Knowledge Content Objects and an attendant Knowledge Content Carrier Architecture - A Brief Introduction
Wernher Behrendt, Salzburg Research, Austria, 22/04/2005.

The Semantic Web is mid-term, potentially the most disruptive technology for traditional “content industries”. Its potential lies in the standardisation of meaning through the use of commonly agreed ontologies and this may lead to a situation where any content can be identified through “semantically enabled” querying, as long as it is ontologically marked up (we view this as synonymous with “having meta data”). Semantic Web research of course, is looking far beyond traditional meta data: once content is adequately described by knowledge structures, the question arises whether or not the description is semantically equivalent to the content and thus, the function of the “content” reduces to the “specific rendering” of the description for some purpose. For example, the descriptions of race tracks in computer games are getting so close to reality that it is difficult to distinguish renderings of a virtual lap on the computer, from real videos e.g. taken by an on-board camera (this is at least true for observers with just slightly impaired eye-sight). This means that an adequate semantic description of a football game, rendered by a computer game engine, may become a reasonable lower-price substitute for a live transmission of the actual game. Although some would argue that these two examples are not "semantic" according to the community’s own principles, they do reflect an issue: the realisation of meaningful situations and events with the help of symbolic representations - it would seem difficult to argue that this should not be at the heart of the semantic web enterprise.

METOKIS addresses a specific issue at the intersection of semantic web and (electronic) publishing technologies: how to combine the potential of semantic representations with (arbitrarily) bounded content assets that are at the heart of any content market, i.e. how can the semantic "knowledge soup" (c.f. Sowa, 2000) be partitioned into "knowledge and content commodities" that become tangible, tradable, and identifiable?

Our approach is motivated by thinking in terms of infrastructures which act as condensation points for meaningful services. There are some analogies with the real world, from which we draw: the notion of freight trucks makes sense when there is an infrastructure of roads and highways. The notion of passenger trains makes sense when there are rail tracks between major cities. The "containers" (trucks or passenger wagons, locomotives) have independent semantics from the stuff that they are carrying, but there are also dependencies which can be made explicit - for example:

Passenger trains have specific semantics (e.g. seats, windows) dependent on the kind of thing they are transporting (i.e. humans). They also have general semantics that are independent of the thing they are transporting - the tracks, locomotives, and couplings between wagons are the same no matter what is carried. All transport services are semantically agnostic about the purpose why some good is being transported from A to B, i.e. there is no dependency between the construction plan of a passenger wagon and the reason why I want to go from A to B.

We argue that the research communities of the World Wide Web and the Semantic Web in particular, need to think of content in a similar fashion. We need to factor out what is rail track, what is wagon, what is pay-load and to what extent there is scope or need for differentiating between types of pay load and types of containers.

Knowledge Content Objects (KCO)

KCOs have six semantic facets that help us define the infrastructure implemented in the Knowledge Content Carrier Architecture - these facets are:

- **Content description** - the ontology through which the content can be described. This model covers all questions of the type: "What is this content about?"
• **Community description** - the ontology through which meaningful use of the content can be described. Sub-facets are user tasks, user roles, and usage history. This model describes who is intended to use this content (roles/communities) and how (tasks). It also allows knowledge content instances to be annotated by what has already happened to the content ("I have been read by all students of course 101 in computer science").

• **Business Description** - the ontology through which contracts, pricing and negotiation about purchasing a KCO is described. This includes the ability to preview and enter into a business transaction with the owner of the KCO.

• **Presentation Description** - the ontology through which the rendering and possible interaction with a KCO is described. For example, static web sites have very simple navigation semantics whereas semantics-based Learning Objects may have complex capabilities for context-sensitive navigation (e.g. depending on the results of self-assessment exercises).

• **Trust and Security** - the ontology through which providers as well as consumers of KCOs can describe guarantees for the content they are selling or purchasing.

• **Self-Description** - the ontology that describes the internal structure and semantics of KCOs. At present, KCOs are based on the DOLCE foundational ontology and extensions to DOLCE, which were developed in the METOKIS project.

**Knowledge Content Carrier Architecture**

Given KCOs with such a defined structure, we can now develop an infrastructure that can inspect and manipulate the KCOs and thus, provide principled ways of interaction between external systems and any kind of (knowledge-enhanced) content.

The KCCA fulfills 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 KCOs (Knowledge Content Objects). The metadata within the KCCA middleware is stored in a RDF [RDF] database with integration with relational databases using frameworks like D2RQ [D2RQ], SWIM [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 Services Container (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 perform operation on KCOs.