

OKS Data Model, version 1.01

[OPAALS PROJECT](#) (Contract number IST-034824)

WP10: Sustainable Research Community Building in the Open Knowledge Space

Contribution to the Milestone M10.10: OKS Data Model (M24)

1 July 2008

Tampere University of Technology (TUT):

Ossi Nykänen, Jaakko Salonen, Jukka Huhtamäki & Matti Haapaniemi

This is a versioned document that may be superseded by other documents. Should this happen, appropriate reference information ought to be added here. Discussion and comments related to this document should be written to [a separate document](#).

1. Introduction

The aim of this article is to specify the Open Knowledge Space Data Model (OKS-DM). The OKS Data Model provides a system for capturing and distributing descriptions of various kinds of OKS resources. Intuitively, these resources might include both objects (e.g. publication, event, people, or organisation) and processes (e.g. project, object life cycle, interaction, or development). From a more technical point of view, the OKS Data Model enables implementing a *query access point* for the OKS, providing an extensible framework for implementing applications working with OKS data.

At the time of writing, the underlying technical *OKS infrastructure* is not yet completely specified. We may to some extent escape the resulting issues by acknowledging that, in the abstract sense, the OKS Data Model simply provides a basic OKS data service or semantic repository layer that interfaces between Peer-to-Peer (P2P) OKS node architecture and more advanced semantic layers. Such layers might be based, e.g., Web Ontology Language (OWL) or Semantics of Business Vocabulary and Business Rules (SBVR).

Note that rather than claiming that the OKS *is* a repository, we simply assert that it is useful to consider that a certain *aspect* of the OKS can be technically modelled as a repository. Intuitive design principles of our work include the given assumptions and dependencies, transparency, simplicity, existing tool support, and established applications.

Intuitively, the specification of the OKS Data Model includes the following components:

1. A simple, general-purpose, extensible syntax of semantic statements for asserting semantic information in the first place.
2. Basic set of "nouns", "verbs", "adjectives", etc. modelling primitives for a common vocabulary to start communicating.
3. A simple, general-purpose, extensible method for asserting conceptual structures in order to integrate information.
4. Visual and abstract means to use the OKS collaboratively so that things fit together, e.g., when new "nouns" and "verbs" are needed.

In this article, these informal ideas are explained in terms of the *OKS Modelling Framework*, the *Core OKS Vocabularies*, the *OKS Upper Concept Model*, and the *Common OKS Process*, in the spirit of Semantic Web technologies [SW].

Since we are talking about a data model in the context of a large heterogeneous system, we perceive our work as a seed not only for the development of the OKS infrastructure and applications, but also for a community process. In particular, this means that the OKS vocabularies are *per se* defined and refined in detail only when the OKS is actually used, based on real needs.

The rest of this article is organised as follows. Section 2 defines the background and the basic scope of the work. Section 3 outlines the underlying dependencies and assumptions at the time of writing. Section 4 provides the abstract specification and recommendations for the abstract OKS Data Model. Section 5 illustrates the abstract OKS Data Model in terms of a simple example and Section 6 finally concludes the article.

The OKS Data Model is developed and published under the [Creative Commons Attribution-Share Alike 3.0 Unported license](#).

Note #1: Advanced audience solely interested in the OKS Data Model might wish to proceed directly to Sections 4 and 5; the preceding sections mainly put the definitions in context.

Note #2: The OKS Data Model specified in this article may be considered to provide requirements for the design of the underlying technical OKS infrastructure. Besides the overall idea of an information repository, few explicit requirements are presented in Subsection 4.5. (included into the abstract recommendations about the Common OKS Process).

2. Purpose and Scope

2.1. Background

Spreading OPAALS's excellence requires activities for communication and dissemination. As the OPAALS processes mature, the main depicted instrument for managing and sharing knowledge is supposed to be the Open Knowledge Space (OKS). The two major stakeholders of the OKS include SMEs and academic institutions.

While the OKS has not yet been completely specified as an information system, its properties have been extensively discussed [OPA06b, OPA07d, OPA07e, OPA07f, OPA07g]. In brief, the OKS is intuitively envisaged as a distributed, open, intuitive, perceivable and collaborative community platform for professionals of any domain. Further, the OKS should also support creation and management of knowledge about identities, activities and document- and content-based information but also as subjective, personal opinions that tie together public content with personal opinions and annotations.

Several more precise usage models have also been described, in saying what the OKS is, including OKS [OPA07g]:

- as a means of collaboration on documents and deliverables,
- as a way to collaboratively define a research topic or item and arrive at a rich and consensual outcome,
- as a way to brainstorm and gather multiple views on a given topic,
- as a way to animate and manage the communities,
- as an ongoing recursive process of identifying, establishing and reinforcing professional/behavioural online ethics, and
- as a navigable, multi-dimensional living map (structured as a multi-layer ontological space and implemented as a distributed Google Earth-like 3D virtual space) of the relationships between the research concepts and activities, the people, and the institutions characterising the OPAALS community.

It is worth recognising that the OKS is not simply a software system; it includes also people, processes, and (references to) informal and tacit knowledge.

Of course, the OKS Data Model can not as such provide the above depicted high-level applications. Instead, it should enable their implementation, with readily available software tools, whenever possible. The working hypothesis is to start from a *repository* kind of application, that can be extended to suit the richer needs of the future OKS. In particular, in this specification we consider processes as second-order objects that are captured either by object representations, or query signatures.

2.2. Informative Example

In order to illustrate the role of the OKS Data Model, let us consider a simple example scenario of using the OKS.

Suppose organising a conference. A conference is an event in which researchers present their work-in-progress and results, perhaps as a series of presentations of pre-prepared articles. Now, given a list of attendees and their topics, we would like to search the OKS for associations between the people and their research interests, perhaps in order to visualise the implicit associations of people and articles in the conference.

One way to compile the data set for this application is to query the OKS for the peoples' professional information and their research interests. If the research interests of two researchers match directly (e.g. both are interested in information visualisation), we may use this information to conclude that the researchers can be associated with a joint research interest.

However, sometimes research interests match only indirectly. For instance, one researcher might be interested in knowledge visualisation and another research in information visualisation. In this case, an association can be established with respect to an (at least locally, for purposes of the conference) agreed concept schema (or a thesaurus, ontology, etc.).

Still, a strict consensus, e.g. global taxonomies, might not exist. For instance, different researchers might classify visualisation concepts differently (see two examples below).

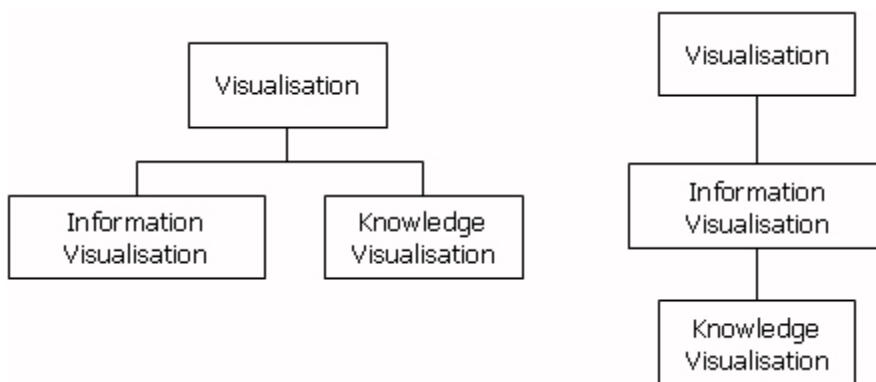


Figure 1. Two ways to model a concept structure

This means that while a particular application needs to decide its modelling stance specifically, the OKS should enable the capturing of various kinds of semantic data and concept systems, using which several kinds of *contextual interpretations* can be performed. In a more complex situation, the challenge of information and knowledge integration includes using synonyms, homonyms, etc. that may require active processing of the information sources. Note that this discussion applies to all recorded data, not simply concept structures. (A concrete example of a concept structure is presented later, in Section 5 -- see Figure 7)

Of course, the conference event itself might be captured by the OKS. In the long term it is indeed desirable to systematically record the activities around several conferences. This would, e.g., enable searching not only individual articles but also tracking the *process* (or trends) of the development of certain research topics.

From the perspective of the computerised OKS Data Model, formal identifiers and structures are needed. Thus, when talking about researchers, articles, research topics, and their descriptions, a precise way of referencing them is required.

Ideally, the OKS Data Model would be indifferent about the underlying OKS infrastructure. In practice, however, the network architecture may have implications. In particular, we have so far talked about researchers and articles, simply assuming that they can be meaningfully referenced in the OKS. We shall next briefly discuss the origin of these names (as formal identifiers).

In brief, we assert that names are assigned with respect to the underlying OKS infrastructure. When a user wishes to exploit OKS services, e.g., in order to search for a resource, he or she connects to the Internet, somehow accesses and registers into the OPAALS OKS network, and requests services using an appropriate agent (see the diagram below).

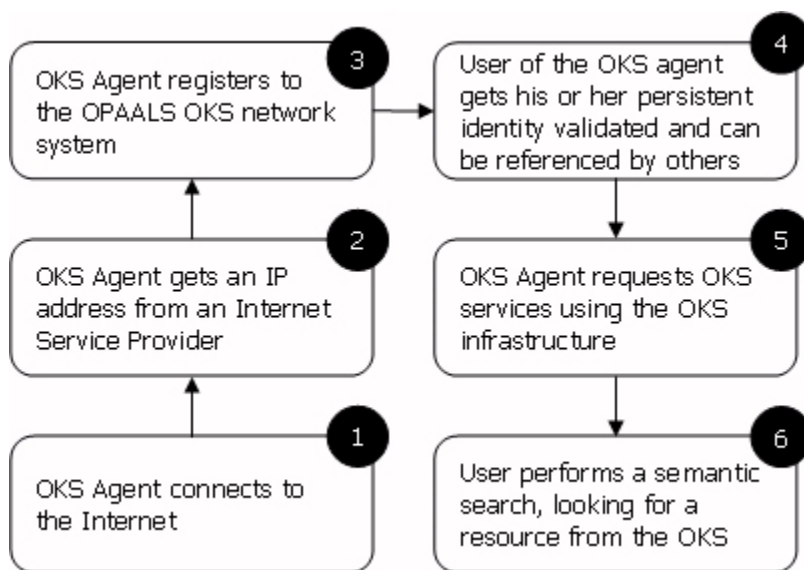


Figure 2. The process of accessing OKS services

In other words, even if technically distributed, replicated, etc. by the OKS infrastructure, e.g., persons and the articles owned by them can be identified with respect to their persistent OKS identity. (In the most straightforward setting, Uniform Resource Identifiers are used; we shall return to this topic shortly.) Note that persistent identifiers *are* needed at some level; otherwise searching would

not be possible at all. The issues of persistent names and Peer-to-Peer architecture are discussed later in the subsection 3.1.

2.3. Abstract Use Cases

We shall next outline simple use cases for the OKS that should illustrate the purpose and scope of the specification of the OKS Data Model in a more systematic way.

In brief, the OKS should be searchable both by navigation (associative/contextual data) and queries (database-oriented data). Typical examples for searchable objects include publications, timetables, events, persons, and tools. As such, it should be possible to perform searches based on the following approaches:

- Search based on metadata descriptions
- Free text search (e.g. matching keywords to the text body, perhaps using regular expressions)
- Search by example/similarity (e.g. vector distance of the OKS term histograms)
- Search based on context descriptions (e.g. references to an object)
- Search based on reviews (special case of the above, e.g. based on data from forum or blogs)

These objectives highlight the need for supporting searching in all data. However, we simplify the OKS Data Model specification by assuming that computation-intensive searches, such as the search by similarity, are in fact implemented on top of the more primitive search services.

For future reference, we identify the six **abstract use cases of the OKS as an information repository**, with respect to the OKS Data Model:

1. Publish a resource (e.g. a document, a service, or contact information)
2. Search resources by descriptions ("metadata"; e.g., for searching people this is the only option)
3. Search resources by content (usually by document content)
4. Retrieve a resource (that can be retrieved via the OKS, e.g. a document)
5. Modify a resource (that can be modified via the OKS, descriptions or content)
6. Remove a resource (that can be removed via the OKS, descriptions or content)

Publishing a resource involves giving an object searchable identity, granting access rights to others, and perhaps informing others about the newly published resource.

Searching resources by descriptions involves querying the OKS Data Model with respect to the available descriptions.

Searching resources by content involves using some content-specific query mechanism that is needed for seeing inside the content resources. Note: In a typical scenario, the search space might first be restricted with a search by descriptions.

Retrieving a content resource involves requesting (a representation of) a content resource based on its identifier or description. Note: Of course, content resources may be retrieved only when suitable digital representations are available.

Modifying a resource involves changing either a description resource or a content resource. Note: In some systems, the underlying implementations might not enable modifying a resource, only instantiating a new version of it. (In general, OKS version control is a significant challenge that is out of scope for this document.)

Removing a resource involves deleting either a description resource or a content resource. In general, referential integrity might not be assumed. This highlights the need for carefully planning the process of publishing resources.

It is important to observe that, in a concrete setting, searching and retrieving information, and modifying and deleting information, might be considered fundamentally different as repository operations. For instance, while searching descriptions can be to some extent harmonised, it seems unrealistic to assume that the editing interface of the different base systems could be harmonised in a similar fashion. For instance, while deleting information from a personal database is usually a primitive operation, a sophisticated content-management system might in principle aim only at versioning (and archiving) information rather than permanently erasing it.

3. Assumptions and Dependencies

3.1. P2P Networks

The OPAALS consortium envisages the OKS infrastructure as a P2P network (see, e.g., [OPA07a, OPA07b]). This provides not only functionality but also special requirements. Below is an attempt to briefly enumerate some of the properties of the underlying P2P infrastructure that have consequences with respect to the OKS Data Model:

1. **P1: Fully distributed environment.** In particular, there must be no opportunity for one organisation, or even a small number of organisations, to be in a position to benefit from, or (intentionally or otherwise) create

risks to, the routine business and knowledge/information or research activities of other members of the network.

2. **P2:** *Fundamentally a scale-free network, or a scale-free network of small-world networks.*
3. **P3:** *Each node has a unique address and identity.*
4. **P4:** *Each transaction has a unique identity.*
5. **P5:** *Consistent replication of data.*
6. **P6:** *A local node has full control over its own data and services.*
7. **P7:** *Primary structure must fully support development of a secure transaction model.*
8. **P8:** *Ability for new members to register with and join the network.*
9. **P9:** *Ability to specify the diameter of the P2P network according to the lookup algorithm.*

Among other things, these assertions seem to yield the following implications, with respect to the OKS Data Model:

1. Each node has a unique name. It is appropriate to use URI names for representing resources.
2. Definition of a node resource repository/DB must be based on some primitive interface of the underlying OKS infrastructure; otherwise the "primary structure" does not fully support "secure transaction model" or simply does not use one.
3. Everything represented/stored according to the OKS Data Model needs to manage the process being copied, while preserving unique identity. We might interpret this so that it implies a sort of versioning: whenever something is copied, it gets a new identity, storing a historical copy of the old metadata descriptions.
4. The OKS infrastructure should be able to manage several user accounts, i.e. several end-users accessing the network from the same OKS Node (access rights, personal workspace). This is simply a pragmatic assumption: New users first register to some bootstrapping node (perhaps hosted by their organisation), get familiar with the OKS network and eventually spawn into genuine OKS nodes, when appropriate.
5. Introduction of shared vocabularies, taxonomies, ontologies, is potentially problematic. If the policy of full distribution is strictly followed, definition of, e.g., a useful vocabulary is known only tacitly (i.e. outside the P2P network).

While speaking about P2P services, it should be acknowledged that OKS nodes typically offer services of various abstraction levels. In particular, some P2P services might be considered primitives, based which more complex services are defined (see Figure 3 below). For instance, several business services might benefit from a set of distributed computing and data repository services.

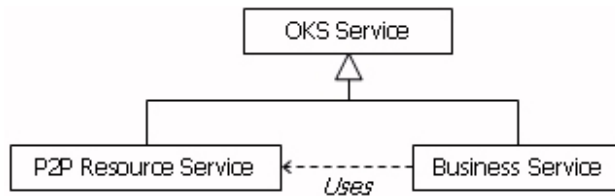


Figure 3. Conceptual view to OKS Services

While it is true that the P2P service infrastructure (e.g. resource sharing and identity management) must not be vulnerable to single-point-of-failures (etc.), *some* (business) services are provided only by single nodes. The P2P architecture itself, however, has a common status which must be acknowledged by every OKS node wishing to benefit from the related services.

This line of thinking easily extends into the realm of data modelling and to some extent allows escaping the issue raised by the implication 5 above (see Figure 4 below).

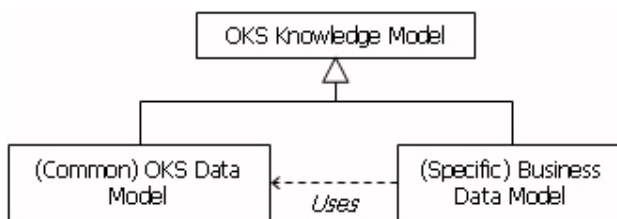


Figure 4. Conceptual view to OKS Knowledge Models

While it is true that using a single data model, ontology, etc. should not be assumed, *some* information structures should be given a common status, otherwise communication becomes impossible. Thus, if knowledge sharing use cases are sought after, the OKS Data Model, as information architecture, has a common status which must be acknowledged by every P2P node wishing to benefit from the related services.

3.2. Web Technologies

In addition to the P2P OKS Infrastructure, transparency of the OKS Data Model is important with respect to contemporary Web technologies and content. This is significant since several networks coexist and some people will continue using the existing (P2P, decentralised, client-server) applications for some time.

In short, the *World Wide Web* (WWW, or simply Web) is an information space in which the items of interest, referred to as resources, are identified by Uniform Resource Identifiers (URI). (Note that URIs include both Uniform Resource Locators such as <http://www.opaals.org>, and Uniform Resource Names such as urn:isbn:0-486-27557-4 .) Any web agent, either people or software, may act

on this information space. User agents act on behalf of a user. Web browser is perhaps the most obvious user agent, but other software agents such as servers, proxies, spiders or multimedia players may act on behalf of user as well. [W3C04a].

The architecture of the World Wide Web introduces three architectural bases of the Web infrastructure: identification, interaction, and format. In short, a Uniform Resource Identifier (URI) identifies a resource. Based on this identifier, representations of *some* resources may be retrieved. Finally, well-established formats enable the parsing and interpretation of representations. The Web architecture is often considered to follow the rational of Representational State Transfer (REST) architectural style for distributed hypermedia systems [REST].

From the perspective of the OKS Data Model, Web technologies provide a well-established technology for capturing knowledge: the Semantic Web. In brief, the Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF). [SW]

From the adoption and tools point of view, it seems natural to reuse the existing knowledge representation specifications and practices. Therefore, the OKS Data Model should follow the design of Web technologies and Semantic Web whenever possible. Among other things, these assertions seem to yield the following implications, with respect to the OKS Data Model:

1. Adopting the REST architectural style.
2. Adopting the RDF Data Model [RDF] as the main modelling principle.

A consensus about this line of thinking was also reached during the [Social Science / Computer Science Meeting in Guildford and London, 6-9 May 2008](#), when the exact nature of the OKS infrastructure was extensively debated.

3.3. Abstract OKS Rerefence Architecture

Because of the certain dependency on the underlying technical infrastructure the specification of the OKS Data Model benefits from briefly introducing our understanding about the *Abstract OKS Reference Architecture*.

A diagrammatic representation of the OKS Data Model and the related OKS concepts is depicted below.

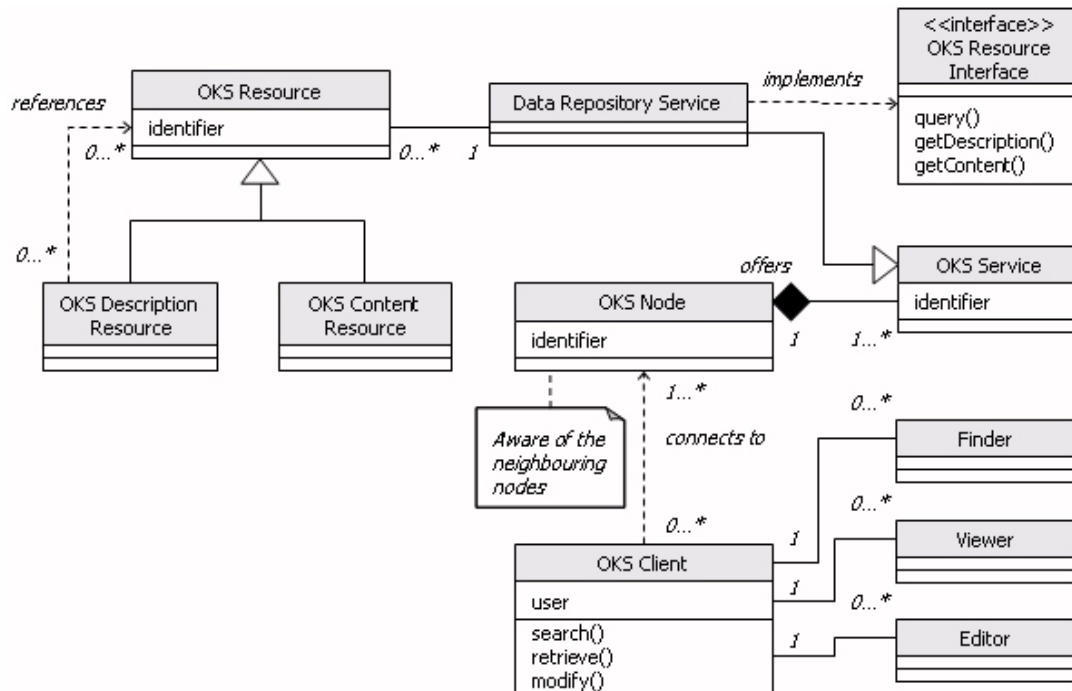


Figure 5. Abstract OKS Reference Architecture (conceptual perspective)

In other words, OKS Nodes offer OKS Services and a Data Repository Service is simply a kind of OKS Service. OKS Nodes typically offer also other kinds of services, but from the perspective of the abstract data model, these are left unspecified (and not highlighted in the diagram). End-users access the Services of OKS Nodes by using different kinds of OKS Clients. From the perspective of the Data Repository Service, users search, retrieve and modify resources using appropriate Finder, Viewer, and Editor Applications. In a P2P setting, besides the specific resources of their own, the OKS Nodes may also provide access to the services of the P2P network as a single entity. These P2P services might include lookup and semantic search services, and a service for requesting unique names from the P2P network (we return to this topic in Subsection 4.5).

The important top-level data repository concepts include the following:

- **OKS resource.** The entity in which data is stored and delivered and which appears logically the smallest data entity within the OKS P2P network. OKS resources can be either Description Resources or Content Resources.
- **OKS Description Resource.** Description resources are mainly used for stating the properties of other (description or content) resources. The common nature of the OKS mainly builds upon the agreements about the common descriptonal properties of OKS resources. The OKS Data Model may be used for describing the properties of entities without digital representations. This means that some OKS description resources may

- thus refer to virtual resources (such as persons, physical objects or concepts) that are (formally) identified solely in terms of given identifiers.
- **OKS Content Resource.** Content resources are resources that have digital representations such as image, video, or document. Depending upon the content type, content resources may also include (metadata) descriptions (such as content size, type and format), availability of which (for, e.g., purposes of the OKS queries) depends upon application.
 - **Data Repository Service.** Access to OKS resources is provided by OKS Data Repository Service that runs locally on an OKS Node, along other node services. By utilising the OKS Node network, data repository services can communicate with each other, providing search, retrieval, modification and other P2P data access services.
 - **OKS Resource Interface.** Data Repository Service provides data access via a well-defined, OKS Resource Interface.

Note that the identifiers and user names are decided "outside" the OKS Data Model (see Figure 2). For instance, identifiers might be based on the users' identities in the OKS network system.

In a simple setting, the OKS Data repository service could simply be implemented in terms of decentralised services (over, e.g., HTTP protocol). However, from the perspective of P2P networks, some descriptions should most likely to be distributed and be transparently available from the network rather than individual nodes. This would allow, e.g., searching for information without having *always* to know *a priori* the Node from which to ask for resources or services (see Figure 3).

As a consequence, the repository services provided by the OKS nodes might transparently include P2P services. For instance, when requesting a particular resource or descriptions of a resource upon identifier(s) or some query criteria, the node the user is accessing might include results that were in fact seamlessly retrieved from its "P2P node neighbourhood" (the exact nature of which is beyond the scope of our treatment). This would thus allow (logical) descriptive data to be (physically) distributed along content and description resources, perhaps in a number of nodes.

This discussion characterises what we mean by a **query access point for the OKS**:

- When a data repository service is implemented in terms of REST, a query access point might simply be a way to query and retrieve resources by semantic descriptions and URI references according to the OKS Data Model.
- When a data repository service is implemented in terms of different kinds of services, a query access point might be implemented via a common

application programming interface, for messaging and remote procedure calls, again following the conventions of the OKS Data Model.

- When applicable, the role of the P2P services is handled in terms of query parameters, i.e. parameters that enable the definition of a search horizon.

Please note that these perspectives are not necessarily exclusive. In particular, providing a REST-compliant interface to the repository service seems favourable since it enables to some extent accessing information with general-purpose Web agents.

4. Abstract Specification of the OKS Data Model

4.1. OKS Modelling Framework

Based on the discussion above, we are now ready to assert the fundamental modelling framework for the OKS Data Model. As suspected, we recommend using the RDF Data Model as a basis of Description Resources [RDF-DM].

Among other things, this implies the following design principles of the OKS modelling framework:

1. Things that are referenced are called *resources* and may be identified with *URI references*.
2. Resources may be referenced using multiple (different) URI references. However, each URI reference identifies exactly one thing (this thing might be, e.g., a bag-like structure, though).
3. Semantic information is asserted as *statements*. Each statement includes three parts: *subject* ("the thing being described"), *predicate* ("the property the description talks about"), and *object* ("the value of the property for the thing being described").
4. Besides URI references, subject and object things in statements can be locally referenced *anonymously* (with local, temporal identifiers). Further, *literal values* (strings with optional *datatype* and *language identifiers*) may be used as objects in statements.
5. As such, the modelling framework indicates that "Anyone Can Make Statements About Any Resource".
6. While arbitrary statements are possible, the modelling framework is insensitive to the multiplicity of identical statements.
7. Semantic associations occur for direct and reasoned matches of identifiers (this is the "Semantic Web"). In particular, reusing the identifiers of commonly used predicates is encouraged.
8. The logical interpretation of the bare RDF data model is rather minimal [RDF-SEMANTICS]. In particular, a set of statements (also called an *RDF graph*) entails each of its subsets of (ground) statements.

9. When necessary, richer logical entailments may be introduced by adding new semantic layers of descriptions (e.g. RDF Schema or Web Ontology Language [RDF-VOC, OWL]).

There are multiple ways to serialise statements. The family of RDF specifications encourages using RDF/XML or Turtle serialisation syntaxes [RDF/XML, TURTLE]. This allows transmitting RDF graphs in terms of text documents.

As a formal language, RDF makes a distinction between two aspects of the language: *syntax* and *vocabulary*. For purposes of a well-established modelling language, the family of RDF specifications introduces a set of vocabularies that are often considered to be a part of the common RDF language [RDF/XML, RDF-VOC]:

- RDF Vocabulary
- RDF Reification Vocabulary
- RDF Container Vocabulary
- RDF Collection Vocabulary
- RDFS Vocabulary

In particular, the RDF Schema (RDFS) Vocabulary [RDF-VOC] is used for asserting properties of RDF vocabularies. The basic idea is that RDF Schema definitions allow additional entailments. For instance, if the domain of a property is specified in a schema, we can entail (with respect to that schema) a type of the object in the related statements.

The above "built-in" vocabularies are used, e.g., to capture the two principal modelling concepts of RDF: *classes* and *properties* (with *subclasses* and *subproperties*, etc.). However, without additional semantic agreements, things may have several types (belong to several classes), things may be both classes and properties, and so on. Of course, practical applications do not usually take full advantage of the modelling freedom of RDF, e.g., restricting to a suitable subset of the model. Further, not every RDF application aims faithfully to follow the formal RDF semantics. For instance, a trivial use case of the RDF is simply syntactically communicating in terms of statements, without actually computing the semantic closures of RDF graphs, etc.

4.2. Relationship between the OKS Data Model and the OKS

At this point it is helpful to consider the relationship between the OKS Data Model and the OKS (see Figure 6 below).

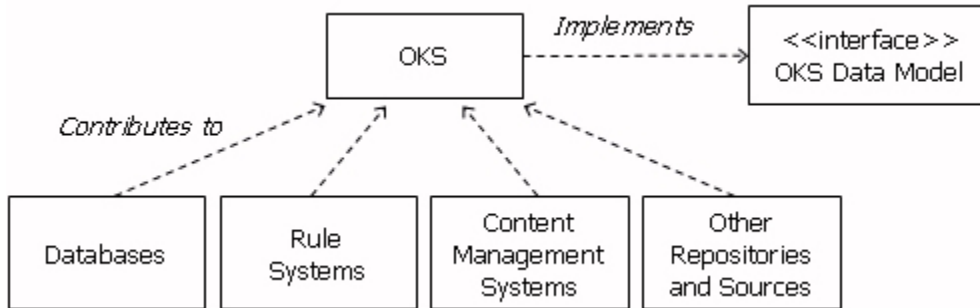


Figure 6. OKS Data Model as an information interface to a contributed knowledge system

In brief, we perceive the OKS as a system for integrating knowledge contributions from several information systems. Thus, rather than forcing information to be captured, e.g., in RDF *per se*, we consider RDF as a standard interface to descriptions. (Note that this conceptualisation is quite similar to Semantic Web.)

Note that besides the common modelling framework, there might not exist a "correct" big picture of the OKS. Rather, the OKS should be perceived as a contributed system, the meaningful interpretation of which is based on *contextual processing*. In particular, OKS agents might accept only certain subsets of the OKS at any one time.

This line of thinking seems to suit the needs of the underlying OKS infrastructure sufficiently well. In particular, the internal representation, distribution, persistence, etc. of the contributing information sources is left unspecified. (Technically, the desired functionality might be accessible using, e.g., *adapters* tailored for each subsystem.)

4.3. Core OKS Vocabularies

After agreeing upon the modelling framework, our next task is recommending a minimal set of properties, used throughout the OKS for describing things. We adopt the strategy of the Semantic Web and recommend using established vocabularies of the Web community as much as possible.

In brief, when describing a resource, we recommend using the following vocabularies:

1. The vocabularies of the modelling framework for general-purpose usage (i.e. the vocabularies from [RDF/XML, RDF-VOC]).
2. Dublin Core (DC) vocabulary for describing documents [DC-RDF].
3. Friend of a Friend (FOAF) vocabulary for describing people [FOAF].
4. RDF Site Summary (RSS) 1.0 for content syndication [RSS10].
5. GeoRSS for describing geographical locations [GEO-RSS].

6. SKOS Simple Knowledge Organization System for describing simple concept systems [SKOS].
7. *More specifications to be added, with refinements, according to the community process.*

In other words, further suitable semantic vocabularies may be introduced on demand. (Please note that an easy way of browsing some the abstract definitions is to use some appropriate schema repository, such as the [SchemaWeb](#) [S-WEB].)

Finally, it is important to recognise that a *controlled vocabulary* has two equally important aspects: a *well-established set of property names* (i.e. predicates), and a *well-established set of property values* (i.e. objects). In the case of RDF serialisations, the former means identifiers of the predicates in statements (e.g., dc:subject). The latter means identifiers of objects in statements (e.g. oks:visualisation). An important special case of well-established lexical space of values is adopting a well-defined datatype, e.g., a standard format for expressing date as Unicode strings (xsd:date, e.g., "2008-01-23").

In particular, adopting, e.g., Dublin Core, provides well-established names for common predicates but does not as such establish a thesaurus of, e.g., OKS concepts. This means that additional agreements are required.

4.4. OKS Upper Concept Model

To support the use cases of search, navigation, and semantic association, it is useful to index individual OKS resources with respect to a common knowledge organisation system. For this task, we propose using a simple collaboratively knowledge model called the *OKS Upper Concept Model* (OKS-UCM). The main virtue of the OKS-UCM is that it introduces the formally captured OKS concepts and their main associations to other concepts.

The backbone of the OKS-UCM is a simple, extensible *thesaurus* (for lack of a better word) that allows the semantic indexing of OKS resources with respect to the important OKS concepts. By a thesaurus we mean the following kind of structure:

- It includes well-defined *terms* that are used to capture *concepts*.
- It defines *term relationships* between terms (usually hierarchical, equivalency, or associative).
- It includes definitions of concepts (or references to definitions).
- It provides an extension mechanism for asserting additional information about terms.

Please note that while hierarchical structures are significant for, e.g., searching, it is not assumed that the "hierarchical" structure of the OKS-UCM is a tree, nor

assumed that relationships are globally "validated". Rather, the OKS-UCM will most likely establish a complex network (e.g. including multiple hierarchies) that is processed upon contextual interpretation. Further, the OKS builds upon the ideology of P2P networks, the OKS-UCM must be specified so that it can be distributed both by the physical and the logical design.

To meet these requirements, the definition of the OKS-UCM is twofold. The first part establishes the common syntax of the framework of defining concepts (identifiers, labels, and relationships) and associating them with different kinds of OKS resources. Considering the definition of the OKS Data Model, we simply recommend using SKOS:

- SKOS Simple Knowledge Organization System [SKOS].

This means that formally, the backbone of the OKS-UCM is technically captured as a SKOS concept scheme. Having said this, it should be acknowledged that SKOS is simply an important special case of a core OKS vocabulary (see the previous subsection). In practice, extensions to the core structure expressed in SKOS might include, e.g., statements using OWL ontologies [OWL]. However, the formal consequences should be carefully examined.

The second part of OKS-UCM definition includes a strategy for exploiting the labels and terms of the existing concept structures, taxonomies, standard ontologies, etc. In brief, when introducing new or referencing existing concepts, we recommend using the terms introduced in the following services/specifications:

1. [SchemaWeb](#) [S-WEB]
2. [The DAML Ontology Library](#) [DAML].
3. [Suggested Upper Merged Ontology \(SUMO\)](#) [SUMO].
4. [SWOOGLE](#) [SWOOGLE]
5. *More specifications to be added, with refinements, according to the community process.*

Note that the above specifications in fact may include several concept systems, using which in a single application may require additional, contextual clarifications. Further, while it is crucial that terms can be formally referenced, aiming for completely formal OKS theories or ontologies might not be always required. In particular, as a contributed system, the OKS might not follow any particular logical framework, besides the modelling framework of the RDF.

Taken together, the terms representing the concepts of the above systems establish the *OKS-UCM vocabulary*. Further suitable semantic vocabularies may be introduced on demand. Reusing existing concepts is of course highly recommended.

Finally, it should be emphasised that due to the P2P nature of the OKS, it is unrealistic to assume that the OKS-UCM would actually globally yield a neat and balanced thesaurus as a tree (for the main concepts, this might be desirable, though). Rather, just like the OKS, also the OKS-UCM should be perceived as a contributed system, meaningful interpretation of which is based on contextual processing (see Figure 6).

4.5. Common OKS Process

The final part of OKS Data Model Definition establishes the community best practice, called the *Common OKS Process*. From the perspective of the OKS Data Model, the Common OKS Process describes how the OKS is used and developed as an information repository by end-users and designers. It is needed for several reasons. In particular, it:

1. Explains how new identifiers are introduced for resources.
2. Suggests means to find out definitions and examples of the currently used OKS vocabularies (both OKS Core Vocabularies and the OKS-UCM Vocabulary), for purposes of reuse and semantic integration.
3. Recommends a best practice for introducing new vocabularies.
4. Suggests functional requirements for the underlying technical OKS infrastructure.

In particular, the process definition must deal with the difficult issues of polysemy, homonymy, synonymy, vagueness, missing concepts, contradictions, versioning and deprecated concepts, errors, access problems, etc. Of course, while these issues can be raised here, from the end users' perspective, tool support and documentation from the underlying technical infrastructure is essential.

Due to its P2P design stance, it should be emphasised that neither the Core OKS Vocabularies nor the OKS Upper Concept Model can not be fully designed as a top-down system with, e.g., flawless referential integrity. For instance, OKS Nodes may disagree about property and concept definitions, or simply define and use new properties and concepts without coordination (see Figure 1). However, it is nevertheless clearly useful to *aim* for an integrated OKS Upper Concept Definition. Therefore, the OKS-UCM Process ought to be actively supported. This might, for instance, include tools for querying, visualising, and locally caching the known OKS-UCM Vocabulary in order to support concept reuse and disciplined definition of new concepts (in particular, with carefully selected relationships to the existing concepts).

We conclude this section with a list of **abstract recommendations about the Common OKS Process** (requirements for the underlying OKS infrastructure are written in cursive):

1. Users and organisations identify resources they possess or produce, using the URI references that they can claim ownership of.
2. In the case when a legal entity (e.g. a person) cannot or does not wish to claim ownership of a specific URI reference, e.g. because it is not associated with an authorised and a trusted identity server, *it can request a previously unassigned reference* <http://www.opaals.org/resources/x> from the OKS infrastructure to reference *itself*, where x is a valid path segment according to [RFC3986]. Once approved, also the subhierarchies of the reference (e.g. <http://www.opaals.org/resources/x/y>) are owned by the legal entity and may be used to reference anything. Note that this mechanism is for naming only, it should be considered a fundamental service of the OKS, and it does not necessarily provide a retrieval mechanism. (The OKS Data Model does not specify the name requesting mechanism but it seems plausible that e.g. user information could be effectively used as a seed for suggesting good references. The structure of the name token x may be specified in more detail elsewhere)
3. When describing important OKS resources, the vocabularies listed in this OKS Data Model specification (and the related documents) should be favoured, when applicable.
4. *OKS Nodes provide an interface through which published OKS data can be searched and retrieved with appropriate access rights. Nodes may also provide interfaces for modifying and deleting descriptions.*
5. Existing identifiers (in particular, URI references) are reused extensively in descriptions and modelling. *The OKS infrastructure provides a query mechanism to ask definitions and examples of the existing identifiers. Further, with appropriate access rights, OKS Nodes may provide indexes about the identifiers they are using (e.g., as RSS feeds).*

Note that Item 5 provides an interface through which also the definitions of the known OKS-UCM concepts may be asked. When this service is decentralised in a P2P setting, this provides a basic mechanism for *knowledge discovery* with respect to the description vocabularies (i.e. asking definitions for locally unknown concepts and properties).

5. Informative Case Study

5.1. Instance Data

To illustrate the OKS Data Model, let us consider a simple example, drawn from the domain of our informative example. Assume publishing user and document information related to a conference (see Subsection 2.2).

The basic idea is to assign resources (including concepts) unique identifiers with URIs, declare the OKS-UCM in terms of a SKOS concept scheme, assert concept definitions (etc.) using the SKOS vocabulary, and associate OKS

resources with the defined concepts. We will next address these topics as a series of concrete examples using the Turtle (or N3) syntax for RDF [TURTLE].

For brevity, we will assume the following namespace definitions throughout the examples:

```
@prefix : <http://www.opaals.org/opaals2008/articles.html#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
```

-

We begin by asserting that there exists an OKS Upper Concept Model identified with the name `http://www.opaals.org/oks-ucm/` and that this model is declared as a SKOS concept scheme:

```
<http://www.opaals.org/oks/upper-concept-model/> rdf:type
skos:ConceptScheme;
  dcterms:title "Open Knowledge Space Upper Concept Model" .
```

-

Having said this, we continue by asserting the existence of and the relationship between two OKS-UCM concepts:

```
<http://www.opaals.org/oks/upper-concept-model/reviewed-proceedings-
article> rdf:type skos:Concept;
  skos:broader
<http://ebiquity.umbc.edu/ontology/publication.owl#InProceedings>;
  skos:definition "Article in peer-reviewed conference proceedings.";
  skos:inScheme <http://www.opaals.org/oks/upper-concept-model/>;
  skos:prefLabel "Reviewed Proceedings Article" .
```

-

In brief, we have said that the identifier `<http://www.opaals.org/oks/upper-concept-model/reviewed-proceedings-article>` is a SKOS concept and it is a subconcept of

```
<http://ebiquity.umbc.edu/ontology/publication.owl#InProceedings>.
```

Further, we have asserted a brief definition and the preferred label of the concept in plain English.

In a typical setting, several definitions about the OKS-UCM are established once and referenced several times. A common use case for the OKS-UCM is associating OKS resources with the concepts of the OKS-UCM.

Let us then assert information about a particular conference article:

```

:id123456 rdf:type foaf:Document;
  dcterms:title "Logics Review" .
  dcterms:abstract "In this article we ...";
  dcterms:creator <http://www.iki.fi/ossi.nykanen>;
  dcterms:format "vnd.oasis.opendocument.text";
  dcterms:hasVersion <http://www.foo-archive1.org/pubs/2008/0507-
pubea.odt>,
  <http://www.opaals.org/opaals2008/articles/ea-nykanen.odt>;
  dcterms:issued "2008-05-16"^^<http://www.w3.org/2001/XMLSchema#Date>;
  dcterms:language "en";
  dcterms:publisher <http://www.tut.fi/hypermedia>;
  dcterms:subject dcterms:BibliographicResource,
  <http://www.opaals.org/oks/upper-concept-model/reviewed-
proceedings-article>, <http://www.opaals.org/oks/upper-concept-
model/evolution-of-language>.

```

-

Intuitively, we are now asserted semantic, machine-readable information about the publication, including the title, abstract, creator, format, version information, publication date, natural language, publisher, and several subjects of the article with a single stroke. In practice, the information might be retrieved from several sources.

Finally, we may assert information about the author, e.g., as follows:

```

<http://www.iki.fi/ossi.nykanen> rdf:type foaf:Person;
  foaf:name "Ossi Nykänen";
  foaf:Organization <http://www.opaals.org>, <http://www.tut.fi>;
  foaf:homepage <http://www.iki.fi/ossi.nykanen>;
  foaf:interest <http://www.opaals.org/oks/upper-concept-model/digital-
ecosystems>,
  <http://www.opaals.org/oks/upper-concept-model/evolution-of-
language>,
  <http://www.opaals.org/oks/upper-concept-model/visualisation>;
  foaf:mbox <mailto:ossi.nykanen@iki.fi>,
  <mailto:ossi.nykanen@tut.fi>;
  foaf:thumbnail <http://matriisi.ee.tut.fi/~onykane/gfx/ossi-
thumbnail.jpg>.

```

-

Note how the property foaf:interest can now be used for, e.g., matching research topics, research interests, and subjects of publications.

Altogether, our example provides a major benefit for a simple use case of searching the OKS. Assuming the P2P infrastructure encapsulates the physical structure of the (local) OKS, we may simply look for proceedings articles in terms of a SPARQL query [SPARQL]:

```

SELECT ?resource
WHERE { ?resource dcterms:subject
  <http://www.opaals.org/oks/upper-concept-model/reviewed-
proceedings-article> . }

```

-
When equipped with inference capabilities, we might use the OKS-UCM also to add intuitive semantics to queries (e.g. "Find conference proceedings documents [including reviewed proceedings documents]."). Further, the concept structures provide a system for defining concepts based on logical operations and concept boundaries (e.g. "Find proceedings documents about the evolution of language.").

The main benefits from using Semantic Web technologies include an established data model and serialisation syntaxes, basic tool support, and a developer community with some application interoperability. However, strictly speaking, the use of RDF data model and SKOS does not include the assumption of managing individual descriptions in some RDF format. We are only assuming that a sufficiently compatible data model is used to capture the statements related to concept indexing, and introduce the RDF data model on the logical level of the common OKS Upper Concept Model (see Figure 6). In other words, when needed, the semantic data establishing the OKS-UCM concept structure might be transformed and adapted from a number of systems and formats, including legacy database systems, content management systems, rule systems, etc. (Of course, these adapters might require implementation effort, except for the use cases of established tools, such as gleaning descriptions from dialects of XML-based languages [GRDDL].)

Finally, please note that the OKS-UCM is not declared as a top-down, global ontology. Rather, the P2P nodes are making contributions to the definition of the OKS-UCM when publishing concept definitions and associations, as they are making contributions to the OKS when publishing content in general (see Figure 6). The techniques that need common recognition as interoperability standards include the abstract RDF data model (in particular, mapping identifiers as URI names) and the policy of declaring and referencing the OKS-UCM Concepts as a SKOS concept scheme. Of course, the technical interoperability as such is not sufficient for a working OKS since (social) agreements about the OKS-UCM Process are needed as well.

5.2. Notes about the Common OKS Process

It should be obvious that new OKS-UCM concept definitions should not be asserted or applied too carelessly. Further, the best practice is faithfully reusing the existing concepts (i.e. term and relationship definitions) as much as possible. Without trying to define the social process at this point, it is nevertheless instructive to briefly outline a potential approach for compiling the OKS-UCM.

In particular, the previous example included statements identifying *some* UCM concepts (see the code fragment copied below). Where did the names of these terms come from?

```
<http://www.opaals.org/oks/upper-concept-model/reviewed-proceedings-  
article> rdf:type skos:Concept;  
  skos:broader  
<http://ebiquity.umbc.edu/ontology/publication.owl#InProceedings>;  
  skos:definition "Article in peer-reviewed conference proceedings.";  
  skos:inScheme <http://www.opaals.org/oks/upper-concept-model/>;  
  skos:prefLabel "Reviewed Proceedings Article" .
```

-

While this question is almost irrelevant from the perspective of the abstract data model, it is of course highly relevant to the community.

A **potential publishing scenario** goes as follows: A designer (or a publisher) of the conference knows that searching OKS resources requires appropriate descriptions. One approach is supporting searching via semantically associating (or indexing) resources with terms drawn from the OKS Upper Concept Model.

The designer happens to know that a "proceedings publication" (as a concept) can be identified with a name

<http://ebiquity.umbc.edu/ontology/publication.owl#InProceedings> (as a term), in the eBiquity Publication Ontology [EBIQUITY]. This piece of knowledge might be due to a simple search, e.g., using the SchemaWeb repository [S-WEB]. (In principle, as noted before, this process of asking common concepts might be considered as a basic service of knowledge discovery, provided by the technical OKS infrastructure to the *end-users*. Further, the current schema and ontology repositories already provide useful search and navigation user interfaces for *designers*.)

However, the designer fails to find a common concept for a "reviewed proceedings article" and decides to introduce a new term for this concept. It should be emphasised that it is crucially important that the designer integrates her new term to the OKS-UCM, by using the property `skos:broader`. Thus, instead of simple semantic *tagging*, a genuine (albeit simple) concept structure is established. In principle, this allows searching resources not only using the terms directly associated with them, but also based on term relationships.

Later, it might happen that more appropriate or already widely used terms are discovered, e.g., <http://purl.org/net/nknouf/ns/bibtex#Inproceedings> from the bibTeX ontology [BIBTEX]. As a result, the existing descriptions may be corrected, or alternatively, explicit term relationships may be introduced. Eventually, this yields *some* concept structure for describing publications in the OKS that may also be used to describe other data (see Figure 7 below).

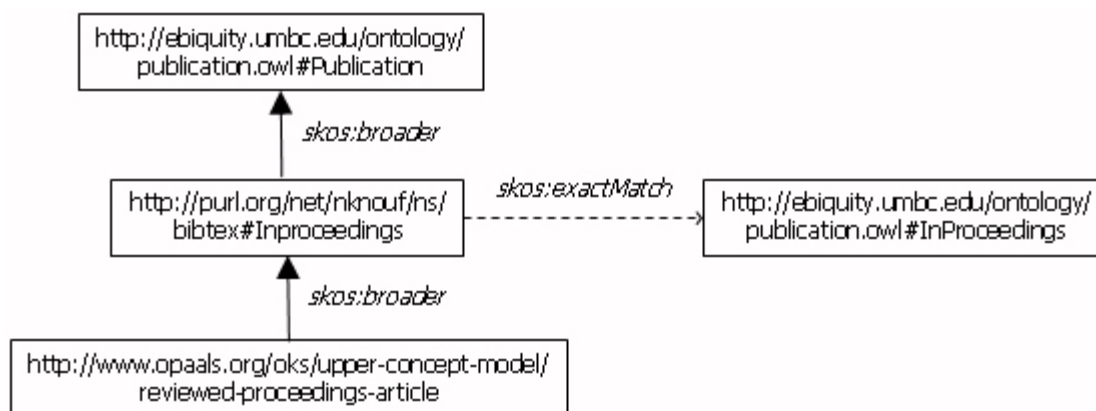


Figure 7. Example of an OKS-UCM fragment (without definitions and labels)

The difference between correcting classifications and declaring relationships lies in applications using the concept structure. In particular, if "correct" terms are used to index resources, little reasoning is required for searching resources. Nevertheless, it is unrealistic to assume that a single concept structure should suit all needs. Again, the notion of a contributed system with contextual interpretations should be kept in mind.

This concludes the informative case study. The discussion about the exact nature of the Common OKS Process is beyond the scope of this document. However, it should be obvious that the needs of the *end-users* are not met with abstract specifications: Tools and user guides are needed.

6. Conclusion

In this article, we have specified the Open Knowledge Space Data Model that provides a system for capturing and distributing descriptions of various kinds of OKS resources. In brief, the definition of the OKS-DM includes the OKS Modelling Framework, the Core OKS Vocabularies, the OKS Upper Concept Model, and the Common OKS Process.

The proposed model has been developed with respect to the known context of the OKS development. In an ideal world, the definition of the OKS-DM would of course be based on an established OKS Infrastructure and mature community needs. In practice, however, this development must be done incrementally, with parallel steps. Due to this incremental nature of the OKS development, we expect that also the OKS Data Model gets refined after the publication of this milestone. However, at this point it seems difficult to see how e.g. the OKS Modelling Framework should change in the process.

This design stance also explains why we do not provide a formal schema or an ontology for describing the OKS Data Model. We feel that at this stage of the OKS development, too strict definitions would provide an artificial constraint for

the community process. Not providing formal schemas is not a major handicap, though, since from the perspective of the OKS-DM, formal schemas should be perceived simply as OKS descriptions that can be used in OKS queries. In other words, valuable schemas, ontologies, rule systems, etc. descriptions can be distributed by the community through the (built-in) knowledge discovery process of the OKS (see Subsection 4.5).

Further, the role of the URI references in the OKS Data Model deserves some attention. In particular:

- URIs are needed.
- URIs do not represent a significant risk of a single point of failure to the OKS.
- The OKS Modelling Framework (the RDF Data Model) provides the main shared medium that internal data representations can be mapped into.
- Well-defined, concrete processes and tools are needed to incrementally build the OKS.

It should be emphasised that the questions of *technically using* URI references and *showing them to the end users* are two separate issues. In an ideal setting, *knowledge objects* (perhaps captured as collections of resources by the OKS Data Model) are referenced via *sufficiently identifying natural properties*, not necessarily by giving them explicit names according to some artificial naming procedure.

For instance, a natural way to identify an email message is to associate it with the date and time it was written, persons who were communicating, and the topics of interest. Forcing *end-users* to explicitly name the "container of knowledge objects" (e.g. name of an email message) before an act of "knowledge creation" (e.g. suggestion about a meeting place) would most probably impair the whole process. Still, being able to identify email messages with URI references (when necessary) is useful and creating URI names might not be a problem for an appropriate *email client*.

However, it seems unrealistic to assume that "advanced end-users" would *never* need to know about the formal things of the OKS Data Model or the Upper Concept Model. Looking around for roughly similar applications may help understanding why. For instance, while it is true that in principle, formal email addresses could often be hidden from the end-users sight, few email clients actually do so, for a number of (good and bad) reasons. Further, trying to find an interesting book from the local library is probably easier when having basic understanding about the library catalogue system, or at least knowing how to ask from the information desk. As a consequence, knowing the basics about these things might well be considered *literacy of the OKS* as an electronic knowledge system.

Finally, we would like to repeat our stance with respect to the fundamentally bottom-up, non-centralised nature of the OKS. Since we are talking about a data model in the context of a large heterogeneous (P2P) system, we perceive our work as a seed not only for the development of the OKS infrastructure and applications, but also for a community process. In particular, this means that the OKS vocabularies are *per se* defined and refined in detail only when the OKS is actually used, based on real needs. We also expect to see an increased interest towards Peer-to-Peer Knowledge Management in general that will eventually turn this discussion into mainstream [P2P-KM].

Acknowledgements

The authors appreciate the OPAALS community for a creative work environment. Further, for version 1.01 of the document, the authors wish to thank Paolo Dini for comments and feedback.

References

[BIBTEX] Knouf, N. (Ed.) 2004. bibTeX Definition in Web Ontology Language (OWL) Version 0.1, 2004. Available at <http://zeitkunst.org/bibtex/0.1/>

[DAML] The DARPA Agent Markup Language Homepage. Available at <http://www.daml.org/ontologies/>

[DBE07] Nachira F., Dini, P., Nicolai A., Le Louarn M., Rivera Lèon L. (Eds.) 2007. Digital Business Ecosystems, European Commission, Directorate General Information Society and Media. Bruxelles. Available at: <http://www.digital-ecosystems.org/book/de-book2007.html>

[DC-RDF] Nilsson, M., Powell, A., Johnston, P., & Naeve, A. (Eds.) 2008. Expressing Dublin Core metadata using the Resource Description Framework (RDF). Dublin Core Metadata Initiative. Available at <http://dublincore.org/documents/dc-rdf/>

[EBIQUITY] UMBC eBiquity: RGB Ontologies. UMBC ebiquity research group. Available at <http://ebiquity.umbc.edu/ontology/>

[FOAF] The Friend of a Friend (FOAF) project. Available at <http://www.foaf-project.org/>

[GEO-RSS] GeoRSS :: Geographically Encoded Objects for RSS feeds. Available at <http://georss.org/>

[GRDDL] Connolly, D. (Ed.) 2007. Gleaning Resource Descriptions from Dialects of Languages (GRDDL). W3C Recommendation 11 September 2007. Available at <http://www.w3.org/TR/grddl/>

[OPA06a] Description of Work for project Open Philosophies for Associative Autopoietic Digital Ecosystems (OPAALS). 2006. OPAALS Internal project documentation - not publicly available.

[OPA06b] Workpackage 9: Communication and dissemination. Del9.1 – Brand Identity. 2006. OPAALS project deliverable.

[OPA07a] Workpackage 3: Autopoietic P2P networks. Deliverable D3.1: Preliminary architecture for P2P network focusing on hierarchical virtual super-peers, birth and growth models. 2007. OPAALS project deliverable.

[OPA07b] Workpackage: 3: Autopoietic P2P networks. Deliverable D3.2: Report on formal analysis of autopoietic P2P network, together with predictions of performance. 2007. OPAALS project deliverable.

[OPA07c] Workpackage 7: Community Networks and Digital Ecosystems. Del7.1 – The relationship between Community Networks and Digital Ecosystems. 2008. OPAALS project deliverable.

[OPA07d] Workpackage 10: Sustainable Community Building. D10.4 – A demonstration implementation of the OKS Technical Platform. 2007. OPAALS project deliverable.

[OPA07e] Workpackage 10: Sustainable Community Building. Del10.6 – A proof-of-concept implementation of a visualisation client. 2007. OPAALS project deliverable.

[OPA07f] Workpackage 10: Sustainable Community Building. D10.1 – The design and organisation of plenary sessions, workshops and Summer/Winter schools based on an on-going process of mapping network interactions. 2007. OPAALS project deliverable.

[OPA07g] Workpackage 10: Sustainable Community Building. D10.5 – Principles, Models and Processes for the Development of the Open Knowledge Space. 2007. OPAALS project deliverable.

[OWL] Web Ontology Language (OWL). World Wide Web Consortium (W3C). Available at <http://www.w3.org/2004/OWL/>

[P2P-KM] Zaihrayeu, I. & Robertson, D. (Eds.) 2005. P2PKM '05 Peer-to-Peer Knowledge Management. Proceedings of the Second Workshop on Peer-to-Peer Knowledge Management July 17, 2005, UCSD, La Jolla, San Diego, California,

USA. Available at <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-139/>

[RDF] Resource Description Framework (RDF). World Wide Web Consortium (W3C). Available at <http://www.w3.org/RDF/>

[RDF-DM] Klyne, G. & Carroll J.J. (Eds.) 2004. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004. Available at <http://www.w3.org/TR/rdf-concepts/>

[RDF-SEMANTICS] Hayes, P. (Ed.) 2004. RDF Semantics. W3C Recommendation 10 February 2004. Available at <http://www.w3.org/TR/rdf-mt/>

[RDF-VOC] Brickley, D. & Guha, R.V. (Eds.) 2004. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004. Available at <http://www.w3.org/TR/rdf-schema/>

[RDF/XML] Beckett, D. (Ed.) 2004. RDF/XML Syntax Specification (Revised) W3C Recommendation 10 February 2004 Available at <http://www.w3.org/TR/rdf-syntax-grammar/>

[REST] Fielding, R.T. 2000. Representational State Transfer (REST), Chapter 5 of "Architectural Styles and the Design of Network-based Software Architectures", Doctoral Thesis, University of California, USA. Available at http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[RFC3986] Berners-Lee, T., Fielding, R., & Masinter, L. (Eds.) 2005. Uniform Resource Identifier (URI): Generic Syntax. IETF Network Working Group, Request for Comments: 3986, The Internet Society. Available at <http://tools.ietf.org/html/rfc3986>

[RSS10] Swartz, A. (Ed.) RDF Site Summary (RSS) 1.0. RSS-DEV Working Group. Available at <http://web.resource.org/rss/1.0/>

[SPARQL] RDF Data Access Working Group. World Wide Web Consortium (W3C). Available at <http://www.w3.org/2001/sw/DataAccess/>

[SKOS] Simple Knowledge Organization System (SKOS) - Home Page. World Wide Web Consortium (W3C). Available at <http://www.w3.org/2004/02/skos/>

[SUMO] Suggested Upper Merged Ontology (SUMO). IEEE. Available at <http://www.ontologyportal.org/>

[SW] W3C Semantic Web Activity. World Wide Web Consortium (W3C). Available at <http://www.w3.org/2001/sw/>

[S-WEB] SchemaWeb - RDF Schemas Directory. Available at <http://www.schemaweb.info/>

[SWOOGLE] Swoogle Semantic Web Search Engine. ebiquity group at UMBC. Available at <http://swoogle.umbc.edu/>

[TURTLE] Beckett, D. 2007. Turtle - Terse RDF Triple Language. Available at <http://www.dajobe.org/2004/01/turtle/>

[W3C04a] Jacobs, I. & Walsh, N. (Eds.) 2004. Architecture of the World Wide Web, Volume One. W3C Recommendation 15 December 2004. Available at <http://www.w3.org/TR/webarch/>