KNOWLEDGE CONTENT OBJECTS AND A KNOWLEDGE CONTENT CARRIER *

W. MAASS

=mcminstitute, University of St. Gallen
Blumenbergplatz 9,
9000, St. Gallen, Switerland
E-mail:wolfgang.maass@unisg.ch

S. GOYAL AND W. BEHRENDT

Salzburg Research Forschungsgesellschaft mbH
Jakob HaringerStrasse 5/III
5020-Salzburg, Austria
E-mail: wernher.behrendt@ salzburgresearch.at

The exchange of semantically rich digital content over distributed applications facilitates knowledge exchanges in mixed human / machine environments as envisaged by scenarios within the scope of Ambient Intelligence and Semantic Web. For this class of domain, we introduce a semantically annotated data structure (KCO) and a distributed, layered protocol-based system architecture (KCCA, KCTP) that is designed for loosely coupled, semantically rich content applications. We relate this to requirements of commercial companies for sharing, exchanging and trading knowledge and content. Characteristics of the distributed system architecture are outlined. The argument is that by defining a reasonable set of semantics for manipulating the content objects in question, one offers a stage at which a market of providers and consumers can develop and is facilitated.

1. From actors and action spaces to e-sharing and trading of (knowledge) content

The background model to the EU project METOKIS is that actors using a digital environment have knowledge and obtain information from various sources, such as the WWW. The actors make use of their existing knowledge, weave into it new information and eventually, interpret their growing "knowledge space" with respect to their current or future "action space". Using the WWW, actors can interact with other actors irrespective of whether they are humans or software. The way to communicate (i.e. have controlled and controllable interaction) is by exchanging meaningful statements between the

* This work is funded by the European Union (6th FWP under the strategic objective IST-2002-2.3.1.7.) and the Swiss Federal Government.

actors. 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. We will introduce a "knowledge content object" (KCO) as such a surrogate.

Projecting this scenario of knowledge exchanges into a commercial context, firm's ability to create innovative products and service increasingly depends on its knowledge exchange capabilities within the organisation and beyond [8]. With increased innovation cycles and decreased product cycles, firms are forced to leverage internal and external knowledge in efficient and effective fashions [1, 5]. A firm is challenged to produce needed knowledge assets, to use it internally in multiple ways, to leverage them as products by knowledge markets for reasons of branding, customer relationship maintenance or direct sales and to distribute them over an increasing variety of channels and devices. Hence, knowledge becomes a scarce and valuable resource that calls for economic steering, control, protection and exploitation within the boundaries of an organisation but more importantly also between organisations and amongst individuals [2, 5]. Within this context, information technology primarily acts as a transfer medium that allows to store and to transmit knowledge assets engrained into data structures independent of time and space [7]. As a result, huge networks of heterogeneous information and data networks have been erected by organisations and individuals. Thereby an important driver for web presences and intranets of commercial firms has been to follow their competitors.

In general users require that information systems deliver dedicated information to application domains in a ready-to-use and ready-to-execute fashion. This means that information becomes "aware" of situations and user's needs which asks for sufficiently precise representations of (1) problem statements and situations, (2) application scopes of digitised information and (3) adaptation and inference mechanisms. Thereby the digital structure by which information is captured is viewed as a rather independent, active, self-describing, object-like entity. In the EU-project METOKIS , we design and test data structures that implement this kind of information objects, called KCOs, together with an integrated architecture for knowledge exchange (KCCA) in distributed environments with a broad variety of different applications. Throughout this article, we sketch the current proposal for the KCO structure and the corresponding architecture. In general, a KCO consists of (1) an envelope for content, such as documents, streaming video and pictures, (2) semantic annotations of different types and (3) associated contracts that describe the scope and procedure of applicability. KCO's semantic annotations are typed, i.e. respond to different aspects of a situation. We distinguish between semantic annotations for multimedia issues, usage contexts, business semantics, trust, access and domain semantics of the content itself. This encompasses initiatives such as proposed by MPEG-7 and Dublin Core and provides a homogenous framework for knowledge exchanges.

We consider semantic annotations on the basis of formal ontologies [3]. Ontologies are particularly used for (1) formal evaluation of system specifications [4], (2) semantic filtering and information mining on repositories [2] or (3) automated reasoning [6].

2. Knowledge Content Objects (KCO)

METOKIS can be regarded as an attempt to create an environment for the exchange of information objects that can be consumed by humans as well as machines. This way, these information objects become a (surrogate) means of communication. The analogy would be that the actors are writing a special sort of letters to each other. The inner structure of these letters helps the machines to separate out what is meant for them and what is for the humans to interpret. This special sort of "letter" - which can be exchanged between humans and humans, humans and machines, as well as machines and machines - we call a knowledge content object.

We believe that a KCO needs to carry the following semantics in order to be a useful unit of value (see table 1).

Table 1. Overview of KCO Structure

Propositional Content		A description of the content using a logic language (e.g. OWL)
Time based spatial rendition		schema for mapping media resources on to a time line
Interaction based spatial rendition		schema for mapping media items on to a discourse / dialogue pattern
Multimedia metadata	IPR information	Supporting Indecs or IPRonto
	Media properties	Supporting MPEG-7, MPEG-21
	Content classification scheme	Supporting e.g. Dublin Core or any other content cataloguing standard
Usage context	User task	Formal description of the tasks in which this content can be used (and how)
	User Community	Intended community of users, the roles that would use the content and the rights one would give to the roles. Not to be confused with the ACTUAL access rights that are defined elsewhere.
	Usage History	Aggregations of how the content has been used and manipulated in the past (trails, history).

Business semantics	License	License terms for using this KCO
	Contract	Contract specification
	Pricing	Information on the pricing scheme of the KCO
	Negotiation	Description of the negotiation protocol
	Trading	Description of the trading protocol
Trust		users can evaluate the trustworthiness of a resource if other users can leave (genuine and verifiable) endorsements. Any kind of quality feedback also allows improvements on the basis of evaluation by users.
Access semantics	User authorisation	who is allowed to do what with this KCO, under which circumstances?
	Processing policies	Which services/processes/agents are allowed to do what with this KCO?
KCO self description		Meta-level description of the KCO schema

We have begun to define operational semantics for KCOs at the level of generic operators that take the whole KCO and its first level structure as operands. These operators are:
(1) ***ADD*** *(kco, db*[c]*);*
(2) ***REMOVE*** *(kco, db);*
(3) ***UPDATE*** *(kco, component, db)*;
(4) ***QUERY****(kco, components, query-expr, db)*;
(5) ***MERGE*** *(kco1, kco2, db)*;
(6) ***CONVERT****(kco1, kco2, db)*

The ***ADD*** and ***REMOVE*** operators work on whole KCOs. The ***UPDATE*** operator changes the designated element of a KCO in the database. The ***QUERY*** operator acts similar to a database cursor and specifies which components of a KCO are to be accessed by the query expression. The ***MERGE*** operator takes two KCOs and fuses their components. The ***CONVERT*** operator takes one of the KCO elements (first level structure e.g. Business Semantics) and transforms the source element into a target element. The operation is defined primarily from the Propositional Description to all other elements and the converse, from all others to the Propositional Description. It means that e.g. a KCO trading information can become a KCO content which is itself tradable. Likewise, the propositional description of some trading information carried in one KCO can be converted into the actual trading information pertaining to some other KCO.
The objective of METOKIS is to arrive at a semantically annotated container structure for arbitrarily complex digital content which can then be used, shared

[c] db: database

or traded over a minimal content infrastructure: the Knowledge Content Carrier Architecture.

Table 2. Example: "query" and "add" operators acting on the business element of a KCO

Operator	Operands / Parameters	Return Value	Description
query	KCO-ID, "BIZ", "license" | "contract" | "pricing" | "negotiation" | "trading", QueryTerm | "all",	KCO-ID, BIZ-Graphs | NULL | FAIL	The query returns the business terms which are known for this knowledge content object, or NULL if no terms are know, or FAIL if the QueryTerm does not match the schema of the SubElement
add	KCO-ID, "BIZ", <SubElement>, Sub-graph	OK | FAIL	If the subgraph to be added does not match the BIZ SubElement ontology then the operation fails. This is to ensure that only consistent sub-graphs are added.

3. Knowledge Content Carrier Architecture (KCCA)

KCOs will be exchanged via a middleware platform for semantic content management systems. The methodology we have used is on the principle that there are wide varieties of systems from simple to complex - and therefore any semantic web middleware should be adaptable and be able to fulfil the middleware role in multiple scenarios. KCCA focuses itself more on interfaces and requirements, which can be implemented using multiple technologies. It consists of two key parts: KCCA Middleware and KCTP (Knowledge Content Transfer Protocol) Protocol. The KCCA Middleware provides the basic middleware components, enables exchange of KCOs and supports the execution of operations (tasks) on KCOs. The KCTP provides interaction and communication support between multiple KCTP enabled systems.

The KCCA fulfils the role of a semantic middleware, provides support for semantic definition of tasks and will also provide specific services (tasks) in the multimedia content management sector (e.g. ontology services, DRM services etc.). In METOKIS we will test the KCCA in three application domains: clinical trials, education and senior executives in the retail sector. The KCCA Middleware consists of the following key components:

- **KCCA Repository:** KCCA Repository provides interfaces with databases for storage of content, metadata, ontologies and KCO's (Knowledge Content Objects). The metadata within the KCCA middleware is stored in a

RDF DB with integration with relational databases using frameworks like D2RQ, SWIM etc.

- **KCCA Middleware Components:** KCCA Middleware Components provide specific components and modules that enable building up of the actual middleware e.g. authentication, workflow engine, session management, inference engine, rule layer and system registry.
- **KCCA Request Broker:** KCCA Request Broker provides support to plug in middleware components and also provides support for system and domain level services. The domain level services include services for all three application domains, services related to multimedia systems (digital rights management etc,), registry services etc. The system level services include services for accessing KCCA Repositories, KCCA Middleware components etc. It also includes KCO services which provide access, query and manipulation of KCOs.
- **KCTP:** KCTP (Knowledge Content Transfer Protocol) is a light-weight request/response protocol implemented by the KCCA Middleware that allows applications to exchange KCOs.

KCCA takes a protocol level view (KCTP) of defining an architecture where the protocol provides a standard way of sharing semantic information (see Figure 2). The KCTP Protocol consists of 2 layers the application layer and a session layer with reference to the OSI Network Reference Architecture. KCTP via using vocabularies enables partial understanding amongst semantic middleware systems. KCTP can be implemented over multiple transport layer protocols (e.g. HTTP, SOAP, Messaging etc.) with the requirement that the requests and responses of the KCTP must be serialized as RDF graphs.

KCTP enables partial understanding between KCTP enabled applications by defining KCTP profiles consisting of a vocabulary of functions (defined in RDF/S). KCTP Profiles are exchanged by KCTP request messages in the format of serialized RDF statements. The following profiles are defined:

- **KCTP-KCCA Profile:** defines the KCCA system characteristics such as protocol bindings, data encodings, repositories etc.
- **KCTP-RDF Data Profile:** provides vocabulary for querying and updating data at RDF data level.
- **KCTP-Service Profile:** This profile provides vocabulary support for service definition. It provides support for system services (described in OWL-S) such as data manipulation services, registry services etc.
- **KCTP-KCO Profile**: The KCTP-KCO Profile consists of the KCO ontology with its defined set of tasks. This enables sharing of KCOs between applications.
- **KCTP-Session Profile:** This profile enables state maintenance across sessions and provides state information.

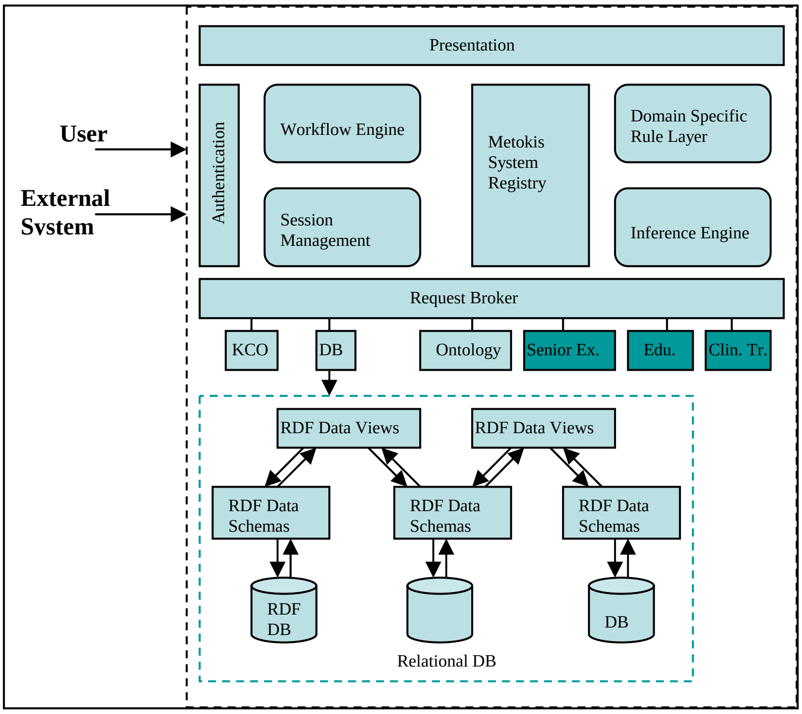

figure 1: Overview of KCCA Middleware

The METOKIS middleware architecture (KCCA + KCTP) provides an infrastructure for semantic web information systems which is extensible depending on the domain of application (e.g., intra or inter-organisation). Because the METOKIS architecture is intended for a broad field of domains, we will deploy it in three different flavours: KCCA-Lite, KCCA-Standard and KCCA-Full.

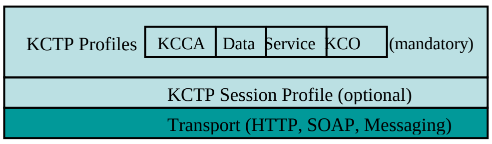

figure 2: KCTP - Knowledge Content Transfer Protocol layering in the OSI Layer

4. Conclusion and further work

We have presented the argument for an advanced distributed information architecture with mixed human/machine and machine/machine loops, in which increasingly more capable machines can act on "the same content" as humans, and can partake in a limited fashion at least, in complex knowledge sharing and e-trading transactions. We proposed that at the heart of such an infrastructure should lie, a kind of object with tangible qualities that humans as well as machines, can act on. At the time of writing, the specification of the METOKIS system has been done and a first implementation lies ahead of us. Project results are expected towards the end of 2005, including demonstrations of the software. Since the proof lies in the pudding, the application showcases will be given high research priority.

References

1. Alavi, M. and Leidner, D.E. *Knowledge management and knowledge management systems conceptual foundations and research issues*. INSEAD, 1999.
2. Apostolou, D., Mentzas, G., Abecker, A., Eickhoff, W.-C., Maass, W., Georgolios, P., Kafentzis, K. and Kyriakopoulou, S., Challenges and Directions in Knowledge Asset Trading. in *Proceedings of the Practical Aspects of Knowledge Management*, (2002), Springer.
3. Baader, F., Calvanese, D., McGuiness, D., Nardi, D. and Patel-Schneider, P. *The Description Logic Handbook: Theory, implementation, and applications*. Cambridge University Press, 2003.
4. Borgida, A. and Premkumar, D., Adding more DL to IDL: towards more knowledgeable component inter-operability. in *Proc. of the 21st International Conference on Software Engineering*, (1999), 378-387.
5. Davenport, T. and Prusak, L. *Working Knowledge: How Organisations Manage What They Know*. Harvard Business School Press, 1998.
6. Horrocks, I. and Patel-Schneider, P., Three theses of representation in the semantic web. in *Twelfth international conference on World Wide Web*, (Budapest, Hungary, 2003), ACM Press, 39-47.
7. Kanter, J. Knowledge mangement, practically speaking. *Information Systems Management*. 7 - 15.
8. Wolfe, R.A. Organizational innovation: review, critique and suggested research directions. *Journal of Management Studies*, *31*. 405-431.