BUILDING EXTENSIBLE 3D INTERACTION METADATA WITH INTERACTION INTERFACE CONCEPT

Jacek Chmielewski
Department of Information Technology, Poznań University of Economics, Poland

Keywords: Metadata, Virtual reality, Interactive 3D, Interactive multimedia, XML.

Abstract: Proliferation of 3D technology is visible in many aspects of current software developments. It is possible to find elements of 3D in many applications, from computer games, to professional engineering software, to end user software. It is especially important in the end user field, where 3D technology is usually used to provide natural and intuitive user interface. Fast development of such new applications can be driven by reuse of interactive 3D objects that are built for one application and reused in many others. Nowadays, number of such objects are stored in shared repositories and pose a challenge of finding right objects for new application. To efficiently search for interactive 3D objects it is required to provide metadata of object geometry, semantics and its interactions. Existing metadata standards can deal only with the first two areas. In this paper a new approach to interaction metadata is presented. The proposed Multimedia Interaction Model is accompanied by an Interaction Interface concept that allows creating interaction metadata not limited to a predefined set of description elements. Interaction Interface concept provides a method and tools for defining new description elements in a way suitable for automatic processing and automatic analysis by search engines.

1 INTRODUCTION

The usage of 3D technology is more and more often. In the entertainment domain 3D is used for years, especially in computer games (Stone, 2001). Also engineering software uses 3D technology extensively. It is used mostly for design and analysis of buildings, cars or other constructions (Ottosson, 2002). Nowadays, applications of 3D technology can be found not only in entertainment and professional software. Performance of current home PCs is so good, that 3D technology is starting to be used in new systems and computer programs for average end user. It is especially noticeable in the area of interactive 3D interfaces that aim to mimic real world and provide user interface, which is natural and intelligible for average home user.

A side effect of fast development of new software that uses 3D technology is a growing number of interactive 3D objects. Potentially, such objects can be reused in new applications without the need to build them from scratch. This fact can significantly speed up the development of new 3D applications and pave the way for faster popularization of 3D technology. On the other hand, the growing number of reusable interactive 3D objects creates new challenges. Large number of objects makes it harder to find an object that is suitable for a particular application. The success of development approach based on reusable objects depends on the availability and effectiveness of tools used to find appropriate 3D interactive objects.

Searching for interactive 3D objects is not an easy task. Most of the important features of an object, especially related to interactions, cannot be extracted from the object definition. It is necessary to introduce additional object description that will provide information about object semantics and object behavior. Such additional description is called metadata. To be useful for search engines, metadata has to be prepared according to a predefined schema. Semantics of each element of such schema has to be known to allow interpretations of user queries. At the same time it is required to allow introducing new metadata elements that will be used in descriptions of interactions specific for a given domain or application. None of existing metadata standards provides such capabilities and therefore there is a need for a new approach to metadata creation.

Chmielewski J. (2009).
BUILDING EXTENSIBLE 3D INTERACTION METADATA WITH INTERACTION INTERFACE CONCEPT.
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pages 160-167
DOI: 10.5220/0002292801600167

2 BACKGROUND

2.1 Types of Interactions

In the field of virtual reality systems interactions can be classified as: human to human, human to object and object to object interactions. The first type – human to human interactions – concern new 3D communication solutions that allow for communication between two or more persons conducted via the 3D virtual environment. The area of human to object interactions refer to human-computer interactions (HCI) or to 3D design and simulation systems, where actions taken by humans on 3D objects are followed by pre-programmed reactions of those objects. The last type of interactions – object to object interactions – is related mostly to 3D virtual worlds, where objects have behavior defined and can act independently of human interventions.

Such classification suggests that interactions of each class should be described differently, which potentially makes interactions metadata creation and analysis more complicated. However, humans are on the real side, while objects are on the computer side and both concepts should not be mixed together. The need is to describe interactions of 3D objects, not humans. Therefore only the last two types of interactions have to be considered. If viewed from the computer point of view, human is just a special case of an object – usually an avatar object (3D representation of a human). Therefore, it is possible to generalize human – object and object – object interactions to a single class: object – object interactions and describe all of them in the same way. Such approach makes it possible to build a single, yet universal model of an interaction that can be used as a base for interaction metadata.

2.2 Interactive Objects

It is possible to distinguish at least three different types of interactive 3D objects (Walczak, 2008). The first type represents objects that are standard 3D geometry figures modified if needed by a virtual environment management process. In case of such objects all interactions are controlled externally and the object itself is static. The second type of interactive objects represents objects whose definition contains implementation of object modifications, but the modification itself is still triggered by a virtual environment management process. Third type of interactive objects represents objects whose definition contains implementation of the whole object behavior. Such objects know when and how they should react to some external events, and therefore can be called: autonomous interactive objects.

In order for the interactive 3D objects to be reusable, they have to be independent of a particular application or virtual scene. To satisfy this requirement behavior of such objects has to be embedded in the object definition and not imposed by a 3D scene or a system that manages a virtual world. The behavior code does not have to be physically part of file with object definition, but it has to be logically bound to this particular object (Walczak, 2006). Only autonomous interactive objects satisfy this requirement. It is possible to state that all interactions of an autonomous interactive object are an effect of execution of object own behavior. As a result, interaction metadata can be tightly associated with a specific object and can be moved and stored with the object.

2.3 Interaction Metadata

The topic of multimedia object metadata is well researched and developed. There are a number of metadata standards that can be used to describe object general properties or object format and purpose specific characteristics. The most general metadata standard is Dublin Core, which is a set of fifteen elements designed to foster consensus across disciplines for the discovery-oriented description of diverse resources in an electronic environment. Dublin Core Metadata Element Set [DCMES] is intended to support cross-discipline resource discovery and it does not satisfy every possible declarative need in every discipline. Thus, in most applications it is used in a combination with more advanced, domain-specific metadata standard. DC elements are usually a subset of resource metadata and are used as a minimal metadata for data exchange and discovery.

Interactive 3D objects usually contain many multimedia components like: images and video as textures, audio, text, complex 3D geometry, therefore the complete description of such objects may be composed of various domain specific metadata. For still images EXIF [EXIF], DIG35 [DIG35] or NISO Z39.87 [Z39.87] can be used. Video can be described with AAF [AAF], MXF DMS-1 [MXF] or P/Meta [P/Meta]. Audio components of the interactive 3D object can use MusicBrainz (Swartz, 2002) or simple ID3 tags [ID3]. However, complex multimedia resources like interactive 3D objects are usually described with the most universal metadata standard for multimedia

objects – MPEG-7 (MPEG-7, Martinez et al., 2004). It addresses not only general and format specific information, but also object semantics, geometry and spatio-temporal composition. MPEG-7 was formulated with audio and video contents in mind and its handling of 3D objects is limited. Nevertheless, in the past few years, some 3D metadata solutions were developed (Bilasco et al., 2005, Lorenz et al., 2006, Pitarello and Faveri, 2006). Especially worth mentioning, is the 3D SEmantics Annotation Model (3DSEAM) (Bilasco et al., 2006) developed by a group from Laboratoire Informatique de Grenoble, lead by Ioan Marius Bilasco. 3DSEAM extends MPEG-7 localization descriptors with Structural Locator and 3D Region Locator tools, making MPEG-7 the best available metadata solution for 3D multimedia objects.

However, even with such great pool of metadata standards and solutions like MPEG-7 with 3DSEAM extension (Bilasco et al., 2006) there is no universal and extensible solution capable of describing interaction properties of interactive 3D objects. Existing metadata solutions can be used for describing object semantics and geometry. However they are not sufficient for describing object interactions. Existing metadata standards for multimedia objects were designed to describe a static or linear content, i.e. still images, documents or movies. In case of interaction descriptions the problem is more sophisticated. An interaction is not a simple 'content'. Interaction influences modification of the content, i.e. the object, but the interaction itself is an action based on the object behavior. The behavior is represented as an algorithm expressed by a computer program. The interaction metadata does not have to describe the whole algorithm, which may be very complex or confidential. To enable analyzing object interaction properties and chains of subsequent interactions it is enough to describe only the result of execution of such computer program. Interactive objects change their state according to interaction results and interaction context, and new object parameter values cannot be determined a priori. Existing metadata solutions are not applicable for describing multiple states of an object (i.e. all possible interaction execution results). Therefore there is a need for a new metadata solution.

To deal with this situation, a new approach to interaction metadata is proposed. The proposed approach is based on the following assumptions. To be interactive, objects need to have a behavior. The behavior is encoded in a form of a computer program. The computer program of an interactive object has at least one communication interface – an API, which allows exchanging information with the 3D virtual world or other interactive objects. Such API includes a number of functions and attributes that reflect the state of an object. Knowing the API and the computer program it is possible to precisely describe object interaction characteristics and possible object interactions, as well as match compatible objects. Following the assumptions stated above the Multimedia Interaction Model is proposed, accompanied by the Interaction Interface concept.

3 MULTIMEDIA INTERACTION MODEL

The Multimedia Interaction Model (MIM) is a conceptual model of an interaction described from the point of view of an object (Chmielewski, IT 2008). It means that the interaction description is build around object reaction and a context that triggers the reaction. This approach comes from areas of computer science dealing with active environments, especially active databases. Research on the active databases resulted in paradigm called Event-Condition-Action (ECA) (Dayal et al., 1988), which was introduced in late eighties by the members of the HiPAC project. The semantics of ECA rules are straightforward: when an event occurs, evaluate the condition associated with the event; and, if the condition is satisfied, trigger the action. The ECA rules are used up to date to express active DMBS features and event-reaction oriented features in other systems (Bry and Patrânjan , 2005, Papamarkos et al., 2003, Thome et al., 2005, Zhou et al., 2004). The MIM model is based on the ECA rules paradigm and uses Event, Condition and Action components to describe the whole interaction. Decomposition of an interaction into these three independent areas is important, as allows using specific tools to describe different aspects of an interaction. In the MIM model there is a metadata element that corresponds to each ECA component. The Event element corresponds to the trigger of an interaction. The trigger is a change of a selected property of some object or a virtual environment in which the interaction occurs. The Condition element describes the context of an interaction. The context represents a state of selected objects or the virtual environment that is required to allow the action. The Action element matches the object reaction. The reaction describes how the object changes itself in a reply to the event in a given context. Each metadata element provides formal and semantic description

tools and is associated with an XML Schema [XML Schema] defining how interaction metadata documents can be encoded in XML format. Details of all three elements are described below.

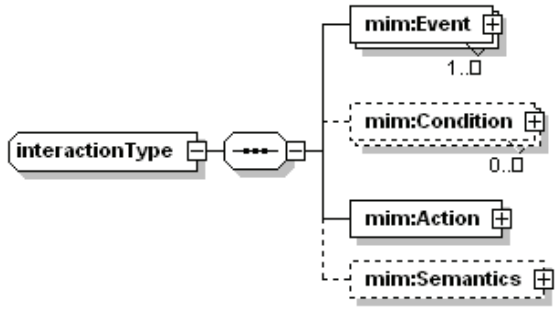

Figure 1: Multimedia Interaction Model overview.

3.1 Event

An event is a change of the state of a trigger object. The state change is defined as a change of a value of any object parameter. In the MIM model, the Event element is related to a change of a single parameter, but interaction metadata can include many Event elements. The Event element contains a set of preconditions that need to be satisfied to trigger a particular interaction. These preconditions include information about what has to change to trigger an interaction, and where in 3D space the change has to occur to be noticed. The first part – what has to change – is described by an object parameter type. To trigger an event, object parameter that is modified, have to be of specified object parameter type. The second part, which describes where in 3D space the change has to occur, is a specification of a fragment of 3D space, called 3D interaction space. To satisfy the location precondition, the event has to occur inside the specified 3D interaction space.

The specification of object parameter type (represented in metadata by Parameter element) is accompanied by optional arguments. It is due to the fact, that some parameters are functions, and therefore need some initial arguments to provide a value. An example of such parameter is 3D Dimension, which is a distance between two furthest points of an object measured along a given vector.

The 3D interaction space is represented by a set of 3D spaces and 3D viewpoints. In the MIM implementation, both elements are defined with tools taken from the X3D [X3D] language. A 3D space is defined by the X3DGeometry, which provides wide range of tools for defining 3D geometry, from primitive geometry forms, to complex sets of triangles or faces. By default, the geometry is positioned in the center of the interacting object. The geometry can be repositioned in space with X3D transformation tools. The transformation is always done relatively to the interacting object coordinate system. If no geometry or viewpoints are specified the 3D interaction space is equal to the whole 3D Virtual World.

3.2 Condition

A condition is a set of requirements set on the states of all objects taking part in an interaction, i.e. the trigger object, the interacting object and the environment. Environment is a special case of an object. It does not have geometry, but can have some ambient parameters of the 3D world that are taken into account in the interaction. All requirements of the condition have to be satisfied to trigger an action.

In the Condition element of the MIM model, the set of requirements is written as a mathematical description of a form of a logical expression. Single logical expression represents all condition requirements. Such approach allows setting logical relations between different requirements, which can be something else than just AND operation. Semantic description of a condition may be used as a substitute or an enhancement of the mathematical description. The Condition element of MIM metadata is optional and can be omitted. In such case the action will be executed at each occurrence of the event.

In the MIM implementation, the Condition logical expression is written in MathML content notation according to mathml:condition.type type. The logical expression may use symbols representing variables and constants. Each variable represents a value of a specified parameter type of the trigger object, the environment or the interacting object. Moreover, it is possible to refer parameter values either before or after the event. It allows using also the relative change of a value, not only an absolute value. A constant is a value defined according to a data type of a particular parameter type.

3.3 Action

An action is a change of the state of an interacting object. In the MIM model, the Action element describes the outcome of an action, a new object state expressed as a mathematical and/or semantic description of new values of object parameters.

There are two possible types of action: simple action and complex action. The simple action is

when only a single object parameter is modified. In such case, mathematical expression can be used to calculate new parameter value, while the semantic description may provide more information about the change. The complex action occurs when more than one parameter value is modified. Again, single parameter change can be described with mathematical and/or semantic tools forming a parameter change descriptor, but there is more. It is possible to group some parameter change descriptors together and form a tree of such groups. Each node of such tree can be described with additional semantic description providing more abstract information on different levels of detail. In the MIM implementation, complex actions are made with recurring schema design, which allows any action element to contain an arbitrary number of sub-actions. Evaluation of an action on different levels of detail makes it possible to match an object even if the query uses high level terms.

The implementation of the Action element of the MIM model uses MathML content notation for mathematical expressions. Expressions are written according to mathml:apply.type type and use variables and constants in the same way as mathematical expression in the Condition element of the MIM model.

4 INTERACTION INTERFACE CONCEPT

The Interaction Interface (II) concept is used to describe the API of an object in terms of functions and attributes related to object interactions. Both elements of the API are represented by object parameter types of the Interactive Interface and are accompanied with semantic descriptions. Interaction interfaces are used in metadata as object interaction characteristic and as a dictionary of parameter types referenced in the Multimedia Interaction Model.

An interaction interface groups object parameter types that have similar semantics, e.g. object parameter types related to object visual characteristics or object parameter types related to object internal properties. An object implements a given interaction interface if the set of object parameter types is a superset of the set of parameter types of this interaction interface. To enable creation of interaction metadata, each interactive 3D object has to implement at least one interaction interface.

Interaction Interface concept provides a schema for definition of new interaction interfaces and parameter types. Each interaction interface is composed of a name, ID and a set of object parameter types. The definition of an interaction interface can be also extended with a semantic description. The interaction interface name is only informative, while its ID is used in metadata descriptions as a prefix in references to parameter types defined in a particular interaction interface. Definition of new parameter types is based on a separate schema described below. The semantic description of an interaction interface does not provide direct information about parameter types. Hence it does not influence efficiency of interactive object searches. However it can be useful for discovery and exchange of definitions of particular interaction interfaces.

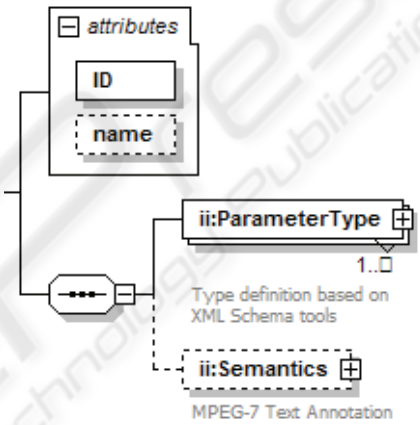

Figure 2: Interaction Interface schema.

The interaction interface schema is implemented as an XML Schema document. All schema information is contained in two attributes: ID and name, and two elements ParameterType and Semantics. The name attribute is optional. Each interaction interface definition may contain one or more ParameterType elements and zero or one Semantics element.

Definition of a parameter type is composed of an ID, definition or reference of parameter values data type, definition of possible arguments, semantic description and relation to a concept in some ontology. The parameter type ID is used to reference a particular parameter type in interaction metadata. The data type information is necessary to allow search engines properly interpret and process values of interaction parameters. The data type definition is based on XML Schema which provides a large set of type definition tools. Additionally, the usage of XML Schema allows using references to data types defined in other standards. Some parameter types may represent object functions that require attributes to provide resulting value (e.g. object dimension

along a given vector). The parameter type attributes are composed of a name used to reference a particular attribute and definition of a data type of attribute value. The data type of an attribute value is defined in the same way as the data type of parameter value – using XML Schema. Apart from technical elements, the definition of a parameter type may contain optional semantic description and optional relation to an ontology concept. Both fields are intended to help search engines interpret the meaning of parameter types and process user queries that include just some keywords and do not include exact identifiers of object parameters.

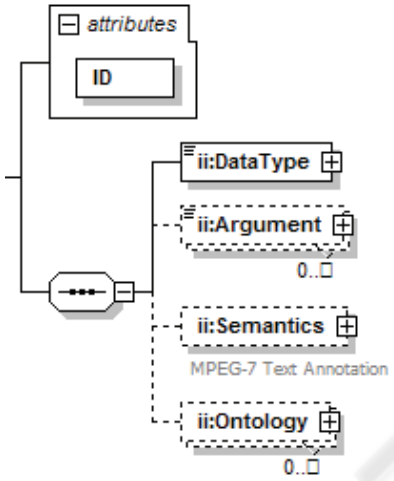

Figure 3: Parameter Type schema.

The ParameterType schema element pictured in fig. 3 contains a mandatory ID attribute and four elements: DataType, Argument, Semantics and Ontology. The DataType element contains a definition of a data type of parameter values and it is mandatory. The Argument element may occur zero or more times and contain name and definition of a data type of a parameter argument. The Semantics element is of the mpeg7:TextAnnotationType type and contains semantic description of a parameter type. The last element, Ontology, can occur zero or more times and it is a relation to a concept in an external ontology system. The Ontology element includes two attributes, which are URIs of an ontology and a concept within this ontology.

4.1 Visual Interaction Interface

The Interaction Interface concept provides a method of defining new parameter types. Specific parameter types should be grouped based on their semantics or usage. Therefore new interaction interfaces will be defined per domain or per application. However, there are some common properties of all 3D objects that allow forming an interaction interface common for all objects. This interaction interface is called: Visual Interaction Interface. It represents object parameter types related to object visual characteristics: geometry, position and appearance and can be used to build metadata of visual and geometrical interactions. The Visual Interaction Interface includes object parameter types for describing variuos aspects of object size (SizeBoundigBox, SizeBoundingSphere, SizeDimension), shape (ShapePrimitive, ShapeFaceSet, ShapeCustom, ShapeContour, ShapeRegion), position (PositionCenter, PositionCollision), orientation and appearance (AppearanceTexture, AppearanceTextureHomogeneous, AppearanceColor, AppearanceColorDominant, AppearanceColorLayout, AppearanceColorStructure, AppearanceTransparency). Data types of these parameter types are based on MPEG-7 descriptors, X3D nodes and custom data types expressed with XML Schema tools.

5 INTERFACE DEFINITION RULES

Apart from the Interaction Interface schema, the Interaction Interface concept contains also a set of Interaction Interface Definition Rules that describe requirements related to defining new interaction interfaces. Especially new object parameter types.

5.1 General Definition Rules

1. II definition consists of an interface name, ID and a set of object parameter types.
2. Name of an II has to be a unique free-text description depicting semantics of object parameter types contained in this II. Examples:
   - Visual Interaction Interface
   - Aural Interaction Interface
   - Inventory Interaction Interface
3. ID of an II has to be a unique and short acronym of the II name. If the name uses a term “Interaction Interface”, then the acronym of that term has to be II and it has to be prefixed with a ‘-‘ character. Examples:
   - V-II
   - A-II
   - I-II
4. Set of object parameter types has to group all object parameter types that are logically related,

i.e. they have similar semantics. The decision on exactly which object parameter types should be grouped together depends on the given application and domain and is left to the author of the interface.

5. New II may be defined as an extension of an existing II. The extending II has new name, new identifier and a set of object parameter types which is a superset of the set of object parameter types of the extended II.

5.2 Object Parameter Type Definition Rules

1. Object parameter type (OPT) definition consists of an identifier, a semantic description, a specification of a data type of values of object parameters of this object parameter type and a specification of arguments required to evaluate the object parameter function. OPT identifier has the following syntax:

```
[interaction interface ID].
[object parameter type name]
```

2. Object parameter type name has to represent its semantics and has to be written in camel-capped notation with a leading upper-case character. Examples:
   - V-II.SizeBoundingBox
   - A-II.Waveform
   - I-II.NumberOfItems
3. Semantic description of an OPT has to depict its meaning. The semantic description has to be expressed in one of two forms:
   - Text annotation based on MPEG-7 TextAnnotation data type, which allows free text, keyword, structured and dependency structure annotations.
   - Relation to an ontology concept formulated as a URI.
4. Data type specification of an OPT has to be written using tools defined in XML Schema specification.
5. OPT arguments specification has to list all arguments required to evaluate an object parameter function. An argument description has to contain argument name and argument data type.
6. Argument name has to depict its meaning and has to be unique in the scope of the OPT.
7. Data type of an argument has to be specified in the same way as a data type of the OPT.

6 CONCLUSIONS

The presented Multimedia Interaction Model, exemplified in (Chmielewski, ECMS 2008), is a complete solution for describing object interaction characteristics. Moreover, the Interaction Interface concept allows using the MIM model in ways that are not limited to any particular domain. For example, the MIM solution can be easily employed for describing 3D representations of body parts used in medicine or for describing some interactive objects in game development industry. With small modifications, namely definition of new interaction interfaces, the presented solution for building interaction metadata can be even more general. It can be used to describe any interactive entity, even if it is not a 3D object, for example a role in automatic negotiations or a software component.

The domain of interaction metadata for autonomous, reusable interactive 3D objects is in early stages of development. There are many areas open for research. First and most obvious area is research of search engines and comparison algorithms for interaction properties. Second is related with interaction properties descriptors. Some of such descriptors could be calculated directly from the object definition making the metadata creation process faster and more efficient. Moreover, different applications will require different interaction interfaces. Therefore the third area of potential research is related to development of universal interaction interfaces and tools for publishing and discovery of such interfaces. With such tools producers of sliding doors could post libraries of 3D representations of their products that are described with a domain specific interaction interfaces. Making these libraries searchable and allowing for CAD system extensions that connect with object libraries spread through the Internet and automatically find objects best suited for a given design, could make architect/designer life much easier.

REFERENCES

AAF, AAF Object Specification v1.1, 2005, Available: http://www.aafassociation.org/html/techinfo/index.htm

Bilasco I.M., Gensel J., Villanova-Oliver M., Martin H. 2005. On indexing of 3D scenes using MPEG-7, in

Proceedings of the ACM Multimedia 2005, Singapore, 2005, pp. 471-474

Bilasco I.M., Gensel J., Villanova-Oliver M., Martin H. 2006. An MPEG-7 framework enhancing the reuse of 3D models, in *Proceedings of Web3D Symposium 2006*, Columbia, Maryland USA, 2006, pp. 65-74

Bry F. Patrânjan P. 2005. Reactivity on the web: paradigms and applications of the language XChange, in *Proceedings of the 2005 ACM Symposium on Applied Computing*, Santa Fe, New Mexico USA, 2005, pp. 1645-1649

Chmielewski J. 2008. Metadata Model for Interaction of 3D Objects. In *Proceedings of the 1st International IEEE Conference on Information Technology IT'08*, Gdansk, Poland, May 1821, 2008, pp. 313-316

Chmielewski J. 2008. Architectural Modeling with Reusable Interactive 3D Objects, in *Proceedings of the 22nd European Conference on Modelling and Simulation ECMS'08*, Nicosia, Cyprus, June 3-6, 2008, pp. 555-561

Dayal U. et al. 1988. HiPAC: a Research Project in Active, Time-Constrained Database Management, Interim Report, Technical Report XAIT-88-02, Xerox Advanced Information Technology, June 1988

DCMES: Dublin Core Metadata Element Set, Version 1.1, 2008, Available: http://dublincore.org/documents/dces/

DIG35: Digital Imaging Group, DIG35 Specification, Metadata for Digital Images 1.0, 2000, Available: http://xml.coverpages.org/FU-Berlin-DIG35-v10-Sept00.pdf

EXIF: Japan Electronics and Information Technology Industries Association (JEITA) Standard, JEITA CP-3451, Exchangeable image file format for digital still cameras: Exif Version 2.2, 2002, Available: http://www.exif.org/Exif2-2.PDF

ID3, ID3 Informal standard, ID3 tag version 2.4.0, 2000, Available: http://www.id3.org/id3v2.4.0-frames

Lorenz B., Ohlbach H.J., Stoffel E.P. 2006. A Hybrid Spatial Model for Representing Indoor Environments, in *Proceedings of W2GIS 2006, LNCS 4295*, Hong Kong, China, 2006, pp. 102-112

Martinez J. M. et al., MPEG-7 Overview, 2004, Available: http://www.chiariglione.org/MPEG/standards/mpeg-7/mpeg-7.htm

MPEG-7, ISO/IEC 15938, ISO International Standard, Information technology – Multimedia content description interface Available: http://www.iso.org/iso/en/prods-services/popstds/mpeg.html

MXF: SMPTE 377M-2004 Television Material Exchange Format (MXF) Standard – File Format Specification, 2004, Available: http://store.smpte.org/

Ottosson S., Virtual reality in the product development process. *Journal of Engineering Design*, Volume 13, Issue 2 June 2002, pp. 159 - 172

Papamarkos G., Poulovassilis A., Wood PT. 2003. Event-Condition-Action Rule Languages for the Semantic Web. *Workshop on Semantic Web and Databases*, 2003

Pitarello F., de Faveri A. 2006. Semantic Description of 3D Environments: a Proposal Based on Web Standards. In *Proceedings of Web3D Symposium 2006*, Columbia, Maryland USA, 2006, pp. 85-95

P/META: EBU Metadata Exchange Scheme (EBU P/Meta) 2.0, EBU Tech 3295-v2, 2007, Available: http://www.ebu.ch/en/technical/metadata/specifications/index.php

Stone, R. J., 2001 Virtual Reality in the Real World: A Personal Reflection on 12 Years of Human-Centred Endeavour, in *Proceedings of the 11th International Conference on Artificial Reality and Telexistence, ICAT '01*, Tokyo, Japan, December 5-7, 2001, pp. 23-32

Swartz A. 2002. MusicBrainz: A Semantic Web Service. *IEEE Intelligent Systems*, vol. 17, no. 1, Jan/Feb, 2002, 76-77

Thome B., Gawlick D., Pratt M. 2005. Event processing with an oracle database, in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, Baltimore, Maryland USA, 2005, pp. 863-867

Walczak K. 2006. Beh-VR: Modeling Behavior of Dynamic Virtual Reality Contents. In *Proceedings of the 12th International Conference on Virtual Systems and Multimedia VSMM* 2006. In: H. Zha et al. (Eds.): Interactive Technologies and Sociotechnical Systems, Lecture Notes in Computer Sciences 4270, Springer Verlag Heidelberg 2006, pp. 40-51

Walczak, K., Flex-VR: Configurable 3D Web Applications, in *Proceedings of the International Conference on Human System Interaction HIS'08*, Krakow, Poland, May 25-27, 2008, pp. 135-140

X3D, 2004. ISO/IEC 19775-1:2004, Information technology -- Computer graphics and image processing -- Extensible 3D (X3D) – Part 1: Architecture and base components

XML Schema, 2004. W3C Recommendation, Available: http://www.w3.org/XML/Schema

Z39.87: NISO American National Standard, ANSI/NISO Z39.87-2006, Data Dictionary – Technical Metadata for Digital Still Images, 2001, Available: http://www.niso.org/kst/reports/standards/

Zhou X., Zhang S.,Cao J., DAI K. 2004. Event Condition Action Rule Based Intelligent Agent Architecture, *Journal of Shanghai Jiaotong University*, Volume 38, Issue 1, January 2004, pp. 14-17