DOI:10.22337/2587-9618-2022-18-4-101-110

EXPLICIT DIGITAL MODELS OF LINEAR COMPLEXES

Aleksandr N. Rozhkov [1,2], Vera V. Galishnikova [1,2]

[1] People's Friendship University of Russia (RUDN University), Moscow, RUSSIA

[2] National Research Moscow State University of Civil Engineering, Moscow, RUSSIA

Abstract: It is obvious that the interoperability of existing digital models is insufficient. Current research on model view definitions and on their semantic enrichment addresses the issue of good interpretation of the results of existing models to improve interoperability. The alternative research presented in this paper is not concerned with interpretation. Instead, the influence of modifications in the geometric and topological concepts of the digital models themselves on their interoperability is investigated. The geometric and topological attributes of the models are made as explicit as possible. Two-dimensional line drawings are replaced by three-dimensional linear complexes to reduce the need for implicit information. The topology of a complex is described with topological tables containing all elements of the model, thus replacing the geometric neighborhood concept of the industry foundation classes. A highly efficient algorithm for the construction of new topological tables of large buildings is presented. The difficulties encountered in modifying existing topological tables are analyzed and solved. Topological and geometric aspects of linear complexes that cannot be treated explicitly with topological tables are identified and presented.

Keywords: linear complex, cell, polyhedral partition, topological modeling, topological tables, topology, neighborhood

ЯВНЫЕ ЦИФРОВЫЕ МОДЕЛИ ЛИНЕЙНЫХ КОМПЛЕКСОВ

А.Н. Рожков [1,2], В.В. Галишникова [1,2]

[1] Российский университет дружбы народов, Москва, РОССИЯ

[2] Национальный исследовательский Московский государственный строительный университет, Москва, РОССИЯ

Аннотация: следует признать, что функциональная совместимость цифровых моделей, используемых в программных продуктах разных производителей, на данный момент не достигнута. Текущие исследования в этой области нацелены на усовершенствование интерпретации данных существующих моделей и их семантическом обогащении. Альтернативное исследование, представленное в данной статье, посвящено влиянию модификаций геометрических и топологических концепций самих цифровых моделей на их функциональную совместимость. Геометрические и топологические атрибуты моделей сделаны максимально явными. Двумерные линейные чертежи заменяются трехмерными линейными комплексами, чтобы уменьшить потребность в неявной информации. Топология комплекса описывается топологическими таблицами, содержащими все элементы модели, заменяя, таким образом, концепцию геометрического соседства, используемую в системе отраслевых базовых классов (IFC). Представлен высокоэффективный алгоритм построения новых топологических таблиц больших строительных объектов. Проанализированы и решены проблемы, возникающие при модификации существующих топологических таблиц. Выявлены и представлены топологические и геометрические аспекты линейных комплексов, которые не могут быть явно обработаны топологическими таблицами.

Keywords: линейный комплекс, ячейка, полиэдральное разбиение, топологическое моделирование, топологические таблицы, топология, соседство

INTRODUCTION

The information used in the design, construction and operation of buildings must be explicit, reliable and complete. Digital models of the buildings are the tool with which the information is assembled, distributed and applied. The models are constructed with commercial software based on international standards. A widely used approach in the building industry are boundary representations of the building components and their assembly in a model using the industry foundation classes IFC [1].

The IFC were originally designed to be interoperable. Successful implementation of this concept would have permitted vendor-neutral automatic exchange of building information models in digital networks [2]. Because this goal has not been reached, the concepts on which the IFC are based and the manner, in which digital models based on IFC are interpreted, are the subjects of intensive current research [3–6].

The IFC are complex and voluminous, as they cover a very large number of topics in the building and construction industry. To permit the model users to focus on the specific information, which they require for their tasks, model view definitions MVD have been introduced [7]. A MVD specifies, which parts of an IFC data model needs to be implemented for a specific data exchange scenario. The MVD approach assumes that software companies will develop IFC export and import subroutines tailored for each MVD.

IFC Certification [8] for consistent and reliable implementations of IFC specifications by software vendors for multiple software platforms was developed. The procedure supports checks for collisions and voids using geometric attributes and element identities. The National BIM Standard initiative [9] facilitates information exchange through MVDs [10]. Interoperability is only guaranteed within a single MVD, not between different MVDs.

The semantic enrichment concept for building information models SeeBIM extends the MVD concept [11]. The concept postulates that IFC-based models contain both implicit and explicit information. To interpret both types of information, if-then rules are formulated using a predefined set of object types and operators. The operations include reading the building model, testing for geometrical and topological relationships, and creating new objects, properties, and relationships. The new and the enriched objects conform to the definitions of an MVD defined for the given subdomain.

Tools based on the MVD and SeeBIM concepts support the interpretation of the results of existing IFC-based models, but do not affect the models themselves. The question arises, whether unsatisfactory interoperability may not be due partly to inherent deficiencies of the IFC concept itself [2]. Such deficiencies cannot be remedied by concepts like MVD and SeeBIM. This paper analyzes the treatment of geometry and topology in the IFC concept, as described by Borrmann et al. [1], and investigates the use of topological tables as an alternative concept that promotes interoperability.

The most widely used approach to the modeling of geometric solids with IFC is Boundary Representation (Brep). Classes IfcFacetedBrep and IfcAdvancedBrep implement flat surfaces and surfaces with curved edges respectively for simply connected domains. Corresponding classes for multiply connected domains are

- IfcFactedBrepWithVoids and
- IfcAdvancedBrepWithVoids.

Solids such as walls and doors are constructed individually and aggregated to construct the IFC model.

The neighborhood of solids in an IFC model is described indirectly using classes that inherit from IfcObjectPlacement [1]. Each IFC solid has a local coordinate system, whose location in the model is specified in a common global coordinate system. This method is a geometric specification of neighborhood, which can increase the risk of collision of solids and voids between solids due to imprecise numerical attributes of the solids, especially their node coordinates.

This paper shows that the problematic geometric specification of neighborhood in IFC can be replaced by a truly topological specification of neighborhood: the contact of topological elements is described in topological tables. This is an example of the replacement of implicit information (the IFC user must convert geometric location to topological neighborhood) to explicit information (the topological tables explicitly name the elements, with which a specified element is in contact).

EXPLICIT AND IMPLICIT INFORMATION FOR BUILDINGS

Buildings are traditionally planned with two-dimensional line drawings showing plans, elevations, sections and details of the project. The line drawing in figure 1 shows the plan and a section of a room. The drawings contain explicit information about the original, such as the dimensions of the building components and their projections to the plane of the drawing. The explicit information in a drawing is not sufficient to create a three-dimensional mental model of the three-dimensional original of the building. For example, building components in a plan are not explicitly associated with the same components in the elevations and sections. Some faces of components are not shown explicitly in the projections. The person reading the drawing must add implicit information to the explicit information to be able to create the mental model. Persons with different background, knowledge and experience add different implicit information. Implicit information is therefore a potential source of inconsistency, inaccuracy and errors in engineering practice. The aim of our research is to make models explicit.

The storage capacity and high speed of the digital environment provide an opportunity to lessen the need for implicit information. The strategy of our research project is to map as many of the topological and geometric properties of the original explicitly to the model as possible. The mapping is bijective, such that the information and insights gained with the model can be applied to the original, and vice versa. Line drawings remain a valuable tool in engineering practice. However, in the digital environment line drawings for selected parts of the project are prepared upon demand and for a specific purpose from a general computer model of the entire original that contains the explicit information describing the original.

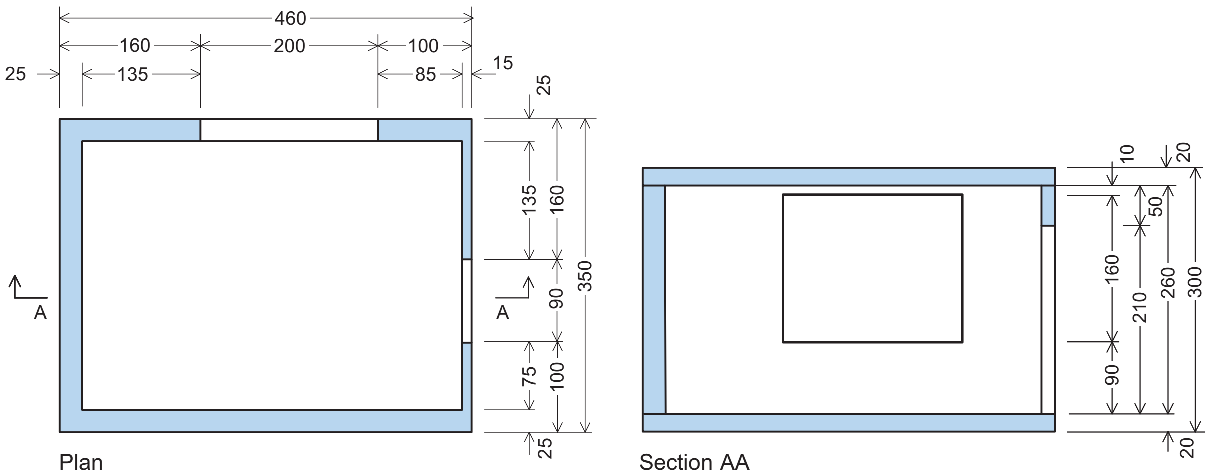

Figure 1. Line drawing showing the plan and a section of a room

LINEAR COMPLEXES

The character of explicit and implicit information is investigated for linear complexes. The linear complex for the room in figure 1 is shown in figure 2. A linear complex is a configuration composed of nodes, edges, faces and cells called the domains of the complex. A node is a single point. An edge is a straight line segment. A face is a plane area bounded by at least one closed polygonal curve composed of edges. A cell is a volume bounded by at least one closed polyhedral surface composed of faces. A rank from 0 to 3 equal to their dimension is assigned to nodes, edges, faces and cells respectively to create a hierarchy.

A domain is described by its boundary. The boundary of a domain of rank n consists of domains of rank n-1. For example, the boundary of a cell consists of faces. The topology of a complex describes relations between its domains. For example, the topology specifies the edges of a cell. The geometry of a complex is specified with the global coordinates of its nodes and the rules for the shape interpolation between to nodes of the domains.

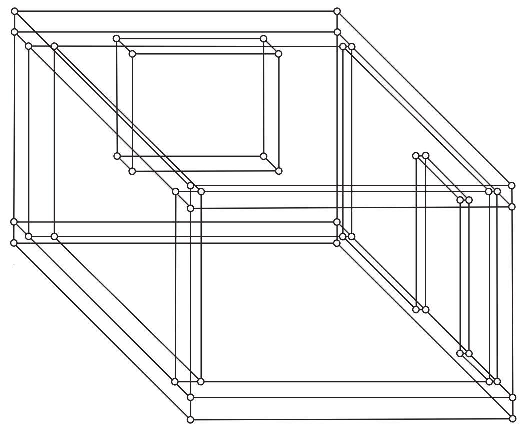

Figure 2. Perspective of the linear complex for the room in figure 1

The topology of a complex is unique. However, complexes with equal topology can have different geometries. For example, four nodes and four edges can form the boundary of a rectangular face. If the coordinates of two adjacent nodes are interchanged, the topology does not change, but two edges intersect at internal points such that they no longer form the boundary of face.

TOPOLOGICAL TABLES OF A COMPLEX

The complex, which is shown graphically in figure 2, is described alphanumerically in a computer model. A unique name is assigned to each domain of the complex. The node objects are collected in a map using the node name as key and the three global node coordinates as value of an entry. The topology of the complex is described with 12 topological tables. The first column of a table contains all components of a given type D_1 in the complex, for example all faces. The other columns of the table contain objects of another type D_2 in the complex, for example nodes. A row of the table contains the objects of type D_2, which are in contact with an object of type D_1. In the example, the table contains the nodes, which are in contact with a face of the complex. A topological table is named with the domain types D_1 and D_2, for example face-node-table.

A complex contains 4 types of domains: nodes, edges, faces and cells, one of which is placed in the first column of a table. Once the type for the first column has been selected, there are three remaining types, one of which is entered in the other columns. Some tables have a constant number of domains per row, for example the edge-node-table, whereas the number varies for others, for example the face-edge-table. The number of type combinations that can be formed for a table in this manner is 3*4. Figure 3 shows the 12 types of topological table arranged in matrix form.

Entry T_{kk} in the matrix is the set of the domains of rank k. It is not a topological table. The entries below the diagonal are tables showing the domains of rank m<k, which are components of the domain of rank k, for example the nodes which are components of a face. The entries above the diagonal are tables showing the domains of rank m>k, which have a common domain of type D_k. For example, T_{13} is the edge-cell-table which contains the cells of the complex that have a common edge.

CONSTRUCTION OF THE TOPOLOGICAL TABLES

Objects that are instances of industry foundation classes IFC describe their own topology. The topological relationship to other objects of the model is specified explicitly in special cases such as a common face of two cells. In general, the overall topology is not specified explicitly. Additional implicit topological information must be derived from the relative location of the

objects. The concept of our project is to describe the topology of all elements of the complex explicitly in common topological tables. This approach demands an efficient algorithm for the construction of large topological tables.
The topological tables in figure 3 are not independent. For example, the edge-node, face-edge and cell-face tables together define the topology of the complex completely. They are called base tables. The other tables can be derived from the base tables.

T_{km}		rank m , domain type			
		0 node	1 edge	2 face	3 cell
rank k, domain type	0 node	T_{00}	T_{01}	T_{02}	T_{03}
	1 edge	T_{10}	T_{11}	T_{12}	T_{13}
	2 face	T_{20}	T_{21}	T_{22}	T_{23}
	3 cell	T_{30}	T_{31}	T_{32}	T_{33}

Figure 3. Matrix of topological tables T_{km} for linear complexes

The base tables are specified by the user with a sequence of commands consisting of a key word *node, edge, face or cell* for the component type, followed by the name of the domain and a set of parameters. Each command defines one domain of the complex. The commands for a complex end with command *do*. The command sequence for the construction of the unit cube in figure 4 is shown in figure 5.

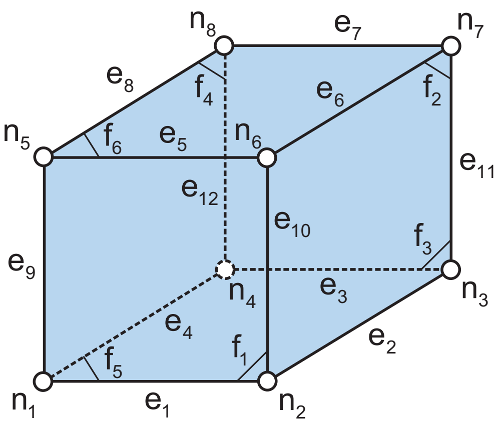

Figure 4. Perspective of a unit cube

Because the commands can be specified in arbitrary order, a command can contain objects that have not yet been defined. For example, the objects for the four edge objects in command *face* f2 (e2, e11, e6, e10) in figure 5 are not yet constructed.

```
cell   c1 (f1, f2, f3, f4, f5, f6)
face   f1  (e1, e10, e5, e9)
face   f2  (e2, e11, e6, e10)
face   f3  (e4, e11, e7, e12)
face   f4  (e2, e12, e8, e9)
face   f5  (e1, e2, e3, e4)
face   f6  (e5, e6, e7, e9)
edge   e1 (n1, n2)
edge   e2 (n2, n3)
edge   e3 (n3, n4)
edge   e4 (n4, n1)
edge   e5 (n5, n6)
edge   e6 (n6, n7)
edge   e7 (n7, n8)
edge   e8 (n5, n8)
edge   e9 (n1, n5)
edge   e10 (n2, n6)
edge   e11 (n3, n7)
edge   e12 (n4, n8)
node   n1 (0.0, 0.0, 0.0)
node   n2 (0.0, 1.0, 0.0)
node   n3 (0.0, 1.0, 1.0)
node   n4 (0.0, 0.0, 1.0)
node   n5 (1.0, 0.0, 0.0)
node   n6 (1.0, 1.0, 0.0)
node   n7 (1.0, 1.0, 1.0)
node   n8 (1.0, 0.0, 1.0)
do
```

Figure 5. Commands for a unit cube

When command *face* is interpreted, a persistent *Face* object is constructed and its edge attributes a persistent *Face* object is constructed and its edge attributes are set to null. The face is entered in a face map with the face name as key and the reference of the *Face* object as value. The names of the edges and the reference of the persistent Face object are stored in a transient shadow edge object. When command *do* is reached, all persistent objects of the complex have been constructed and entered in the map. The set of shadow objects is traversed. The names of the edges are used to read the references of the persistent *Edge* objects in the map, which are then stored in the persistent *Face* object. After command *do* has been executed, the independent topological tables exist on the data base.

For navigation in the complex, the other topological tables in the matrix in figure 3 must be constructed. An efficient algorithm has been developed to derive the dependent tables form the specified independent tables as follows. Because any object of class *Edge*, *Face* or *Cell* refers only to the objects describing domains of the next lower rank, the 12 tables can be constructed in four nested loops. The algorithm, which is used to add an element to a map, must automatically suppress multiple entries of the same object. The following operations are performed in the loops:

1. Loop over the cells *c* of the complex.
2. Loop over the faces *f* of cell *c*.
– add face *f* to the cell-face map with key *c*
– add cell c to the face-cell map with key *f*
3. Loop over the edges *e* of face *f* of cell *c*
– add edge *e* to the cell-edge map with key *c*
– add edge *e* to the face-edge map with key *f*
– add face *f* to the edge-face map with key *e*
– add cell *c* to the edge-cell map with key *e*
4. Loop over the nodes *n* of edge *e* of face *f* of cell *c*
– add node *n* to the cell-node map with key *c*
– add node *n* to the face-node map with key *f*
– add node *n* to the edge-node map with key *e*
– add cell *c* to the node-cell map with key *n*
– add face *f* to the node-face map with key *n*
– add edge *e* to the node-edge map with key *n*

The innermost loop 4 can be avoided by arranging the six add-operations in a method with node *n* as parameter, and invoking the method twice in loop 3 to treat the two nodes of the current edge. The outer loop over the cells constructs the cell-face and the face-cell tables. The first nested loop over the faces constructs the cell-edge, face-edge, edge-cell and edge-face tables. The second nested loop constructs the remaining tables. As an example, the topological tables for the unit cube in figure 4 are presented in tables 1 to 4.

The complexity of the table construction algorithm is determined by counting the number of add-operations for the tables:

– Loop 1 is performed N_c times, where N_c is the number of cells in the complex
– Loop 2 is performed N_f times per cycle of loop 1, where N_f is the average number of faces per cell. The total number of traversals of loop 2 is $N_c{\cdot}N_f$. Two domains are added per cycle.
– Loop 3 is performed N_e times per cycle of loop 2, where N_e is the average number of edges per face. The total number of traversals of loop 3 is $N_c{\cdot}N_f{\cdot}N_e$. Four domains are added per cycle.
– Loop 4 is performed twice per cycle of loop 3, where 2 is the number of nodes per edge. The total number of traversals of loop 4 is $2{\cdot}N_c{\cdot}N_f{\cdot}N_e$. Six domains are added per cycle.

Table 1. Node-edge, Node-face and Node-cell Tables

node	edges	faces	cells
n1	e1, e4, e9	f1, f4, f5	c1
n2	e1, e2, e10	f1, f2, f5	c1
n3	e2, e3, e11	f2, f3, f5	c1
n4	e3, e4, e12	f3, f4, f5	c1
n5	e5, e8, e9	f1, f4, f6	c1
n6	e5, e6, e10	f1, f2, f6	c1
n7	e6, e7, e11	f2, f3, f6	c1
n8	e7, e8, e12	f3, f4, f6	c1

Table 2. Edge-node, edge-face and edge-cell tables

edge	nodes	faces	cells
e1	n1, n2	f1, f5	c1
e2	n2, n3	f2, f5	c1
e3	n3, n4	f3, f5	c1
e4	n1, n4	f4, f5	c1
e5	n5, n6	f1, f6	c1
e6	n6, n7	f2, f6	c1
e7	n7, n8	f3, f6	c1
e8	n5, n8	f4, f6	c1
e9	n1, n5	f1, f4	c1
e10	n2, n6	f1, f2	c1
e11	n3, n7	f2, f3	c1
e12	n4, n8	f3, f4	c1

Table 3. Face-node, face-edge, face-cell tables

face	nodes	edges	cells
f1	n1, n2, n5, n6	e1, e5, e9, e10	c1
f2	n2, n3, n6, n7	e2, e6, e10, e11	c1
f3	n3, n4, n7, n8	e3, e7, e11, e12	c1
f4	n1, n4, n5, n8	e4, e8, e9, e12	c1
f5	n1, n2, n3, n4	e1, e2, e3, e4	c1
f6	n5, n6, n7, n8	e5, e6, e7, e8	c1

Table 4. Cell-node, cell-edge, cell-face tables

cell	nodes	edges	faces
c1	n1, n2, n3, n4, n5, n6, n7, n8	e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12	f1, f2, f3, f4, f5, f6

The total number N_t of add-operations for the construction of the topological tables is:

$$N_t = 2N_cN_f + 4N_cN_fN_e + 12N_cN_fN_e \approx \approx 16N_cN_fN_e \quad (1)$$

Usually the average number of cells per face N_f and the average number of edges per face N_e are independent of the size of a complex. The complexity of the algorithm then is O(N_c), which is highly efficient.

MODIFICATION OF LINEAR COMPLEXES

Engineering design proceeds in cycles of work steps, during which a complex changes continuously. Three types of modification occur in a design cycle:

– addition of new domains
– removal of old domains
– modification of attributes of old domains.

The identification of the old values of type D_2, which must be removed in the topological tables due to a modification, requires extensive searches. It is more efficient to group the commands of a modification and to construct new tables for the modified complex defined by the command group. This concept is followed in the project.

An additional command type *remove* nameset is defined for the removal of domains. The entries for the objects in the set are removed from the base tables. The existing command types *node, edge, face* and *cell* are used to add new domains to the base tables and to construct their persistent objects as before. The same command types are used to modify the attributes of old domains whose entries in the base tables already exist. The attributes of these domains are modified in their persistent objects. When command *do* is executed, the modified base tables are used as input stream for the construction algorithm described in section 4. The complexity for a modification group equals the complexity of the construction of the initial topological tables.

Figure 6 shows a modified unit cube. Figure 7 shows the command group for the modification of the unit cube. Node n_6 is removed. Edges e_5, e_6 and e_{10} as well as faces f_1, f_2 and f_6 are modified. Nodes n_9 to n_{15}, edges e_{13} to e_{21} and faces f_7 to f_9 are new.

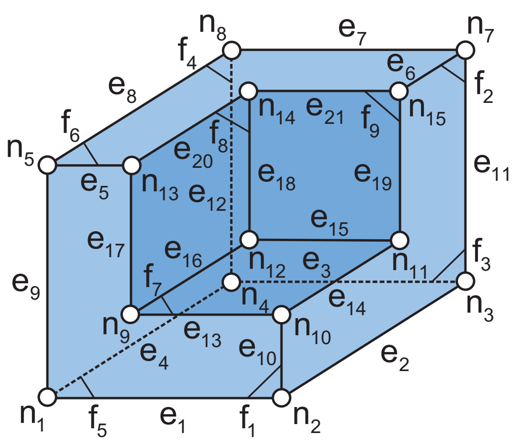

Figure 6. Perspective of the modified unit cube

Cell c_1 is modified. The constructed tables are similar to those shown for the unit cube in tables 1 to 4.

remove node n6

edge	e5	(n5, n13)
edge	e6	(n7, n15
edge	e10	(n2, n10)
face	f1	(e1, e10, e13, e17, e5, e9)
face	f2	(e2, e11, e6, e19, e14, e10
face	f6	(e5, e20, e21, e6, e7, e8)
node	n9	(0.5, 0.5, 0.0)
node	n10	(0.5, 1.0, 0.0)
node	n11	(0.5, 1.0, 0.5)
node	n12	(0.5, 0.5, 0.5)
node	n13	(1.0, 0.5, 0.0)
node	n14	(1.0, 0.5, 0.5)
node	n15	(1.0, 1.0, 0.5)
edge	e13	(n9, n10)
edge	e14	(n10, n11)
edge	e15	(n11, n12)
edge	e16	(n12, n9)
edge	e17	(n9, n13)
edge	e18	(n12, n14)
edge	e19	(n11, n15)
edge	e20	(n13, n14)
edge	e21	(n14, n15)
face	f7	(e13, e14, e15, e16)
face	f8	(e16, e17, e18, e20)
face	f9	(e15, e18, e19, e21)
cell	c1	(f1, f2, f3, f4, f5, f6, f7, f8, f9)

do

Figure 7. Commands for the modification

CONCLUSIONS

The investigation has shown that topological tables can be constructed efficiently for linear complexes representing entire buildings. The necessity to combine the individual topology of a large set of standardized building components in the model is thus eliminated. The complexity of the table construction algorithm developed in the project is linear in the number of cells and thus very efficient. Topological tables for modified complexes are constructed with the same algorithm as the tables for the initial complex.

Topological tables do not make all of the properties of an original explicit in the model.

The closed polygonal curves of the boundaries of faces and the closed polyhedral surfaces of the boundaries of cells are not specified explicitly in the tables. Faces and cells therefore cannot be oriented in the tables. As a result, topological tables do not show explicitly whether the bounded or the unbounded area defined by a closed curve in a plane is the interior of the face. Similarly, the tables do not show explicitly whether the bounded or the unbounded volume defined by a closed surface is the interior of a cell. Due to these deficiencies, it is not possible to differentiate explicitly between simply and multiply connected faces and cells of complexes. Other topics for further research are the robustness of the algorithms for complexes, unbounded domains for complete models of the environment of buildings and handling of concave domains without the necessity for convex triangulation. These topics are being investigated in an associated research project.

ACKNOWLEDGEMENTS

The research is funded by the Russian Foundation for Basic Research (RFBR) – Project Number 20-57-12006.

REFERENCES

1. **Borrmann A., Beetz J., Koch C., Liebich T., Muhic S.** Section 5.7. Geometric Representations. In: Borrmann A., König M., Koch C., Beetz J., eds. Building Information Modelling. Cham: Springer; 2018, pp. 101-111, DOI: 10.1007/978-3-319-92862-3_5
2. **van Berlo L., Krijnen T., Tauscher H., Liebich T.** Future of the Industry Foundation Classes: towards IFC 5, in: Proc. of the 38th International Conference of CIB W78, 2021, pp. 123-137
3. **Huhnt W., Galishnikova V.** Partitioning of space as basis for data structures to describe digital building models, in: Proc. of the 17th International Conference on Computing in Civil and Building Engineering, 2018, pp. 42-49
4. **Huhnt W.** Reconstruction of edges in digital building models. // Advanced Engineering Informatics, 2018, V. 38, pp. 474-487. DOI: 10.1016/j.aei.2018.08.004
5. **Huhnt W., Hartmann T., Suter G.** Space classification from point clouds of indoor environments based on reconstructed topology. In: Smith I., Domer B., eds. Advanced Computing Strategies for Engineering, 25th EG-ICE International Workshop. Cham: Springer; 2018, V. 10863, pp. 82-102. DOI: 10.1007/978-3-319-91635-4_5
6. **Kraft B., Huhnt W.** Geometrically Complete Building Models // Proc. of the 21th International Workshop of the European Group for Intelligent Computing in Engineering, 2014, pp. 1-11
7. **Belsky M., Sacks R., Brilakis I.** Semantic Enrichment for Building Information Modeling // Computer-Aided Civil and Infrastructure Engineering, 2016, V. 31, No. 4, pp. 261-274. DOI: 10.1111/mice.12128
8. **Steinmann R.** IFC Certification 3.0: Specifications and Terms and Conditions. BuildingSMART International Ltd., UK; 2018. URL https://www.buildingsmart.org
9. National BIM Standard-United States Version 2 (NBIMS-US 2012). The National Institute of Building Sciences
10. **Hietanen J., Find S.** IFC model view definition format, in: Proceedings of the 24th International Conference of CIB W78, 2007, pp. 1-29
11. **Bloch T., Sacks R.** Clustering Information Types for Semantic Enrichment of Building Information Models to Support Automated Code Compliance Checking // J. Comput. Civ. Eng., 2020, V. 34, No. 6, 04020040. DOI: 10.1061/(ASCE)CP.1943-5487.0000922

СПИСОК ЛИТЕРАТУРЫ

1. **Borrmann A., Beetz J., Koch C., Liebich T., Muhic S.** Section 5.7. Geometric Representations. In: Borrmann A., König M., Koch C., Beetz J., eds. Building Information Modelling. Cham: Springer; 2018, pp. 101-111. DOI: 10.1007/978-3-319-92862-3_5
2. **van Berlo L., Krijnen T., Tauscher H., Liebich T.** Future of the Industry Foundation Classes: towards IFC 5, in: Proc. of the 38th International Conference of CIB W78, 2021, pp. 123-137
3. **Huhnt W., Galishnikova V.** Partitioning of space as basis for data structures to describe digital building models, in: Proc. of the 17th International Conference on Computing in Civil and Building Engineering, 2018, pp. 42-49
4. **Huhnt W.** Reconstruction of edges in digital building models. // Advanced Engineering Informatics, 2018, V. 38, pp. 474-487. DOI: 10.1016/j.aei.2018.08.004
5. **Huhnt W., Hartmann T., Suter G.** Space classification from point clouds of indoor environments based on reconstructed topology. In: Smith I., Domer B., eds. Advanced Computing Strategies for Engineering, 25th EG-ICE International Workshop. Cham: Springer; 2018, V. 10863, pp. 82-102. DOI: 10.1007/978-3-319-91635-4_5
6. **Kraft B., Huhnt W.** Geometrically Complete Building Models // Proc. of the 21th International Workshop of the European Group for Intelligent Computing in Engineering, 2014. pp. 1-11
7. **Belsky M., Sacks R., Brilakis I.** Semantic Enrichment for Building Information Modeling // Computer-Aided Civil and Infrastructure Engineering, 2016, V. 31, No. 4, pp. 261-274. DOI: 10.1111/mice.12128
8. **Steinmann R.** IFC Certification 3.0: Specifications and Terms and Conditions. BuildingSMART International Ltd., UK; 2018. URL https://www.buildingsmart.org
9. National BIM Standard-United States Version 2 (NBIMS-US 2012). The National Institute of Building Sciences
10. **Hietanen J., Find S.** IFC model view definition format, in: Proceedings of the 24th International Conference of CIB W78, 2007, pp. 1-29
11. **Bloch T., Sacks R.** Clustering Information Types for Semantic Enrichment of Building Information Models to Support Automated Code Compliance Checking // J. Comput. Civ. Eng., 2020, V. 34, No. 6, 04020040. DOI: 10.1061/(ASCE)CP.1943-5487.0000922

Aleksandr N. Rozhkov, Senior Lecturer at the Department of Fundamental Education, branch of National Research Moscow State University of Civil Engineering in Mytishchi, 129337, Russia, Moscow, Yaroslavskoe Shosse, 26; assistant of the Department of Civil Engineering of the Engineering Academy of the People's Friendship University of Russia, 117198, Russia, Moscow, st. Miklukho-Maclay, 6; phone: +7(925)023-83-76. e-mail: rozhkovalex@hotmail.com.

Vera V. Galishnikova, Professor, Dr.Sc.; Vice-Rector of the National Research Moscow State University of Civil Engineering, 129337, Russia, Moscow, Yaroslavskoe Shosse, 26; Professor of the Department of Civil Engineering of the Engineering Academy of the People's Friendship University of Russia, 117198, Russia, Moscow, st. Miklukho-Maclay, 6; phone: +7 (495) 287-49-14. e-mail: galishni@yandex.ru.

Рожков Александр Николаевич, старший преподаватель кафедры Фундаментального образования филиала НИУ МГСУ в г. Мытищи, 129337, Россия, г. Москва, Ярославское шоссе, д. 26; ассистент департамента строительства инженерной академии Российского университета дружбы народов, 117198, Россия, г. Москва, ул. Миклухо-Маклая, 6; тел. +7(925) 023-83-76. e-mail: rozhkovalex@hotmail.com.

Галишникова Вера Владимировна, профессор, доктор технических наук, проректор Национального исследовательского Московского государственного строительного университета, 129337, Россия, г. Москва, Ярославское шоссе, д. 26; профессор департамента строительства инженерной академии Российского университета дружбы народов, 117198, Россия, г. Москва, ул. Миклухо-Маклая, 6; тел. +7 (495) 287-49-14. e-mail: galishni@yandex.ru.