

Short Paper

TRANSFORMATION FROM IFC DATA MODEL TO GML DATA MODEL: METHODOLOGY AND TOOL DEVELOPMENT

I-Chen Wu and Shang-Hsien Hsieh*

ABSTRACT

To facilitate the incorporation of 3D CAD objects into the 3D-GIS environment, this paper proposes a feasible approach to transform the 3D geometric information of CAD objects expressed by the IFC model into geometric objects for the GML model. The discussions include not only the proposed model transformation methodology but also the development of a software tool that implements the proposed methodology. In addition, a simple model transformation example is provided to demonstrate the feasibility of the proposed methodology and the functionalities of the developed tool.

Key Words: IFC, GML, CAD, 3D-GIS, interoperability.

I. INTRODUCTION

In recent years, for creating a more navigable, interactive, and visually realistic information management environment for building construction, there has been an increasing interest in addressing the issues related to interoperability and integration between 3D CAD data and 3D GIS data. Because the CAD systems and the GIS systems (Zlatanova *et al.*, 2002) are primarily designed for different purposes, with their own specialized functionalities, and have been employed by different user groups for a long time, it is not feasible to expect development of any system soon that can support both CAD and GIS capabilities and provide a seamless CAD/GIS integration. Therefore, a more feasible approach at this point is to achieve CAD/GIS interoperability at the data level. However, because the 3D building objects created by the CAD systems and the geo-data objects managed in the GIS systems are quite different in their representation approaches and data storage formats, their data integration still presents a challenging task.

Two approaches may be employed for the CAD/GIS data integration. The first one is to communicate

geo-data from the GIS systems to the CAD systems. For example, the on-going development of the Industry Foundation Classes (IFC) for GIS, called IFG¹, attempts to provide a solution for the passage of information about terrains, land parcels, local planning, road access, etc., from various GIS standards to the CAD systems. The second approach is to transform 3D CAD data from the CAD systems into the GIS systems. For example, Benner *et al.* (2004) have transformed CAD data into a GIS-based model called the QUASY (Quartierdaten-Managementsystem) model, which is structurally very similar to the GML (Geography Markup Language) model (OGC, 2004) and is designed as a new 3D semantic building model dedicated to urban development. Based on the QUASY model, new applications for 3D city models can be realized in the area of town planning and urban management.

In this paper, a methodology for model transformation from the IFC data model to the GML data model is presented and it belongs to the second approach. The IFC data model, instead of a traditional CAD data model, such as DWG, DXF and DGN formats, is employed here because it has become an international industry standard for CAD data exchange and sharing. The IFC model defines a set of internationally standardized objects for building construction based on the STEP (ISO, 1992) technology. It represents not only tangible building components such as, walls, doors, beams, ceilings and furniture, but

*Corresponding author. (Email: shhsieh@ntu.edu.tw)

The authors are with the Department of Civil Engineering, National Taiwan University, Taipei, Taiwan 106, R.O.C.

¹ More information can be found at <http://www.iai.no/ifg/index.html>

also abstract concepts such as schedules, activities, spaces, organization, construction costs and so on (Liebich, 2004). For the GIS data model, the GML model is employed here as a base model to provide an integrated ontology covering the CAD and GIS model concepts in one framework. GML is the most widely supported open specification for representation of geographic (spatial and location) information and its latest version (GML 3.1.1) includes several 3D geo-data types.

The remainder of this paper is organized as follows. The proposed methodology for model transformation from IFC to GML is presented first. Then, the design and implementation of a software tool to realize the proposed methodology are discussed. Finally, a simple example is provided to demonstrate the application of the proposed methodology and tool for CAD/GIS data integration.

II. METHODOLOGY FOR MODEL TRANSFORMATION FROM IFC TO GML

In CAD systems, 3D building models are usually created using the Constructive Solid Geometry (CSG) or Swept Solid method. The CSG method combines a set of the so-called primitive solids (e.g., boxes, tetrahedrons and pyramids) through regularized Boolean operations (e.g., addition, subtraction and union) to create complex solid models. The Swept Solid method sweeps an object (e.g., a bounded planar surface) along a path through space to create solid objects. On the other hand, the GIS systems (as well as GML) represent the 3D objects using the B-Rep (Boundary Representations) method. Therefore, the transformation of building objects from the CAD systems to the GIS systems mainly involves transformation from the CSG or Swept Solid models to the B-Rep models.

In this Section, a methodology for such transformation is presented and discussed. First, the 3D geometric information of building components from the IFC object model that is generated from the CAD system is extracted. Then, the extracted 3D information is converted into the corresponding GML objects for integrating with geo-data in 3D-GIS.

1. Geometric Representation of IFC Building Model

Geometric Representation of IFC model includes the axis, walkingline, shape and boundingbox. In this paper, the focus will be on the shape representation of building elements. In the IFC2x3 model, there are three main types of solid model representation defined: (1) Standard geometric representation: It is defined using the simple "SweptSolid" representation

which uses the `IfcRectangleProfileDef` or `IfcCircleProfileDef` to define the profile of sweeping. (2) Advanced geometric representation: It is defined using the "SweptSolid" (enhanced by additional profile types) or CSG representation. (3) B-Rep: This is used for representation of objects with complex geometry, such as, `IfcDoor` and `IfcWindow`. There is no need to go into details about how to transform the B-Rep models of IFC to the B-Rep models of GML in this paper because it is only the transformation of file format. Also, with more than sixteen types of Profile Definitions supported by IFC for the SweptSolid models, most of building elements can be well represented by the Swept Solid method. Therefore, this paper pays little attention to the transformation from the CSG models to the B-Rep models, but instead, focuses mainly on the transformation from the SweptSolid models to the B-Rep models.

The schema of IFC has complicated structure with Defined Types, Enumerations, Select Types, Entities, and Attributes. A sketchy class diagram and its corresponding file format are shown in Fig. 1 to demonstrate the structure of the IFC building objects that are related to the transformation of building shapes discussed in this paper. Any building object in IFC has the following two attributes to help in describing its geometric representation. One is called "ObjectPlacement", which describes the placement of the object in the spatial context. The other one is called "Representation", which defines object representation (e.g., SweptSolid, CSG or B-Rep) and object definition shape (e.g., `IfcRectangleProfileDef`, `IfcCircleProfileDef`, `IfcIshapeProfileDef` and so on).

The IFC data file uses the Clear Text Encoding of the Exchange Structure as defined in ISO 10303 Part21. IFC specifies information in terms of the Entities which have associated attributes and constraints. Every entity instance is represented by the name of the entity in capital letters, followed by the attribute values in the defined order within parenthesis in the IFC file (see Fig. 1). The major difficulty encountered in the handling and manipulation of the geometric data of IFC building objects is on parsing and interpreting the IFC files. In this research, the IFCsvr² ActiveX Component, an open source library for handling IFC data, is employed. This library helps us develop an IFC-Based tool which has such features as a dictionary of IFC schema information and functionality for representing and manipulating instances of IFC objects.

2. Generation of GML Object Model from IFC Object Model

As shown in Fig. 2, the Swept Solid method

² More information can be found at <http://groups.yahoo.com/group/ifcsvr-users/>

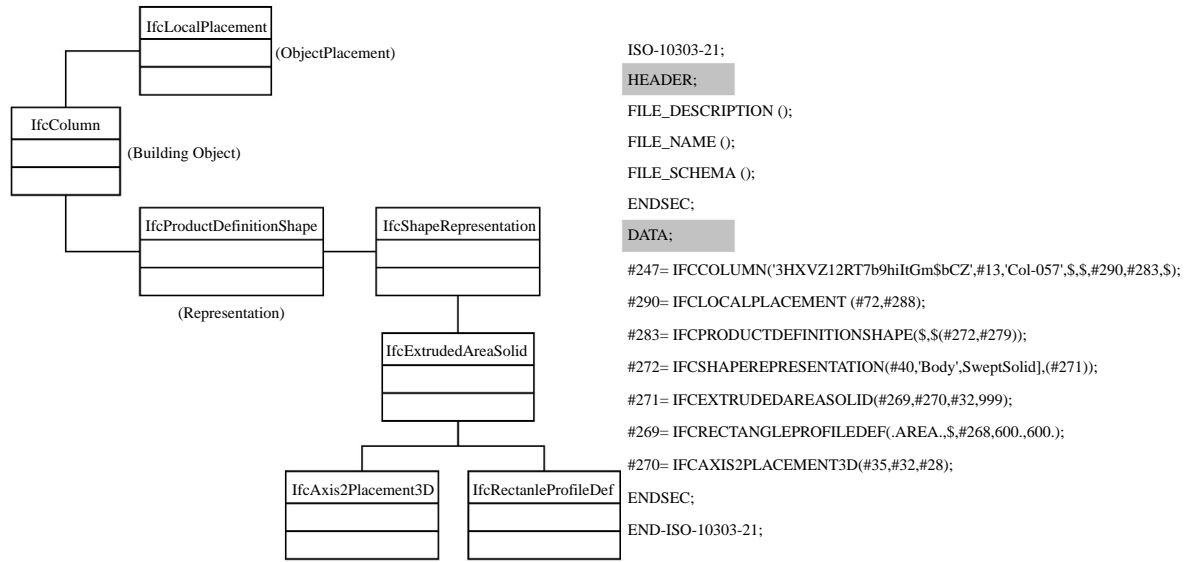


Fig. 1. Sketchy class diagram and corresponding file format

describes the shape of an object by sweeping a surface along a specified direction for a specified distance, while the B-Rep method describes the boundary faces of a 3D shape using a set of the edges, faces and vertices. With the understanding of the principle of the 3D representations in both methods, the algorithm for the model transformation from the Swept Solid representation (of IFC objects) to the B-Rep representation (of GML objects) can be developed.

The model transformation algorithm employed in this work is described as follows:

STEP 1: Extracting geometric data of desired building objects from the IFC file. The required data are extracted from the IFC file for model transformation, such as definition of the local coordinate system, geometry of the SweptArea, distance to be swept, direction vector of sweeping, definition of the real world coordinate system and the placement coordinates of the object. For a parallelepiped example (see Fig. 2), the data extracted include the origin ($X1, Y1, Z1$) of the shaded rectangle in the local coordinate system ($Z1 = 0$ is automatically assigned), the length and width of the rectangle (X Dim and Y Dim), the distance to be swept (Depth), direction vector of sweeping, the definition of the real world coordinate system and the placement coordinate of the parallelepiped.

STEP 2: Calculating the coordinates of vertices of the objects in the local coordinate system. The coordinate of the vertices on the surface is already defined as (x, y, z) of the SweptArea. The following equation can be used to calculate coordinate of the vertices of the surface, defined as (x', y', z') on the opposite side:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = D \cdot \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} + \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (1)$$

in which D is the sweeping distance and (V_x, V_y, V_z) is the direction vector of sweeping.

STEP 3: Transforming coordinates from the local system to the real world system. A coordinate transformation is a mathematical process whereby coordinate values expressed in one system are transformed into coordinate values for another coordinate system. Here is an example to illustrate the coordinate transformation. Assume a point P can be expressed in the X, Y, Z coordinate system (the local coordinate system) as $(x1, y1, z1)$. Likewise, in the X', Y', Z' coordinate system (the real world coordinate system), the point is defined as $(x2, y2, z2)$. We have the following transformation relationship between $(x2, y2, z2)$ and $(x1, y1, z1)$:

$$\begin{bmatrix} x2 \\ y2 \\ z2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ z1 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \quad (2)$$

in which $\theta_x, \theta_y, \theta_z$ are the angles of rotations about the X' -axis, Y' -axis, and Z' -axis, respectively, for making the axes of the X, Y, Z coordinate system parallel to the corresponding axes of the X', Y', Z' coordinate system (Foley *et al.*, 1995; Chen *et al.*, 1994); $(\Delta x, \Delta y, \Delta z)$ is the translation vector from the origin of the local coordinate system to that of the real world coordinate system.

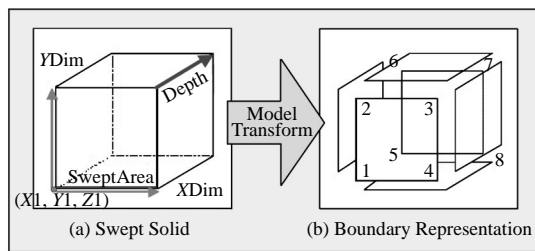


Fig. 2 Swept Solid Model v.s. B-Rep Model

STEP 4: Generating the GML object Model

Once all of the vertices are transformed to the real world coordinate system, the boundary faces (i.e., boundary representation) of the swept solid can be constructed using the following procedure:

- Assume V_i is the i^{th} vertex of the SweptArea and A_{sa} is the area of the SweptArea that can be defined by an ordered set of vertices $\{V_1, V_2, \dots, V_n\}$. The i^{th} vertex of the area on the opposite side of the SweptArea, V'_i , can be calculated using Eq. (1). The ordered set of vertices, $\{V'_1, V'_2, \dots, V'_n\}$, defines the area of the SweptArea on the opposite side, A'_{sa} .
- Each of the side areas, A_i , connecting A_{sa} and A'_{sa} can be obtained as the vertex set $\{V_i, V_{i+1}, V'_{i+1}, V'_i\}$ by looping $i = 1 \sim n$ and the value of i is set to one if $i = n + 1$.

To illustrate the process of model transformation, a column object is used as an example here. The left column of Fig. 3(a) shows the original IFC file format for describing the column object. When the data extraction process is executed, the necessary data, as shown in the right column of Fig. 3(a), are extracted from the IFC file (i.e., STEP 1) and the local coordinates of the vertices of the column object are calculated (i.e., STEP 2). Then, the local coordinates of the vertices are transformed to the real world coordinates system using the methodology discussed in STEP 3.

Finally, the boundary surfaces of the column object in the GML file are constructed using the procedure discussed in STEP 4. As shown in Fig. 3(b), the column object consists of six boundary surfaces, each of which is defined using an exterior LinearRing object. The LinearRing objects are defined using the vertices transformed from the IFC file.

III. DESIGN AND IMPLEMENTATION OF THE MODEL TRANSFORMATION TOOL

In this work, a user-friendly model transformation tool is developed to facilitate the IFC-to-GML transformation of selected building objects. The key requirements for the model transformation tool are identified at the early stage of the software development. They are summarized as follows:

- Allow the user to select the building objects from the IFC file for viewing and transformation. This means that the tool should support object-to-object transformation, instead of file-to-file transformation.
- Display the information of the selected IFC building objects, such as the definition of the coordinate system, geometric data, object placement, and so on.
- Transform the major IFC building objects, such as columns, beams, walls, and floors, into GML objects automatically.

The object-oriented technology is employed for developing the model transformation tool, which is named IFC2GML. Fig. 4 depicts the basic class diagram of the system design of IFC2GML. Besides the **frmMain** class responsible for the graphical user interface, there are four main classes defined in IFC2GML for transforming the IFC objects into GML objects. Their responsibilities and collaboration are briefly explained as follows. The handling of the IFC2GML workflow begins with the **CEntityTypeLister** class, which is responsible for parsing the IFC file and allowing the user to filter out undesired types of IFC objects then select objects from the filtered object list for viewing and transformation. Next, the **CShapePrinter** class is responsible for getting all of the geometric information of the selected IFC objects and displaying it. The **CGmlConverter** class uses the coordinate transformation method discussed in the previous section and the **CGeoCalculator** class to transform the local Cartesian coordinates of the vertices of an IFC object into real world coordinates. Finally, the **CGmlConverter** class groups the related vertices to form the boundary polygonal faces of the object and outputs the corresponding GML file.

As already mentioned earlier, the implementation of IFC2GML employs IFCsvr, an ActiveX component for handling IFC data. IFCsvr supports both Visual Basic (VB) and Visual C++ programming. In this research Microsoft VB.NET environment is adopted for its ease of use and for its object-oriented support on user-defined classes and data types, as well as reusable object classes for program extension.

IV. DEMONSTRATION

A demonstration of what can be accomplished visually through IFC2GML is presented here and a very simple building model with two beams, four columns, and one floor is used to test and demonstrate the functionality of IFC2GML. Fig. 5(a) shows the user interface of IFC2GML. Currently IFC2GML supports both IFC2x3 Final (Dec. 2005 released) and ifcXML data files. After the name and path of the IFC file are specified, IFC2GML automatically identifies the version of the IFC schema used and adopts the corresponding data model of the schema to parse and input

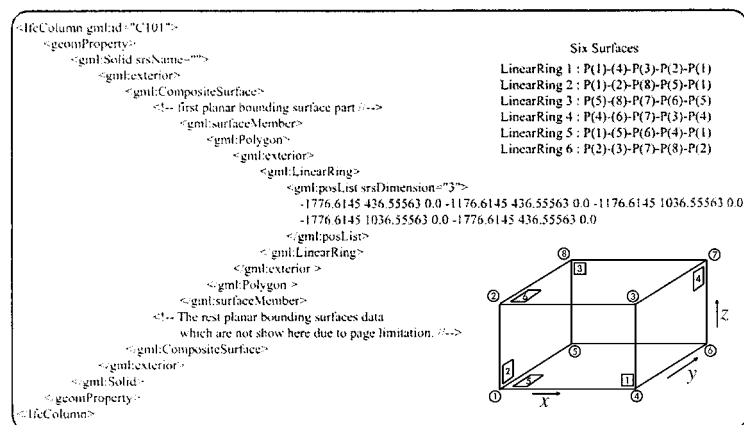
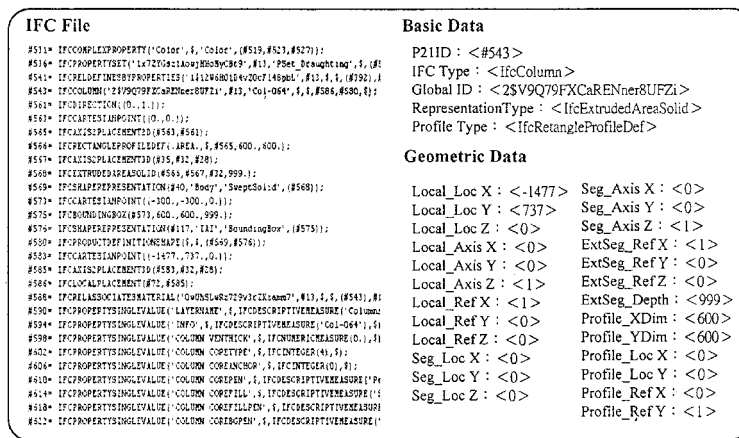
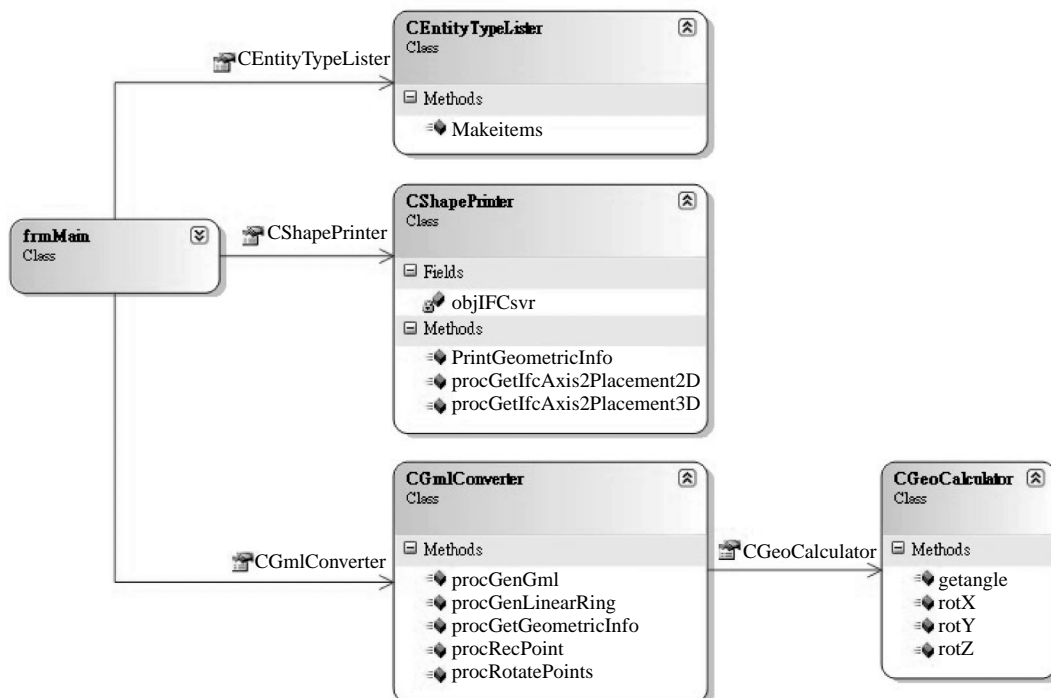


Fig. 3 Segments of IFC and GML files for the column example



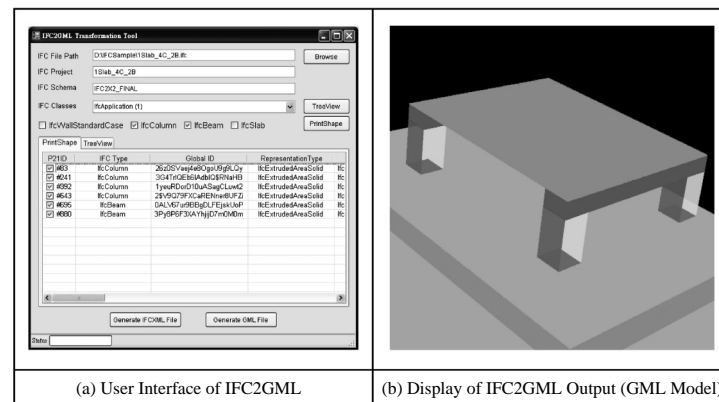


Fig. 5 IFC2GML

the IFC entities and their inherited attributes. It also counts the objects it collects from the IFC file. The users are allowed to toggle on the check boxes for several IFC object types so that only specified types of IFC objects are listed. Then, by checking the box in front of each object (see the P21ID column in Fig. 5 (a)), the users select the desired IFC objects in the list for information viewing and model transformation.

After the model transformation is completed, IFC2GML can generate the GML file for 3D-GIS to use, as shown in Fig. 5(b). The GML3 3D viewer application Aristoteles 0.9³ is used to test and verify the GML file in Fig. 5(b).

V. CONCLUSIONS

In this paper, an approach has been proposed for transforming the IFC model objects to the GML model objects so that 3D models constructed by CAD systems can be easily reused in 3D-GIS applications that support the GML model format. The proposed model transformation methodology is feasible and useful for transforming major building components of regular shapes from the IFC SweptSolid model into the GML B-Rep model to support a more realistic 3D-GIS environment for construction planning, information management, and decision making. A computer-aided model transformation tool called IFC2GML, that implements the proposed methodology and provides a friendly graphical user interface, has been developed and used to verify and demonstrate the feasibility and applicability of the proposed approach. The present implementation of the tool can transform the swept solids with either the rectangle or arbitrary profile for the shape of the SweptArea, and the straight sweeping line. Although transforming more IFC objects with more complicated SweptSolid geometry into GML model can be implemented, the proposed

methodology and tool have been shown to have practical engineering values for 3D-GIS applications.

ACKNOWLEDGMENTS

The authors would like to thank the R.O.C. National Science Council for the financial support under Grant No. NSC94-2211-E002-065.

REFERENCES

- Benner, J., Geiger, A., and Leinemann, K., 2005, *Flexible Generation of Semantic 3D Building Models*, Institute for Applied Computer Science, Research Center Karlsruhe, Germany.
- Chen, C. F., Chou, W. S., and Chen, Y. C., 1994, "The Mark-Based Approaches for Motion Estimation from 3-D Point Sets," *Journal of the Chinese Institute of Engineers*, Vol. 17, No. 1, pp. 43-53.
- Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F., 1995, *Computer Graphics: Principles and Practice (Second Edition in C)*, Addison-Wesley Professional, Reading, MA, USA.
- ISO 10303, 1992, "Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 1: Overview and Fundamental Principles," *ISO TC 184/SC4*.
- Liebich, T., 2004, *IFC 2x Edition 2: Model Implementation Guide*, International Alliance for Interoperability (IAI).
- OGC, 2004, *Geography Markup Language (GML) Encoding Specification 3.1.1*, Recommendation Paper, Open Geospatial Consortium (OGC).
- Zlatanova, S., Rahman, A. A., and Pilouk, M., 2002, "3D GIS: Current Status and Perspectives," *Proceedings of the Joint Conference on Geo-spatial theory, Processing and Applications*, Ottawa, Canada.

Manuscript Received: Sep. 11, 2006

Revision Received: Mar. 23, 2007

and Accepted: Apr. 16, 2007

³ More information can be found at <http://www.ikg.uni-bonn.de/aristoteles>