Project Number: 032518

QVIZ

Query and context based visualization of time-spatial cultural dynamics

Specific Targeted Research Project

Information Society Technologies

Knowledge Content Model

Due date of deliverable: 31/10/2007

Actual submission date: 14/12/2007

Start date of project: 01/05/2006 Duration: 24 month

Umeå University Final

Abstract

Number and name	Project Number: 032518 QVIZ, Query and context based visualization of time-spatial cultural dynamics
Work Package	WP 3
Task	3.4
Date of delivery	**Contractual:** 31/10/2007 **Actual:** 14/12/2007
Code name	032518 **Version** draft ☐ final ☑
Nature	Knowledge Content Objects Report
Distribution Type	Restricted Deliverable A revised version of this deliverable (due at M24) is a public deliverable.
Authors (Partner)	Rupert Westenthaler, Wernher Behrendt, Tobias Bürger, Violeta Damjanovic, Andreas Gruber, Bob Mulrenin (SRFG)
Contact Person	Wernher Behrendt, wernher.behrendt@salzburgresearch.at
Abstract	This document presents the generic Knowledge Content Model reflecting the QVIZ requirements. It introduces the business objectives of developing KCOs, describes the generic KCO model and its specialization for QVIZ, and provides an example of its use within the project.
Keywords List	Knowledge Content Object

Table of contents

1. Executive Summary

This project report and its attached schema addresses the operational objective of the QVIZ project "to apply state-of-the-art knowledge management technologies to the QVIZ archival knowledge infrastructure" (Technical Annex, p.6) by specializing knowledge content objects for the QVIZ domain. These objects encapsulate creative work including some business semantics for their exchange.

QVIZ proposes a novel format for knowledge exchange between information systems that is based and inspired by a model for *intelligent content,* which was pioneered in the METOKIS[1] project (2004 - 2005).

A KCO is a structure that contains those descriptions about content objects that are essential for communicating and trading digital content. It is a machine-readable thematic classification and context description that can contain a usage history and planned usage for a target community, license and contract information as well as a presentation description that turns digital content from a simple file into added value for use within information systems.

This document briefly outlines the relevant findings of this project and specifies what parts are relevant for a KCO within the QVIZ project. This process of specialization has two major parts.

The first is the specialization of the model through a domain ontology – which provides the propositional context and the means of thematic classifications of content objects.

The second part is to specify, what kind of knowledge is available from the actual application and how this can be mapped onto a generic knowledge model. In the case of QVIZ, this means, that some of the more advanced facets of a KCO, such as the presentation facet and the trust and security facet are not addressed.

The document is organized as follows: After a short introduction about the KCO as a unit of value for QVIZ, the methodology for building knowledge content systems for communities of practice and the KCO base model is described. The specialization of the KCO is provided in the following chapter. The final chapter describes a simple example of a QVIZ KCO.

[1] http://metokis.salzburgresearch.at [last visited: 01.11.2007]

2. Introduction

The 6th framework IST research project METOKIS proposed an infrastructure that would support web based content services to be seamlessly merged with knowledge structures as proposed by the Semantic Web initiative.

The research assumption was that such an infrastructure would need to provide a generic yet semantically rich (i.e. *meaningful*) container structure that allows the capture of machine-interpretable descriptions of arbitrary content. This container structure is called *KCO, Knowledge Content Object*. Each KCO has firstly, the generic semantics of all KCOs and additionally, provisions for adding domain specific descriptions of the content that the KCO refers to (Figure 1).

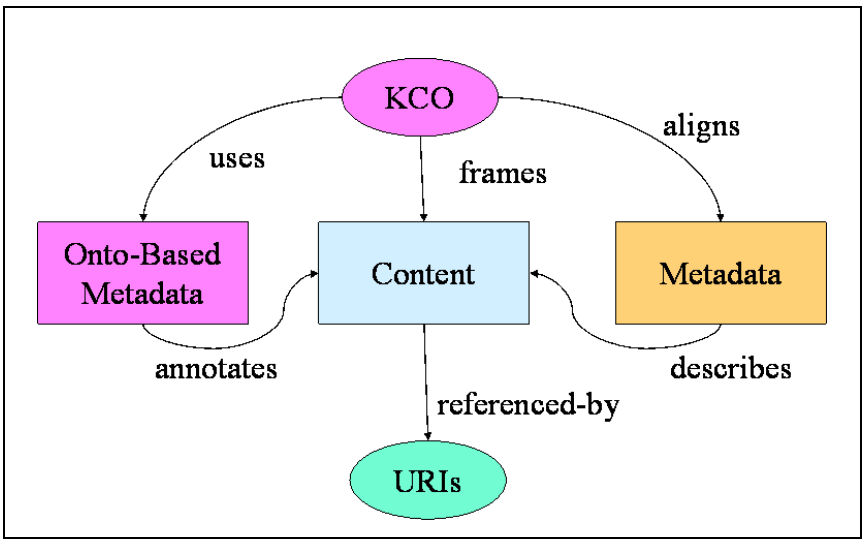


Figure 1: The motivation - knowledge content objects create a "unit of value" for the Semantic Web

Since there was no such infrastructure available, the METOKIS project attempted to design and implement a prototypical infrastructure in the form of a framework which caters for the generic semantics and which allows for merging of the infrastructure with different supporting technologies. METOKIS assumed a combination of web based and intranet-based distribution of resources. The Knowledge Content Carrier Architecture (KCCA), a knowledge management system for KCOs [METOKIS D21] is therefore designed as a federated, distributed system which is "held together" by a common protocol and the KCO as a schema for algebraic objects upon which the nodes of a METOKIS federation can operate.

In order for a METOKIS federation to achieve certain system tasks, collaboration between the distributed nodes is necessary. For this purpose, a high-level request/response protocol was defined for messaging between nodes and for remote manipulation of KCOs. This protocol is called KCTP (Knowledge Content Transfer Protocol).

One of the far-reaching goals of the METOKIS project was to investigate whether this infrastructure could make a contribution to the visions of Ambient Intelligence[2] where communication and service provision would need to rely on mixed human-machine-machine-human loops. In such an advanced environment, the system would also need to cater for the definition of user tasks ("services" at a business level and "workflow" at the operational level) and because of the extreme distribution and the peer-to-peer character of many interactions, probably the safest place to keep state information would be in the manipulated object itself, i.e. the KCO. To this end, the KCO, KCCA and KCTP were designed in a way that distinguishes between content, knowledge about the content, knowledge about the structure, and knowledge about how the structure and the content can be used in different work environments. The knowledge about how KCOs can be managed is formally embedded in the KCCA. The knowledge about how the content can be *used* in some specific work environment is a matter for domain experts to define and this knowledge can be added to the generic definition of KCOs, thus "pre-formatting" domain-specific KCOs for further use. How this is done and how it can be used is described in the METOKIS Methodology Handbook [METOKIS D20].

[2] "The vision of 'ambient intelligence' (interactive intelligent environment) places the user, i.e. the human being, at the centre of the future development of the knowledge-based society." FP6 IST Introduction (http://ec.europa.eu/research/fp6/index_en.cfm?p=2)

Specializing the KCO for the QVIZ domain

The main intention of the QVIZ system is to manage descriptions about archival resources and about so-called social objects representing community based knowledge about these resources. This fact is visualized in the conceptual model of the QVIZ domain ontology (Figure 2)

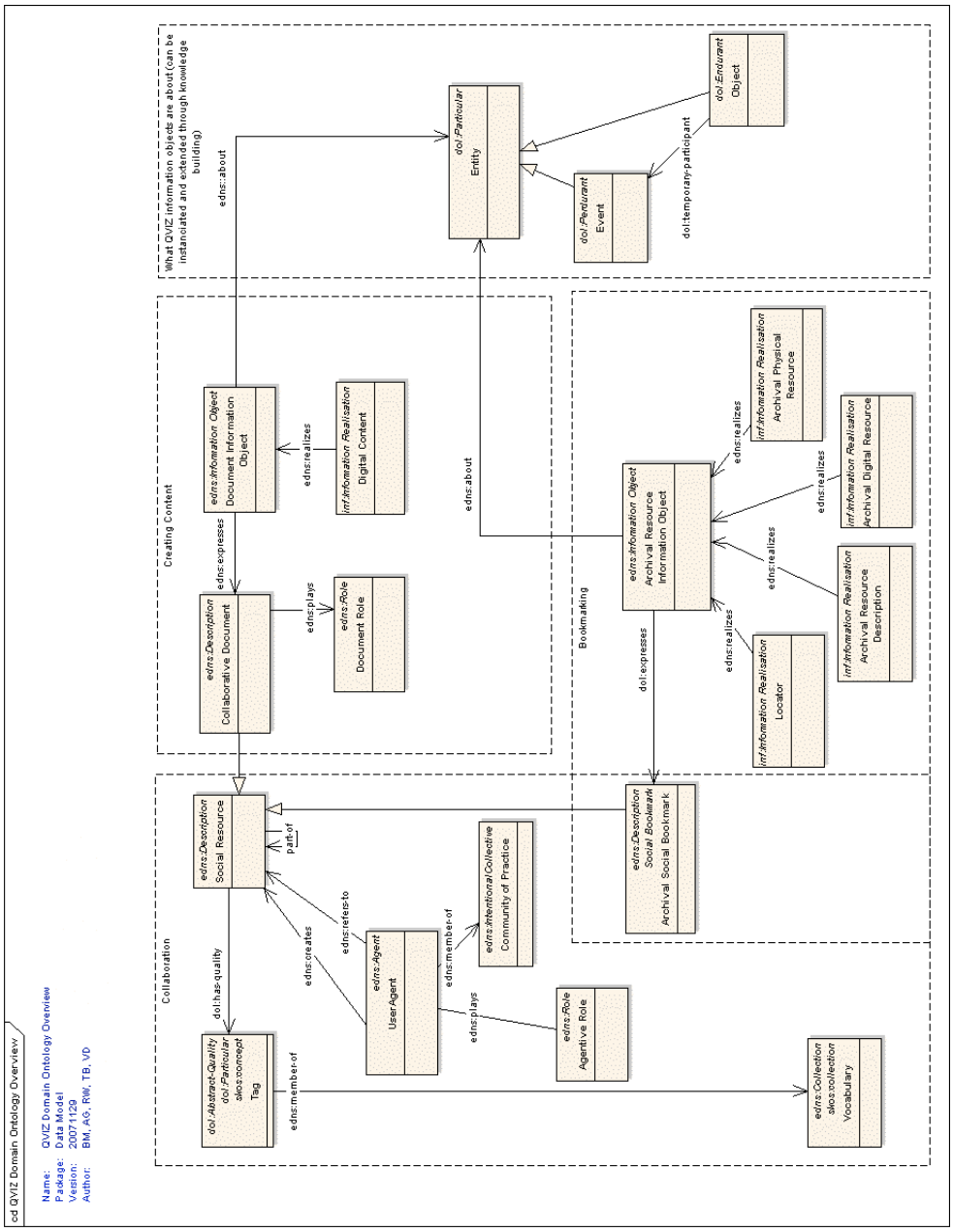


Figure 2: The conceptual model for archival resource descriptions

The figure above shows the parts of the domain ontology of QVIZ. For a more detailed description, please see Deliverable 3.3, the *Domain Ontology Report*.

One of QVIZ' objectives is to build tools to support knowledge building and to create knowledge content and knowledge content objects (KCO).

KCO as an unit of value for QVIZ

The main goal of QVIZ is to enhance access to archival material. The role of the collaborative environment is to create a semantic network over the use of the material in new contexts and thereby enhance access to the archive material. Because archive material and their associated administrative units are highly important to QVIZ, collaborative social knowledge content is also a means to add user based social information to archival resource descriptions and to enhance existing archival metadata about archive materials using the administrative unit and one or more domain ontologies in use by a community. The idea is that so-called social objects will become *social knowledge content* that can be aggregated into one or more Knowledge Content Objects (KCO), a unit that also contains business and community descriptions and that can be exchanged between systems. Social knowledge content created by social software, e.g. blogs, blog entries, and their multimedia attachments that are being built in QVIZ, can be aggregated into Knowledge Content Objects (KCOs). Social knowledge content can enhance access to archive materials by providing semantic descriptions and interrelationships for archival materials, collaborative resources or topic classification related resources related to one or more Communities of Practice (CoP).

An *atomic KCO* is defined as an object that has all available knowledge about one "Archival Resource Information Object"[3].

A more complex, i.e. *composed KCO* that can be assembled by users of the QVIZ system, can be about a specific content and contain social bookmarks and/or publications and/or digital objects. It is assumed that the interrelations between these objects are already available based on the information in the social bookmarks, publications or digital objects.

This collection of different QVIZ items is exchangeable between different systems and available on request i.e. there is a query facility for exporting KCOs.

[3] Definition for these terms can be found in the QVIZ Domain Ontology Report.

3. A Methodology for Building Knowledge-Content Systems for Communities of Practice

We apply the KCO Service Development Methodology as developed in the METOKIS project to QVIZ [METOKIS, D20]. In principle, the methodology distinguishes three development phases each consisting of a number of activities:

- Analysis
- Design
- Realization

The following tables give an overview of these phases and activities:

Main Phase	Activities	Explanation
Analysis	Knowledge Sharing Communities and Environments	Discover who the stakeholders of the knowledge sharing community are and how their processes work
	Motivation structures in knowledge-sharing communities	Knowledge sharing communities have their own motivation patterns for working together, e.g. reciprocity or altruism.
	Existing initiatives to knowledge sharing	Through introspection, the organization collects its own current approaches - the ultimate objective is consolidation of these approaches
	Organizational readiness for knowledge sharing	This is done at management, organizational and technological level, and also includes an assessment of the learning capability of the organization
	Primary Processes	Given the characterization of the knowledge sharing community in the previous activities, we now develop descriptions of the working situations and the main processes that govern these situations. Standard process modeling techniques can be used here.

Figure 3: Methodology - Activities in the Analysis Phase of knowledge and content applications

What becomes evident from this methodology is the focus on collaborative processes which generate knowledge and which help to distribute knowledge within an organization. Eliciting this information is a first step towards encoding it in the community facet of the KCO.

In the design phase, we use our understanding of the knowledge sharing community and the processes of the actors to develop concrete value generating options that are assessed for their business impact. For those service options which are chosen as promising a detailed description of interactions is done and this leads to the specification of the application specific task ontologies which in turn, form the basis of the business services provided by the target system.

Design	Identification of leverage options	Possible approaches are either resource-based (higher productivity) or market based (better services or products)
	Development of business models	There are four elements to be considered: Network environment (defining one's own position in the market); the value creation offered; the economic control one has over the development of the business options; the identification of revenue sources.
	Cost accounting model for semantic information goods	This model was developed to get some quantitative handle on a cost benefit analysis for semantic information goods.
	Description of interactions	Each interaction is characterized by the roles which interact; the purpose of the interaction; its outcome; the knowledge that is exchanged through the interaction
	Development of task ontologies	At this stage, the tasks are formally modeled using the DDPO (DOLCE + Descriptions and Plans Ontology)
	Service Design	At this stage, we can determine which facets of the KCO will have to specialized in which way and which additional services will have to be provided through the KCCA (e.g. wrapper services to legacy systems)

Figure 4: Methodology - Activities in the Design Phase of knowledge and content applications

In the Realization phase, the designs have to be mapped to the KCO and KCCA, and are then implemented as specializations of the ontological framework of the KCO and the services framework of the KCCA.

Realization	KCO Specialization	The generic KCO Ontology is extended by domain specific modeling constructs
	KCCA Specialization	The wrappers of the KCCA are specialized to interoperate with KCO-unaware, external systems.

Figure 5: Methodology - Activities in the Realization Phase of knowledge and content applications

Analysis of the KCO Environment from a QVIZ point of view

As outlined in the table above, the activities to go through are:

- Identifying the Knowledge Sharing Communities and Environments
- Analyzing the motivation structures in knowledge-sharing communities
- Identifying existing initiatives to knowledge sharing
- Assessing organizational readiness for knowledge sharing
- Initial modeling of knowledge sharing processes

Most of these tasks were done in the requirements analysis phase of QVIZ and the usage scenarios and use cases are already described in the technical specification [QVIZ D4.1.3] and the domain ontology [QVIZ D3.3.]

Design of KCO-based QVIZ applications

The design activities proposed by the original METOKIS methodology were geared towards a commercial knowledge sharing environment, whereas the QVIZ case appears to be more in the altruistic or reciprocating region, coupled with a primarily non-commercial motivation. The service design steps are therefore subject to adaptation:

- Identification of leverage options
- Development of business models
- Cost accounting model for semantic information goods
- Description of interactions
- Development of task ontologies
- Service Design

Identification of leverage options means in the QVIZ context, identifying where there would be most public value gained from offering archival resources for public use.

Development of business models for collaborative content would require negotiation with publishers to further refine business description of the QVIZ KCO. The licensing aspect of the business semantics for common public users is described by the use of Creative Commons.

The cost accounting models would likewise, be developed together with the business models, but are not a primary concern for the application domain of QVIZ and have thus, little influence on the design of the QVIZ KCOs.

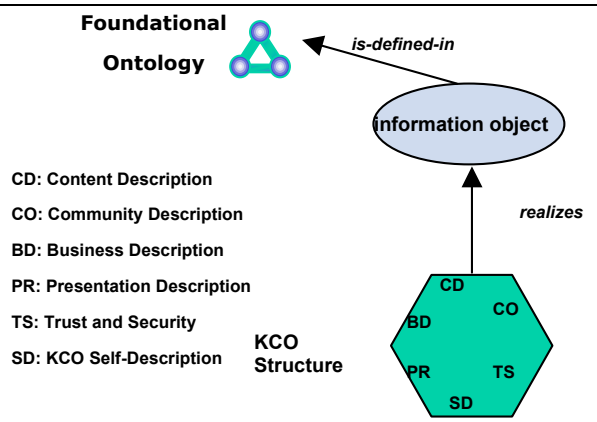
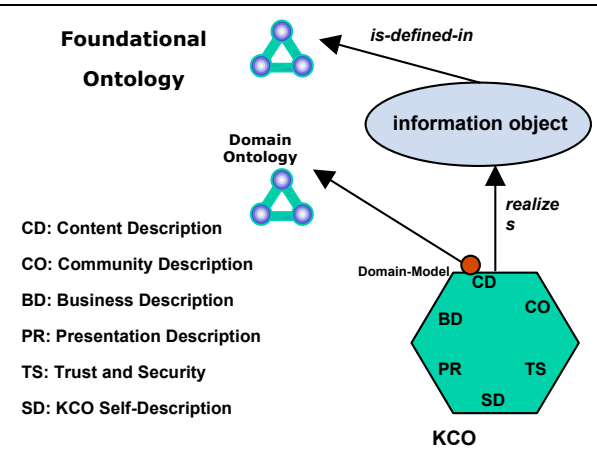
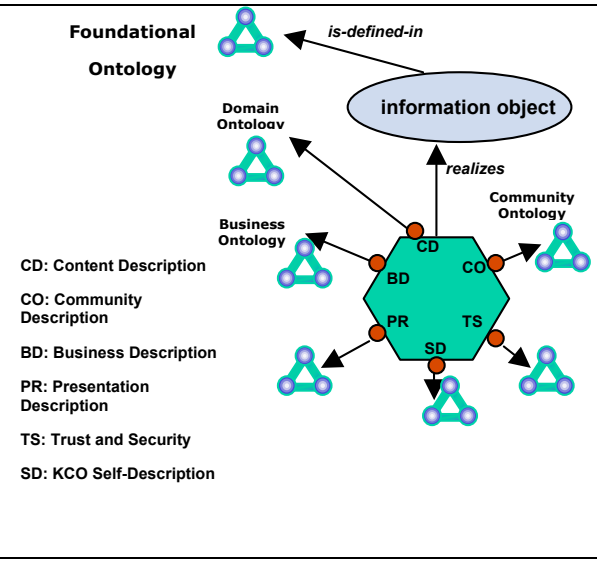
Realization - KCO Specialization from a QVIZ point of view

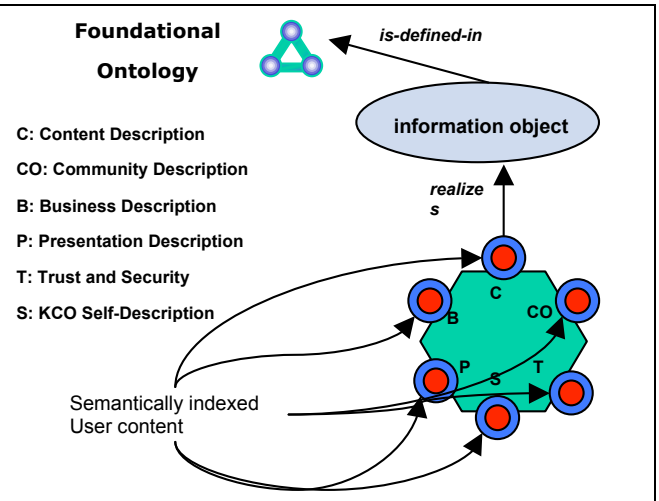
The KCO is intended to be a generic container for digital content and computational knowledge structures. Before it can be used by an application there are two specialization steps necessary:

1. Sector- or domain specific specialization - this means adding a (possibly sector-specific) domain ontology to the general model

2. Application-specific specialization - this means the provision of transformations (wrappers) between non-KCO and KCO-compliant systems

These steps are best illustrated by diagrams:

Fundamental KCO - the KCO has six facets and these are successively specialized. The starting point is the domain ontology, followed by the community and business ontologies.	
Domain specialization - the generic KCO and its foundational ontology are specialized to provide a universe of discourse for the domain or the business sector. Example: "archival resource context" is a domain concept	
Application-specific customization of the KCO - each of the facets is specialized to cater for the needs of the application type Example: "Archival resource description" or "Social bookmark" is a term that is invented (and has meaning) at the level of QVIZ, but is not part of everyday language in the world of archives.	

The KCO in use - user content is semantically indexed through the ontological structures provided by the system.	

4. KCO base model

This section first gives an introduction to the general KCO architecture, then summarizes requirements for a QVIZ KCO, and finally presents a model for KCOs that will be implemented in QVIZ.

KCOs were designed in response to an environment where actors have knowledge and receive information. Furthermore, the actors have access to a large content space from which they can draw further information. The actors make use of their existing knowledge space, weave the new information into their existing knowledge and eventually interpret their growing knowledge space with respect to their current or future *action space*. If the interpretation is done with reference to a future action space then it is called planning, with respect to the current action space it is called doing. Actors communicate with each other by exchanging meaningful statements: Human beings are capable of using *natural language* for this task. When machines are involved, *surrogates* must be found for natural language and for the notion of meaningful statements. KCOs can be understood as a surrogate language that can be exchanged between humans and humans, humans and machines, as well as machines and machines.

To align this general definition with QVIZ, one has to state that most of the features of this environment go along with the setting as defined for QVIZ. The users of the QVIZ Collaborative Environment are actors working with knowledge. They have access to large content spaces – the archives – and use this information to draw further information. They create new knowledge – such as by creating social bookmarks, collaborative documents, qualities or named entities – and share this knowledge with other users within their own community and also with other communities. However, to go beyond the confines of a local application environment, the knowledge and content created by QVIZ users needs to be shareable with people and applications external to the QVIZ Collaborative Environment.

Where the QVIZ domain ontology as described in D3.3 is mainly concerned about providing a formal model to support users in the QVIZ Collaborative Environment the QVIZ Knowledge Content Model has a focus on defining these surrogates needed to share the knowledge and content created by the QVIZ users with external applications.

The KCO model as defined in METOKIS is a highly formalized model based on the DOLCE foundational ontology for the description of knowledge about content, whereas the foundational grounding establishes a basis for common understanding of the structure of the information between foreign systems. The idea of the KCO model is to describe all aspects of content, including the content itself, but also business related aspects or workflows related to the content in one tradable object.

Knowledge Structure of the generic KCO

KCOs are based on the information object design pattern, which was developed within the METOKIS project as an extension to the DOLCE foundational ontology. The information object design pattern is embedded in the DOLCE foundational ontology by extending the description and situation (dns) pattern by the ability to describe information objects. The following figure shows an overview of this pattern.

*Information-Object*s are *Social-Objects*. From a communication perspective, an *Information-Object* plays the role of a message. From a semiotic perspective, it plays the role of an expression. But since communication theory and semiotic theories are different, it is more correct to say that a message role specializes an expression role.

An *Information-Object* is defined by the *Agent* who *interpret*s it, the *Description* forming the propositional content, some *Physical-Realization* that supports the assertions taken by the *description expressed-by,* the *Information-Object,* and the *aboutness* of the *Information-Object*.

That means that an *Information-Object* is a representation (sign) that

- *expresses* (stands for) an *Description conceived-by (created-by or adopted-by)* an *Agent*. This description defines a formal conceptualization of a *Situation* as *internally-represented* in the mind of the *Agent*. That means that a *Description expressed-by* an *information-object* is dependent on the interpretation of the *Agent*.
- is *realized-by* some content *realizing* the sign represented by the *information-object*. These relations support to state that a content object – the *Information-Realization* – supports the sign represented by the *Information-Object*. The *Information-Object* is therefore a kind of a mediator linking the different interpretations of *Agent*s with possible different content objects realizing these interpretations. Note that a content object may tell more than one story and therefore *realizes* more than one *Information-Object*.
- is *about* a topic. The *about* relations provide the possibility to state, that an *information-object* is *about* some entity (*particular*). The difference to the *expresses* relation is, that the *about* relation is not dependent on an interpretation of an *Agent* but can be assumed as "global" truth.

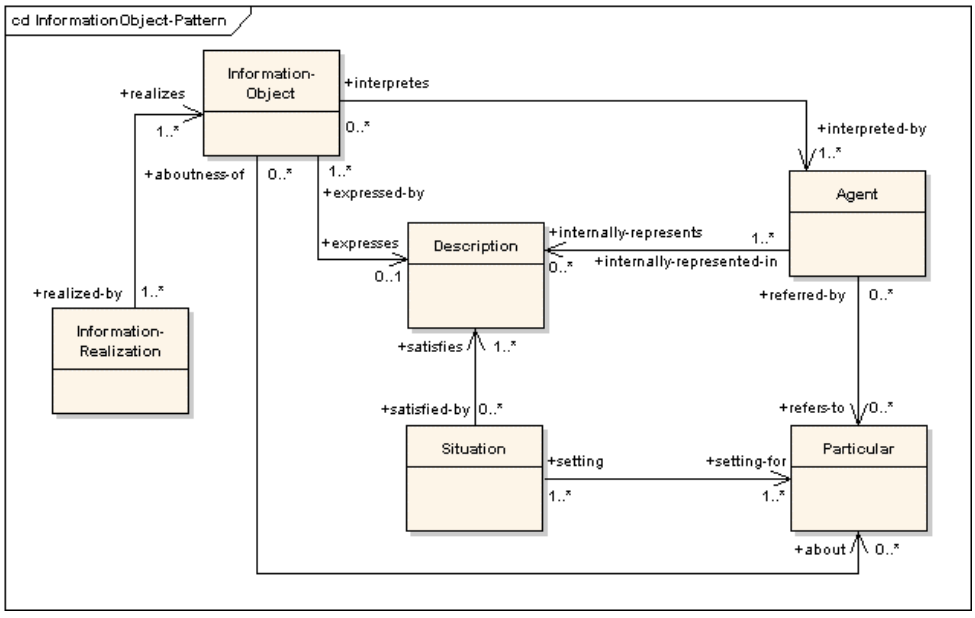


Figure 6: Information Object Design Pattern

As implicitly described by the above description, this pattern is useful to model two settings. First those where different agents (e.g. users of a system) create (*internally-represent*) different interpretations (*Descriptions*) on the same topic (an *Information-Object about* a *Particular*) and second that content (an *Information-Realization*) may support/communicate different messages (*Information-Objects)* representing expressions with a different context (the *Situation*) *interpreted-by* the same or different *Agent*s.

More easily: The first use case describes that different agents may tell the same story differently. The second use case describes that content can tell different stories (to the same or different users).

As already noted, Knowledge Content Objects (KCO) define an intelligent content model which is based on the information object design pattern. Knowledge described by KCOs is structured along three levels.

1. **Resource Level:** This level provides the functionality of uniquely identifying the actual content object (File, Stream, image, text and structured text like XML or MS Office documents).
2. **Meta Level:** This level refers to knowledge describing features of the content object, e.g. frame rate, compression type, color coding scheme... but also publication rights, access restrictions describe the content object itself and not the subject of the content object.
3. **Subject Matter Level:** This level comprises knowledge about the topic (subject) of the content as interpreted by an actor. The content object realizes this interpretation.

The knowledge model defined for KCOs models these three layers by a nested usage of the information object design pattern. The following figure provides a simplified visualization of how the information object design pattern is applied to build up the three levels described above.

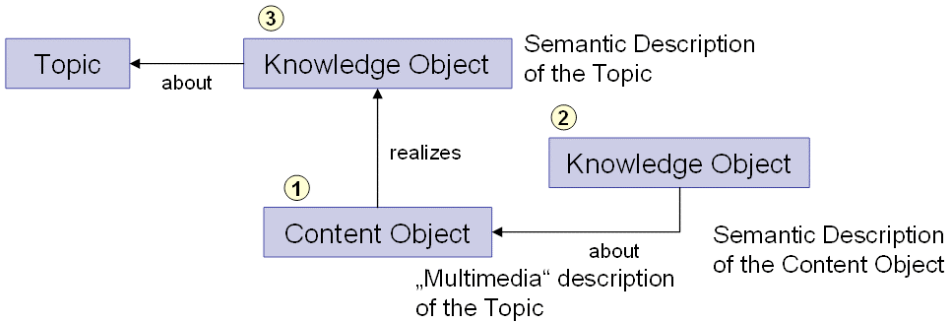


Figure 7: Overview about the KCO Model

Knowledge Object in the above figure is a placeholder for an *Information-Object interpreted-by* an *Agent* and *expresses* a *Description conceived-by* that *Agent*. The numbers in the figure represent the levels described above.

The Content Object – an *Information-Realization* – represents the first level, because it defines the unique reference, which makes a content item reference able in the knowledge space.

The meta level is modeled by Knowledge Object(s) *about* the Content Object. Based on the information object design pattern, that means that no interpretation of an *Agent* is required to state, that such a Knowledge Object is *about* the Content

Object. Such a modeling perfectly fits for the information described by the meta level like frame rate, compression type, publication rights, access restrictions …

The subject meta level is modeled by Knowledge Objects which are *realized-by* the Content Object of the KCO. That corresponds to the second usage scenario described for the information object design pattern, describing that an *Information-Realization* can support/communicate different stories. The *aboutness* of Knowledge Objects of the subject matter level is the subject matter of the content.

Facet Structure of the generic KCO

The KCO as a whole is divided up into six *facets*. Each facet is dedicated to providing information for a fundamental function of the KCO. The KCO *facets* group the "knowledge" of a KCO into five fundamental domains:

1. The *Content Description Facet* specifies which identifiable content we refer to, what its media properties are and what that content is about.
2. The *Presentation Facet* declares how the content should be presented in time and space, and it specifies at which points interactions between users and the rendition of the content are needed or possible.
3. The *Community Facet* declares the potential scope of usage of the object, by defining user roles and actions (tasks) that are relevant for the knowledge and content covered by the KCO. As an example a KCO of the publishing domain would define roles like "editor" and tasks like "publishing".
4. The *Business Facet* constrains the meaning of a KCO to express three things: firstly, available license models for usage of the KCO and its content, secondly, available pricing policies for usage of the knowledge and the content part of the KCO, and thirdly, negotiation protocols to reach agreement on a licensing model and its associated pricing policy.
5. The *Trust and Security Facet* addresses Tim Berners-Lee's vision of a "Web of Trust" by defining on the one hand, mechanisms to ensure that the consumer can have confidence in the content and knowledge encapsulated in a KCO when it is "consumed" and on the other hand, mechanisms to ensure that the intellectual or virtual property of the vendor is protected until a transfer of that virtual property (e.g. through a usage license) has been done.
6. The last facet covers the *Self Description* of the KCO and can be used to formally express domain requirements on the above facets and the KCO in general.

The faceted structure of the KCO has the aim to foster inter domain interoperability by providing a standardized views – the facet – on the relevant parts of the knowledge of the KCO. Applications using KCOs therefore need not process/understand the whole KCO, but only facets (or even facet elements) relevant to their tasks.

The following table lists all facets and their elements, supplemented by a short description.

Facets	***Elements***	***Short Description***
Content Description (CD)	Propositional Content	The knowledge realized by the content or segments of the content part of the KCO
	Content Classification	Keywords and concepts assigned to the content object
	Multimedia Characterization	Media type, encoding, access information, of the content part of the KCO
Presentation Description (PR)	Spatio-temporal rendition	Description of how the content (and the Knowledge) of the KCO is presented to users
	Interaction-based rendition	
Community Description (CO)	User task	Description of Plans, Tasks, Roles and Goals in the context of a community which uses KCOs
	User community	
	Usage history	List of actions performed with the KCO during its lifecycle
Business Description (BS)	Negotiation protocol	Process by which a trade will be settled
	Pricing scheme	Economic constraints on the settling of the trade
	Contract information	Legal constraints on the settling of a trade
Trust & Security (TS)	None	Not covered in the current implementation
Self-description (SD)	None	Specification of the KCO in machine-interpretable form (here: DOLCE:OWL).

Figure 8: KCO facets and its elements

Referring to the basic knowledge structure of the KCO, each facet is built up by one or more *Knowledge Object*s. Most of this *knowledge object*s has a one-to-one mapping with the different elements defined for the six facets. Each facet is defined in a separate ontology and defines its own namespace. Facets may depend on other facets and may import other ontologies.

From the described KCO base model, one can specialize the model further, as described in the next section.

5. A specialized KCO for QVIZ

Content Description Facet

The content description facet contains three elements: The propositional description, the content classification, and the Multimedia characterization.

The propositional description and the content classification element of this facet are directly covered by the domain ontology of QVIZ. This is the typical case, because both the propositional description and the classification are highly dependent on the specific application domain.

Propositional content and content classification

Propositional descriptions are *knowledge object*s on the *subject matter level* of the KCO. Due to the fact that the domain ontology of QVIZ already uses the Information object design pattern, all *social object*s can be considered as propositional descriptions for the particular content objects.

The content classification is already provided by a feature of the domain ontology, that all *social objects* have the *has-quality* relation to the *quality* class.

Media characterization

The third element is the multimedia characterization. The following figure shows the design of this element.

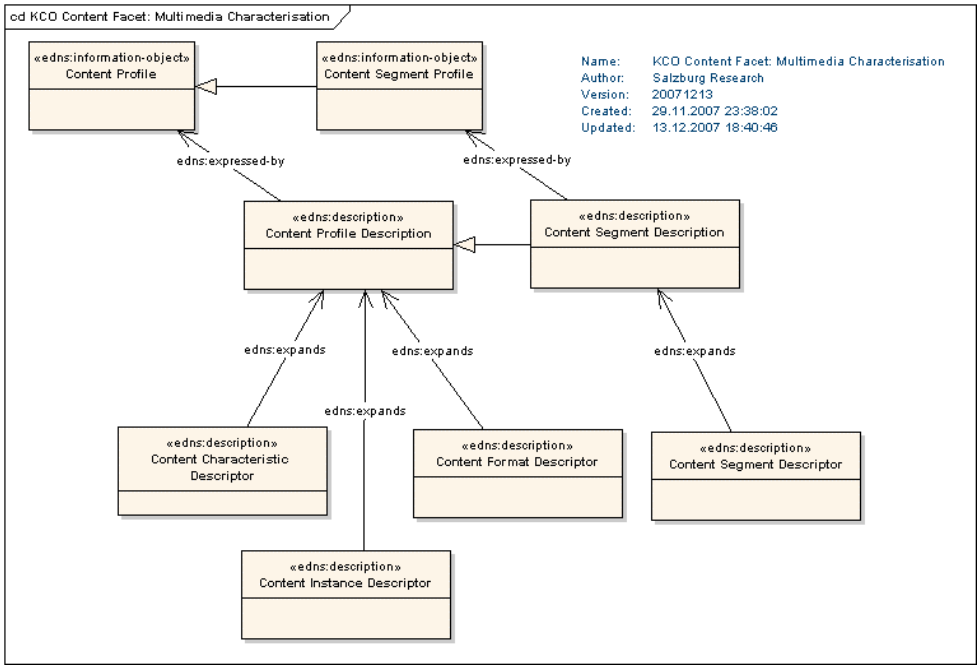


Figure 9: The media characterization element of the content description facet

The main concept of this element is the *Content Profile*. The *Content Segment Profile* is a specialization of the *Content Profile* and is used if characteristics of a content object segment have to be described. As predetermined by the information object design pattern, the actual description of the features for the content is

modeled with the *Content Profile Description* and its expansions. The *Content Characteristic Descriptor* defines a simple, but rough set of attributes defining general attributes of the content like media type, mime type, or file encoding. The *Content Format Descriptor* can be used to give specific information about a content object. The *Content Instance Descriptor* provides the link to storage locations of the Content Object. In the case of a *Content Segment Description,* the *Content Segment Descriptor* can be used to describe the spatial and temporal boundaries of the segment.

For a better understanding of the intended usage of this element within QVIZ, the following figure describes an example where a *Digital Archival Resource* is annotated with a *Content Profile*.

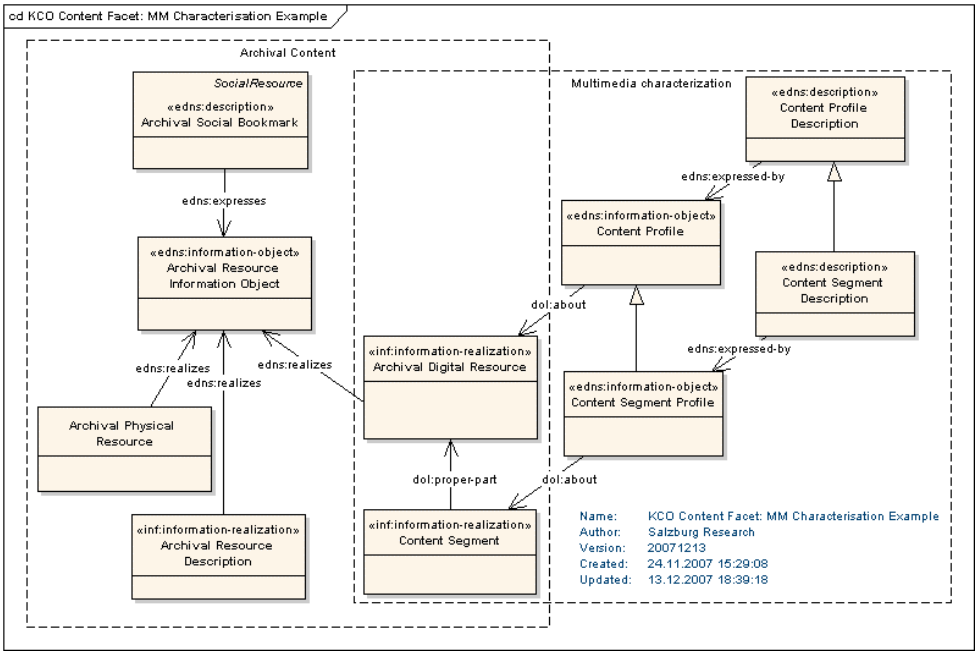


Figure 10: The KCO Content description facet: An Archival Digital Resource is described by a Content Profile; a Segment of that Archival Digital Resource is described by a Content Segment Profile.

Community Description Facet

The community description describes the organizational context in which knowledge and content can be used. This facet has three elements. Firstly, its gives information about the tasks typically performed with the knowledge and content; secondly, it provides information about the community including roles users can play in the context of this community; and thirdly its allows the tracking of activities performed in form of an usage history.

Two design patterns of the DOLCE foundational ontology and the *edns* (extended Description and Situation) extension are basis for the modeling of this facet. In the following figures, these two patterns are described in more detail. The first figure, Figure 12, shows the principle layout of the participation pattern.

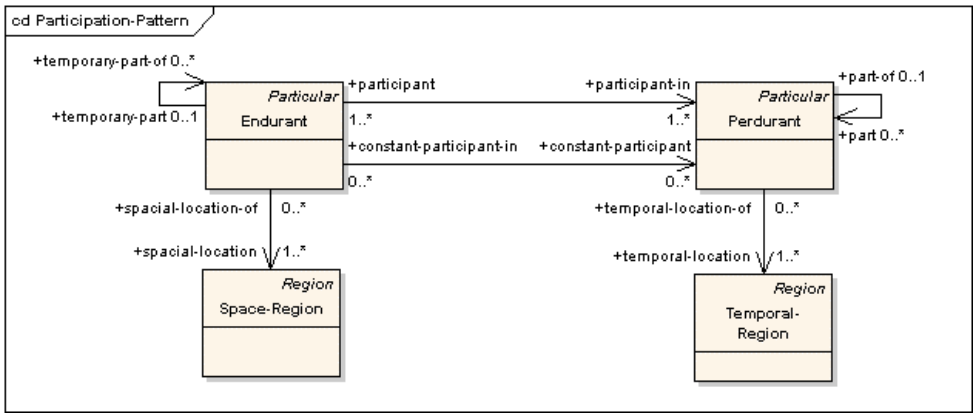


Figure 11: The Participation Pattern

The participation pattern defines possible relations between the top-level concepts in the DOLCE foundational ontology. It describes that *Endurants* (physical and abstract objects and individuals) can participate in *Perdurants* (any kind of happening, event, activity etc.). Both *Perdurant*s and *Endurant*s have *Region*s assigned. These *Region*s define their occurrence in space and time.

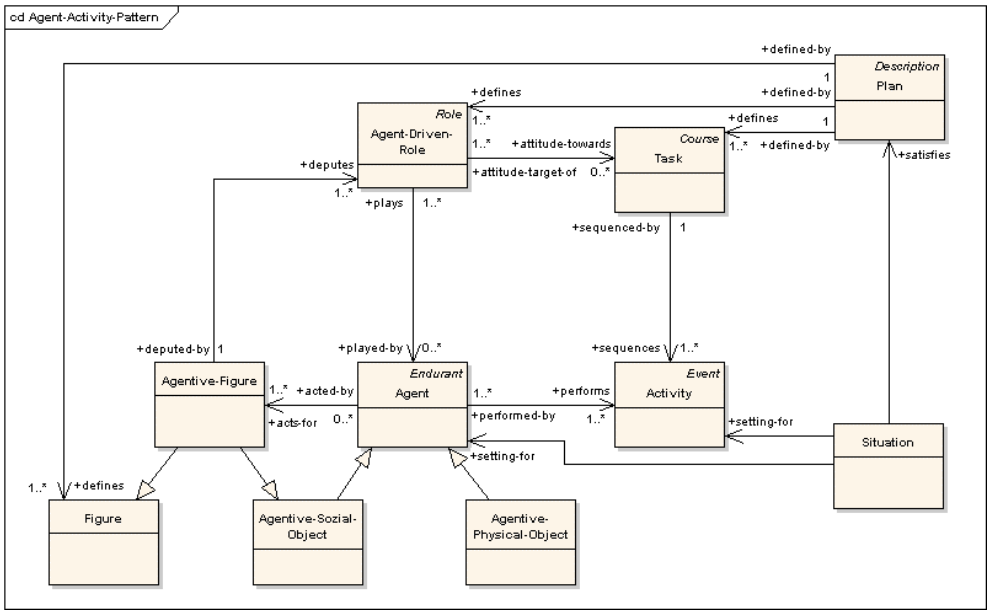


Figure 12: Agent Activity Pattern

The Agent-Activity-Pattern (Figure 13) is a specialization of the description and situation pattern. This defines *Agent* as an *Endurant*, which plays an *Agent-Driven-Role*. *Agent*s perform *Activities*. *Agent*s are divided in *Agentive-Physical-Object*s that mainly include natural persons and *Agentive-Social-Objects* that cover all sorts of organizations, teams and communities. This pattern also states that *Agent* may act on behalf of an *Agentive-Figure*. In that case the *Agentive-Figure* (e.g. a Community) must *depute* some power to a *Role* (e.g. Moderator) which is then played by the *Agent* (e.g. Sarah the Moderator acts-for Education Community of Practice).

Perform is a sub relation of the *participation* relation defined by the participation pattern described above. That means that by performing an action, an *Agent* is also *Participant* in an *Activity*. The above figure also indicates that *Activity* is a sub concept of *Event*.

Activities are *sequenced-by Task*s. This relation can be used for modeling a kind of typology for *Activities*.

A *Situation* provides the *setting-for Activities* and their participants. *Situation* can comprise a single *Activity* and all its participants, or a single *Agent* and all its *Activities*. But also *Situations* with several *Activities* and *Agents* are possible.

The last important statement of the agent activity pattern is that *Situation* can *satisfy Description*. This can be used to state, that an actual *Situation* corresponds to a planned one that is described in a *Description*. This relation is not of importance for the community facet, but will come into play if reality (the *Situation*) needs to confirm some regulations (the *Description*)

The community facet defines two information objects modeling the three elements. Firstly, the *content-usage* information object and the *content-usage-description* hold all planned interactions with the content. That includes the *Community*, *Roles* and *Tasks*. Secondly, the *usage-history* information object and the *usage-history-description* hold the information about the actual *activities* performed with the content. In addition, this description also allows the referring to the *Agents* that have interacted with the content.

User task, community and usage history

The following figure shows the layout of the community facet in the example of a collaborative document.

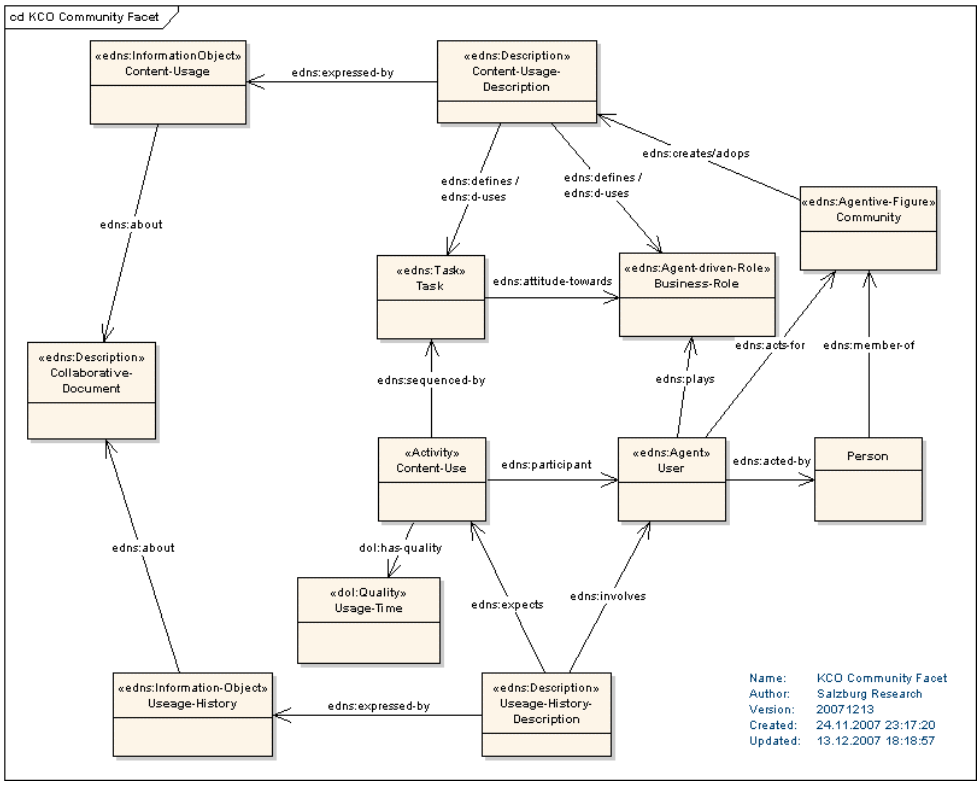


Figure 13: The Community facet for a single collaborative document

As with all the other facets, the *about* relation is used to attach the facet to the objects in the domain ontology. The two information objects and their description provide the linkage to the agent activity design pattern, which is used to express the knowledge of the *community facet*. The above figure also shows how the agent activity design pattern provides the linkage between the planned interactions (*content-usage-description*) and actual interaction (*usage-history-description*) with a *content-object*.

Business Description Facet

The business description contains a specification of the business semantics associated with the KCO. This comprises three elements: First the negotiation protocol that describes the business scripts by which a contract and the price is being negotiated. Second the pricing scheme, which is used for restricting the price policies that can be applied during the negotiation. The third element covers the resulting contract.

This facet is a realized as a domain specific variation of the community facet. It is based on the identical foundational design patterns like the community facets – the participation and the agent activity design pattern. All three elements of the business description facet are covered by the *Business-Profile* information object and the *Business-Profile-Description*. However the actual activities performed by agents on the content during the negotiation process can be stored by using the *Usage-History* element of the community facet.

Contract information and negotiation protocols

The following figure shows the design of the contract element for the business Description facet. Please refer to the description of the community facet for a better understanding.

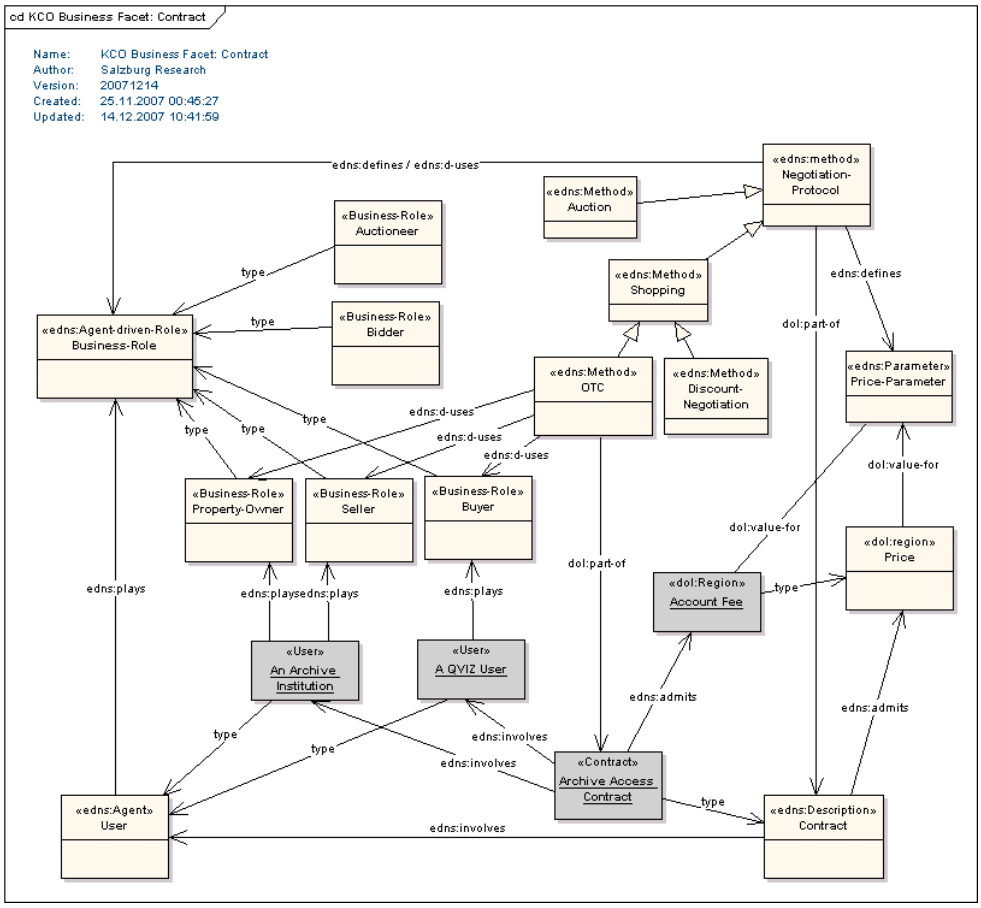


Figure 14: Business description facet

The figure above shows the more prominent parts of the business ontology of the KCO and also gives an example of a contract made by a QVIZ user to be able to access resources of an archive (gray rectangles).

The business ontology can define many types of *negotiation-protocols*. The above figure shows only the *OTC* and the *discount negotiation* as part of *shopping*, but there are also *auction* based negotiation parts of the ontology. Each of these negotiations define/use a set of *business-roles*, *business-tasks* (the tasks are not shown in the above figure) and may define additional *parameters*. The most prominent parameter of each negotiation is the *Price-Parameter.* The *contract* links the negotiation protocol with the parties involved in an actual negotiation situation. The *involves*-relation between *contract* and *agent* is actually a mediated relation where an *agent plays* a *role d-used-by* the *description* (in that case the *negotiation protocol* is part of the *contract*). The *admits*-relation is similar. It holds between a *region value-for* a *parameter d-used-by* a *description.* There is also a third relation of that type, which is not shown in the figure. The *expects*-relation mediates an *activity sequenced-by* a *task d-used-by* the *negotiation protocol.*

By the means of these three relations the contract can link directly to the contractors (agents) and the price.

In the example of an *Archive Access Contract* as shown in the above figure an *Archival Institution* and a *QVIZ user* are the contractors. Because of that, the *involves* relation holds in between these entities. In addition the *Archive Access*

Contract defines an *Account Fee.* Because of that the *admits* relation holds in between the *Archive Access Contract* and the *Account Fee*.

Licence information

In addition to contracts, the business ontology also defines licenses. This part of the ontology is shown in Figure 16.

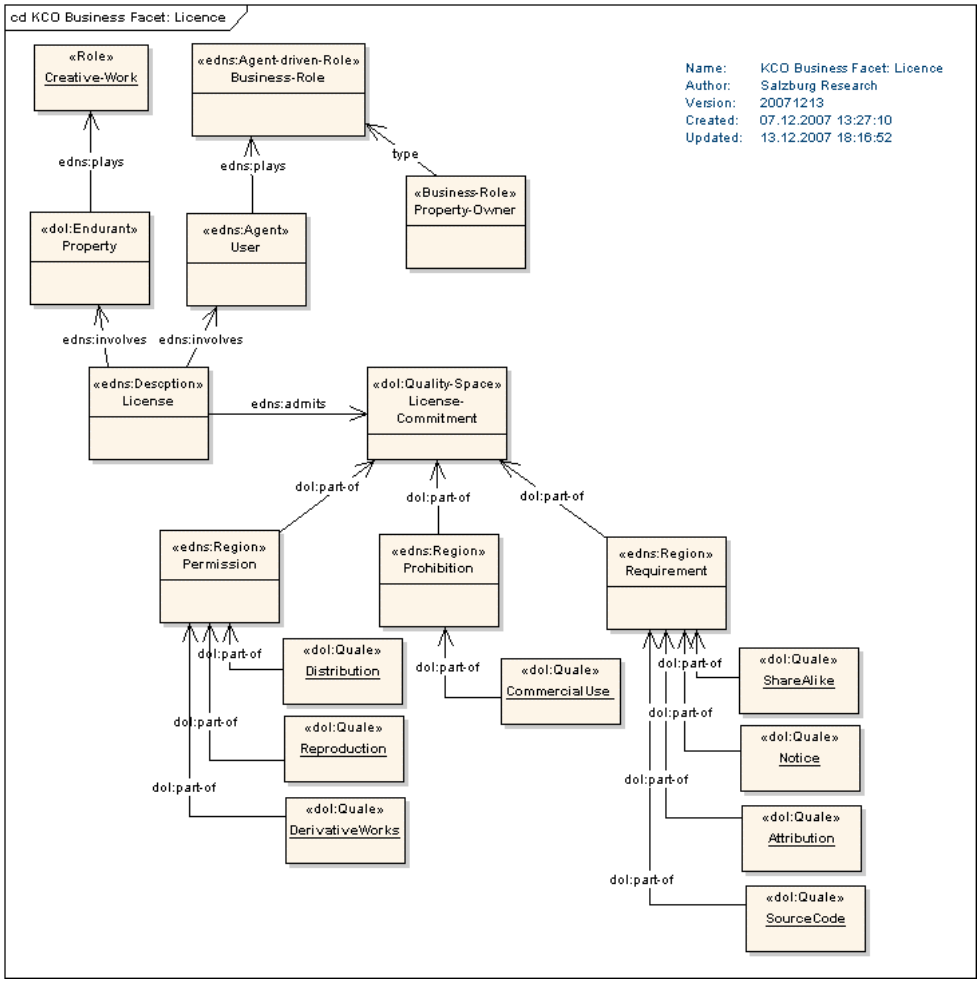


Figure 15: A license is a description.
The modeling of a license is very similar to the model of contract.

A *license* defines three important things. Firstly, each *license involves* a *property* paying the role of the *creative work* to be licensed. The *property* is modeled as *endurant* and therefore covers physical as well as non-physical objects. In the context of the business ontology it is important that such a *property* can only be licensed if it can play the *functional role* of a *creative work*. Secondly, each *license involves* at least one *agent.* Typically this a*gent plays* the *business role* of the *property owner.* Thirdly, each *license admits* some *license commitments*. The set of all possible *license commitment*s is modeled as a *quality space.* The different commitment types build *regions* within the *quality space*. Unique commitments within these *regions* are modeled as *quale*. Based on DOLCE a *quale* is an atomic part of a *region.* That means that a *quale* cannot be divided. The set of *regions* and *quale*, shown in the above figure, maps with the set of

Permissions, *Prohibitions* and *Requirements* as defined in the Creative Commons ontology.

The following figure provides an example of a license for a collaborative document as defined by the QVIZ domain ontology.

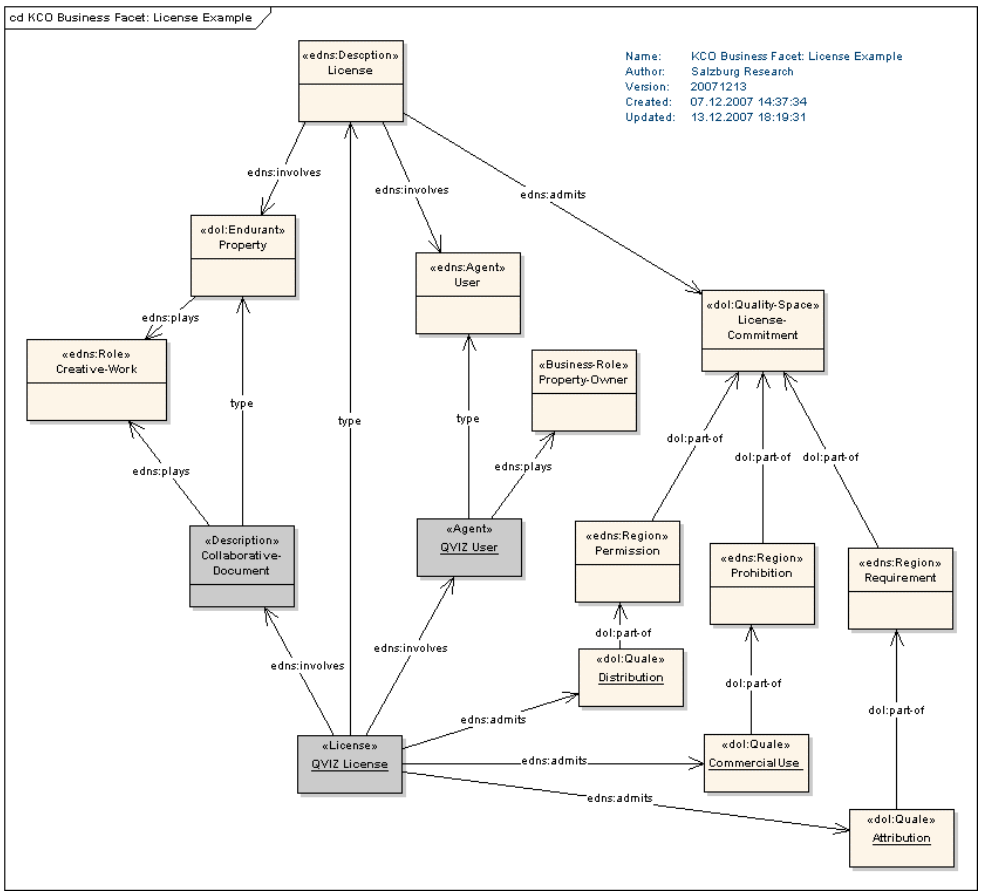


Figure 16: License example within a business facet

The gray rectangles show entities representing an example. All other objects are part of the business ontology as presented above.

The example presents a *QVIZ license* for a *collaborative document*. The *license* is granted by the *QVIZ user* who owns the *collaborative document*. Usually this will be the Author of the document, but the business ontology uses the *business role "property owner"* to represent that fact.

The *QVIZ license* in the above example *admits* three *license commitments.* The *license* grants the *permission* for *distribution,* prohibits *commercial use,* and requires *attribution*.

Business profile

After describing how the knowledge of the business facet is encoded in the business ontology of the KCO, the next figure shows the information object and the description of the business description facet.

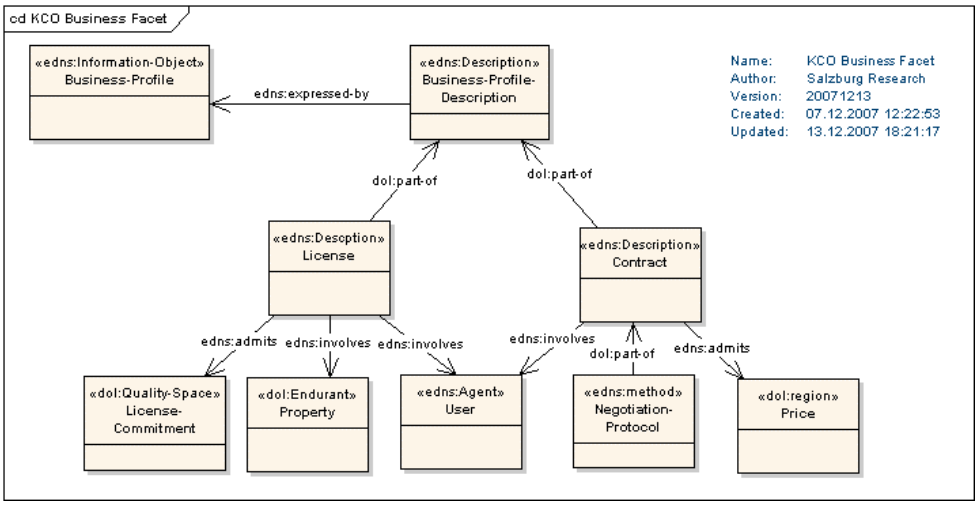


Figure 17: The business profile description

The business description facet defines a single information object – the *Business-Profile* and a single description – the *Business-Profile-Description*. The *Business-Profile-Description* can have multiple *license*s and *contracts* as its part.

Presentation Description Facet

There is a pronounced mismatch between the conceptual model for the rendering of, and interaction with knowledge based content. While most people would intuitively agree that presentation and interaction are closely intertwined, and many would also agree that a common model would be of benefit, reality looks different: Cascaded style sheets (CSS) are the chosen level of abstraction for rendering static content while keeping layout and design separate. Interaction as well as design is dealt with by various Web-toolkits such as GWT, Fresnel, Ajax. The most ambitious approach for presentation and interaction comes from the SMIL 3.0 standard for the Synchronised Multimedia Integration Language[4]. However, there is at present no implementation of 3.0 and even its predecessor SMIL 2.1 only boasts one single implementation[5].

The reason for the mismatch has evident causes, because the lack of "intelligence" in computers requires everything that has a bearing on the actual system output or user input, to be explicitly stated by a program. This leads to a huge number of parameters required to be specified, checked and mutual constraints to be taken care of. The issue is compounded by the fact that the (web-based) user interface is the place where the “browser-wars” rage between the big players. Even a large-scale initiative such as SMIL 2.1 cannot take off because two of the major players (Microsoft and RealNetworks) do not allow integration of their proprietary players with an open source offering.

No single project is capable of handling this Pandora's box and QVIZ must therefore accept that the project has to use more traditional user interface engineering methods and tools to bring the knowledge-enhanced content of QVIZ to the user. As a result, rather than providing a detailed ontological model of

[4] http://www.w3.org/TR/SMIL3/, retrieved 2007-11-26.

[5] http://www.cwi.nl/projects/Ambulant/distPlayer.html, retrieved 2007-11-26.

presentation and interaction, it describes the principle that can be used as a reference for actual implementations.

Time based spatial rendition

This sub-facet of the KCO specifies how each of the identified media items (full items or specified segments) will be rendered in time and space.

The rendition process is described by partial functions that combine knowledge items, the time line, and a representation of the output channels of the interface. The reason for proposing abstract output channels is that they can associate properties with the output channels that give further control over the rendering process.

An output channel can be associated with an interface space in four dimensions, where *x,y,z* are the spatial dimensions of an assumed 3D space and *p* is a time index denoting time periods which can be associated with Allen's temporal logic of time intervals [Allen 1983].

Temporal rendering can be composed in *lanes* or *tracks* for each specified media item or media segment. The primitive operators start(at time), pause(duration), mute(duration), and stop(at time**)** need to be implemented. *Mute* means *not audible* for audio tracks, and *not visible* for video tracks, images and text.

Interaction based rendition

This component of the KCO specifies how each of the identified media items interacts with an end user (if such interaction is defined for the type of media and/or for the knowledge structure described by the logic description). The semantic annotation specifies whether the presentation is entirely pre-programmed, whether it is entirely open (e.g. web based navigation) or whether it follows some dialogue pattern where humans and the system take conversational turns in order to navigate in the knowledge/information structure.

For the interaction, *dialogues* between *actors* are specified. A dialogue is a named situation in which several actors are *engaged*. The dialogue is described with reference to *events* that are points in time (either defined absolute or relative to some other event).

Issues arising from combining time-based presentation and interaction

In the case of differing rendition directives between time based spatial rendering and interaction based rendering, a *mode priority* flag in the KCO self-description will decide how to resolve the conflict. By default, interaction has a higher mode priority than time based spatial rendition.

If there is an interaction that includes a decision by the user then there is a need to describe *possible worlds*, i.e. follow different paths of interaction. Combined with the spatiotemporal model this leads to complex descriptions that can only be handled by a full-blown ontology that is rooted in some form of modal logic.

For QVIZ, the project does not envisage this type of complex multimedia navigation scenario and for the reasons given earlier, it does not engage in deep ontological modeling of the presentation and interaction facet.

Trust and Security Facet

A KCO based system supports the possibility to define novel business models between consumers and producers of content. The KCO structure aims at offering both sides flexibility and the possibility to engage in negotiation about mutually acceptable terms of use. What is important to note is a change of philosophy: QVIZ is interested in defining a business model and its semantics, rather than putting big locks on the content. This is due to a separation of concerns between the business description facet and the trust and security facet: the trust and security facet of the KCO is the place where one can put the *big locks* if needed, whereas the business description facet only describes negotiation, licensing and pricing, but not the enforcement of any such regulations.

There has recently been a shift even in research on business models and rights management for digital content, acknowledging that there needs to be a balance of trust and security between producers and consumers, otherwise a market will not perform optimally. The separation of he concepts for business model, trust and security in the KCO is an attempt to address this separation of concerns.

Trust

QVIZ defines trust as the need for feeling secure. This need holds for anybody moving around in virtual worlds and committing to actions such as buying or even just viewing content.

The *trust* aspect of this facet makes it possible for the KCO to carry metrics of usage around: how often has this KCO been copied, but not paid for? How well has the vendor reacted to customer queries? For this, the trust sub-facet can use information from the usage history. However, the usage history need not necessarily be used (indeed, it can be disabled if the vendor wants to guarantee anonymity to the users of KCOs). In current systems, it is the portal which offers such trust related statistics. In a future system, it is quite conceivable that each content object could carry a reference to the trust statistics that may be collected by a mutually trusted third party! It is in our view, important to provide an infrastructure that is able to support restrictive as well as liberal uses of content and where the terms of these uses are transparent to all parties involved in the contract.

Security

QVIZ defines security as any measure taken for some stakeholder to feel secure. In the definition of KCOs, we identified the security facet as the place where digital rights management systems should interface with the KCO model. If there was any requirement in QVIZ to address digital rights management, then an ontological subset of a rights expression language would need to be developed specifically for QVIZ.

The balance between trust and security

In the context of a KCO, the vendor of some content may require consumers to first give their personal details and email address before they are allowed to access the content, and the content may be watermarked to stop people from reselling it without paying. Such a model is heavy on security, but does nothing for trust. The seller signals "we don't trust you, the customer". The customer gets nothing in return for giving away his or her personal details. Therefore, such a model would be fairly secure, but probably sub-optimal with respect to market potential.

For QVIZ, some of these considerations are of interest: for example, communities of practice which are based on free sharing (and contributing) may get out of balance if their content is used by others to create paid content from it.

However, at present, there are no trust and security considerations in any of the usage scenarios which would suggest a deeper need to address this facet of KCOs in QVIZ further.

6. An example for using KCOs in QVIZ

The main motivation for creating KCOs is to export content together with bits of contextual knowledge to external systems. Exporting to an external system requires some overlapping in the syntax and the semantics of the content. In one of the QVIZ professional usage scenarios (D3.3 Domain Ontology), Peter is using the QVIZ archival search and its community features to prepare a feature article on "The shootings in Ådalen".

The transmission of the content of the article from the community site is done via copy-paste and simple references. Using KCOs provide more advanced features to move content and knowledge from one information system to another.

Imagine that Peter not only works for one newspaper but needs the flexibility to transfer his work to several authoring systems. As he works as freelancer, he will not get an account for the advanced content management systems of all magazines, but he has to submit a package, that allows semantic enabled systems to interpret the information.

The following table describes an example of what information the KCO of on one specific article - "The shootings in Ådalen" - will contain:

KCO Facet	**KCO Sub-Facet**	**The example KCO contains the following ontologies, schemas and data.**	**What one can has done before at the QVIZ collaborative environments KCO engine.**
Content Description	Propositional Content	The QVIZ domain ontology as a means of describing objects that is related with the Peter's main popular article, such as "Archival Social Bookmark" or "Community" or "Administrative Unit".	Any resources can be typed based on the propositional content. Assertions can be added to the knowledge base of the propositional content facet Any typing system is allowed if it is defined in the ontology. If a "related" or "part-of" concept is introduced in the domain ontology then it can be used to refer to other resources.

	Content Classification	A content classification schema and the actual content classifications that include all "keywords" of the article. If the tags are aligned to vocabularies, these would be referenced here.	Allows for the definition of metadata schemes Resources can be classified based on controlled vocabularies, arbitrary schemes can be defined.
	Multimedia Characterization	Multimedia-Schema and media-types derived from it such as TEXT as well as mime type information. This sub-facet will also directly link to the actual HTML content file.	E.g. selection of player depending on multimedia type Metadata and references to the content are stored in the multimedia characterization facet
Presentation Description	Spatio-temporal rendition	A standard CSS for simple rendering of HTML pages.	A choice of presentation mode depending on the presentation descriptors (this may match with CSS or some time-based rendering e.g. SMIL
	Interaction-based rendition	Not applicable for not interactive contents in this example.	Interaction leads to "possible worlds" of presentations
Community Description	User task	Schema of tasks valued by the vocabulary of document roles such as scientific article, popular article.	All stages of a content workflow can be described and supported by the definition of user tasks.
	User community	A description of the designated communities for this content.	Any community of practice can be defined and tasks/workflows can be associated with the CoP.
	Usage history	The usage history reveals whether a KCO is referred to by one or more communities of practice.	The usage history is built up from logging the actions of users and it is thus possible to make inferences about the publication state.
Business Description	Negotiation protocol	There is at present, no requirement to model the negotiation between a requester and a provider of archival resources.	Negotiation protocols such as eBay.

	Pricing scheme	Not applicable for this example, because this would need to integrate the archives business models.	Pricing schemes.
	Contract information	Holds Creative commons license for the Document.	The business facet holds any static contractual information, e.g. the type of license under the creative commons model.
Trust & Security	None	The KCO will include the anonymous access to the QVIZ community together with its access policy, but not specific roles within a community.	Access policies can be defined and enforced
Self-Description	None	The "table of contents" of the KCO, which will indicate, which (sub-) facets will populated.	The KCO contains a semantic header that gives information about the KCO's content.

Figure 18: An example of a QVIZ KCO.
The Article "Shootings in Ådalen" as a package for external systems.

7. Conclusion

This report introduces "intelligent content objects" and their business model, and then describes the base model of a KCO and its specialization for the needs of QVIZ. The facets, for which the QVIZ application will provide data, are defined in detail.

These semantic descriptions will provide the information for the domain context, the content description and its classification, a licensing model for creative content, the community and usage history.

The KCO schema will be used in the QVIZ final prototype to create example KCOs by using the QVIZ tools for handling knowledge content in KCOs. Its application will be described and delivered in connection with the final prototype and its respective software component

The schema included with this document provides the model for the QVIZ KCO, and imports the domain ontology (QVIZ D3.3). It will guide the generation of actual instances of KCOs from the QVIZ collaborative environment.

8.References

[Allen 1983] J. Allen: Maintaining Knowledge about Temporal Intervals; in Communications of the ACM, 26(11), pp.832-843, November 1983.

[Maass 2007] Maass, W., Behrendt, W., Gangemi, A, (2007). Trading digital information goods based on semantic technologies. To appear in JTAER (Journal of Theoretical and Applied Electronic Commerce Research).

[Metokis D20] Metokis Deliverable D20 - Final Methodology and Business Principles for knowledge sharing (Restricted Report, 2005), http://metokis.salzburgresearch.at

[Metokis D21] Metokis Deliverable D21 – Final Design Specification for a Knowledge Content Carrier Architecture (Restricted Report, 2005), http://metokis.salzburgresearch.at

[SMIL] Synchronised Multimedia Integration Language. http://www.w3.org/TR/SMIL3/, retrieved 2007-11-26.