19

An XML Implementation for Data Exchange of Heterogeneous Object Models

X.Y. Kou[1] and S.T. Tan[2]

Department of Mechanical Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, China
Email: [1]kouxy@hku.hk, [2] sttan@hku.hk

Abstract
Heterogeneous objects refer to objects with spatially different material compositions or structures. Tremendous research efforts have been devoted to modelling heterogeneous objects and many heterogeneous object representations have been proposed. Regardless of the diversity of these CAD models, there are needs to transport and exchange the included *geometry*, *topology* as well as *material distribution* between CAD modellers, CAE tools and CAM facilities. In literature and practical applications there have been lots of STEP (STandard for the Exchange of Product model data) based tools and implementations for the exchange of the geometric/topological data. However, there has been only limited research on the data exchange of material distributions. This chapter focuses on an XML implementation for data exchange of heterogeneous CAD models. The proposed heterogeneous CAD model is described by Extensible Markup Language and detailed approaches to represent the *voxel based*, *explicit function based* and *heterogeneous feature tree based* models are described. The idea is to introduce self-descriptive, customised tags/vocabularies to fit the specific needs of material modelling. The structure of the heterogeneous CAD model is specified with XML schemas and related data validations can accordingly be checked to ensure the model correctness. A prototype CAD module is developed to construct XML-based heterogeneous material model, and the XML model is then exported to SolidWorks to test the validity of the proposed approach. Results show the proposed XML based model can facilitate the data exchanges of heterogeneous material distributions.

19.1 Introduction

Heterogeneous object modelling [19.1,19.2] is a relatively new research direction in the CAD community. Different from traditional homogeneous solid modelling, in which the material distribution of an object is assumed uniform in geometric domain, heterogeneous object modelling incorporates and utilises spatially varying material distributions as additional design freedoms.

The advantages of using heterogeneous material distributions in CAD design have been getting increased recognition in recent years. One primary reason is that the users' design requirements are usually manifold and can seldom be fulfilled with a single homogeneous material. For instance, in artificial finger joint replacement, the implanted finger joint should be both strong and biocompatible. Homogeneous metals (stainless steel, for instance) can provide sufficient strengths; however, the bonding of the artificial joint with the human bones are not ideal. Using biocompatible materials (e.g. titanium, etc.) can enhance the bonding of the artificial joints with the fingers, but the wear resistance of the joint is relatively poor, which may deteriorate the finger movement accuracy and long life use. To alleviate the problems, a heterogeneous finger joint can be developed, as shown in Figure 19.1. In this heterogeneous finger joint, two primary materials are used: material A (cylindrical in shape) has good biocompatibility and can gradually grow into bones (thus offering enhanced integrity with the fingers); and material B (kidney shape) has good strength and wear resistance properties and can contribute to the long life use of the joint. The gradient material region in Figure 19.1 is designed to eliminate possible cracks due to abrupt material changes in the material interfaces.

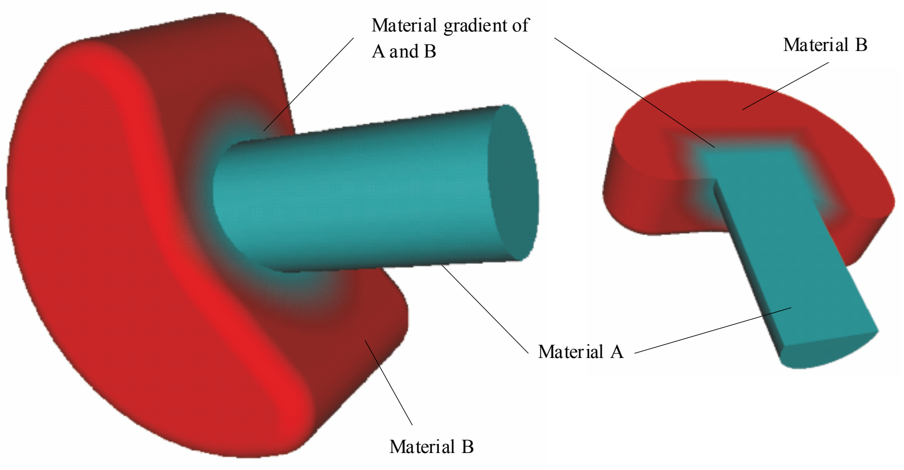


Figure 19.1. An example heterogeneous object

The example shown in Figure 19.1 demonstrates a typical heterogeneous object application, in which the material heterogeneities contribute significantly to the object's functionality and performance. In recent years, similar applications have also been found in mechanical, electrical, thermal and other interdisciplinary areas [19.1].

The wide applications of heterogeneous objects have aroused very active research in modelling, analysis and fabrication of heterogeneous objects in the past few decades. In the literature, many heterogeneous object models have been proposed [19.1,19.2]. These heterogeneous object models differ in their *representational capacities*, *design intuitiveness*, *model exactness*, *compactness, etc*. and many of them target to one or two specific application fields, for instance, *visualisation* or *finite element analysis* of heterogeneous objects.

In practical engineering applications, a heterogeneous object model usually needs to be converted to another format to share/exchange the geometry, topology as well as the associated material heterogeneities. This naturally calls for a standard data format to accomplish such goals. The exchange of geometric/topological data has been extensively studied by many researchers and engineers, mostly under the framework or STEP (STandard for the Exchange of Product model data), and numerous such papers have been published, as reviewed in other chapters of this book. In the literature, however, there is only limited research which tackles the issues of data exchange of material distributions/heterogeneities. This chapter discusses XML based heterogeneous CAD models to cater for such needs.

A brief discussion on XML (Extensible Markup Language) and ISO 10303 is first provided in Section 19.2, where the scopes of current XML-based data exchange are analysed. Before delving into the proposed XML-based heterogeneous model, a concise review on existing heterogeneous object representations is presented to provide sufficient backgrounds in this field. Our Heterogeneous Feature Tree (HFT) based model is particularly elaborated as it is taken as an example to demonstrate the proposed approach. Section 19.4 focuses on the implementation issues and a case study is provided. Finally this chapter is concluded in Section 19.5.

19.2 XML Technologies and ISO 10303

The Extensible Markup Language is a general-purpose specification for creating custom markup languages. It allows users to define semantic constraints for specific applications and enables versatile data to be described in plain text. The plain-text nature facilitates data sharing and exchange across different languages and platforms. By defining meaningful markups, the data can be self-descriptive and much easier to understand.

Using XML to describe/exchange product data is relatively new [19.3, 19.4] and EXPRESS language (ISO 10303-11) [19.5] has always been the formal language to describe product data and their relationships. Recent studies, however, show that XML based data exchange is much easier, more flexible and systematic [19.4–19.7]. In the past few years, there have been many investigations that target on mapping EXPRESS schema/data to XML schema/data for information exchanges. Stephen Chan et al. [19.4] discussed product data exchanges using XML-based mediators, and they analysed the pros and cons of "early and late binding mappings from STEP to XML". Barkmeyer and Lubell [19.7] discussed the issues of reformulating EXPRESS model as XML and tackled some mismatch problems during the mapping, for instance the issues encountered during name mapping, attribute mapping and element mapping. ISO 10303-28:2007 standard [19.8] documents detailed specifications on the use of XML to represent EXPRESS schema as well as the data governed by EXPRESS schemas. The latest release of ST-Developer SDK v12 [19.9], developed by STEP Tools Inc., provides programmatic interfaces to read and write STEP Part-28 XML files to and from many commonly used data formats (inclusive of STEP Part-21 file modelled in EXPRESS). These efforts take advantage of both the rich semantics of EXPRESS language and the widespread infrastructure of XML; however these existing approaches are mostly targeted to objects/products with homogeneous material definitions. Although ISO 10303 has included many specific Application Protocols (AP), for instance, the AP 202 (Associative draughting), AP 203 (Configuration –controlled design) , AP207 (Sheet metal die planning and design), AP 227 (Plant spatial configuration), etc. [19.10], so far there are no application protocols which can be used for data exchanges of heterogeneous object models. To the best of our knowledge, the series of work done by Patil et al. [19.11, 19.12] seem to be the only available investigations along this research direction. Patil et al. [19.11, 19.12], however, use EXPRESS-G language (a subset of EXPRESS family language) to construct STEP-compliant heterogeneous object models and therefore cannot take full advantage of XML's merits, for instance self-descriptive property and explicit hierarchies [19.4, 19.7]. The approach presented in this chapter reflects our attempt towards this goal and a prototype CAD module is proposed to construct and parse XML-based heterogeneous CAD model, as is detailed below.

19.3 An XML Implementation for Data Exchange of Heterogeneous Object Models

19.3.1 Existing Heterogeneous Object Models

Most conventional CAD models assume the material of the product under design is homogeneous on and inside the object's boundaries. The modelling space is usually three-dimensional Euclidean space E^3 or its subspace [19.1], and the major focus in geometric modelling is the shape and spatial relations [19.13,19.14]. In addition to geometric information, heterogeneous object modelling also deals with material heterogeneity defined over the geometric domain. Under such assumptions, for any two points inside a heterogeneous object, their material compositions might be, in general, distinct.

To model the material compositions at a given location, the most common way is to user a k-dimension vector $(r_1, r_2, ..., r_k)$, where r_i represents the volume fraction of the i-th primary material of interest. Within the framework of heterogeneous object modelling, a heterogeneous object can be regarded as a point set $\{P\}$ [19.1]:

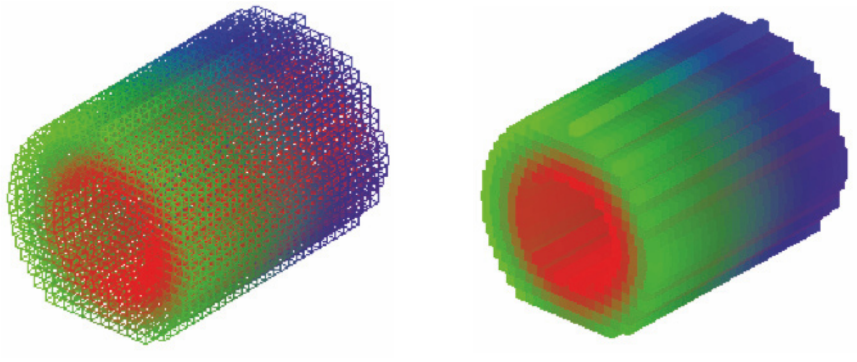
$$P = (P_g, P_m),\ P_g = (x, y, z) \in \Omega_g \subset E^3,\ \ P_m = (r_1, r_2, \cdots r_k) \in \Omega_m \subset E^k$$
$$0 \le r_i \le 1,\ \ 1 \le i \le k, \sum_{i=1}^{k} r_k = 1 \qquad (19.1)$$

where P_g is the location of the point P in the geometric domain Ω_g, P_m is the material composition defined at P_g, and Ω_m is the material domain (subspace of E^k) [19.1]. Note that all the scalars r_i are constrained to sum up to unity such that the material composition P_m is physically meaningful.

There are many ways to represent heterogeneous material distributions in E^3 or its subspace. The voxel based model is perhaps the most intuitive one. In a voxel model, a heterogeneous object is represented as a collection of heterogeneous voxels, each of which represents a small cube in space with a homogeneous or interpolated material distribution [19.15–19.18]:

$$O = \{V_i\} = \{(x_i, y_i, z_i, m_i)\},\ 1 \le i \le n \tag{19.2}$$

where V_i is a representative voxel in object O, (X_i, Y_i, Z_i) is the voxel's location in space, m_i denotes its material compositions and n is the number of voxels. Figure 19.2 shows an example heterogeneous object modelled in a voxel array.



(a) (b)

Figure 19.2. A voxel representation of a heterogeneous hollow cylinder (a) The wireframe view (b) The shaded view

In the above voxel model, it is seen that the material heterogeneity is represented as enumeration of material compositions, as rendered in different colours in Figure 19.2. Apart from such direct enumerations, it is also natural to use analytic functions $V = f(x, y, z)$ to denote the material composition at the position (x, y, z), for instance power-law functions [19.19] and exponential and parabolic functions [19.20]. Figure 19.3 shows a component whose heterogeneity follows a linear material gradation along the Y-axis.

Despite many merits such as the intuitiveness and easy implementations, it is not always possible, or sometimes non-trivial, to describe the material distributions with explicit mathematical functions, for instance, the hollow cylinder shown in Figure 19.4. The cylinder is "extruded" from a heterogeneous disk (Dsk, see Figure 19.4), which has a graded material transition between two circles (Cir1 and Cir2). Along the extrusion direction, the material distributions of these two circles are governed by two heterogeneous lines (L1 and L2). The lines' material are in turn dependent on the location and material definition of the point pairs (P1, P2) and (P3, P4) respectively. The final material distributions of the hollow cylinder are shown in Figure 19.4a from two different viewing directions.

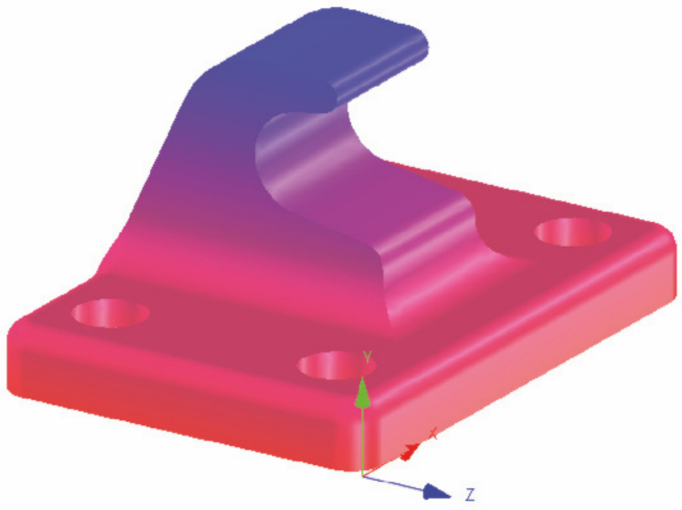


Figure 19.3. Material heterogeneity represented with analytic functions

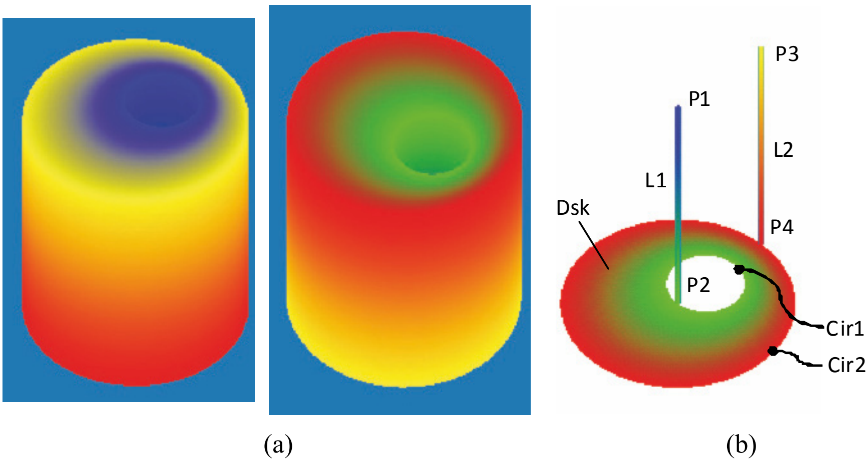


Figure 19.4. A heterogeneous cylinder with trivariate material gradations (a) 3D shaded view from two viewing directions (b) Dependent heterogeneous features

To represent the material heterogeneity shown in Figure 19.4, an explicit function based model is usually inadequate or less flexible. In our previous papers [19.21,19.22], we proposed a generic data structure called Heterogeneous Feature Tree (HFT) to model such material variations. The key idea is to encode the material variation dependency into a tree structure, where the parent feature's material distribution is defined to be dependent on its child features' spatial locations as well as their material definitions. For instance, the material distribution of the cylinder is dependent on the heterogeneous disk (from which it is extruded) and therefore, the disk is modelled as the child feature of the cylinder, as shown in Figure 19.5. The disk's material is defined as a linear material gradation between the two circles, so Cir 1 and Cir 2 are saved as the children of the "Dsk" feature. Such hierarchies are continuously constructed until the leaf nodes (the four points, whose material definitions do not depend on any other features, i.e. without any child features) are populated into the HFT structure, as illustrated in Figure 19.5.

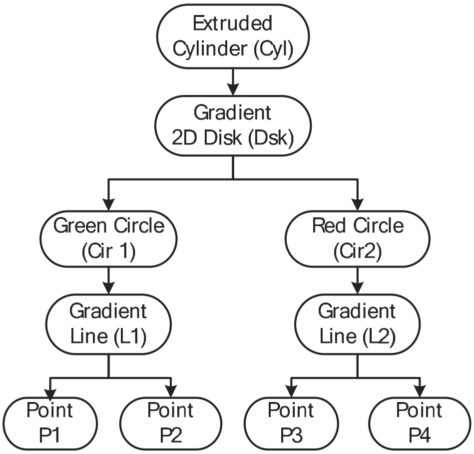


Figure 19.5. A simplified heterogeneous feature tree structures for representing the material heterogeneity shown in Figure 19.4

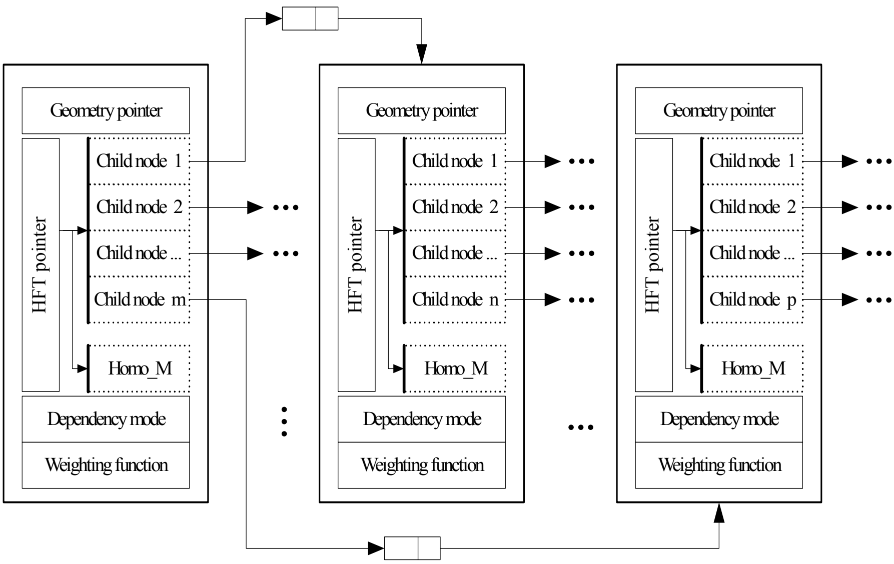


Figure 19.6. The Heterogeneous Feature Tree structure (HFT)

The heterogeneous feature tree structure shown in Figure 19.5 only qualitatively describes the material variation dependency relationships. To represent the material heterogeneity of interest precisely, we propose to associate a list of helper directives with each HFT node, as is graphically illustrated in Figure 19.6. For instance, each HFT node has an associated material gradation function which controls how child features' material compositions are blended to define the parent's material heterogeneities. More detailed descriptions of these parameters and their usages are beyond the scope of this chapter and readers can refer to [19.21–19.23] for more details.

Apart from the aforementioned data models, there are many other different heterogeneous object representations and it is beyond the scope of this chapter to offer an exhaustive analysis on each of the models. For vigorous computer models in the field of heterogeneous object modelling, refer to [19.1] for details.

19.3.2 Representing Material Heterogeneity with XML

Because of the many merits of XML, we propose to use XML based models to describe heterogeneous material distributions defined over the geometric domain. The voxel model, the explicit function model and the HFT based model are taken as examples to illustrate the proposed approach.

Representing Homogeneous Materials with XML

To represent the material vector (r_1, r_2, ..., r_k), a custom defined XML element type called "HomoMaterial" is proposed. In what follows, we assume k=3, i.e., only three primary materials will be considered. An example XML element for the description of a homogeneous material is shown in Figure 19.7a. The "HomoMaterial" element has three sub-elements, each of which represents the volume fractions of three primary material. The underlying data type of Material 1, 2 and 3 is the built-in simple data type specified in the standard XML schema [19.24], as shown in Figure 19.7b.

```
<HomoMaterial>
  <Material1>1</Material1>
  <Material2>0</Material2>
  <Material3>0</Material3>
</HomoMaterial>
```

(a)

```
<simpleType name="double">
  <restriction base="double" />
</simpleType>
```

(b)

Figure 19.7. An example XML element for the description of a homogeneous material (a) XML elements (b) Related XML schema snippet.

Note that the markup "*HomoMaterial*", "*Material1*" etc. in Figure 19.7a are custom tags that are specifically defined for representing material definitions. The vocabulary and structure of these custom data are defined in a proposed XML schema, as shown in Figure 19.9.

Representing Voxel and Explicit Function Based Models with XML

Given the definition of the “HomoMaterial” as described above, a voxel model can be easily represented using XML aggregate or collection data types. Figure 19.8a illustrates an XML element containing two voxels. As the element names and attributes are self-descriptive, we will not elaborate on these details. By sequentially populating all the voxels in similar formats, the heterogeneous object shown in Figure 19.2 can be easily modelled.

```
- <VoxelModel VoxelSize="0.1">
  - <Voxel>
      <XCoordinate>0</XCoordinate>
      <YCoordinate>0</YCoordinate>
      <ZCoordinate>0</ZCoordinate>
    - <HomoMaterial>
        <Material1>0.6</Material1>
        <Material2>0.4</Material2>
        <Material3>0.0</Material3>
      </HomoMaterial>
    </Voxel>
  - <Voxel>
      <XCoordinate>0.1</XCoordinate>
      <YCoordinate>0</YCoordinate>
      <ZCoordinate>0</ZCoordinate>
    - <HomoMaterial>
        <Material1>0.5</Material1>
        <Material2>0.5</Material2>
        <Material3>0.0</Material3>
      </HomoMaterial>
    </Voxel>
  </VoxelModel>
```

(a)

```
- <math>
  - <mrow>
    - <msup>
        <mi>x</mi>
        <mn>3</mn>
      </msup>
      <mo>+</mo>
    - <mrow>
        <mn>0.5</mn>
        <mo />
        <mi>x</mi>
      </mrow>
    </mrow>
  </math>
```

(b)

Figure 19.8. XML based heterogeneous model (a) XML element of an illustrative voxel model; (b)) XML element for the representation of an explicit function

For the case of explicit function based models, the key issue is to represent the mathematical functions with XML markups. By using the MathML [19.25], the recommended XML specification for describing mathematics by the World Wide Web Consortium (W3C), there is no need to reinvent the wheel from scratch. For instance, the function $f(x) = x^3 + 0.5x$ can be represented with the XML elements shown in Figure 19.8b. Here the “*msup*” tag indicates superscript, “*mi*” indicates “identifier”, “*mn*” means “number” and “*mo*” refers to “operators”. Following the specification of MathML, it is straightforward to transform more complex mathematical formulae into XML elements and then use the functions to model precisely the desired material distributions.

Representing HFT-Based Model with XML

Transforming the hierarchical HFT structures into XML requires the introduction of additional custom data types, since there is no off-the-shelf constructs in standard XML schema.

```
 <?xml version="1.0" encoding="UTF-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  - <xs:element name="Children">
    - <xs:complexType>
      - <xs:sequence>
          <xs:element ref="HeterogeneousFeature" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  + <xs:element name="GeometryID">
  - <xs:element name="HeterogeneousFeature">
    - <xs:complexType>
      - <xs:sequence>
          <xs:element ref="Name" />
          <xs:element ref="GeometryID" />
          <xs:element ref="MaterialDescriptor" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  + <xs:element name="HomoMaterial">
  + <xs:element name="Material1">
  + <xs:element name="Material2">
  + <xs:element name="Material3">
  - <xs:element name="MaterialDescriptor">
    - <xs:complexType>
      - <xs:choice>
          <xs:element ref="Children" />
          <xs:element ref="HomoMaterial" />
        </xs:choice>
      </xs:complexType>
    </xs:element>
  + <xs:element name="Name">
  </xs:schema>
```

Figure 19.9. A snippet of the proposed XML schema for the representation of HFT based material distributions

Figure 19.9 shows a snippet (parts of the schema are suppressed for brevity) of the proposed XML schema, where the required kernel data types are defined. In this schema, the "*Children*" type is defined as XML aggregate type with zero or more element items (see the "maxOccurs" attribute value in Figure 19.9). The item of the "Children" sequence is of type "*HeterogeneousFeature*", which contains a "*Name*" element for the textual description of the feature, a "*GeometryID*" element which contains the identification of the geometry referred to (i.e. the XML equivalent of the "Geometry pointer" in Figure 19.) and a sub-element of "*MaterialDescriptor*". The "MaterialDescriptor" type element, as indicated by the keyword "xs:choice", is either a "*HomoMaterial*" element or contains a "*Children*" element for modelling the material distributions, as described earlier. Note that the type "xs:choice" is prefixed by the *namespace prefix* associated with the XML Schema, indicating that the data type is of a predefined one.

This XML schema defines an XML equivalent of the data structure and constraints depicted in Figure 19.6. Based on such definitions, the object in Figure 19.4a can be then modelled with an XML file shown in Figure 19.10.

```
- <HeterogeneousFeature xmlns="http://www.CAX4D.com/HOM_XML">
    <Name>Extruded Feature 1</Name>
    <GeometryID>8</GeometryID>
  - <MaterialDescriptor>
    - <Children>
      - <HeterogeneousFeature>
          <Name>Region 1</Name>
          <GeometryID>3</GeometryID>
        - <MaterialDescriptor>
          - <Children>
            - <HeterogeneousFeature>
                <Name>Arc 1</Name>
                <GeometryID>1</GeometryID>
              - <MaterialDescriptor>
                - <Children>
                  - <HeterogeneousFeature>
                      <Name>Line 3</Name>
                      <GeometryID>6</GeometryID>
                    + <MaterialDescriptor>
                    </HeterogeneousFeature>
                  </Children>
                </MaterialDescriptor>
              </HeterogeneousFeature>
            + <HeterogeneousFeature>
            </Children>
          </MaterialDescriptor>
        </HeterogeneousFeature>
      </Children>
    </MaterialDescriptor>
  </HeterogeneousFeature>
```

Figure 19.10. A snippet of XML file for the model shown in Figure 19.4.

The hollow cylinder is assigned a descriptive name "Extruded Feature 1", and the cylinder's geometry points to an entity whose ID is "8". By tracing the entity with the provided ID, the geometric details of the hollow cylinder can be interrogated. In our implementation, the "ID" is modelled with ACIS custom attribute which is associated with internal geometric entities, such as BODY, FACE, and EDGE, etc. [19.26]. The sub-element "MaterialDescriptor" corresponds to the HFT representation shown in Figure 19.5. For the hollow cylinder, the sole child feature on which its material distribution depends is the "Region 1". Similarly, in terms of material heterogeneity, "Region 1" relies on "Arc 1", which corresponds to the "Green Circle (Cir 1)" in Figure 19.5, and "Arc 1" relies on "Line 3" and so forth.

Note that the XML file in Figure 19.10 is not illustrated in full details and some less important elements are suppressed in the XML tree for brevity reasons. In addition, some low level details such as the "Weighting function" etc. (see Figure 19.6) are not elaborated here to avoid unnecessary complexities. Technically, there should be no difficulties in integrating such information into the proposed XML model using MathML languages, as described in previous sub-section. Throughout this chapter we assume that linear material blending function is used for all the HFT nodes by default.

Using XML File for Exchanges of Material Heterogeneities

The merits of representing the material distributions with XML include enhanced interoperability, flexibility and ease of use. The interoperability comes from XML's plain text nature which enables computer models constructed with different languages to be shared and exchanged. The flexibility is benefited from XML's widespread infrastructure and extensibility. The ease of use is attributed to its self-descriptive property, many ready-for-use markup tags and the available rich development tools offered by commercial vendors as well as a huge user community.

In our previous work [19.21–19.23], we have proposed a prototype heterogeneous CAD modeller based on the hierarchical HFT representation. The proposed modeller, CAD4D [19.27], uses ACIS [19.26] as the geometric modelling kernel and the HFT structure is implemented using C++ class. The native CAD4D model is of a binary format, which is efficient in data serialization, but not convenient for information exchanges with other CAD/CAE packages. For instance, we have encountered quite a few challenges when attempting to export the material distributions to COMSOL Multiphysics [19.28] to perform numerical analysis on some designed heterogeneous objects. In the work reported in [19.29, 19.30], material compositions of points at regular grids (imported from COMSOL Multiphysics) must be evaluated by CAD4D and then exported to COMSOL Multiphysics via text files, as illustrated in Figure 19.11.

The approach illustrated in Figure 19.11 has certain limitations. First, the heterogeneous CAD model is not self-contained and the material composition evaluation relies on CAD4D's functionalities. In other words, such a heterogeneous model is meaningless outside CAD4D and its semantics cannot be interpreted by applications other than CAD4D. Second, as CAD4D runs on Windows platform only, CAD/CAE packages running on other platforms such as Linux etc. are therefore unable to utilise such models in downstream applications. The reason for such problems is that our heterogeneous CAD model is private and not interchangeable. By definition, however, the proposed hierarchical HFT representation could be implementation and platform independent, i.e. the model can be constructed by C, C++, C# or other well known programming languages without any constraints imposed, and such CAD models should also be accessed and maintained by software packages running on Windows, Linux and other operating systems.

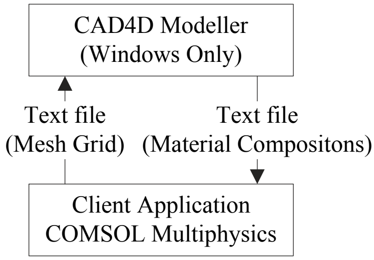


Figure 19.11. Data exchanges between CAD4D and COMSOL Multiphysics

One way to accomplish this goal is to exchange such information using XML files. To do this, heterogeneous CAD modellers need to convert their internal data representations using the data types defined in the XML schemas. Other applications such as COMSOL Multiphysics or SolidWorks, etc. can then retrieve the data from the neutral file, as shown in Figure 19.12. In this approach, XML is taken as an intermediate layer to exchange the material information between different applications.

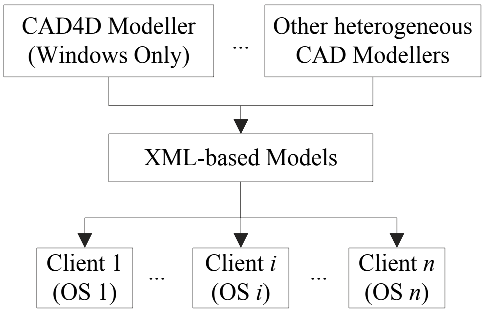


Figure 19.12. Data exchanges using XML-based heterogeneous object model

Note that when using the XML-based models, it is no longer the role of CAD4D to offer the material interrogation services; instead, the role is passed to the clients which retrieve, parse and utilise the data from the XML models.

Material Composition Interrogation from XML Files

Given an XML-based heterogeneous model, for an arbitrary point of interest, its material composition can be interrogated.

For the voxel-based model, all the XML elements are first read into a dynamic list and the voxel which contains the point under interrogation (i.e. whose distance to the input point is smaller than the voxel size) is identified, if it exists. The point under evaluation is said to be contained by the voxel and the "HomoMaterial" sub-element of the voxel (see Figure 19.8) is then returned as the interrogated material composition. If the material composition inside a voxel is also represented by interpolation functions [19.15–19.18], the point can be input as the function parameter, and the function value is returned as the output material. This is essentially the same as the material evaluation process for the explicit function based models.

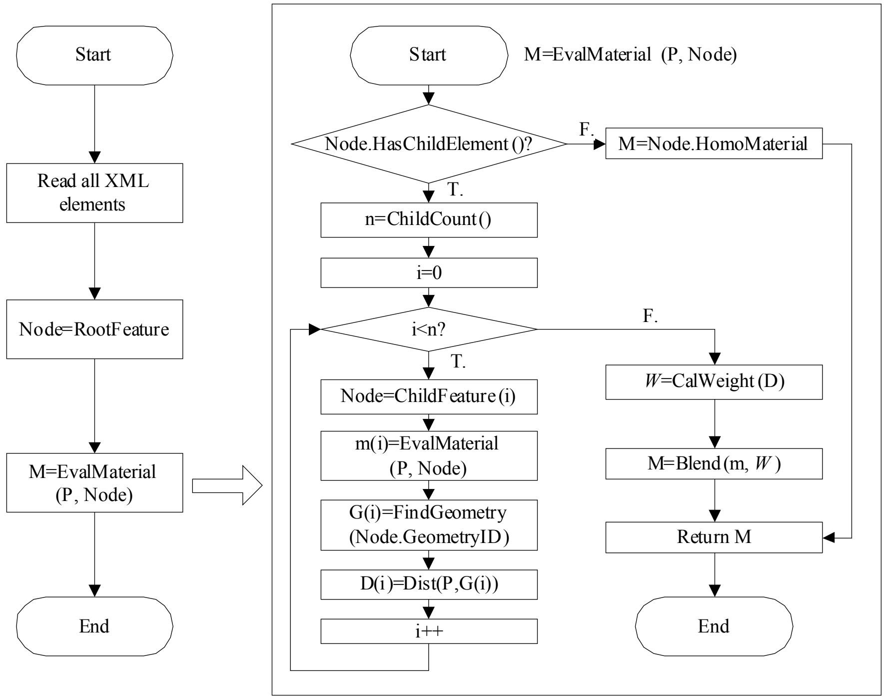


Figure 19.13. Material evaluation from XML-based HFT model

For the heterogeneous feature tree based model, the material evaluation can be obtained by calling a recursive function [19.21], as explained in Figure 19.13. The material composition evaluated on the root XML node is returned as the output material. According to the definition of the heterogeneous feature tree representation, the parent-child relationship is used to encode the material variation dependencies. So, to get the material evaluated on a specific node, the materials evaluated from all its sub-nodes (child nodes) must be known. This recursion continues until a leaf node which has no child element is reached. For a leaf node, its material can be directly retrieved from its HomoMaterial sub-node.

The materials evaluated from all the XML sub-nodes contribute to the material evaluated on their parent XML node, and the weights are related to the distances from each feature's geometry to the input point. The feature's geometry can be traced via the node's GeometryID element, which is associated with the geometric entity by custom attributes. Finally, all the materials evaluated from each sub-element are blended with calculated weights applied, and such blending is repeated until the material evaluated from the root XML node is returned. Figure 19.13 shows a flow chart depicting the above algorithm.

19.4 Implementations and a Case Study

To test the validity of the proposed XML-based heterogeneous object model, we have developed a prototype module to convert our private model files to XML files. The geometry and material information in the CAD4D model are respectively represented with ACIS B-Rep and the HFT structures. CAD4D is developed with Microsoft Visual C++ 6.0 on Windows XP platform, as shown in Figure 19.14.

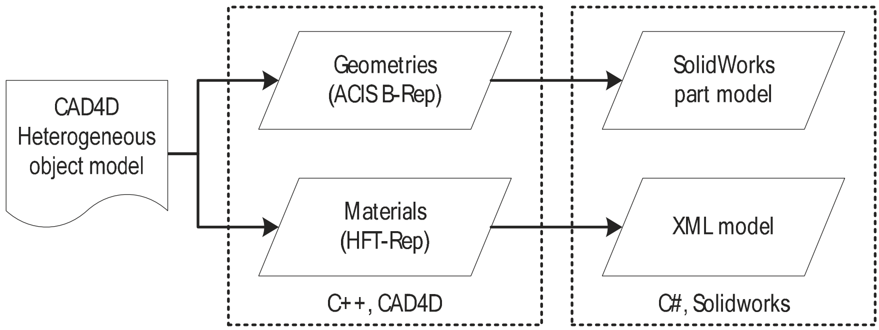


Figure 19.14. Implementation of the proposed software module

In CAD4D, the heterogeneous feature tree structure is populated to a XML file using MSXML API 4.0 SP1 [19.31]. The XML file is then imported by SolidWorks using SolidWorks API. Several C# classes are developed to de-serialise the material information from the XML file[19.32], and the internal C# representations of these classes are shown in Figure 19.15.

In Figure 19.15, the classes "*HeterogeneousFeature*", "*MaterialComposition*" and "*MaterialRep*" correspond to the XML constructs "*HeterogeneousFeature*", "*HomoMaterial*" and "*MaterialDescriptor*" in the XML schema shown in Figure 19.9.

The ACIS based geometry is converted to SolidWorks [19.33] part model using SolidWorks API and C# language.

As discussed in Section 19.3, the "GeometryID" element in the XML file refers to a feature's geometry ID and such geometric information is necessary in the material composition evaluation process. In our implementation, we attached such ID information to relevant ACIS entities, and to trace the associated geometric entities in SolidWorks we developed a custom ACIS attribute parser to enable entity tracking via these ID attributes.

An example heterogeneous model is first constructed in CAD4D, as shown in Figure 19.16. The geometry and material data are then exported to an ACIS SAT file and an XML file. Custom C# Addin is then developed to read, parse and manipulate the related data in SolidWorks. Figure 19.17 shows the translated heterogeneous model in SolidWorks graphics area. The right side of the figure shows the source XML file which is used to represent the material heterogeneities. This example shows that the proposed model is valid and effective.

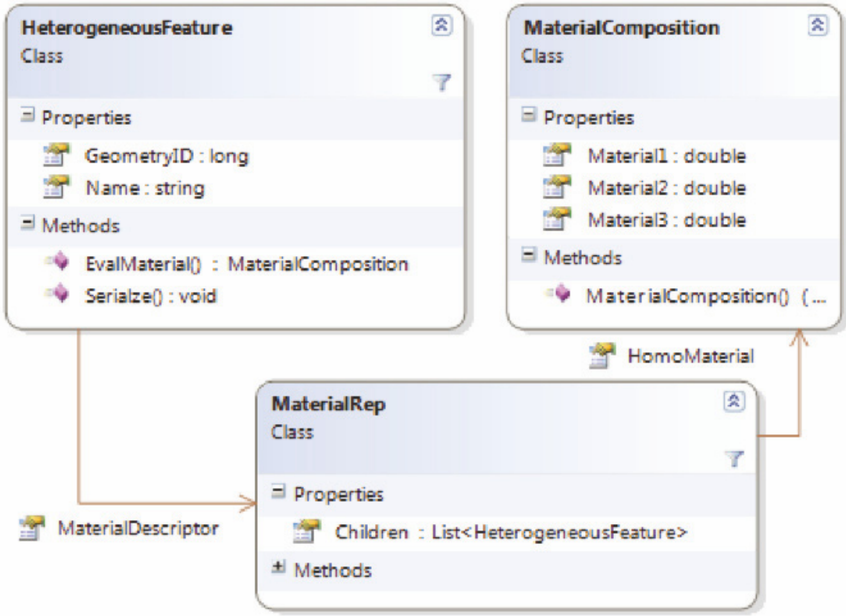


Figure 19.15. A diagram of related C# classes

Note that the geometry of the heterogeneous object is not represented in XML format because, even if it is, it will ultimately be translated to SolidWorks part file in this case study. Technically, there is no difficulty in representing the geometries in XML and there have been mature data conversion tools for such purposes. For instance, SolidWorks has related APIs to convert the part model to Dassault Systèmes 3D XML format [19.34]. In this regard, there is no difficulty in representing the entire heterogeneous object (inclusive of the geometries and material distributions) in a single, unified XML file.

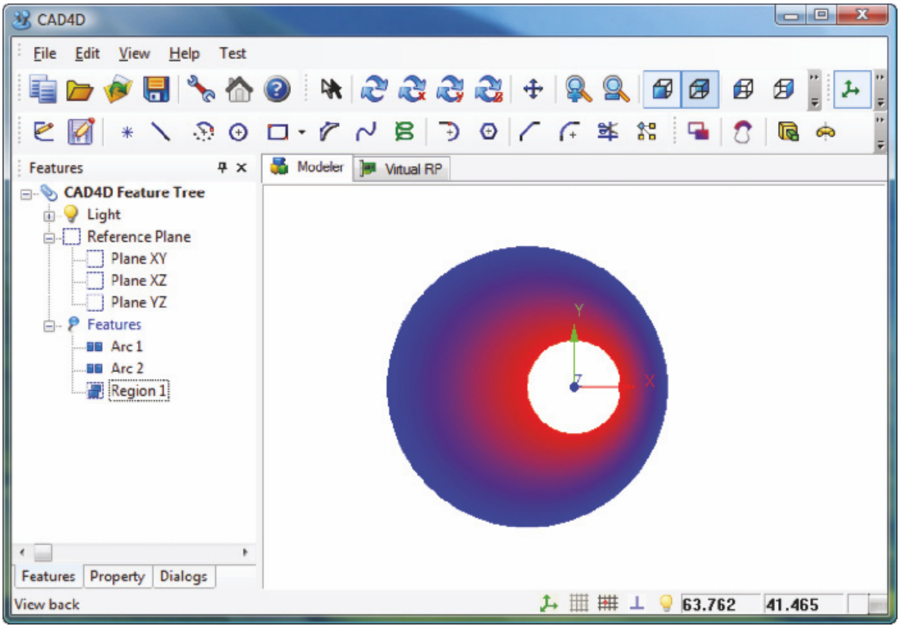


Figure 19.16. A heterogeneous model constructed by CAD4D

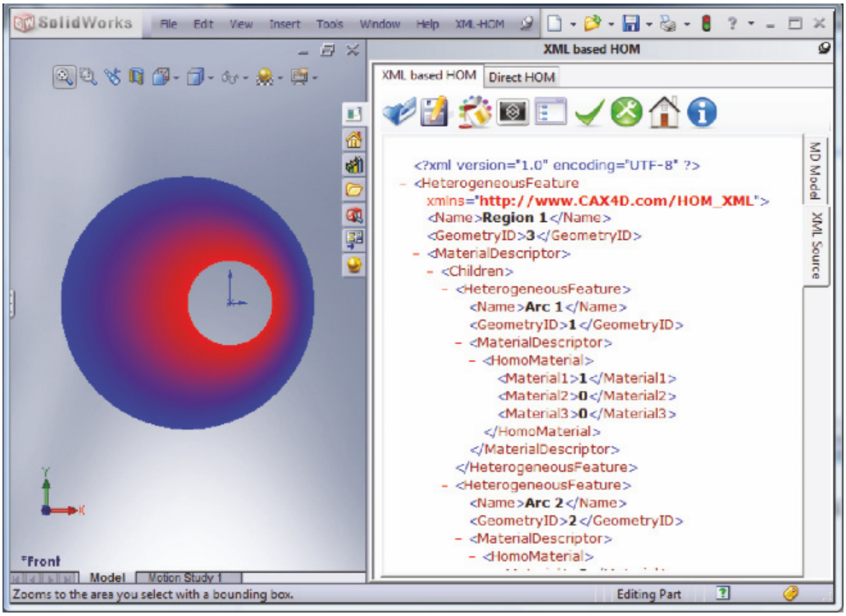


Figure 19.17. A heterogeneous model imported by SolidWorks and parsed by the XML-HOM Addin

19.5 Conclusions

Data exchanges under STEP are mostly targeted to share and exchange products with homogeneous material definitions. In STEP, there are no application protocols which can be used for data exchanges of heterogeneous object models. This chapter presents an XML-based heterogeneous CAD model to represent the material heterogeneities defined over 3D Euclidean space. Using XML language to represent material heterogeneities is much easier, more flexible and more systematic. Detailed approaches to represent the voxel based, explicit function based and heterogeneous feature tree based models are described. The idea is to introduce self-descriptive, customized tags/vocabularies to fit the specific needs of material descriptions. The structure of the heterogeneous CAD model is specified with XML schemas and related data validations can be accordingly checked to ensure the model correctness. A prototype CAD module is developed to construct XML-based heterogeneous object model, and the XML model is then exported to SolidWorks to test the validity of the proposed approach. Results show the proposed XML based implementation can facilitate data exchanges of heterogeneous material distributions.

Acknowledgment

The authors would like to thank the Department of Mechanical Engineering, The University of Hong Kong and the Research Grant Council for supporting this project (Project No: HKU 7200/04E). We would also like to thank Mr. Olivier Lejardinier, from TraceParts S.A. (http://www.traceparts.com) for his discussions and suggestions during the prototype software development. Thanks also to the anonymous reviewers and the editors of this book for their many constructive comments and suggestions.

References

[19.1] Kou, X. Y. and Tan, S. T., Heterogeneous object modeling: A review, Computer-Aided Design, vol. 39, pp. 284-301, 2007.

[19.2] Pasko, A., Adzhiev, V., and Comninos, P., Heterogeneous Objects Modelling and Applications, in Collection of Papers on Foundations and Practice Series, Lecture Notes in Computer Science. vol. 4889 2008.

[19.3] Burkett, W. C., Product data markup language: A new paradigm for product data exchange and integration, CAD Computer Aided Design, vol. 33, pp. 489-500, 2001.

[19.4] Chan, S. C. F., Dillon, T., and Ng, V. T. Y., Exchanging step data through XML-based mediators, Concurrent Engineering Research and Applications, vol. 11, pp. 55-64, 2003.

[19.5] ISO10303-11, Industrial automation systems and integration — Product data representation and exchange— Part 11: Description methods: The EXPRESS language reference manual, 2004.

[19.6] Peak, R. S., Lubell, J., Srinivasan, V., and Waterbury, S. C., STEP, XML, and UML: Complementary technologies, Journal of Computing and Information Science in Engineering, vol. 4, pp. 379-390, 2004.

[19.7] Barkmeyer, E. and Lubell, J., XML Representation of EXPRESS Models and Data, in ICSE Workshop on XML Technologies and Software Engineering Toronto, Ontario, Canada, 2001.
[19.8] ISO10303-28, Industrial automation systems and integration--product data representation and exchange, 1st ed. Geneva: ISO, 2007.
[19.9] http://www.steptools.com/.
[19.10] Michael, J. P., Introduction to ISO 10303 - the STEP standard for product data exchange, J. Comput. Info. Sci. Eng., vol. 1, pp. 102-103, 2001.
[19.11] Patil, L., Dutta, D., Bhatt, A. D., Jurrens, K., Lyons, K., Pratt, M. J., and Sriram, R. D., Representation of heterogeneous objects in ISO 10303(STEP), American Society of Mechanical Engineers, Manufacturing Engineering Division, MED, vol. 11, pp. 355-363, 2000.
[19.12] Patil, L., Dutta, D., Bhatt, A. D., Jurrens, K., Lyons, K., Pratt, M. J., and Sriram, R. D., A proposed standards-based approach for representing heterogeneous objects for layered manufacturing, Rapid Prototyping Journal, vol. 8, pp. 134-146, 2002.
[19.13] Requicha, A. A. G. and Voelcker, H. B., Solid modeling - A historical summary and contemporary assessment, IEEE Computer Graphics and Applications, vol. 2, pp. 9-16, 18-20, 22-24, 1982.
[19.14] Requicha, A. G., Representations for rigid solids: theory, methods, and systems ACM Computing Surveys, vol. 12, pp. 437-464, 1980
[19.15] Hu, Y., Blouin, V. Y., and Fadel, G. M., Design for manufacturing of 3D heterogeneous objects with processing time considerations, Proceedings of ASME 2005 Design Engineering Technical Conferences, September 24-28, 2005, Long Beach, California USA, 2005.
[19.16] Cho, J. R. and Ha, D. Y., Optimal tailoring of 2D volume-fraction distributions for heat-resisting functionally graded materials using FDM, Computer Methods in Applied Mechanics and Engineering, vol. 191, pp. 3195-3211, 2002.
[19.17] Shin, K.-H., Representation and process planning for layered manufacturing of heterogeneous objects: PhD Thesis, University of Michigan, 2002.
[19.18] Chen, M. and Tucker, J. V., Constructive volume geometry, Computer Graphics Forum, vol. 19, pp. 281-293, 2000.
[19.19] Elishakoff, I., Gentilini, C., and Viola, E., Three-dimensional analysis of an all-round clamped plate made of functionally graded materials, Acta Mechanica, vol. 180, pp. 21-36, 2005.
[19.20] Eraslan, A. and Akis, T., On the plane strain and plane stress solutions of functionally graded rotating solid shaft and solid disk problems, Acta Mechanica, vol. 181, pp. 43-63, 2006.
[19.21] Kou, X. Y. and Tan, S. T., A hierarchical representation for heterogeneous object modeling, Computer-Aided Design, vol. 37, p. 307, 2005.
[19.22] Kou, X. Y., Tan, S. T., and Sze, W. S., Modeling complex heterogeneous objects with non-manifold heterogeneous cells, Computer-Aided Design, vol. 38, pp. 457-474, 2006.
[19.23] Kou, X. Y., Computer-Aided Design of Heterogeneous Objects: PhD Thesis, The University of Hong Kong, 2006.
[19.24] http://www.w3.org/2001/XMLSchema-datatypes.
[19.25] http://www.w3.org/Math/.
[19.26] Corney, J., 3D modeling with the ACIS kernel and toolkit. Chichester ; New York: J. Wiley & Sons, 1997.
[19.27] Kou, X. Y. and Tan, S. T., An interactive CAD environment for heterogeneous object design, in Proceedings of ASME 2004 Design Engineering Technical Conferences, September 28-October 2, 2004, Salt Lake City, Utah USA.
[19.28] http://www.comsol.com/.

[19.29] Kou, X. Y. and Tan, S. T., Heterogeneous Object Design: An Integrated CAX Perspective, in Heterogeneous Objects Modeling and Applications, Lecture Notes on Computer Science, V. A. Alexander Pasko, Peter Comninos, Ed., 2007, pp. 42-59.

[19.30] Kou, X. Y. and Tan, S. T., A systematic approach for Integrated Computer-Aided Design and Finite Element Analysis of Functionally-Graded-Material objects, Materials & Design, vol. 28, pp. 2549-2565, 2007.

[19.31] http://www.microsoft.com/downloads/details.aspx?familyid=3144B72B-B4F2-46DA-B4B6-C5D7485F2B42&displaylang=en.

[19.32] Albahari, J., Albahari, B., and Drayton, P., C# 3.0 in a nutshell, 3rd ed. Beijing ; Cambridge: O'Reilly, 2007.

[19.33] http://www.solidworks.com/.

[19.34] http://www.3ds.com/3dxml.