Metadata for 3D Models
How to search in 3D Model repositories?

Stefan Boeykens (1), Elena Bogani (2)
(1) Katholieke Universiteit Leuven, Department of Architecture, Urbanism and Planning, Kasteelpark Arenberg 1/2431, B-3001 Heverlee, Belgium
(2) Politecnico di Milano, Department BEST – Building Environment Science and Technology – ProTeA Research Unit – Computer Assisted Component Design, Via Bonardi 3, 20133 Milano, Italy.
stefan.boeykens@asro.kuleuven.be - elena.bogani@polimi.it

Abstract

In architectural education and practice, students, teachers and architects increasingly rely on online repositories with architectural information. This includes product model data, exemplary architectural projects and technical documentation, in a wide variety of formats. To be able to retrieve information from such repositories, they have to be well structured, preferably adhering to common classification standards. This article discusses how access to architectural repositories can be facilitated using optimized metadata. In particular, metadata describing 3D content is described, to improve structured searches. This research is elaborated within the MACE project.

Keywords
3D Models, Building Information Modeling, Content Repositories, Metadata, Classification

1. Introduction

This article looks at architectural repositories, mainly in the context of architectural education and learning. There are several 3D Model repositories available on internet. Typical examples include the Google 3D Warehouse [1] and the TurboSquid community [2]. Unfortunately, the major part of these 3D models exists in individual repositories and they are not accessible to geographically distributed professionals and students who need them. Some of them are freely accessible, while others require registration or even subscription. In many cases, artists, designers, students, architects and engineers created these models. Since future architects and engineers will increasingly rely on product model repositories as their project database, much work is to be done to make such repositories easy to use, intuitive and effective.

When architects and students in architecture query information, they might want to look for 3D Models of architecture, including buildings, building details, city models, technical installations, furniture and component design. However, to retrieve these Models, they need to look into many different repositories, each structured differently. Moreover, to be able to retrieve relevant models, it is important that information about them is available, through their Metadata. Some repositories allow for a structured search, e.g. looking for style, function or location, while many others are generic in nature, providing only a very crude classification, when considering architecture. They might have a "building" category, but seldom use additional classification. This severely limits their usefulness in an architectural context. This article discusses metadata to help structure repositories with architectural 3D content, to enhance structured queries.

The authors are involved in the EU-project MACE (Metadata for Architectural Contents in Europe) [3], which utilizes metadata to improve and enrich architectural content repositories. This article extends this approach with further specific information on the classification of 3D Models.

Example

To illustrate this, a first example is given from the Google 3D Warehouse. This online repository allows users to freely upload almost any 3D model, to make them available to other users. The process is fairly smooth, yet little attention is paid to properly edit or review the correctness or completeness of the given information. When searching for the term "glass façade", a first, rather short list is returned, in which the TNO Office was chosen, as shown in Figure 1. Some *tags* are also displayed, which are freely assigned by users uploading content. However, with this particular search, the “façade” term was retrieved from the model description.

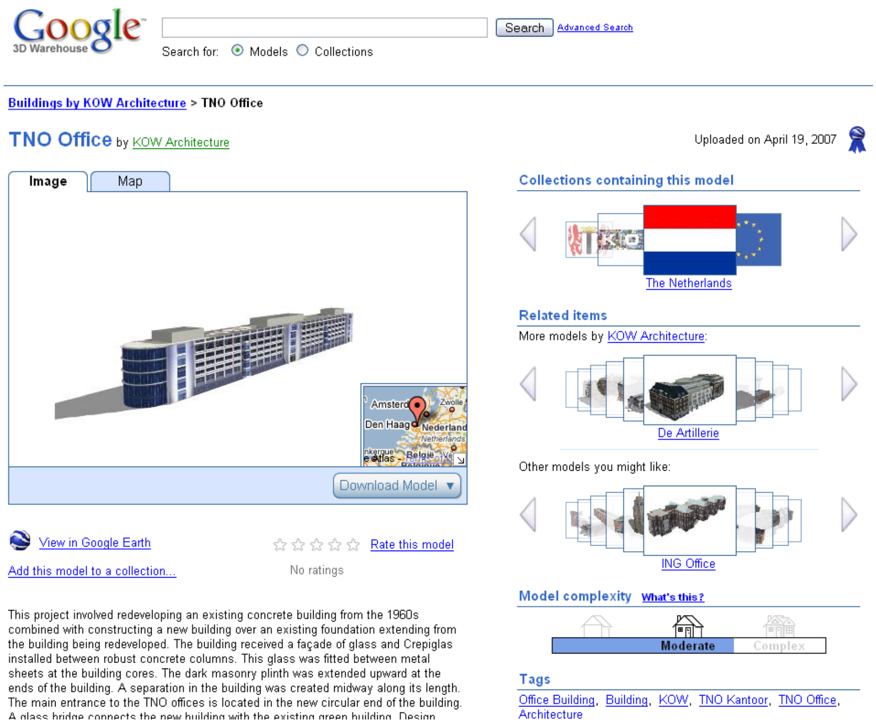

Fig. 1. TNO Office Building from the Google 3D Warehouse

While the search did yield a result, some remarks can be given. The use of free tagging is a simple approach for metadata collection, but it is mostly unstructured. The description field, while also searchable, is left to the creativity of the end user. A geographic location is provided in many cases, which makes it easy to position the 3D model on its actual location.

2. Architectural 3D Content

Architects, students, designers and graphic artists increasingly create 3D content. They apply a variety of software applications, ranging from generic 3D and CAD software, over dedicated architectural design applications, following the *Building Information Modeling* (BIM) approach. The BIM concept presumes a single integrated database of information about a building. However, it includes much more than geometry or a detailed 3D model. The digital building model has the potential to include additional information about the building and its usage. Through the correct application of a BIM model, it might be possible to retrieve who will occupy an office, what capital assets are located in which rooms, which routes lead to the nearest fire exits and other vital information. Objects in the building model are parametric, “intelligent” objects, with established relationships with their surroundings. Walls behave as walls, they are aware of their relationship with other walls. They also know how they relate to doors, windows and other objects in the model. But Building Information Modeling is much more than just object intelligence.
While many architectural practices usually avoid sharing their models, to respect confidentiality and privacy of their clients, designers and students are more eager to make their models available. There are even dedicated market sites where models can be shared or sold.
However, to be able to accurately retrieve models – apart from endlessly browsing through the repositories – metadata are required.

3. Learning Objects and Metadata

This article describes a suggested approach to properly classify and structure metadata for Architectural 3D Content. To facilitate this, the MACE project utilizes the *Learning Object Metadata*

(LOM) standard. By definition, a *Learning Object* (LO) is any entity, digital or non-digital, that may be used for learning, education or training. The *IEEE 1484.12.1 Standard for Learning Object Metadata* (LOM) [4] defines how the metadata describing Learning Objects have to be structured. In LOM, there is a complete description of the metadata that are required, such as articles, documents, images or courses. However, in the definition of LOM, no particular attention was paid to the description of 3D content and there are some limitations within LOM with regard to the description of Media Files.
Within the MACE project, the LOM standard has been extended into the MACE Application Profile, combining three different taxonomies: MEDIA, DOMAIN and APPLICATION. The Domain taxonomy is a combination of different architectural classification systems, optimized and reorganized into a single taxonomy, covering the whole architectural domain. This is used to classify content.

4. Metadata for Architectural 3D Content

To classify content, metadata are used. There are several metadata standards available, such as the *Dublin Core* Metadata structure [5], as shown in table 1. This is usable as a generic and agreed upon standard for the structuring of metadata. However, as can be noticed, no particular attention is given to the type of digital media files, such as 3D objects. While it would be possible to use this standard, there are too many ambiguities, to only rely on DC.

Table 1. Dublin Core Metadata

Data Element	Definition
Contributor	An entity responsible for making contributions to the content of the resource
Coverage	The extent or scope of the content of the resource
Creator	An entity primarily responsible for making the content of the resource
Date	A date associated with an event in the life cycle of the resource
Description	An account of the content of the resource.
Format	The physical or digital manifestation of the resource.
Identifier	An unambiguous reference to the resource within a given context.
Language	A language of the intellectual content of the resource.
Publisher	An entity responsible for making the resource available.
Relation	A reference to a related resource.
Rights	Information about rights held in and over the resource.
Subject	The topic of the content of the resource.
Title	A name given to the resource.
Type	The nature or genre of the content of the resource.

This article presents the MACE Application Profile, as an extension of the LOM standard. In general, three different groups of metadata are discussed:

1. Metadata about the *Learning Object* (**LO**), utilizing the LOM scheme (such as title, description, keywords, authors, copyright etc.).
2. Metadata about the *Media Object* (**MO**), containing information about file, size, location and required software. In this article, we specifically extend the generic Media object with *3D Objects* (**3DO**), which require information such as format, amount of geometric objects and scale. Media and 3D objects are digital files, which represent a building, a design, an item or a site;
3. Metadata about the *Real World Object* (**RWO**), represented by this model. In many cases, this will be the building, but it could as well be a furniture object from a manufacturer catalog. This mostly contains classification information.

These three types are illustrated in more detail in the following paragraphs.

Ambiguities within LOM

In LOM and the derived MACE Application Profile, there is an ambiguity when defining the author for a LO. It is suggested here to distinguish between the author of the RWO, the MO or 3DO and the LO.

The RWO is commonly a building. The author will be the architect or the design team who invented the building. The 3DO Author is the creator of the 3D Model. This could be the architect or someone from the architect's office, but it could as well be a graphical artist. The author of the LO is the person who included the 3DO into the repository and who created the metadata, possibly a student or a teacher. The three possible authors might be one and the same person, but in many cases, they are different. That is one of the reasons to make a clear distinction between the Real World Object, the 3D Object and the Learning Object.

Learning Object Metadata (LO)

Table 2 describes a few additions that could be applied within the MACE Application Profile, to improve its usefulness in the context of 3D Architectural models.

Table 2. Fragment from the MACE Application Profile, to be improved for usability with 3D Models

	Field Name	Explanation	Value Space
1.9	learning object kind	The kind of the learning object: to distinguish between a real world object (i.e. a building) or 3DO	real object (RWO) media object (MO) 3D Object (3DO) learning object (LO)
2.1	version	The version of the 3D Model	
2.3	contribute/ authors	Architect of the real object or Authors of the 3D Model	see remarks about ambiguity of author!
2.3.1	role	Specifies the kind of contribution in respect to the 3D Model and extending the value space with architecture specific roles	Extend the vocabulary with architecture specific roles, e.g. designer, architect, constructor, engineer, 3D Artist
4.4	requirement	The technical capabilities necessary for using this learning object.	
4.4.1.1	type	The technology required to use this 3D Model (e.g., hardware, software)	
4.4.1.2	name	Name of the required technology to use this 3D Model.	The vocabulary is not specified, but some examples are ArchiCAD, Allplan, Bentley Architecture, Revit, AutoCAD Architecture, SketchUp, Ifc Viewer, QuickTime, Windows Media Player, VRML viewer…
5.2	learning resource type	Defining the specific kind of LO. Describes the educational kind of the LO (e.g. if it is a building, a project, etc.)	Extending the vocabulary with architecture specific types of learning resources: drawings, graphic representation, render, 3D Model, technical drawing, photo

The Media Taxonomy and Media Objects (MO)

At the same time, the *Media Taxonomy* from the MACE Application Profile has many fields that are meant to describe common media objects, such as images, sounds and movies. However, in the case of 3D objects, some of these characteristics are irrelevant, such as image resolution, display scale, color depth and animation duration. Table 3 displays an example of the application of this taxonomy for 3D Models.

Table 3. The MACE MEDIA Taxonomy, applied on 3D Models

MACE ID	Name	text	E.g.: Ventilated Façade or Basilica S. Pietro or Sagrada Famila
M.02	Author	text	Name of the 3D Model Author
M.03	Source	URL	The Web Site where the 3D Model is available
M.05	Creation date	date	Creation Date of the 3D Model

M.07	Mime type	text	The file format
M.08	Representation		Attributes describing the quality, the size and software for the representation
M.08.1	Scale		Representation scale of drawings
M.08.2	Duration	seconds	Time duration of videos - for example VRML Video
M.10	Topic type	text	The type of the 3D Model topic for example: technical drawing or building
M.12.1	Version		Identification of the Software version
M.13	IPR		Intellectual Property Rights of the media
M.13.1	Owner		The owner of the asset/content of 3D Model
M.13.2	Access terms	text	Under what terms it can be accessed
M.13.3	License costs	Boolean	License costs

It can be noticed that the application of the Media Taxonomy on a 3D model lacks many of the specific characteristics of common 3D Models. This explains why the next section presents possible extensions to this taxonomy.

3D Object Taxonomy for 3D Objects (3DO)

The digital 3D Object is commonly created in a particular design application and is stored in a particular file format. The actual data is this file, while the metadata could be used to describe the content of the file in a searchable, textual format, usable for browsing and querying.
While it might be possible to create search mechanisms that index information directly from the digital 3D files, it is not trivial. A first reason is the complexity of acquiring this information. Many different formats exist and they are commonly connected to particular applications. While some of these formats are documented and available as Open Standards (e.g. VRML [6], Collada [7], STEP [8]), many others are proprietary, closed formats, which are usually not documented and might even be protected, using encryption techniques. The developer of the creation software might provide viewing applications to enable the inspection and visualization of these models.
A particular example is the format for *Autodesk 3ds Max* [9]. This is a *Digital Content Creation* (DCC) application, for modeling, rendering and animation and is widely used in architectural visualization. The *.max format is not documented and can only be read by the 3ds Max software. It directly reflects the model content, including information stored in the applied plug-ins. To be able to recreate the model, it is required to have access to all plug-ins that were in use when the document is stored, which in practice leads to the requirement of having the complete application installed, together with all external plug-ins, which might not be included with the software by default.
The Probado project [10] does provide such an approach, but it is not applicable to all 3D objects.

Methodology
The next sections illustrate different aspects of 3D Models. To derive these characteristics, we have examined two main sources of information: online 3D repositories and 3D file formats information.
Online 3D repositories, such as the already mentioned Google Warehouse and Turbosquid, already have some way to classify their content. They typically include a unique identifier, the author name, a date (used to indicate submission to the system) and common fields such as description and keywords. These characteristics are adequately dealt with using the LOM standard. In addition they commonly provide information about the file format and version, for which the Media Taxonomy is well suited. Finally, they provide some, rather limited, information about the 3D model itself, such as Booleans indicating the availability of textures, materials or animation and numbers to indicate the amount of vertices and polygons.
3D File formats on the other hand, clearly define the possible type of data that could be found in a 3D model, stored in that format. In particular, the support of textures, mapping and animation is not available in all formats. A format using the *OpenNURBS* specification [11], as an example, can store mathematically unambiguous descriptions of curves and double-curved geometry, which is simply not available in a typical polygonal format, such as the old 3ds format from *3D Studio for DOS*, by Autodesk.

Common characteristics of 3D Models
To enable access to a 3D Model, for repository searches, a group of metadata has to be associated with the file. This includes common fields, such as title, description, creation date and file size, for

which the above-mentioned taxonomies are adequate. Additionally, several fields could be defined which are specific for 3D models, such as the number of vertices, the references textures and the geometric primitive types that have been used. Table 4 displays an example of the information that could be attached to a particular 3D model.

Table 1. Example of common metadata for a particular 3D model

Field Name	Explanation	MACE Application Profile
Title		1.2
Description		1.4
Type	real, planned, historical, imaginary	not available
Creation Date		3.3.2 (Creatio of MO/3DO)
Submission Date		3.3.2 (Creation of LO)
File Format	3ds, Dwg, IFC, VRML…	4.1
Software	3ds Max, ArchiCAD, SketchUp…	4.4 (and 4.4.1.1/4.4.1.2)
File Size	202,240 bytes	4.2
Compression	zip, gz, tar, rar, none	4.4 (and 4.4.1.1/4.4.1.2)
Author	the 3D artist	2.3 (contributor of MO/3DO)
Number of Vertices	250.000	not available
Geometry Type	polygonal, NURBS, CSG	not available
Number of Triangles	2.957,936	not available
…		

Other labels or Metadata that further classify the 3D Model can include free keywords, separated by a comma or a point, as specified within the LOM standard. This can include more descriptive information about the object, such as its function or intention: library, ancient construction, warehouse, railway, re-use.

Table 5 provides information about the software that was used to create the model. It is important to include the software version too, as incompatibilities between different versions of an application are very common.

Table 2. Creation Information

Field Name	Format	Explanation	MACE Application Profile
Application	Text	Software in which the 3D Model was created (e.g. ArchiCAD, Allplan, Revit Architecture, Bentley Architecture, etc..)	4.4.1.2 software
Application Version	Text	Specific Release (e.g. 11). Only up to the level that compatibility might be at stake, so no need for 11.123.xyz	could use 4.5 and 4.6, but ambiguous

Table 6 displays information about the digital file itself. It is important to provide information about the file version as well, to help with accessing the file with the correct software and software release. There might be a difference between the application version and the file version, e.g. AutoCAD 2007, 2008 and 2009 share the same file format (even though minor differences occur, when new objects are introduced).

Table 3. File Format information

Field Name	Format	Explanation	MACE Application Profile
Format	Text	e.g. AutoCAD Drawing	4.1
Extension	Text	File extension (e.g. dwg, dxf, dwf)	not available
Version	Text	e.g. 2008	could use 4.5 and 4.6, but ambiguous
Size	Number	x Bytes	4.2

While these first tables are partially supported within the MACE Application Profile, the following tables are specific for 3D Objects and are simply not covered in the current taxonomy.

Table 7 displays common information about the 3D model, e.g. the number of vertices or faces. However, it is not trivial to define a series of values that apply to most 3D models. They might have a completely different way to define geometry. CAD applications and DCC applications are usually not working with the same types of geometry. However, it is questionable if this information is really useful to the end user. The file size might indicate complexity, while the file format might indicate the expected type of geometry.

Table 4. Geometric information

Polygonal Mesh	Number	Number of faces or n-sided faces or vertices
NURBS	Boolean	Are NURBS used in the model?
Parametric Objects	Boolean	Are parametric relationships available in the model?
Complexity	Value	Coarse – medium – fine/complex

Table 8 describes the existence of particular type of file content. This is expressed as a *Boolean* value (yes/no) or a number (e.g. 24 layers).

Table 5. File Content

Materials	Boolean	Is material information available?
Textures	Boolean	Are textures being used? (texture maps and map projection information is available, possibly in external files)
Shades	Boolean	Are advanced shaders being used? (e.g. Renderman Shading Language shaders)
Physically Correct Materials	Boolean	Do materials represent physically correct values?
Layers	Boolean/Count	Is layer information available? How many layers are used?
External Files	Text	listing of all possible external documents that are required to load this file e.g. xref, textures, library objects (preferably using correct resource location)
Blocks/Instances/Components	Boolean	Are instances being used?
Lights	Text	Information about light sources? For example: Light sources – typology or ubication
Physically Correct Lighting	Boolean	Is physically correct lighting information available?

Much of this information can be actually retrieved from the descriptions of commonly known file formats. E.g. the *.3ds format from the old 3D Studio for MS-DOS only supports polygonal meshes, but is still widely used as a file exchange format. We created such a list for many common formats, where it is explained which kind of data they support (e.g. 2D, 3D, animation, visualization, building model) and the characteristics of the file format (e.g. ASCII, xml, Open Standard). However, while the format might enable the storage of certain information, the actual file is not per definition using all of it. E.g. while a particular format might support meshes, splines and textures, it is possible that the model only contains a polygonal mesh without any material information applied.
Table 9 describes the scaling factor. This defines the physical size of one file unit. This is extremely important in architecture and is often missing in current model repositories.

Table 6. Size and Scaling information

Unit size	Real	1 unit in the file equals X meters

Table 10 gives information about the availability of certain types of animation information. While this is not too common in architectural models, it is useful to be able to indicate the existence of animation information in the metadata.

Table 7. Animation information

Key frame Animation	Boolean	Is Keyframe animation used?
Motion Capture Animation	Boolean	Is Motion Captured animation used?

Real World Object Metadata (RWO)

There is often no obvious difference between metadata describing a building project and metadata from a digital model of that building. The values for fields such as designer, location, date (timeframe),

context, building type, style, construction and others pertain to both the RWO and the 3DO and, eventually, the LO.
However, there is the possibility that this 3D Model lacks a counterpart in the real world. An imaginary building could belong to a design studio assignment or could be created from the imagination of a graphics artist. Another example is the model of a particular chair, from the catalog of a manufacturer. It represents a specific model, but there is no singular relationship with a particular real world chair. It might even be a generic product, to be used in a technical document or as part of a visualization, e.g. an unbranded chair.
To specify the relation between the 3DO and a possible RWO, it is suggested to distinguish between different levels of "reality":

- *Real object*, which exists as a tangible subject;
- A *planned* or *designed* object, which could be realized. Typical examples include models for architectural competitions, from which usually only a single winning design will finally be realized as a RWO;
- An *historical* object, such as a reconstruction of a demolished or perished building. It could also pertain to the recreation of a real building in a particular phase in time, e.g. before a certain restoration phase.
- An *imaginary* object, which could be almost anything, without the necessity of it being realized. This not only includes architectural design studio results, but even pure imagined models, such as a spaceship or a fantasy dungeon.

5. File Formats and Open Standards

It could be argued that the access to repositories with 3D Content would be facilitated if they would all adhere to a common format, preferably an Open Standard. To develop and implement a new interoperative standard to share 3D files of different formats without loosing any information is too ambitious and probably not desired. This is not only caused by the complexity of defining a standard which supports all common characteristics of 3D models, but also because many different 3D formats already exist and are actively being used. There is a large variety of both proprietary and open standard formats e.g. DWG/DXF/DWF, VRML/X3D, IGES, STEP, OpenNURBS, Collada, FBX, OpenFlight, JTOpen, 3Dxml, HMF/HSF, IFC and many others. There are also some widely used older formats, such as 3ds and OBJ, which are originating from a previous generation of design applications, but which are still being used as file exchange formats. Introducing another format/standard will also place a huge barrier for the development of content repositories. It not only requires content providers to convert their existing databases into such a new format, but it also demands the development of additional conversion tools by the software manufacturers or by third-party developers.
In summary, the last two decades have proven that no single 3D standard seems to be accepted by the different involved parties.

Conversion of 3D Content

To be able to retrieve information from different content providers, it is thus required to cover many different existing formats. Creating conversion tools for all these formats is a huge undertaking. However, some commercial software developers already have this file conversion/translation services in place, such as *Okino* [12], with their Polytrans/Nugraph applications and *Right Hemisphere* [13] with the Deep Exploration and related tools. In the case of Polytrans, an internal geometric database is applied to serve as a common translation target for all conversion modules. This is extended with domain specific modules, such as CAD and Digital Content Creation.
The ambition to convert 3D data without loosing any information is problematic, though, since the different formats simply do not cover the same geometric information. In the case of architectural content, targeting only geometrical information might be insufficient. There are more benefits in coping with product model information, as used in BIM applications. However, while it seems wise to adhere to a neutral product model format, such as IFC, it might severely limit the amount of 3D Models that could be covered. Most 3D models in content repositories are created in generic modeling software, containing only geometry and no information about the building.

6. Structured Building Models

It is important to stress the value of structured building models. While it is theoretically possible to define shape recognition algorithms to retrieve information about pure geometric 3D or 2D models, it is much easier and more efficient to search through structured data. A digital building model contains actual information about the represented building. It would be feasible—while not trivial—to generate 3D model metadata from a structured digital building model.

Generating such metadata from regular CAD or 3D models is more cumbersome. There is only geometry to look through. E.g. two parallel lines on a layer called "walls" might represent a wall, but that is about the amount of information that can be retrieved from a regular 2D CAD document. 3D CAD models can also provide surfaces, meshes and solids. When such a file is properly structured, it might be feasible to retrieve quantitative information about the materials that have been used and the function of certain shapes or objects. In many cases, however, 3D models are mostly unstructured.

IAI and IFC

The *International Alliance of Interoperability* (IAI [14][15]) is active in developing and promoting the use of a digital building model, through the development of the *Industry Foundation Classes* (IFC).

The IAI is organized in Chapters. A Chapter represents a country or a group of countries acting together and organizations within the alliance are members of regional Chapters. In Europe there are German, French, Italian, Iberian and Nordic (Denmark, Finland, Norway and Sweden) Chapters. The IAI promotes interoperability in the AEC/FM industry by publishing an open, freely available, non-proprietary data model specification, known collectively as the Industry Foundation Classes. Software applications supporting the IFC data model are able to exchange data with other applications that also support this model. The advantage of a non-proprietary data model is that the content, integrity, and reliability of the data can be independently verified.

The IFC protocol became an international standard: *ISO/PAS 16793* (PAS: Publicy Available Specification).

In the IFC format, digital objects represent “real” objects in a building. They have geometry, including a 3D description. The IFC format allows passing this information to all participants throughout the different design stages of a building project. The principal benefit of IFC lies in the object description. The format preserves the full geometric description in 3D, but it also transfers information about location and relationships, as well as all properties (or parameters) of each object, such as finish, serial number, material description, thermal conductance and cost.

The scope defined by the IAI for the IFC Object Model is “enabling interoperability between AEC/FM applications from different software vendors”. The AEC/FM industry is, by its nature, fragmented and distributed. It also encompasses a very large set of object model requirements. The aim of the IFC Model is to be able to support the exchange and sharing of information throughout the project lifecycle.

IFC utilizes the EXPRESS-G graphical modeling notation developed within STEP. Currently, IFC2x3 is the latest official IFC release recommended for implementation.

IFC versus Geometric 3D Models

With regard to the availability of architectural content, it is assumed that 3D Model repositories mostly contain geometric 3D models. The public creating these models is limited and only a handful of applications are able to create structured digital building models to begin with. Typical examples include ArchiCAD, Revit, AutoCAD Architecture, Bentley Architecture and Allplan. By far, the large majority of architectural models on internet are geometric models, made in software such as AutoCAD, MicroStation, FormZ, 3DS Max and obviously SketchUp.

In the case of the popular SketchUp software, it is interesting to note that IFC support is being developed, but this is currently only to import IFC models, not to create them. At this moment, there is a beta version available of IFC2SKP [16], which works inside SketchUp and has the ability to load IFC data created in BIM applications. The plug-in will load geometry from the IFC file format, but it will also display the BIM data of imported objects.

7. Searching through Data or Metadata?

The discussion of retrieving information from 3D Model repositories is quite similar to the retrieval from other repositories. In current architectural repositories, containing project images or articles, the same reasoning can be followed.

Inherently, a photograph or an image is an unstructured series of pixels. No information about the objects is available. Currently, shape recognition techniques are still improving. This includes techniques such as *Optical Character Recognition* (OCR) and *Facial Feature Recognition* (e.g. tracking eyes and mouth from video streams). That said, these techniques are still complex and require extensive calculations. On the other hand, the retrieval of shape information from a vectorial drawing or model is almost trivial. So having the information in a usable format is probably the best strategy. The *Probado* Project [10] already developed methods for using non-textual data inside libraries and still derive meaningful information (metadata) from the models, directly.

However, by properly tagging photographs and drawings, search mechanisms can retrieve proper information through Metadata searches. A similar strategy would be beneficial when searching through 3D Model libraries.

Current 3D Model repositories usually only provide a limited amount of usable metadata, as explained in the initial example of this article. Efficient methods and strategies have to be developed to generate and manage metadata for such repositories. While this might be a considerable effort to add to existing libraries, it will make the models and the information they carry retrievable by search mechanisms and online queries.

8. Conclusion/Evaluation

This article proposed the creation of a Metadata structure for 3D Models. The proper classification and tagging of 3D Content will assist the providers of this content to improve the structure of their repositories and make the content easier to search and retrieve.

The creation of specialized 3D Formats is deemed unnecessary, as the current range of formats will likely continue being used in the foreseeable future. However, the importance of structured building models should not be underestimated. It is therefore of utmost importance that the development of the IFC initiative will continue and that the usage of Building Information Modeling applications increases in the AEC domain.

The combination of structured content and optimized metadata will increase the accessibility and availability of 3D content, which will ultimately benefit the content providers and their end users.

Acknowledgments

Parts of this research have been carried out within the MACE project [3]. MACE (“Metadata for Architectural Contents in Europe”) is a research project started in September 2006. It is a project co-financed by the European Commission within the “eContentplus” program, a multi-annual European Union initiative to make digital architectural content in Europe more recognizable, accessible, selectable on the base of shared and dynamic ontology, reusable within multiple university education and life- long learning curricula for professional user.

The main assumption of MACE is that the expert knowledge sets of a number of stake holders, protagonists in the architectural domain, have to perform as the back-bone of the desired boundary-less repository of architectural knowledge, to be shared by the members of a world-wide community of users. Thence MACE initiates the proposition of a commonly shared ontology, terminology, series of concepts structured in a domain index, conceptual maps for the representations of the dense tissues of relationships connecting the multiform and trans-disciplinary contents of architecture.

Its main objective consists of the content enrichment of a huge quantity of knowledge resources dedicated to architectural education all over Europe, already expressed in the digital form, thence eventually available to a vast audience of students, teachers, professionals, researchers through the web. The aims of MACE concern the definition of a number of users’ profiles, based upon the nature of their practice as stakeholders operating in the architectural domain: user profiles the designated requirements of whom have to be compared with the access path to selected learning objects in the web-connected repositories.

References

[1] Google 3D Warehouse, http://sketchup.google.com/3dwarehouse
[2] TurboSquid, http://www.turbosquid.com
[3] MACE (Metadata for Architectural Contents in Europe), ECP-2005-EDU-038098, http://www.mace-project.eu
[4] IEEE 1484.12.1 Standard for Learning Object Metadata, http://www.imsglobal.org/metadata/mdv1p3/imsmd_bestv1p3.html
[5] Dublin Core Metadata Standard, http://dublincore.org/documents/1999/07/02/dces
[6] VRML / Virtual Reality Markup Language, International Standard ISO/IEC 14772-1:1997, http://www.web3d.org/x3d/specifications/vrml
[7] Collada / COLLAborative Design Activity, http://collada.org/
[8] Jon Owen, STEP - An Introduction, Information Geometers, 1993.
[9] Autodesk 3ds Max, http://www.autodesk.com/3dsmax
[10] H. Krottmaier, Frank Kurth, Thorsten Steenweg, Hans-Jürgen Appelrath, Dieter Fellner PROBADO - A Generic Repository Integration Framework, Research and Advances Technology for Digital Technology: 11th European Conference, ECDL 2007, Budapest, Hungary, September 16-21, 2007. Proceedings, 9/2007
[11] OpenNURBS, http://www.opennurbs.org
[12] Okino Polytrans/Nugraph, http://www.okino.com
[13] Right Hemisphere Deep Exploration/Server tools, http://www.righthemisphere.com
[14] International Alliance of Interoperability (IAI), http://www.iai-international.com
[15] Model Support Group MSG within the IAI – International Alliance of Interoperability http://www.iai-tech.org
[16] IFC2SKP, http://www.ohyeahcad.com/ifc2skp/index.php