an entity (and optionally its subparts) either when it is created or later by identifying that entity using the superior-element identification scheme. Furthermore, in an interactive graphic environment, it is desirable to be able to select and manipulate entities via graphic utilities, such as screen selections, etc.

In this work, the realization of a label is a pointer to a data structure containing several attribute-value pairs. The two primary attributes in this data structure are "name" and "view." The values for the name attributes can be any arbitrary strings of characters (within a reasonable range) defined by the user, such as beaml3G or zoneAC3. Meanwhile, the view attribute facilitates the classification of spatial elements based on some preset or user-defined functional criteria, such as floor-systems or HVAC-zones. Note that an element is uniquely labeled by a distinct combination of its name and view attribute values, i.e., two elements in the same view may not have the same name.

3 Linkage to a Database Management System

So far, this paper has concentrated on the representation of mixed-dimensional spatial attributes of the facility components. In addition to a component's spatial attributes (i.e., its topology and geometry) each component has other types of attributes, such as material properties, behavior, etc., that are needed by different functional views of the constructed facility. These attributes, which are referred to as the non-spatial attributes, are defined either explicitly by their values or by reference to items in some standard component libraries. Management of the non-spatial attributes of the facility components in a computer-integrated environment is as important as dealing with the spatial attributes. Much work has been done in the area of data modeling and information management for the domain of constructed facilities, primarily for dealing with the general organization of the facility representation and of the non-spatial attributes [9,10,11,12]. The contribution of the present work in this area is to provide a linkage between the spatial and non-spatial attributes of the facility components while maintaining separate data models and representation schemes for each of the information types.

As shown in the proposed framework of Figure 1, the spatial attributes of the facility components are handled by a geometric modeling system, while the non-spatial attributes are dealt with in a separate database management system. An object-based client program, designed and implemented specifically for storing and retrieving the facility information, interacts with the geometric modeler and the database management server programs and therefore encapsulates the underlying representation schemes and the functional behaviors of the two server programs. This client program consequently provides an object-based functional interface with which users can specify new information and query existing information about a facility to be respectively stored in and retrieved from the server programs. Each of the server programs, as well as the client program, have their own data representation schemes, and it is the client program's responsibility to coordinate the translation between these different representation schemes. A somewhat similar but more general architecture has been developed in KADBASE [13] for linking various database and knowledge-base systems in the structural engineering domain.

Separation of the component attributes into two distinct types, i.e., spatial and non-spatial, and managing each type of information by a specialized system has the major advantage of using data representation schemes and algorithms that are most appropriate for the particular type of data being considered. For example, it is extremely difficult, if not impossible, to represent and manipulate mixed-dimensional geometry by using any existing database technology, such as relational or object-oriented, as effectively and consistently as it can be achieved by using a non-manifold geometric modeling system; and vice versa, performing database operations such as join and select are extremely cumbersome without the underlying relational algebra provided by a relational database system. Thus, by appropriately linking a geometric modeling system and a relational database management system via a high level, object-based program, a powerful information management system is created to effectively handle various types of information about a constructed facility in a computer-integrated environment. On the other hand, there is a performance price to pay when data is partitioned across different programs and has different representation schemes. This drawback is quite severe when one type of information is dominant over the other types, e.g., when the geometric data associated with the facility components is relatively simple, or one functional view primarily deals with only one specific type of data. For such cases, it is possible to either store duplicate (but consistent) data in different programs, or for an extreme case, store all the information in the program that is most relevant.

The current implementation of the proposed framework shown in Figure 1 links a relational database with the NOODLES non-manifold geometric modeler [2]. A menu-driven, interactive graphic prototype program has been developed that uses the high-level, object-based functional interface in order to define the facility components by mixed-dimensional geometry of any desired abstraction and optionally attach non-spatial attributes to these components. The linkage to the relational database enables users to automatically transfer data from or to the database via the object-based client program without the detailed knowledge about the underlying table organizations or the SQL commands issued. Every component is uniquely identified by its name and functional view, as described in the preceding section. Furthermore, a unique internal id is attached to every component once it is defined, and the spatial attributes (represented in terms of the geometric entities of NOODLES) and the non-spatial attributes (represented in terms of the tuples in relational tables) of a component are linked together via this internal id. The object-based client program in turn is responsible for maintaining the integrity and consistency of the component attributes by using this identification mechanism. Furthermore, the client program encapsulates the underlying data representations and functional interfaces provided by the geometric modeler and the relational database, thus providing a higher level of abstraction for dealing with the facility information. In a sense, the client program provides a flexible yet powerful object-based functional interface that combines the rigorous and well-defined paradigm of relational databases with the unique geometric modeling capabilities of the new vertex-based, non-manifold representation scheme in order to facilitate the modeling and the communication of various abstractions of artifacts, including constructed facilities.

4 Summary and Future Work

This paper mainly discusses issues with regards to the representation of mixed-dimensional spatial abstractions of the components of constructed facilities. These issues are of great importance in order to develop the proposed framework for modeling and communicating the facility spatial information in a computer-integrated environment. As a result, a general representation and identification scheme is developed to deal uniformly with the topological relationships between mixed-dimensional geometric abstractions of components used by different functional views throughout the lifecycle of a constructed facility. This scheme is based on a new maximal boundary representation technique that is provided on top of the minimal boundary representation of the underlying non-manifold geometric modeler. This maximal representation scheme (here referred to as the "superior elements" scheme) provides a symbolic way of modeling and identifying spatial configurations and their subparts consistently and uniformly regardless of the dimensionality of the configuration's geometric entities. Furthermore, the underlying vertex-based, non-manifold geometric modeler provides the mechanism for retrieving topological relationships, such as adjacent-to, contains, or overlaps, between mixed-dimensional abstractions of components.

A number of issues that are specifically important in dealing with adjacency relationships and mixed-dimensional geometry of the abstractions used for modeling the components of constructed facilities have prompted the development of the superior element representation scheme. The two most important of these issues are:

- Invariance of topological relations with respect to dimensionality of the geometry.
- Distinction between the explicit topological relationships in the geometric model and the implicit ones perceived by the functional views.

The first issue is addressed by using a non-manifold geometric modeling paradigm in which geometric entities of different dimensionalities can coexist in the same model and can be manipulated consistently and uniformly with a set of dimensionally polymorphic operations. For example, the fact that specific pipes are contained in a particular room is independent of the dimensionality of the geometric entities representing various abstractions of these components. The superior element scheme intends to address the second issue by providing a higher-level representation technique than that of the underlying non-manifold modeler so that the facility components can be defined in terms of some common skeletal frame. This skeletal frame, defined in terms of superior elements, thus provides a basis for resolving topological relationships between the facility components at higher levels of abstractions than the actual topological relations associated with the various geometric entities that represent different abstractions of a facility component. For example, two rooms sharing a wall are often perceived to be adjacent spaces by architectural or HVAC functional views, although in a purely topological sense they are adjacent to a common wall which may have a certain thickness; but the perceived adjacency relationship of the two rooms can be extracted by geometric reasoning about the higher-level abstractions (spaces and partitions) defined in terms of the superior-elements and mapped into their corresponding minimal, non-manifold representations;

The operations for spatial decomposition and functional view creation discussed in this paper are closely related to similar operations used in database management systems for developing appropriate taxonomies and groupings of information. Our initial findings suggest that there exist parallel hierarchical structures in the geometric model and the database organization of a constructed facility. In other words, the spatial decomposition of a facility and the creation of functional views and aggregate components is paralleled by the parent-child representation in relational databases.

Several issues will be studied closely and possibly implemented in the remaining time frame of this research. These issues include (but not limited to): associating multiple, mixed-dimensional geometric entities to a single component while providing consistent topological relationships between components; expansion of lower-level geometric entities to higher-level ones to deal with the evolution of the facility throughout its lifecycle; and formal specification of the functions supplied by the object-based client program in order to provide a high-level, functional interface that facilitates the exchange of facility information in a computer-integrated environment.

Acknowledgments This research has been supported under the cooperative agreement number ECD-8943164 by the Engineering Design Research Center, an NSF Engineering Research Center at Carnegie Mellon University.

References

[1] W. Gielingh. General AEC reference model (GARM). Technical report, AEC committee of ISO/STEP, P.O. Box 49,2600 AA Delft, The Netherlands, (1988).

[2] E. L. Gursoz, Y. Choi, and F. B. Prinz. Vertex-based represenation of non-manifold boundaries. In M. J. Wozny, J. U. Turner, and K. Preiss, editors, *Geometric Modeling for Product Engineering, pages* 107-130, New Yoik, September (1990). IHP WG 5.2/NSF, North-Holland.

[3] W. Kim. *Introduction to Object-Oriented Databases.* The MIT Press, (1990).

[4] A. A. G. Requicha. Representations for rigid solids: Theory, methods, and systems. *Computing Surveys,* 12(4):437-464, December (1980).

[5] M. K. Zamanian and S. J. Fenves. An information exchange protocol for constructed facilites. Technical Report 12-37-90, Engineering Design Research Center (EDRC), Carnegie Mellon University, (1990).

[6] S. J. Fenves and S. A. Liadis. A data structure for computer-aided design of buildings. *APEC,* pages 14-18, Fall (1976).

[7] C. M. Hoffmann. *Geometric and Solid Modeling: An Introduction.* Morgan Kaufmann, (1989).

[8] G. Stiny. Intoduction to shape and shape grammars. In *Environment and Planning B 7,* pages 343-351, (1972).

[9] S. J. Fenves, U. Hemming, C. Hendrickson, M. L. Maher, and G. Schmitt. Integrated software environment for building design and construction. *CAD,* 22(l):27-36, (1990).

[10] C. M. Eastman. The evolution of CAD: Integrating Multiple Representations. *Building and Environ*men*, 0(0):00-00, (1990).

[11] B. Bjoik and H. Penttila. A scenario for the development and implementation of a building product model standard. In *Current Research and Development in Integrated Design, Construction and Facility Management,* Stanford, CA, March (1989). CIFE, Stanford University.

[12] G. H. Powell and R. Bhateja. Data base design for computer-integrated structural engineering. *Engineering with Computers,* 4:135-143, (1988).

[13] H. C. Howard. *Interfacing Databases and Knowledge-based systems for Structural Engineering Applications.* PhD thesis, Camegine Mellon University, Pittsburgh, PA, (1986).