Hybrid Modeling Based on the Implicit Complexes

Elena Kartasheva[α], Valery Adzhiev[β], Peter Comninos[β], Oleg Fryazinov[β], Alexander Pasko[β]
(α) Moscow Institute for Mathematical Modeling, Russian Academy of Science, Moscow, Russia
(β) The National Centre for Computer Animation, Bournemouth University, Poole, UK
ekart2005@gmail.com, {vadzhiev, peterc, ofryazinov, apasko}@bournemouth.ac.uk

Abstract

We present a novel approach for modeling heterogeneous objects containing entities of various dimensions and representations within a cellular-functional framework based on the implicit complex notion. We provide a brief formal description for implicit complexes and describe their structure; in particular we describe both the geometry and topology of cells of different types. Then we consider the necessary basic operations over IC's components and the set-theoretic operations over objects represented by ICs. Finally, we present a case-study showing how to construct a hybrid model integrating both boundary and function representations.

***Keywords: Computational geometry, object modeling, implicit complexes, hybrid modeling, object representations*.**

1. INTRODUCTION

Heterogeneous objects modelling is becoming an important research topic in different application areas such as volume modelling and rendering, modelling of objects with multiple and varying materials in CAD and in rapid prototyping [6]. Such objects that can represent the results of physical simulations, geological and medical modelling, etc. are heterogeneous from the points of view of their internal structure and their dimensionality. This paper presents our recent advances in developing a novel framework for heterogeneous object modelling which is based on the concept of an Implicit Complex (IC) that was first introduced in [1]. Within this framework, we proposed a way of constructing a hybrid model which is not supposed to serve just for the combined usage of separate representations but is genuinely unified. In particular, we tackle the very practical problem of the integration of the boundary (BRep) and function (FRep) representations - along with a cellular representation - into a unified model thus keeping all the advantages that these particular representations have and supporting descriptions of different material and other non-geometric properties.

The IC describes composite heterogeneous objects consisting of several components that can differ in their dimensionality, geometric representation and non-geometric attributes. The IC-based hybrid framework provides a unified description of a heterogeneous object's geometry and topology. An object is described as the union of cells of various representation types and dimensionalities along with the relations between them. The main relations characterizing mutual locations of cells are the boundary and the "to contain" relations. Non-geometric attributes are independently described by functional or cellular models and are associated with IC's cells by means of relations provided that some constraints are satisfied.

In our previous papers [1], [8], we introduced an IC's general structure and described some basic procedures of IC-based model construction along with suitable discretization methods. In this paper, we consider the application of the set-theoretic operations over heterogeneous objects within the IC framework. Our particular focus in this paper is on the construction of hybrid models unifying components represented by BReps and FReps.

The paper structure is as follows. Section 2 reviews some related works concerning the modeling of heterogeneous objects. We give an outline of a formal IC framework in Section 3. The necessary basic operations over IC's components are considered in Section 4 and the set-theoretic operations are described in Section 5. Section 6 presents some examples.

2. RELATED WORKS

A typical technique for describing heterogeneous objects is to represent them as collections of homogeneous components. Both topological subdivisions and constructive procedural methods can be used to combine such components. The former include geometric complexes [13], CW-complexes [9], selective Nef complexes [5], and others. The latter are Structured Topological Complexes (STC) [12], the constructive hypervolume model [11], and the hybrid constructive trees [2],[3].

In the STC framework [12], a composite object is defined using a combination of layers each of which is described by a geometric complex, which is homogeneous with respect to the representations of the components. The hypervolume model supports uniform constructive modelling of point set geometry and attributes using real functions of point coordinates. The HybridTree [2] is a constructive tree with leaves defined by a number of representations (implicit surfaces, polygonal meshes, and discrete point sets). Both the function evaluation and the surface mesh generation are provided for modelled objects. Depending on the user's query, corresponding conversions between representations are applied. A hybrid constructive tree in [3] has leaves with both implicit and parametric representations. To polygonize the surface of a complex object, surface meshes of primitives are classified against the subtree defining function, trimmed, and merged into the resulting mesh. However, both these approaches do not support heterogeneous objects with components of different dimensionalities and do not provide a description of the topological structure of the object being modelled.

In [1] we introduced a hybrid cellular-functional model based on the notion of an IC, which provides a valid topological description of heterogeneous objects and allows for the flexible combination of cellular and functional representations of the geometry of objects and their attributes. In [8], we showed how this framework can be exploited for representing some heterogeneous models without using set-theoretic operations. In this paper, we describe an extended structure of an IC and then concentrate on the constructing the ICs, especially using set-theoretic operations. Set-theoretic operations on polyhedral topological complexes were previously discussed in [5]. Here we consider the specifics of these operations caused by the heterogeneous structure of ICs containing BRep and FRep components.

3. IMPLICIT COMPLEXES

In this section we provide a brief description of the theoretical framework that is based on the Implicit Complex (IC) notion.

3.1 IC basic definition

We consider a hybrid model of a geometric object $D \subset E^3$ defined in Euclidian modelling space E^3 as the union of cells $g_i^{q_i}$ under the following conditions:

1) Each cell $g_i^{q_i}$ is a closed point set $g_i^{q_i} \subset E^3$ unambiguously described by some known geometric representation which provides geometrically and topologically correct discretization of the cell. Here *i* denotes the index number of cell and q_i is its dimension.

2) The boundary $\partial g_i^{q_i}$ of each cell $g_i^{q_i}$ is the union of a finite number of cells of lesser dimensions;

3) Cells can overlap each other but the intersection of any two cells is either the union of a finite number of cells or is empty. Note that we call the cells satisfying conditions 2 and 3 as properly joined cells.

A collection K of cells satisfying the above conditions is called *an implicit complex* (i.e.: $K = \{g_i^{q_i}\}_{i=1}^{N}$). The point set union of all cells of an IC *K* is denoted by |*K*| and called a *carrier* of *K*. Thus formally, the *hybrid representation* for a geometric object *D* is defined as $D = |K|$. The dimension of an IC is the maximal dimension of its cells. In accordance with the IC definition, *polyhedral*, *cellular* and CW complexes are also represented in the IC framework.

The ICs allow us to describe objects of different dimensionalities consisting of components represented by various geometric models. The above conditions of the IC definition actually ensure the ability to convert an arbitrary IC *K* into a polyhedral complex, which approximates, geometrically and topologically, the object D being modeled. In fact, the reducibility of each of the used representations to a polyhedral one guarantees a correct execution of any operation over objects described by various representations. However, we strive to exploit advantages of different types of representations; that is why we keep the initial representations for components of the model and use meshes only for the implementation of various numerical procedures applied in topology analysis, computational geometry and finite element analysis.

The support of overlapping cells allows for inserting components of a composite object into its IC model without any subdivision. To satisfy the IC definition, extra cells describing mutual intersections of the components are added to the complex. This allows for the preservation of the initial representations of components, which is useful for heterogeneous object modeling.

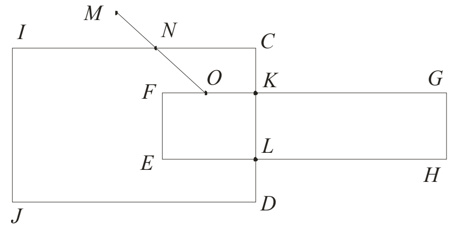


Figure 1: A 2D implicit complex.

Fig. 1 shows an example of an IC based 2D model. The model describes an object consisting of two 2D components (rectangles *DCIJ* and *EFGH*) and one 1D component (segment *OM*). We assume that the components are represented by different methods and have different non-geometrical properties (attributes). One of the possible IC representations of this heterogeneous model is described by the following complex *K* consisting of three 2D cells: rectangles $g_{DCIJ}^2, g_{EFKL}^2, g_{EFGH}^2$; seven 1D cells including two closed polylines g_{DCIJ}^1, g_{EFGH}^1, one non-closed polyline g_{KFEL}^1, three segments $g_{KL}^1, g_{OM}^1, g_{ON}^1$; and five 0D cell points $g_K^0, g_L^0, g_O^0, g_N^0, g_M^0$. According to the IC definition, the complex *K* includes cells describing the initial components ($g_{DCIJ}^2, g_{EFGH}^2, g_{OM}^1$), their boundaries ($g_{DCIJ}^1, g_{EFGH}^1, g_O^0, g_M^0$), and cells representing the mutual intersections between the listed initial ones. Thus the cells g_{DCIJ}^2, g_{EFGH}^2 preserve the initial representations of the corresponding components and the cell g_{EFKL}^2 describes their intersection.

We also introduce the concept of a subcomplex, which is a means of aggregation inside the IC. Formally, an IC *L*, is called a *subcomplex* of IC *K*, if the collection of cells of IC *L* is a subset of the collection of cells of IC *K*. For example, the following cell collections define the subcomplexes of IC shown in Fig.1: $L_1 = \{g_{DCIJ}^2, g_{EFKL}^2, g_{DCIJ}^1, g_{EFKL}^1, g_{KFEL}^1, g_{KL}^1, g_K^0, g_L^0\}$, $L_2 = \{g_{EFGH}^2, g_{EFGH}^1\}$, $L_3 = \{g_{DCIJ}^2, g_{DCIJ}^1\}$

We call two implicit complexes properly joined if their cells altogether satisfy conditions 1 and 2 of the IC definition. Thus, the subcomplexes L_2 and L_1 of the IC shown in Fig.1 are properly joined but L_3 and L_2 are not.

An IC provides a consistent description of both the geometry and the topology of a modeled object. Its geometry is represented by the geometry of the individual cells and its topology is described by means of the relations between cells.

3.2 The IC Topology

The general structure of a 3D IC is illustrated by Fig. 2. By definition, a 3D IC consists of 0D, 1D, 2D, and 3D cells.

Let G^p be a set of p-dimensional cells g_i^p. There are two main types of relations that establish connections between cells of different dimensionalities: the boundary relation and the "to contain" relation. According to the conditions of the IC definition the mutual disposition of any of the IC cells can be evaluated through queries to its main relations.

We denote by Rb^{ps} the boundary relation between p-dimensional and s-dimensional cells, $Rb^{ps} \subset G^p \times G^s$, $s < p$. The pair (g_i^p, g_j^s) belongs to Rb^{ps} if g_j^s belongs to the boundary of g_i^p and does not lie in the interior of any other boundary cell of g_i^p. The relation "to contain" is denoted by Rc^{ps}, $Rc^{ps} \subset G^p \times G^s$, $s \le p$. The pair (g_i^p, g_j^s) belongs to Rc^{ps} if $g_j^s \in g_i^p$ and $g_j^s \cap \partial g_i^p \ne g_j^s$. The entire structure of 3D IC is defined by six different boundary relations and nine different "to contain" relations. Other relations are the *co-boundary*, the "*to be contained*", the *incidence* and the *adjacency* relations. These can be derived from the boundary and the "to contain" ones using various operations on relations.

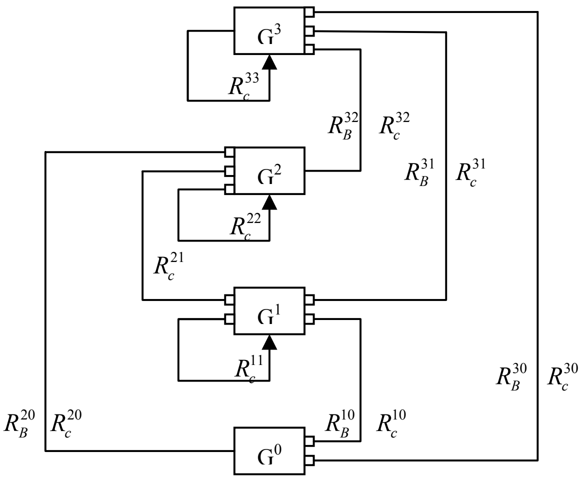


Figure 2: The general structure of a 3D IC.

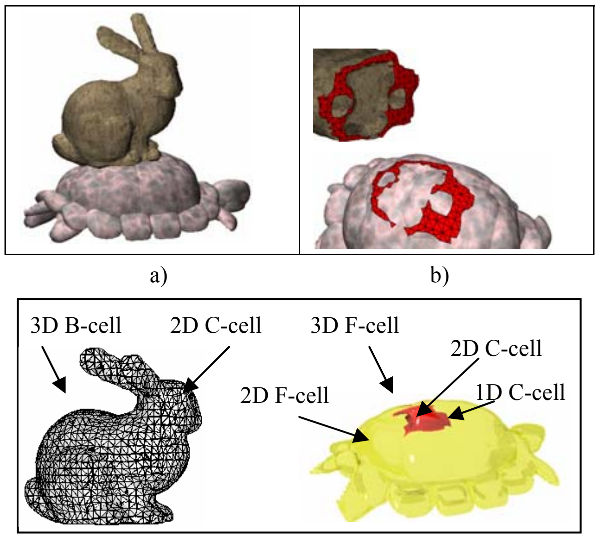
Thus the description of an implicit complex K consists of the collection $G = \{g_i^{q_i}\}_{i=1}^N$ of cells and the sets of boundary Rb relations and "to contain" Rc relations, i.e. $K = < G, Rb, Rc >$.

3.3 The IC Geometry

According to the IC definition, a variety of representations can be used for the description of cell shapes, but all of these representations should guarantee a conversion into the mesh described by a polyhedral complex.

We have introduced and implemented the following five types of IC cells that differ in their geometric representations but are topologically uniformly related to each other. This set of types can be extended in the future and currently includes:

- The P-cell, which is an explicit cell representing a simple polyhedron of an appropriate dimensionality.
- The B-cell, which is a cell representing a manifold defined by its boundary. B-cells describe segments of parametric curves, patches of parametric surfaces and boundaries defining 3D solids A 1D (2D) B-cell is defined by its supporting curve (surface) and by its oriented boundary. A 3D B-cell is defined by its oriented boundary only. In the general case the boundary of a B-cell can consist of cells of all the other types supported in the IC framework.
- The F-cell, which is an implicit cell described by the FRep that is a constructive representation by real-valued functions in the form of an inequality $F(X) \ge 0$ [10]. We restrict a valid variety of 2D and 1D FRep objects by *s*-dimensional F-cells (*s*<3) to those which are represented as subsets of the boundaries of 3D manifolds. Thus a 2D F-cell is described by a pair (F, F_M) and a 1D F-cell is described by triple (F, F_{M1}, F_{M2}) where F, F_M, F_{M1}, F_{M2} are real-valued functions of point coordinates, such that function F describes the point set of the corresponding cell and other functions are used to describe the underlying 3D manifolds in the form $F_{Mi}(X) \ge 0$.
- The C-cell, which is a composite cell aggregating cells of various types. Each C-cell is defined as the carrier of an implicit complex *T* differing from the complex *K* containing this C-cell. The complex *T* can consist of the cells of all types supported in the IC framework. The complex *T* is not a subcomplex of *K*. Its cells are not properly joined with respect of the cells of *K*.
- The T-cell which is a cell described by a constructive tree. Its leaves represent objects described by cells of all the other types. The tree nodes represent operations admissible for the IC – in particular, some bijective geometric transformations, non-regularized set-theoretic operations and trimming by 3D manifold. The T-cells allow for the description of the results of applying set-theoretic operations to cells of different types without the need for converting representations of one type into another.



c)

Figure 3: The unified hybrid model combining an FRep Turtle (courtesy of G. Pasko) and a BRep Bunny (Stanford 3D Scanning Repository) a) The general view of the model b) The contact zone c) The structure of the IC

Various types of cells are illustrated by Fig. 3 showing an example of the IC describing a hybrid model combining FRep and BRep components. This IC consists of two 3D cells and three 2D cells (see Fig. 3c). The 3D B-cell describes the bunny's body and the 2D C-cell represents its boundary defined by a triangular mesh. The turtle's body and its boundary are represented by the 3D F-cell and the 2D F-cell, correspondingly. Finally, the intersection of the bunny and the turtle called a *contact zone* is described by the 2D C-cell. The contact zone is illustrated by Fig. 3b.

3.4 The Attribute Model

In this section we briefly describe a cellular-functional representation of attributes associated with an IC. Each attribute Λ is described by a set N_Λ of its values embedded into a multidimensional real number space $\Re^{m_\Lambda}$ of a proper dimension m_Λ. For the sake of uniformity, we assume that the value set N_Λ of each attribute is supplemented with a special "empty" value θ and so all the attributes are defined at each point of the modeling space Ω. The attribute values are assigned to geometric object points described by the IC K using a collection of *attribute functions* $S_\Lambda = \{S_{\Lambda j}\}_{j=1,J}$ and a set of *attribute relations* $R_\Lambda = \{Rs^3_\Lambda, Rs^2_\Lambda, Rs^1_\Lambda, Rs^0_\Lambda\}$. So an attribute Λ of an IC K is represented as $\Lambda = < S_\Lambda, Rs_\Lambda >$. Each function $S_{\Lambda j}$ of an attribute Λ maps the modeling space Ω into the attribute value set N_Λ, $S_{\Lambda j} : \Omega \to N_\Lambda$. Attribute functions can be analytic, piecewise analytic or be defined by an interpolation methods [1].

The relations Rs^p_Λ (p=0,1,2,3) associate functions of an attribute Λ with cells of the IC K, that is $Rs^p_\Lambda \subset G^p \times S_\Lambda$, where G^p is a set of all p-dimensional cells g^p_i of the complex K. If $(g^p_i, S_{\Lambda j}) \in Rs^p_\Lambda$ then the value of the attribute Λ at any point $X \in | g^p_i |$ is defined as $S_{\Lambda j}(X)$.

Only one function of each attribute can be associated with a cell. Taking into account that IC cells can overlap each other, we propose *priority* and *additive schemes* for calculating the value of an attribute Λ at an arbitrary point X of the object represented by the IC *K*. According to the *priority scheme*, we look through all the cells associated with the attribute Λ and containing the point X and select one cell of the lowest dimension which does not contain other cells associated with Λ within it. The value of the attribute function defined on that cell is used for calculating the attribute value at the given point. According to the *additive scheme* the value of the attribute Λ at the point X is calculated as a blend of the attribute functions associated with all cells containing the point X.

3.5 Implementation Model

We have implemented software for cellular-functional modeling of heterogeneous objects within an object-orientated framework. Let us outline the principal classes which are directly derived from the presented theoretical description.

The basic *IComplex* class represents an implicit complex data structure (Fig.2 is an illustration). Its attributes represent six boundary relations and nine "to contain" relations as well as cells of various dimensionalities. The methods of the *IComplex* class realized the operations over cells of the complex (see Section 4) as well as operations over ICs (see Section 5). The *IComplex* class includes operations for modifying the relations as well as inquiry operations on the relations.

Each relation is described by the object of the *Relation* class which contains all the pairs of numbers of related cells. The operations of the *Relation* class allow us to get the indices of all the related cells as well as to add and delete pairs of cells.

The IC geometry within the *IComplex* class is specified using objects of classes inherited from the abstract *Shape* class that contains virtual operations for defining the point membership as well as for rendering and discretization.

4. THE BASIC OPERATIONS IN THE IC FRAMEWORK

Here we introduce basic operations over cells, subcomplexes and entire implicit complexes. These operations are especially important in the context of constructing and manipulating implicit complexes. For each operation, there are constraints on input data that should be checked before their actual evaluating.

Cell_adding adds a new cell to IC. Cell g^r can be added to IC K if g^r is properly joined to all the cells of *K* and the boundary of g^r is represented as the union of cells of *K*. The input data of the procedure include the cell geometry description and the list of cells of K related to the cell being added.

Attribute_adding defines an attribute on *K.* The input data include a collection of attribute functions and the list of cells of K being associated with this attribute.

Cell_removing deletes a cell from IC. It is follows from the IC definition that cell g^r_c can not be deleted if one of the following conditions or both of them are satisfied

1) Cell g^r_c has co-boundary cells in K.

2) Cell g^r_c represents the intersection of some other cells of K.

These conditions are checked automatically using IC relations.

Cell_cutting removes a cell with all its boundary cells from IC. It is implemented under the same restrictions as the *Cell_removing* operation.

IC_adding implements sum operations over properly joined complexes. Given any two properly joined implicit complexes C and T, the sum of C and T is a complex M consisting of all the cells of the complexes C and T, this is denoted as $M = C \oplus T$. The boundary relations and the relations "to contain" of M are automatically formed on the base of the same relations defined on the complexes *T*, *C*. Attributes are established on the IC *M* by combining those ones defined on the initial complexes *C* and *T* using an appropriate mixing function. Suppose an attribute Λ is described on both the ICs *C* and *T*. To set this attribute of *M* we look through all the cells of *M,* and for each g^r_m of *M* we find its equivalent cells in the complexes *C* and *T*. If g^r_m has only one equivalent cell then it inherits an attribute function from this cell. Otherwise, the attribute function is calculated by blending the attribute functions defined on the cells equivalent to g^r_m.

IC_transform implements *bijective geometric transformations* over ICs including affine transformation and nonlinear bijective transformations.

The *IC_meshing* procedure implements the conversion of the IC representation into the simplicial one. The discretization of IC models are guaranteed by the IC definition according to which the appropriate mesh generation methods have to be available for all types of the IC cells. The discretization of an implicit complex *K* is implemented as an iterative process. The mesh generation of each kind of cell is implemented using specific meshing algorithms. An extensive survey of discretization methods are presented in the book [4]. We subdivide IC cells in the order of their increasing dimensionality. Among the cells of the same dimensionality we first subdivide those ones which do not contain other yet unprocessed cells. Thus, at the moment of the meshing of a cell we already know the discretization of its boundary and the subdivision of all the cells lying inside the considered one. Then we subdivide the cell into mesh elements such that they are compatible with other meshes which already belong to it. The corresponding incremental mesh generation approaches that allow for preserving existing mesh elements can be found in works [2], [7] and in the references contained in these works.

5. THE SET-THEORETIC OPERATIONS

Set-theoretic operations are the main mechanism for constructing composite geometric objects starting from more primitive ones. In the IC framework, we introduce the union and intersection operations over ICs with attributes. Given two implicit complexes *A* and *B*, the *intersection* of *A* and *B* is an implicit complex *C* whose carrier is equal to the set-theoretic intersection of the carriers of *A* and $B, C = A \cap B$. The *union* of two implicit complexes is a complex whose carrier is equal to the set-theoretic union of the initial complexes. Thus we consider the non-regularized set-theoretic operations. The *difference* operation is more problematic as a mere set-theoretic difference of the carriers results in non-closed objects. So we consider the restricted version of the difference operation, namely, *trimming* with a 3D manifold. Let the IC *B* represent a 3D manifold; then the result of trimming the IC *A* by the complex *B* is a complex *C* whose carrier is equal to the set-theoretic intersection between the carrier *A* and the point set described as the inversed carrier of *B* (that is the cavity in the whole solid space). We denote this operation as $C = A - B$.

We propose four procedures for the implementation of the set-theoretic operations. Each of them takes two input ICs *A* and *B* and returns the IC *C*. The intersection and the trimming operations are implemented as the procedures *IC_intersection* and *IC_trimming*, correspondingly. There are two procedures implementing the union operation, namely, *IC_union* and *IC_subtractive_union*. The procedure *IC_union* returns the IC *C* involving all the cells of the input ICs *A* and *B* and is realized as $C = A \oplus (A \cap B) \oplus B$. An example of this procedure is shown in Fig.4.

The procedure *IC_subtractive_union* calculates the union of the ICs *A* and *B* according to the formula $C = (A \oplus (B - A))$. Thus the resulting IC *C* does not include those cells of IC *B* which lie inside the carrier of *A*. The example of this procedure is shown in Fig.5.

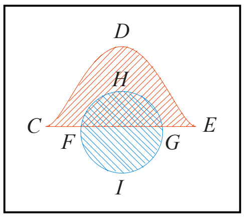


Figure 4: The union of two ICs *A* and *B* is calculated as $C = A \oplus (A \cap B) \oplus B$. The IC *A* represents an area CED and the IC *B* describes disk IGHF.

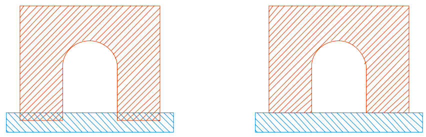


Figure 5: The union of two ICs *A* and *B* is calculated as $C = (A \oplus (B - A))$. The IC *A* represents a rectangle and the IC *B* describes an arch. a) input objects; b) the result of the union

The attributes are associated with the resulting IC *C* through the relations between *C* and the initial complexes *A* and *B*. The attribute functions associated with the cells of *C* belonging to both ICs *A* and *B* are calculated as a blend of the initial attributes functions associated with the same cells on ICs *A* and *B*. Other cells of *C* inherit attributes from their preimages in the initial complexes. For example, let us consider the example in Fig. 4. Here the attribute "hatch" is defined on the initial ICs *A*, B. Then in the resulting IC *C* the cells CDE and IGHF preserve those attribute values that were defined on the initial ICs. As to the attribute value in the cell FGH created as the result of the intersection of the cells CDE and IGHF, it is calculated as a blend of the attribute values defined on those cells.

A detailed description of algorithms for the set-theoretic operations on ICs is beyond the scope of this paper. Here we briefly consider the implementation of the *IC_intersection* operation. Other set-theoretic operations on ICs are implemented on the base of this one. To build the IC $C = A \cap B$, we at first perform the discretization of the initial ICs and then convert them into simplicial complexes. Then we evaluate non-regularized set-theoretic intersection between these simplicial complexes. To do this, one can apply the methods similar to those developed for geometric complexes [13]. After that, we aggregate the simplexes of the resulting complex using the following rules: any two *n*-dimensional cells can be merged only if they have a common boundary, the same co-boundary cells and if they are shared by the same cells in the initial ICs. In all the previously described operations we support information about relationships between complexes. So it is known for each aggregated cell what was an initial cells in the implicit complexes *A* and *B*. Preserving such an information allows us to build boundary, constructive or functional description of the aggregated cells depending on the types of the initial cells whose intersection these aggregated cells represent. Thus we convert aggregated cells into IC cells to get the desired complex *C*.

The *IC_trimming* operation is realized as the intersection of the IC *A* with the IC *B'* describing the boundary representation of the point set -|*B*| which is a cavity in the modeling space. The complex *B'* is formed using the discretization procedure of the IC *B* followed by the boundary evaluation operation producing the boundary description of the cavity.

According to our intersection algorithm, the resulting IC *C* is properly joined to complexes *A* and *B*. This allows us to implement the union of ICs through the sum of properly joined ICs *A, B,* and $(A \cap B)$ (*IC_union)* or through the sum of properly joined ICs *A* and *(B-A)* (*IC_subtractive_union).*

6. CASE STUDIES

Here we present examples of heterogeneous objects that were constructed based on the theory presented above.

6.1 A flower model

This example illustrates how to create the IC describing the hybrid model of the object D, which is sequentially being built of the heterogeneous components d_j using set-theoretic union operation over initial components. The object D also has the non-geometric attributes.

The construction process for IC K which represents the hybrid model of the object D ($D = |K|$) is performed in a component-wise manner, starting from $K_0 = \varnothing$. At each step *j,* the component d_j is represented by the complex *Tj* containing just one cell describing d_j together with one or a few cells of lesser dimensionalities describing the boundary ∂d_j. Complex *Tj* is formed using the *Cell_adding* operation. Then we get IC K_j as the result of the set-theoretic union between the already created IC K_{j-1} and IC *Tj* representing the new component d_j. In our example we use the *IC_subtractive_union* operation for construction $K_j = K_{j-1} \cup T_j$. As the result we get the IC *K*.

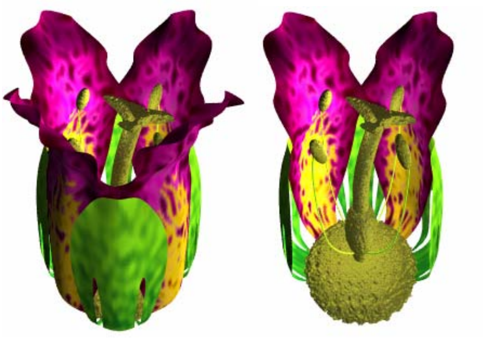


Figure 6: A flower modelled as an IC.

To create a model of such a composite object as a flower (see Fig. 6), we start from modelling its separate components (see Fig. 7).

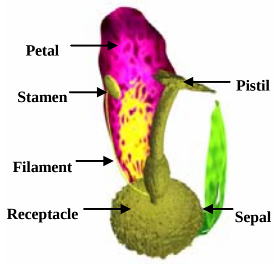


Figure 7. Components of a flower

The receptacle, the pistil and the stamens are defined by the functional representation. The receptacle is described as a half-ellipsoid combined with a solid noise function (algebraic sum with Gardner's noise function). The pistil is defined as the result of the blending union of ellipsoids. Each stamen is defined as an algebraic sum of an ellipsoid and Gardner's solid noise function. ICs *Tj* describing the listed components consist of the 2D and 3D F-cells.

The 2D petals and sepals are described by BRep surface patches which are modeled in two steps. First, an object is described in 2D space as an area bounded by a large ellipse and two smaller ones (representing the holes). Then, tapering and general space mapping deformations are applied to the object. Complexes *Tj* describing the petals consists of the 1D and 2D B-cells.

The filaments are explicitly defined by the spline curve segments defined in 3D space. The corresponding ICs consist of the 1D B-cells.

We consider the flower as a heterogeneous object which has a color property. This is represented by an RGB colour attribute which is described by the function $S = E^3 \rightarrow \Re^3$ in a piece-wise manner, so that $S = \{S_i\}$, where each S_i maps the corresponding subset of E^3 into the RGB space. We use the *Attribute_adding* operation to associate color with cells of ICs *Tj* representing the flower components

Complex K describing the entire heterogeneous model of the flower is calculated automatically as the result of iterative application of the *IC_subtractive_union* operation. Complex K consists of cells corresponding to different flower components and includes auxiliary cells providing the validity of the IC model. These auxiliary cells describe interconnections between the flower components. Figure 8 shows a fragment of the IC representing the flower model. The receptacle is described by the 3D F-cell t_1^3, the pistil by the 3D F-cell t_2^3, the stamen by the 3D F-cell t_0^3, the filament by the 1D B-cell e_0^1, and the petal by the 2D B-cell t_0^2. The auxiliary cells are e_0^0, e_1^0, t_1^2, t_1^1, t_0^1, t_0^0, t_1^0.

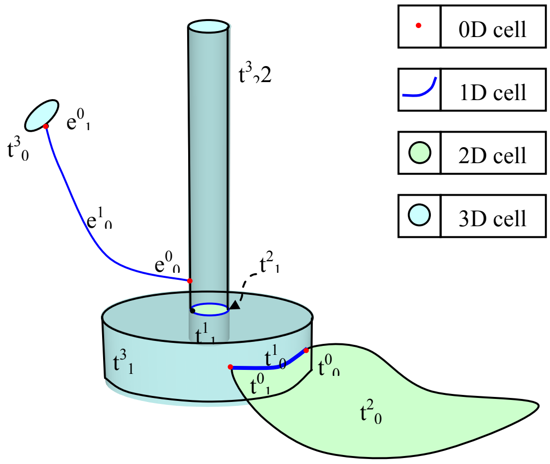


Figure 8: The IC structure for the 'Flower' model.

6.2 A model of a bunny riding a turtle

When we model heterogeneous objects, different sequences forming their components as well as their contact areas are possible. In the previous example, the contact zones (mutual intersections) between components have been calculated automatically (i.e. without user efforts) on the basis of object components data. Let us consider an example where contact zone modelling is crucial as it influences the disposition of heterogeneous model components. Such an application can be called "fitting the object *A* to the object *B*". We suppose that we initially have two separately designed objects *A* and *B*. First, we form a desired contact zone *C*. Then we translate the object *A* and simultaneously deform it to make it possible to intersect with the object *B* along the contact area *C*. To do so, it is necessary to specify the correspondence between the points of *A* and *C*. The deformation itself also needs to be specified. Both physically-based and ad hoc methods can be used here.

As an illustration, let us consider a case study representing the Stanford Bunny riding a turtle (see Fig. 3) where the BRep object *A* defines the bunny model and the FRep object *B* represents the turtle model (see Fig.9a). According to the IC construction procedure, we start with IC representing the initial components. The IC T_1 describing the object *A* consists of one 3D B-cell (the bunny's body) and one 2D C-cell (the triangulated boundary surface of the bunny). The IC T_2 specifying the object *B* consists of one 3D F-cell (the turtle's body) and one 2D F-cell (the turtle's boundary).

The procedure allowing us to attach the bunny to the turtle and to construct the hybrid IC model includes the following steps:

I. Construction of the contact zone. In this example the components have to intersect each other along their boundaries. So the contact zone is a surface patch.

1. Extract the mesh on the horizontal bottom of the bunny (see Fig. 3b, upper left), keeping the correspondence between the extracted part and the main mesh and copying it to a separate mesh – this will be the future contact surface C called "a saddle". The initial mesh selected on the bottom of the bunny is shown in Fig. 3b (left);
2. Select two corresponding local coordinate systems, one on the turtle back and one on the saddle;
3. Using a linear transformation that combines a number of rotations and translation match the saddle's local coordinate system with that of the turtle;
4. Put the saddle on the turtle back by finding projection of all the saddle vertices on to the boundary surface of the turtle (all the vertices are moved in the direction defined by the gradient of the function describing the turtle boundary). The resulting mesh representing the contact zone on the back of the turtle is shown in Fig. 3b (right);
5. Form the IC representation T_3 of the contact zone. It consists of one 2D C-cell combining all the elements of the contact zone mesh and one 1D C-cell specifying the boundary of the mesh.

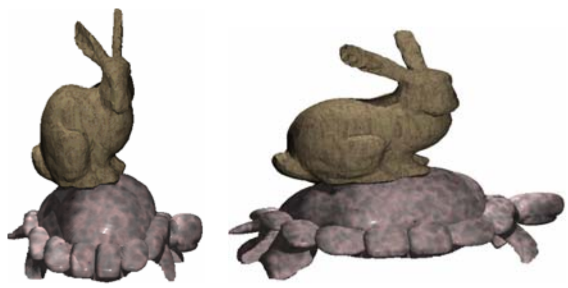
II Deformation of the bunny.

1. Move the bunny to the turtle applying the same transformation as in the step I.3;
2. Using the free form space deformation technique deform the entire bunny so that points of the bunny bottom coincide with the corresponding points of the contact zone.

III Construct the resulting IC *K* describing the hybrid model (see Fig.3c). We get the IC *K* as the sum of the properly joined ICs $K = T_1 \oplus T_2 \oplus T_3$ applying the *IC_adding* procedure.

The unified hybrid model combining the bunny and the turtle is shown in Fig. 3. The total run time for this example is about 3.5 seconds on a PC with a Pentium 4 3.2GHz processor and 1Mb RAM. Finally, to show how one can make a transformation of a unified model, we deform the produced heterogeneous IC model applying some non-linear bijective transformations implemented by the *IC_transform* procedure. The total run time for the construction of this hybrid model with transformations is about 4 seconds. The corresponding model after such transformations is shown in Fig. 9b and 9c.

a)



b) c)

Figure 9: The unified hybrid model combining the FRep turtle and BRep bunny after the attachment (a) and after the subsequent deformations (b, c)

7. CONCLUSION

Implicit complexes provide a novel framework that makes it possible to model heterogeneous objects exploiting hybrid representation schemes. In this paper we outlined the IC model and operations defined on ICs. We proposed a step-by-step construction procedure for important particular cases involving the integration of components BRep and FRep components using a union operation and an attachment operation over implicit complexes. We have also demonstrated that the unified model can be further transformed. These case-studies have allowed us to show such benefits of this approach as preserving the initial representation of all the components and guaranteeing topologically correct definitions for all parts and relationships.

The main advantages of the proposed way of modeling complex assemblies include the preservation of the original representations of all the components (independent of how different they may be) and the ability to guarantee topologically correct definitions for all parts and relations of the hybrid model (in particular for problematic regions belonging to the boundaries). This approach also allows us to handle conformity between the object's geometry and its attributes representing non-geometric properties that are crucial for heterogeneous modeling. At first sight, our approach, relying on non-trivial topological constructs may appear as too complex and intimidating to the ordinary user. This inherent complexity of the method will eventually be hidden from the user by the provision of a set of library routines. Thus, the end user will be blissfully unaware of the underlying complexity and will only have to deal with conceptually simple high-level features of the model whose internal structure has to be generated automatically and can be made transparent through a number of high-level queries.

8. REFERENCES

[1] ADZHIEV V., KARTASHEVA E., KUNII T., PASKO A., SCHMITT B.: Hybrid cellular-functional modelling of heterogeneous objects, *Journal of Computing and Information Science in Engineering, Transactions of the ASME 2*, 4 (2002), 312-322.

[2] ALLEGRE R., GALIN E., CHAINE R., AKKOUCHE S.: The HybridTree: mixing skeletal implicit surfaces, triangle meshes and point sets in a free-form modeling system, *Graphical Models 68*, 1 (2006), 42-64..

[3] FOUGEROLLE Y., GRIBOK A., FOUFOU S., TRUCHETET F: Boolean operations with implicit and parametric representation of primitives using R-functions, *IEEE Transactions on Visualization and Computer Graphics* 11, 5 (2005), 529-539.

[4] FREY, P. J., GEORGE, P. L. Mesh Generation Application to Finite Elements. Hermes Science Publishing, Oxford, Paris, 2000, 816 pp.

[5] HACHENBERGER P., KETTNER L.: Boolean Operations on 3D selective Nef complexes: optimized implementation and experiments, In *Proc. of 2005 ACM Symposium on Solid and Physical Modeling (SPM)*, (2005), 163-174.

[6] Heterogeneous object models and their applications, journal Special issue, A. PASKO and V. SHAPIRO (Eds.), *Computer-Aided Design*, 37, 3 (2005).

[7] KARTASHEVA E., ADZHIEV V., PASKO, A., FRYAZINOV O., GASILOV V.: Surface and volume discretization of functionally-based heterogeneous objects. *Journal of Computing and Information Science in Engineering, Transactions of the ASME* 3, 4 (2003), 285-294.

[8] KARTASHEVA E , ADZHIEV V., COMNINOS P., PASKO A., SCHMITT B.: Construction of Implicit Complexes: A Case Study. In *Proc. WSCG 2005*, Czech Republic, (2005), 219-226

[9] KUNII T.: Valid computational shape modeling: design and implementation. *International Journal of Shape Modeling 5,* 2, (1999), 123-133.

[10] PASKO, A., ADZHIEV, V., SOURIN, A., SAVCHENKO, V.: Function representation in geometric modeling: concepts, implementation and applications, *The Visual Computer*, 11, 8 (1995), 429-446.

[11] PASKO, A., ADZHIEV, V., SCHMITT, B., SCHLICK, C.: Constructive hypervolume modelling, *Graphical Models* 63, 6 (2001), 413-442.

[12] ROSSIGNAC J.: Structured topological complexes: a feature-based API for non-manifold topologies. In *Proc. the ACM Symposium on Solid Modeling,* (1997), 1-9.

[13] ROSSIGNAC J., O'CONNOR M.: SGC: A dimension independent model for pointsets with internal structures and incomplete boundaries. In *Geometric modeling for product engineering,* ed. by M. Wozny, J. Turner, K. Preiss, (1990), 145-180.