

Towards an Ontology-Based Distributed Architecture for Paid Content*

Werner Behrendt¹, Aldo Gangemi², Wolfgang Maass³,
and Rupert Westenthaler¹

¹ Salzburg Research GmbH, Jakob-Haringer Strasse 5/II,
5020 Salzburg, Austria
{Werner.Behrendt, Rupert.Westenthaler}@salzburgresearch.at

² Institute of Cognitive Sciences and Technology (CNR),
Via Nomentana 56, 00161, Roma, Italy
a.gangemi@istc.cnr.it
<http://www.mcm.unisg.ch>

³ mcm^{institute}, University of St. Gallen, Blumenbergplatz 9,
9000 St. Gallen, Switzerland
wolfgang.maass@unisg.ch
<http://www.mcm.unisg.ch>

Abstract. Business models on the basis of digital content require sophisticated descriptions of that content, as well as service-oriented carrier architectures that allow to negotiate and enforce contract and license schemes in heterogeneous digital application environments. We describe Knowledge Content Objects (KCO), that provide expressive semantic descriptions of digital content, based on an ontology of Information Objects, built under the DOLCE, DnS and Plan Ontologies (DDPO). In particular, we discuss how this structure supports business requirements within the context of paid content. Interactions between agents are embedded into digital infrastructures that are implemented on an advanced knowledge content carrier architecture (KCCA) that communicates via a dedicated protocol (KCTP). We show how this architecture allows to integrate existing digital repositories so that the content can be made available to a semantically rich digital environment.

1 Introduction

The WWW can be perceived as a huge information market where supply and demand meet. If content obtains high value to certain demand sides, it will generate market prices. This kind of content is generally termed "paid content" as a special form of information goods and is viewed as a digital product ([4, 16]). [12] define the term *information good* very broadly. Based on the definition of [16] anything one can send

* This work is part-funded by the European Union (6th Framework Programme under the strategic objective IST-2002-2.3.1.7. - Semantic-based Knowledge and Content Systems) and the Swiss Federal Government.

and receive over the Internet has the potential to be a digital product. The term *paid content* is in this article used as the non-free sales and distribution of information-based content products.

A hurdle for effective markets for paid content is the non-existence of appropriate transaction mechanisms that support search, usage and control of paid content so that suppliers can implement sustainable business models and users can efficiently obtain information about content properties and can efficiently use purchased content. Paid content needs to support two kinds of situations: *trading situations* and *usage situations* [11]. During a trading situation a consumer and a vendor negotiate the terms under which a consumer gains rights that can be executed on a particular content. To gain advantages of search products in a trading situation, consumers must be supported by product information on the (1) *utility of content* in respect to intended application situations, (2) *resource restrictions* and (3) *application requirements* that are given by the intended application environment. Resource restrictions encompass organisational, temporal, spatial, presentational and financial dimensions that are relevant during the trading act. The latter category describes in which application environments a particular instance of a paid content is intended to be used in principle. This encompasses its (1) situational requirements, i.e. when it can be used by whom, (2) business requirements, i.e. which contractual obligations, pricing and license schemes apply, (3) constraints on the technical environment in which it can be used, (4) how it presents itself and (5) which requirements are to be met while using it.

Any mismatch of these three categories decreases the utility of a particular content and influences the consumer's buying decisions negatively.

Throughout this article we will explore the potential of semantic annotations of paid content that provide on one hand an opportunity for interoperable markets for paid content and on the other hand a means for product self-descriptions which has strong influence on consumer buying decisions. First, we will briefly discuss application situations from which we will derive requirements for the intended semantically annotated content structure and the underlying technical content carrier architecture. In subsequent sections, we will introduce a semantically enriched content carrier structure (KCO) that is used as a flexible and expressive container for digital content. KCOs are exchanged over a transmission infrastructure that is based on a generic content carrier architecture (KCCA), which enables the interoperation of heterogeneous content repositories. Finally, we will summarize the status of our current work and give an outlook on our future work.

2 Semantic Modeling of Content Objects

An analysis of several hundred existing paid content business models ([13]) resulted in a classification of five central elements to which digital content has to respond during different phases of its life cycle.

1. *Content descriptions*: provides the propositional content that is announced by a digital content on an abstract level.

2. *Community descriptions*: information about the organisational structure (roles, rights and obligations) by which a content product can be used and information that influences trust such as certificates and brand name information.
3. *Business descriptions*: describes the business and legal requirements during information and negotiation phases. After a purchase, contracts will be enforced according to mutually agreed rights and obligations.
4. *Presentation descriptions*: describes the presentation modes to which the information of a content product can be adapted to by rendering and other application-specific means.
5. *Trust and Security descriptions*: content must be associated with some measure of trust for the end user, and for the content provider, there must be some security features which guarantee that the content will not be illegally copied or otherwise misappropriated.

As these results show, digital content needs to be semantically annotated so that it can respond to these five elements. We will now incrementally introduce the concept of a Knowledge Content Object (KCO) that is intended to provide this structure on a computational level because it is intended to be implemented in digital infrastructures. From the foundational ontological viewpoint given by DOLCE and its extensions (a modular library called DDPO), a KCO is to be distinguished from the abstract concept of an information object that carries meaning on cognitive and abstract level, independent on any technical realisation. Because we want to leverage the advantages of foundational ontologies for the exchange and translation of meanings on technical but also non-technical level, KCOs are embedded into DDPO.

2.1 An Ontology of Information Objects

We lay down here a semantic foundation for KCOs, based on an ontology of information objects.

Our ontology for information objects is an extension of *DOLCE* (Descriptive Ontology for Linguistic and Cognitive Engineering), *DnS* (Ontology of Descriptions and Situations), and *Plans* Ontologies. Parts of the reused ontologies have been originally developed within the WonderWeb [8] and Metokis EU [6] projects. We will refer here to this extended ontology as DDPO [6].

The main distinctions in the reused ontologies, which are imported here, include:

- the topmost class is called *particular* (any entity)
- *objects* (e.g. a *dog*) and *events* (e.g. a *barking*) belong to disjoint classes
- *physical* (e.g. a *brick*) and *social* (e.g. a *contract*) objects belong to disjoint classes
- attributes of particulars (e.g. a *color*, or a *spatio-temporal location*) are represented as *regions* within *quality spaces*, with a possible associated metrics
- social objects include *descriptions* (the public, communicable counterpart of agents' conceptualizations, including also *plans*), which can define *concepts* (the *customer role*), encode (or be expressed by) *information objects* (a *sentence*, or a *music chart*), provide unification criteria for *collections* (a *group of people*), etc.
- concepts can be either *roles* played by objects, *tasks* executed during actions (e.g. a *door opening task*), etc.

- concepts from descriptions provide constraints for other particulars: if a configuration of particulars satisfy those constraints, a *situation* emerges that satisfies the concepts' description (typical applications of constraint unification include *regulations, plans, social relationships*, etc.).

The previous distinctions are supported by a large axiomatization that cannot be reported here.¹ We'd rather concentrate on their application as a foundation for KCO implementation and deployment.

For example, a usage context of a content object may require to talk about the digital reproduction of a painting that is owned by an institution, and such institution is willing to commercialize the reproduction at certain conditions that include differentiation for users, pricing, regulations to be followed, inclusion of content metadata, explanations, interpretations, ways of rendering it, etc. This context is complex, and require a subtle understanding of the different entity types involved in it.

According to DDPO, a content (information) transferred in any modality is a kind of social object called *Information Object* (IO). Information objects are *spatio-temporal reifications* of pure (abstract) information as described e.g. in Shannon's communication theory, hence they are assumed to be in time, and realized (materialized) by some entity.

Information objects are the core notion of a *semiotic ontology design pattern*, which employs typical *semiotic relations*, as explained here. The complete IO ontology is quite complex, and is presented elsewhere [6][8].

We present the axiomatization of KCOs in OWL abstract syntax. We firstly present the definition of `DnS:information-object`, which encodes the basic axioms of an ontology of semiotics extending the basic DDPO ontology:

```
Class(DnS:information-object complete
  intersectionOf(
    DOLCE:social-object
    restriction(DnS:about allValuesFrom(DOLCE:particular))
    restriction(DnS:realized-by
someValuesFrom(DOLCE:information-realization))
    restriction(DnS:interpreted-by
allValuesFrom(Actions:agent))
    restriction(DnS:expresses allValuesFrom(DnS:description))
    restriction(DnS:ordered-by someValuesFrom(DnS:information-
encoding-system)))
```

The definition says that information objects:

- are necessarily *encoded* by some information encoding system
- must be *realized* by some particular
- can *express* a description, and, if that description is satisfied by a situation
- can be *about* that situation, or some entity in its setting
- can be *interpreted* by agents that can conceive the description expressed by said IOs.

¹ The full OWL axiomatization of DOLCE, DnS, DDPO, IO, etc. can be downloaded from: <http://dolce.semanticweb.org>.

For example, Dante’s *Divine Comedy* is an IO, it is *ordered* by Middle Age Italian language (the information encoding system), is *realized* by e.g. a paper copy of the 1861 edition with Doré’s illustrations, *expresses* a certain plot and its related meanings (literal or metaphorical), as *interpreted* by an agent with an average knowledge of MA Italian and literary studies, and it is *about* certain entities and facts.

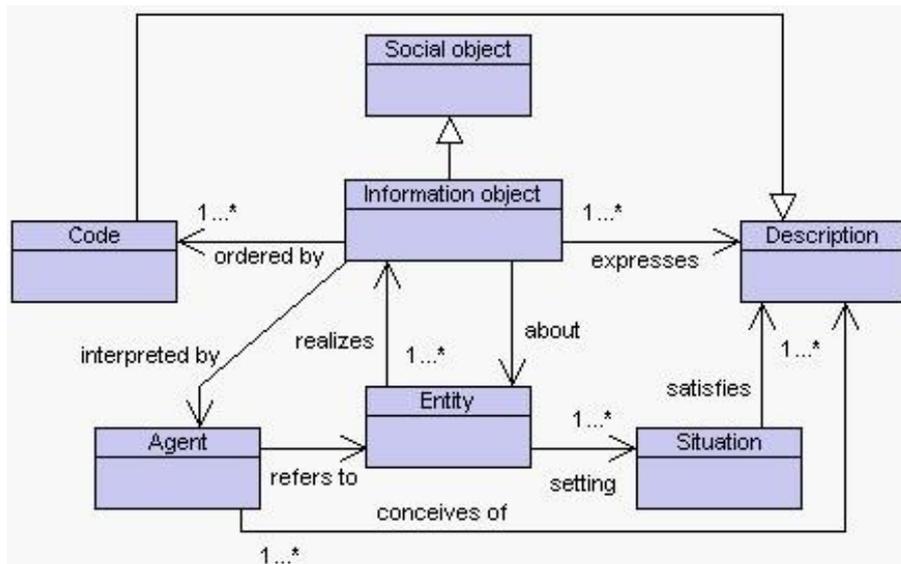


Fig. 1. The IO pattern: an information object is a social object, *ordered* by a code, and *realized* by some concrete entity. It *expresses* a description *conceived* by some agent, *about* some entity. Situations exist for the setting of *realization* (“communication”), as well as for *aboutness* (“reference”); agents *refer* to entities that IOs are about while interpreting them

The relations *realizes*, *expresses*, *about*, and *interprets* must be taken as *temporally indexed*, but such indexing cannot be expressed directly in the OWL property definition; for example, the definition of `DnS:realizes`:

```
ObjectProperty(DnS:realizes
  inverseOf(DnS:realized-by)
  domain(DOLCE:information-realization)
  range(DOLCE:information-object))
```

needs to be complemented by an OWL axiom stating that something that realizes an IO must be present at least in some time interval at which that IO is also present:²

² The axiom does not completely catch the semantics of a real ternary relation: `realizes(x,y,t)`, where `t` is a time interval, but it is a useful approximation anyway. Further refinements could be made by using SWRL [7].

```

SubClassOf(
  restriction(DnS:realizes      someValuesFrom(DnS:information-
object))
  restriction(DOLCE:present-at someValuesFrom(intersectionOf(
  DOLCE:time-interval
  restriction(DOLCE:time-of-presence
someValuesFrom(DnS:information-object))))))

```

These *semiotic* relations constitute a typical *ontology design pattern*, so that any composition of relations can be built starting from any node in the pattern or in an application of the pattern.

The pattern has also some additional axioms, for example, the property *interprets* implies that an expressed description is *conceived* by the agent (i.e., when an agent interprets an IO, it conceives the description expressed by the IO; of course two agents can conceive different descriptions, then resulting in different interpretations).

Once introduced the concept of an information object, we will now describe informally, the intuition and resulting general structure of a Knowledge Content Object (KCO). This work is based on results from predecessor projects: in the CULTOS project (www.cultos.org) we introduced the notion of an enhanced (by explicitly stated domain knowledge) multimedia meta object (EMMO). The logic description of KCOs is based on these ideas, firstly formulated in [9] and initially formalized without ontological grounding by [10]. The business related aspects of KCOs are based on lessons learned in the INKASS project ([1]).

3 Semantic Modeling of Knowledge Content Objects (KCO)

The notion of knowledge content objects is based on business requirements (see section 1), and builds upon previous approaches to multimedia and hypermedia document models. Related work includes [3, 14, 17]. The strength of KCOs lies in the combination of business- and domain-specific semantics that are tied into DDPO.

We have come to distinguish four abstraction levels of a KCO:

1. *Generic KCO (physical) schema*: a physical data structure (subClassOf “computational object” in DDPO) that realizes an abstract data structure (subClassOf “information object” in DDPO).
2. *Tradeable domain KCO (physical) schema*: a a physical data structure that specializes of a generic KCO schema, including an ontology of a domain (e.g. a CD ontology), and an ontology for the related business semantics. By default, we assume that this abstraction layer also features the description of a particular business semantics (see next section). There can be several levels of domain schema specialization.
3. *Instantiated KCO Prototype (Master Copy)*: a physical data structure implementing the same facets (and business semantics) from a tradable domain KCO schema, but that also contains, at a certain point in time, a particular digital content including all semantic annotations required by the corresponding tradable domain KCO schema.
4. *KCO Instance (Copy)*: the clone of an instantiated KCO prototype. It is distinguished here for business reasons; for example, when a user is granted access to a content object, and depending on the contract semantics, this copy could

change over time, through alterations made by the owner. For example, somebody may buy a backing track for a pop song, in order to add her own voice to the recording.

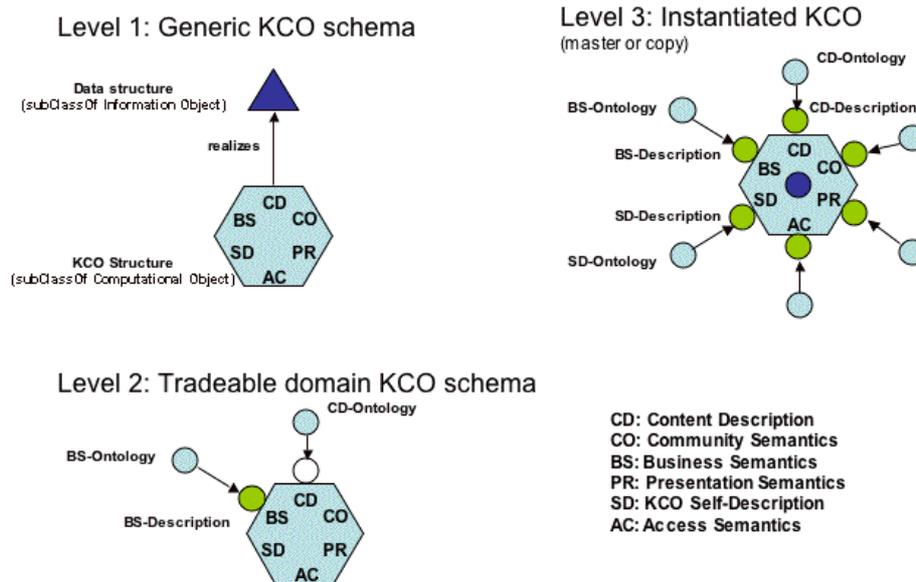


Fig. 2. Levels of KCO abstraction and KCO facets. All levels include physical data structures

Derived from the analysis of business models and paid content products, we have developed a KCO structure consisting of eight facets. Several of these facets are subdivided into further KCO sub-facets. At the "atomic" level, it is intended that each of the leaf elements is associated with well-defined operational semantics, in order to enable organisations to quickly deploy KCOs as part of their information infrastructure.

While KCOs are also rooted in semantic web technology - using an extension of DDPO for their definition - our application interest is more strongly geared towards the following question: "what information and knowledge exchange processes can be actively (i.e. operationally) supported by semantic web technology". In particular, we are interested in how traditional digital content can be enhanced in order to qualify as "knowledge" content.

The first facet is what we call the *content description*. The KCO carries a list of media references which are intended to point to real media files. So the collection of these referenced media files is actually, the full intended content of the KCO. In order to make this content accessible for machines, we provide a simple referential mechanism to associate arbitrary logic descriptions to the media files. For this, the propositional content facet is linked to a domain ontology which represents the universe of discourse for all content descriptions of this KCO. When an instantiation

of a KCO is created, then arbitrary selections of the multimedia assets can be associated with statements that are valid according to the ontology. The semantic annotation is very flexible as it can relate to segments of a media asset (e.g. a scene in a video or a region in an image) or even to a relationship that holds between some media assets. (For example, it holds for the novel "Don Quixote", that it is a parody of the chivalrous romantic epic of "Orlando Furioso". This relationship holds between the "prototypes" of the two novels, and between all derivatives of the two. So by stating it once, we assert this knowledge to all instantiations (i.e. copies) of the novels. The details of creating media semantic networks have been described elsewhere [e.g. 14].

The second facet is the specification of time-based spatial *presentation* of complex content. Given some media tokens, we specify on one or more temporal "tracks" which describe *when* the associated media data will be rendered, and *where* they will be rendered (in terms of spatial arrangements). In an operational environment, this component may use elements of the SMIL multimedia synchronisation language for its implementation.

Table 1. KCO facets

Elements	Facets	Sub-Facets
Content Description (CD)	(1) Propositional Content	Media references
		Logic descriptions
Presentation Description (PR)	(2) Spatio-temporal rendition	
	(3) Interaction-based rendition	
	(4) Multimedia characterisation	Media properties Content classification
Community Description (CO)	(5) Usage context	User task
		User community
		Usage history
Business Description (BS)	(6) Business and legal semantics	Negotiation protocol
		Pricing scheme
		Contract
Trust & Security (TS)	(7) Semantics and pragmatics of confidence in virtual goods	
Self-description (SD)	(8) The description of the KCO's semantic structure (schema)	

The third facet deals with *interaction* and *dialogue*. Here, 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 system take conversational turns in order to navigate the knowledge/information structure. This description defines one or more discourse structures that can be associated with the content for its rendering.

The fourth facet contains *interfaces to existing metadata standards*, notably those in the cataloguing and media management areas. Its purpose is to enable the migration

of media and their associated meta data into the KCO structure. We envisage this to be the place where meta data harmonisation can be done.

The fifth facet, *community description*, describes the context in which a content can be used. This covers three sub-facets: *user tasks*, which are formally described by reference to an ontology of plans and tasks; *user community*, which describes the situations with corresponding roles (rights and obligations) that users would take in order to manipulate or consume the content; and *usage history*, keeping traces of previous use in order to support workflow systems as well as collaborative filtering systems. The latter can be achieved by keeping track of user data when the KCO is being "touched" by that user. Depending on legal and contractual aspects of this facet, the filtering may be more or less anonymous.

The sixth facet, *business description*, contains a specification of the business semantics associated with the KCO. This comprises the sub-facet negotiation protocol which describes the business scripts by which a contract is being negotiated. A negotiation protocol is described as a DDPO *plan* that can be represented and processed in OWL-DL format [6]. The pricing scheme is used for restricting the price policies that can be applied during the negotiation. It is grounded in DDPO as the *regulation* concept. In the simple case of a fixed-price scheme, the negotiation is reduced to a simple over-the-counter (OTC) purchase. The pricing scheme is required for price differentiation strategies that are defined by the seller on the basis of a differentiating factor such as age, quantity discount or date of content origin (see [13, 15]). The resulting contract is also a plan in the DDPO ontology that describes the situation in which agentive roles can be taken by agents and act by using described tasks. We distinguish between pre-existing content and prospective content. Pre-existing content is already available at contracting time while prospective content is produced during execution time of the contract.

The seventh facet, *trust and security*, is currently deemed out of the scope of our project although we acknowledge the need for inclusion of the issue, in the overall framework of METOKIS.

The framework is rounded off with the eighth facet, *self-description*, which exports the basic structure and formal semantics of the KCO to external systems that may want to make use of the KCO structure and its supported semantics.

In the following section, we will translate the KCO model into an OWL ontology, which specializes the IO branching of DDPO foundational ontology.

3.1 KCO Formalization

The IO design pattern can be used and extended in order to characterize Content Objects (CO) and Knowledge Content Objects (KCO). As introduced, IOs can be *realized* by any sort of entities. The realization relation provides us the expressivity to talk about KCOs: a KCO is described here as a *physical data structure*, a subclassOf *computational object*, which is a subclassOf *information realization* (that can be any entity, ultimately relying on some physical object, substance, event, etc.). A physical data structure realizes a (abstract) *data structure*, which is a kind of IO.

KCOs are distinguished from digitalized COs (*content objects*), which are another kind of computational object. A KCO provides an implemented data structure (a *frame*) to COs. The relation between a KCO and a CO is provided by the property: `KCO:realizes-frame-for`:

```
ObjectProperty(KCO:realizes-frame-for
  domain(KCO:physical-data-structure)
  range(KCO:content-object))
```

Such a property is complemented by an appropriate axiom that states that the KCO realizes a data structure that is a *frame* for a content object.

These assumptions allow us to give a foundation to the operationalized KCO model in the KCCA architecture: each *facet* in that model is formalized here as an OWL property, linked to the DDPO ontology.

Some basic distinctions are made firstly to catch the different states of the implemented data structures: in the KCO ontology, the implementation of the most generic data structure (the facets without any values) is called `KCO:generic-KCO`. The class of domain-oriented data structures (the facets with specified “types” for the value of the facets), and a given business semantics, is called `KCO:domain-KCO`. The class of KCOs themselves (the implemented physical data structure with at least one value filling a facet type) is called: `KCO:KCO`. Two properties of KCOs are conceived: `KCO:master-of`, used to characterizes the first implementations of each KCO (this is specially relevant with reference to the masters of content objects that are framed by the KCO data structure), and `KCO:copy-of`, which is used to characterize the copies of the master. Different modification rights, contracts, and pricing schemes apply to the masters.

Once introduced the intended meaning of KCO, and its reference to abstract data structures and content objects, we lay down our characterization of facets in terms of OWL properties.

Firstly, we summarize the properties as from the OWL definition of KCO:

```
Class(KCO:KCO complete
  KCO:physical-data-structure
  restriction(DnS:realizes someValuesFrom(intersectionOf(
    KCO:data-structure
    restriction(KCO:instantiates someValuesFrom(intersectionOf(
      KCO:data-structure
      restriction(DnS:realized-by someValuesFrom(KCO:domain-
KCO)))))))
  restriction(KCO:realizes-frame-for
someValuesFrom(KCO:content-object))
  restriction(KCO:content someValuesFrom(DOLCE:particular))
  restriction(KCO:time-based-rendition
someValuesFrom(KCO:script))
  restriction(KCO:interaction-based-rendition
someValuesFrom(KCO:script))
  restriction(KCO:usage-context someValuesFrom(DnS:plan))
  restriction(KCO:user-task someValuesFrom(DnS:task))
  restriction(KCO:content-user someValuesFrom(unionOf(
    DnS:organization Collectives:collective DnS:agent-driven-
role)))
  restriction(KCO:contract-semantics
```

```

    someValuesFrom(CoreLegal:contract))
  restriction(KCO:negotiation-semantic
someValuesFrom(negotiation-protocol))
  restriction(KCO:pricing-semantic
someValuesFrom(KCO:pricing-scheme))
  restriction(KCO:trust-value           someValuesFrom(KCO:trust-
region))
  restriction(KCO:mappable-to
allValuesFrom(KCO:ForeignClass))

```

We mean that:

- KCO is a subclass of the class *physical data structure*. All the facets of a KCO data structure are modelled in OWL as “restrictions”:
- a KCO realizes a data structure that *instantiates* the data structure provided by a domain KCO (this represents the “self-description” facet)
- a KCO realizes a frame for one or more content objects (even past or future)
- a KCO propositionally represents (in this case, through OWL) the content of the content objects which it realizes a frame for
- a KCO provides scripts for the time-based, and interaction-based renditions of the content objects
- a KCO provides (eventually propositional) representations of the usage context (the *plan*) in which the content objects are supposed to be involved. Plans are axiomatized in the *plan ontology*, another extension of DOLCE and DnS [1][7]
- a KCO has at least one user task. Tasks are also defined in the plan ontology
- a KCO has at least one content user, that can be either organizations, roles played by agents (e.g. *author*), or collectives of any kind
- a KCO implements at least one (eventually propositional) contract semantics for the content objects: such semantics is representable within a *contract*, which is a kind of DnS:description
- a KCO implements at least one (eventually propositional) negotiation semantics for the content objects: such semantics is representable within a *negotiation protocol*, which is a kind of DnS:description
- a KCO implements at least one (eventually propositional) pricing semantics for the content objects: such semantics is representable within a *pricing scheme*, which is a kind of DnS:description
- a KCO provides a trust value for the content objects, here represented with reference to a *trust region* (an attribute), but in principle, it is possible to provide explicit (propositional) descriptions of *trustworthiness*, on which basis the trust regions can be parametrized
- a KCO can be mappable to one or more instances represented according to foreign classification schemes or ontologies. For example, given the CIDOC-CRM classification scheme [8], a KCO framing a digital edition of Dante’s Comedy can be mapped to an individual that has `rdf:type:CIDOC:E73.Information_Object`. Also parts of the content object, or its interpretations, references, etc. can be mapped using the same style.

All the properties that have been introduced have complementary axioms that allow to formally explicate their intended meaning on the basis of DDPO, and the IO extension. For example, the property `KCO:content` has the following complementary

axioms, which state that anything the information of a content object is *about*, or *expressed by* it, is a content for the KCO:

```
SubClassOf(
  restriction(KCO:realizes-frame-for someValuesFrom(
    restriction(DnS:realizes someValuesFrom(
      restriction(DnS:about someValuesFrom(DOLCE:particular))))))
  restriction(KCO:content someValuesFrom(DOLCE:particular)))

SubClassOf(
  restriction(KCO:realizes-frame-for someValuesFrom(
    restriction(DnS:realizes someValuesFrom(
      restriction(DnS:expresses
someValuesFrom(DOLCE:particular))))))
  restriction(KCO:content someValuesFrom(DOLCE:particular)))
```

Another example of axiomatic complementation of properties is given by the property *user task*, which is formalized as *equivalent* to having as usage context a *plan that defines (that) task*:

```
EquivalentClasses(
  restriction(KCO:user-task someValuesFrom(DnS:task))
  restriction(KCO:usage-context someValuesFrom(intersectionOf(
    DnS:Plan
    restriction(DnS:defines someValuesFrom(DnS:task))))))
```

The formalization detail of KCOs is justified by the intricate relationships holding between content, information, users, tasks, contexts, regulations, business requirements, etc. Being precise about these relationships helps the implementation of KCOs, the eventual interoperability with other knowledge management and content metadata systems, as well as paving the way towards ontology-driven management of content objects.

4 Architecture for a Distributed Content Infrastructure

In order to make use of the semantic richness that can be expressed with KCOs we need an infrastructure whose components support the functionality afforded by the KCO. The Knowledge Content Carrier Architecture (KCCA) does this in the shape of services which are logically clustered in the KCCA's components. Assuming a three-layer conceptual architecture with presentation / interaction, business logic and back-end data storage the KCCA specifies the middleware of the business layer. This gives rise to the following structural core components:

- KCO Service API** - offering the functions described by the facets in table 1
- KCCA Registry and Manager** - managing a federation of KCO-aware nodes
- KCTP Service** - a protocol to exchange service requests across KCCA nodes
- KCCA Profiles** - Services for the wrapping and integration of data sources

One of the assumptions of our work is that eventually, most information systems will make use of two further components: firstly, reasoning services based on ontologies and secondly, a task execution environment that will support the definition and execution of flexible workflows. KCOs are designed to support such an

architecture through their content description (this is where reasoning services can access the KCO) and through their community description (describing the tasks for which this KCO is useful and the roles of actors that would do the tasks). We envisage future publishing environments to use an integrated framework consisting of the components described. This will leave the application builder to focus on application and domain specific adaptations, and on the tailoring of the presentation /interaction layer to the needs of the customer.

The following architectural overview shows the full picture combining KCCA components, reasoning and task execution environment, as well as domain specific adaptations and the application layer.

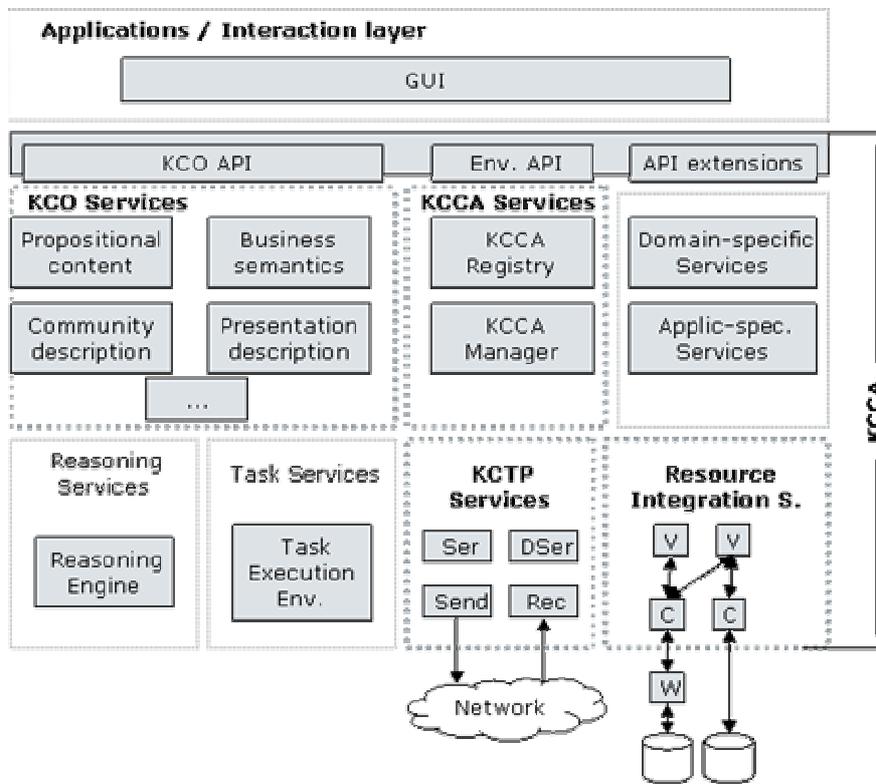


Fig. 3. Knowledge Content Carrier Architecture (KCCA)

The KCO services offer access to the operational semantics of the KCO facets. To achieve this, the structures that are defined in the *generic* KCO are mapped to an according O-O Model which is used for the implementation of the services. For *domain* KCOs, both the O-O model and its attendant services can be extended to cater for the application-specific functionality and semantics. It is assumed that the assignment of instance data (provided by KCO prototypes/instances) to types

(defined at the generic or domain level of the KCO) is done outside this building block (e.g. by reasoning based on the semantics defined at *generic KCO* and *domain KCO* Level).

The KCCA Registry and Manager component keeps track of how a federation of KCO aware information systems is set up. The KCCA environment keeps information about information sources, wrappers and maintains state in user sessions that may span requests and transactions across the federation.

The KCTP Services define a stateful protocol that allows communication between KCCA nodes by exchanging serialised RDF graphs. One specific service is the serialisation and de-serialisation (marshalling/unmarshalling) of KCOs in messages. The protocol is FIPA-based and can be implemented on top of SOAP or http.

The KCCA Integration Services give assistance in binding non-KCCA resources to a KCO aware system. This is done by a two-stage mapping process. The external information source is first mapped into an equivalent RDF schema which we call "context profile". This can be a "naive" mapping to RDF. Next, a view is defined over the context profile and this view is made KCO compliant. We call this the "view profile". The provider of an external information source needs to write a wrapper which provides the context profile for the resource. The KCCA integrator uses the context profile to create the view profile.

5 Summary and Open Issues

KCO are a flexible container structure for paid content that is enriched by dedicated semantic annotations grounded in foundational ontologies. They provide a solid basis for interoperable applications.

The foundational approach is used on one hand as a guideline for an efficient design of domain ontologies for content annotation, and provides on the other hand a minimal but shareable model for content interoperation between heterogeneous applications. The latter property will be leveraged by semantic search queries across KCCA infrastructures, i.e. such queries can be formulated on web objects classifiable as KCO level 2 (domain-specific KCOs) or KCO level 3 (instantiated KCOs). On level 2, partial instantiations can be added to a domain-specific KCO and used as requests.

The formal foundation of KCOs also secures that accessible KCO-compliant content repositories will deliver valid responses only. Because of the rich semantic structure of a KCO, requests can be defined on all content object aspects, i.e. descriptive content, reference community, business elements, and presentation issues.

The development of corresponding query languages and infrastructures that provide operational semantics is part of our current research within the EU project METOKIS (e.g. for KCO matching and composition, and interoperability with Semantic Web Services infrastructures). This will be used to integrate commercial content repositories in operational settings for three different domains: educational content production, clinical trial design and distributed news publishing.

References

1. Abecker, A., Apostolou, D., Maass, W., Mentzas, G., Reuschling, C. and Tabor, S., Towards an Information Ontology for Knowledge Asset Trading. in *ICE 2003 - 9th International Conference of Concurrent Enterprising*, (Espoo, Finland, 2003).
2. Bloom, P. A Decision Model for Prioritizing and Addressing Consumer Information Problems. *Journal of Public Policy & Marketing*, 8 (1). 161-180.
3. Boll, S. and Klas, W. ZYX - A Multimedia Document Model for Reuse and Adaptation. *IEEE Transactions on Knowledge and Data Engineering, DS-8 Special Issue*, 4.
4. Clarke, R. Electronic Commerce Definitions, 2000.
5. Franke, G., Huhmann, B. and Mothersbaugh, D. Information Content and Consumer Readership of Print Ads: A Comparison of Search and Experience Products. *Journal of the Academy of Marketing Science*, 32 (1). 20-31.
6. Gangemi, A., Borgo, S., Catenacci, C. and Lehmann, J. Task Taxonomies for Knowledge Content, METOKIS Deliverable, D07, 2004.
7. Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosz, B.N. and Dean, M. SWRL: A Semantic Web Rule Language Combining OWL and RuleML <http://www.w3.org/Submissions/SWRL/>, 2004.
8. Masolo, C., Borgo, S., A., G., Guarino, N. and Oltramari, A. The WonderWeb Library of Foundational Ontologies, 2003.
9. Reich, S., Behrendt, W., Eichinger, C. and M., M.S., Document Models for Navigating Digital Libraries. in *International Conference on Digital Libraries*, (Kyoto, 2000), 277-284.
10. Schellner, K., Westermann, U., Zillner, S. and W., K., CULTOS: Towards a World-Wide Digital Collection of Exchangeable Units of Multimedia Content for Intertext-tual Studies. in *Conference on Distributed Multimedia Systems (DMS 2003)*, (Miami, Florida, 2003).
11. Schmid, B.F. and Lindemann, A., Elements of a Reference Model for Electronic Markets. in *HICSS*, (Kohala Coast, Hawaii, 1998), IEEE Computer Society, 193-201.
12. Shapiro, C. and Varian, H.R. *Information rules - A Strategic Guide to the Network Economy*. Harvard Business School Press, 1999.
13. Stahl, F., F., S. and Maass, W. Paid Content - Paid Services: Analysis of the German Market and Success Factors of 280 Business Models, =mcmstitute, University of St. Gallen, St. Gallen, 2004, 163.
14. van Ossenbruggen, J., Geurts, J., Cornelissen, F., Rutledge, L. and Hardman, L., Towards Second and Third Generation Web-Based Multimedia. in *The Tenth International World Wide Web Conference*, (Hong Kong, 2001), 479-488.
15. Varian, H.R., Markets for Information Goods. in *Monetary Policy in a World of Knowledge-Based Growth, Quality Change, and Uncertain Measurement*, (2000).
16. Whinston, A.B., Stahl, D.O. and Choi, S.Y. *The Economics of Electronic Commerce*. Macmillan Technical Publishing, Indianapolis, 1997.
17. Zillner, S., Westermann, U. and Winiwarter, W. EMMA - A Query Algebra for Enhanced Multimedia Meta Objects. *CoopIS/DOA/ODBASE*, 2. 1030-1049.