Object Management Group

140 Kendrick Street
Building A Suite 300
Needham, MA 02494
USA

Telephone: +1-781-444-0404
Facsimile: +1-781-444-0320

SOA4KB

Request For Proposal

OMG Document: ~~<taskforce>~~ontology/~~YYYY-MM-NN~~2010-03-01

Letters of Intent due: <day><month> <year>
Submissions due: <day><month> <year>

<Note to RFP Editors: spell out month name; e.g., January>

~~*<Note to RFP Editors: Use this template for preparing the short version of the RFP for review purposes. After the RFP is accepted in its final form the document that is issued must be based on the complete template, ab/08-08-01. When your RFP is completed, ensure that all the red text has been either replaced with details from your RFP (in black), hidden, or deleted.*~~

~~*There must be no red text whatsoever in your completed RFP. Really. **None.***>~~

Objective of this RFP

~~*< Note to RFP Editors: Provide a brief statement of the problem>*~~

The intent of this RFP is to request proposals for Application Programming Interfaces (API) to Knowledge Bases (KB). According to Wikipedia (http://en.wikipedia.org/wiki/Knowledge_base), “a knowledge base is a special kind of database for knowledge management, providing the means for the computerized collection, organization, and retrieval of knowledge”. This RFP targets mainly the retrieval and the modification of knowledge in a KB, as concepts (classes) and roles (properties and relationships) as well as individuals

(reification of concepts and roles, instances). This RFP scopes so APIs dealing with what is often called TBox (schema axioms) and ABox (ground facts).

While this API shall be independent from the way the KB was populated and how the KB is designed and organized, the outcome API shall be designed in line with the RDF and OWL metamodels of the Ontology Definition Metamodel OMG standard (ODM – http://www.omg.org/spec/ODM/1.0/).

This RFP solicits proposals for the following:

- API to retrieve raw information from a KB (resources, properties…);
- API to modify raw information in a KB;
- API to query knowledge in a KB (i.e., through reasoning);
- API to reify resources.

For further details see Chapter 6 of this document.

< Notes to RFP Editors. (1) Instructions to RFP authors are included in this red text. Delete or hide all red notes in your finished RFP. NO RED TEXT SHOULD REMAIN IN YOUR RFP! (2) When the actual RFP is in draft form, a truncated document comprising of this cover page , Chapter 6 and Appendix A suffice for review purposes. However, all chapters and appendices must be present in the published version. (3) You MUST replace the running header and footer with the name, document number and date of the RFP. (3) If additional chapters beyond Chapter 6 and appendices beyond Appendix B are added to the RFP, make sure to include them for the truncated review document, and make sure to insert a brief description of each additional chapter and Appendix in section 1.2. (4) Do not change the contents of any sections other than those mentioned in item (2) above. >

1.0 Introduction

1.1 Goals of OMG

The Object Management Group (OMG) is the world's largest software consortium with an international membership of vendors, developers, and end users. Established in 1989, its mission is to help computer users solve enterprise integration problems by supplying open, vendor-neutral portability, interoperability and reusability specifications based on Model Driven Architecture (MDA). MDA defines an approach to IT system specification that separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform, and

provides a set of guidelines for structuring specifications expressed as models. OMG has established numerous widely used standards such as OMG IDL[IDL], CORBA[CORBA], Realtime CORBA [CORBA], GIOP/IIOP[CORBA], UML[UML], MOF[MOF], XMI[XMI] and CWM[CWM] to name a few significant ones.

1.2 Organization of this document

The remainder of this document is organized as follows:

Chapter 2 - *Architectural Context* - background information on OMG's Model Driven Architecture.

Chapter 3 - *Adoption Process* - background information on the OMG specification adoption process.

Chapter 4 - *Instructions for Submitters* - explanation of how to make a submission to this RFP.

Chapter 5 - *General Requirements on Proposals* - requirements and evaluation criteria that apply to all proposals submitted to OMG.

Chapter 6 - *Specific Requirements on Proposals* - problem statement, scope of proposals sought, requirements and optional features, issues to be discussed, evaluation criteria, and timetable that apply specifically to this RFP.

< Note to RFP Editors: Additional RFP-specific chapters may also be included following Chapter 6. If additional chapters are included, please insert brief description of each such chapter here. Insert the additional chapters immediately following Chapter 6, and preceding Appendix A. >

Appendix A – *References and Glossary Specific to this RFP*

< Note to RFP Editors: Please insert any references that are specific to this RFP in section A.1 as per the instructions that appear in that section.

Note to RFP Editors: Please insert any glossary items that are specific to this RFP in section A.2 as per the instructions that appear in that section. >

Appendix B – General References and Glossary

< Note to RFP Editors: Additional RFP-specific appendices may also be included following Appendix B. If additional appendices are included, please insert brief description of each such appendix here. Insert the additional appendices immediately following Appendix B. >

1.3 Conventions

The key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.4 Contact Information

Questions related to the OMG's technology adoption process may be directed to *omg-process@omg.org*. General questions about this RFP may be sent to *responses@omg.org*.

OMG documents (and information about the OMG in general) can be obtained from the OMG's web site *(http://www.omg.org/*). OMG documents may also be obtained by contacting OMG at *documents@omg.org*. Templates for RFPs (like this document) and other standard OMG documents can be found at the OMG Template Downloads Page at *http://www.omg.org/technology/template_download.htm*

2.0 Architectural Context

<RFP writers shall not change this section>

3.0 Adoption Process

<RFP writers shall not change this section>

4.0 Instructions for Submitters

<RFP writers shall not change this section>

5.0 General Requirements on Proposals

<RFP writers shall not change this section>

6.0 Specific Requirements on Proposals

~~*< Notes to RFP Editors:*~~

~~*(1) Chapters 1 to 5 are designed to be independent of particular RFP content and should not normally need to be changed or extended. Chapter 6 should contain all information specific to this RFP. Additional chapters beyond Chapter 6 may be added if required, for example, if each distinct RFP item is more clearly covered separately.*~~

~~*(2) The red text within angle brackets below is provided for guidance and should be deleted from the RFP, as should these notes. >*~~

6.1 Problem Statement

~~*< Note to RFP Editors: Describe the nature of the problem or need that this RFP is addressing. Include contextual information that will help the understanding of the reader. >*~~

Most important usage scenarios for these APIs are the semantic desktop, the semantic web, the semantic web services and ~~some~~ numerous domain-specific applications.

6.1.1 Semantic Desktop

The semantic desktop is an answer to the following problems:

- Firstly, the lack of meta-data on all documents stored in a computer as well as the heterogeneity in the way to encode these meta-data hampers the effective search, filtering and in general acting upon the information stored in these documents.
- Secondly, information stored on a computer can only be accessed or sorted in a way related to its format. For example, depending of the operating systems, e-mails, contacts, files, tasks, notes, planned activities and so forth are stored separately and must be accessed through different programs, whatever the need of all this information to perform specific tasks. Moreover, even for simple files, different programs may be need to access and search in these files: PDF, PostScript, Microsoft Word, ASCII files and so forth.

  A semantic desktop so provide to the use an integrated view on stored knowledge.

[Sauermann2005] provides a definition of a semantic desktop:

"A Semantic Desktop is a device in which an individual stores all her digital information like documents, multimedia and messages. These are interpreted as Semantic Web resources, each is identified by a Uniform Resource Identifier (URI) and all data is accessible and queryable as RDF graph. Resources from the web can be stored and authored content can be shared with others. Ontologies allow the user to express personal mental models and form the semantic glue interconnecting information and systems. Applications respect this and store, read and communicate via ontologies and Semantic Web protocols. The Semantic Desktop is an enlarged supplement to the user's memory."

In that context, the API4KB standard will provide a standard way to access and query ontologies to the software services that form the semantic desktops.

~~… TBD…~~

6.1.2 The Semantic Web

Tim Berners-Lee originally stated the long-term vision of the Semantic Web (cf. [BernersLeeFischetti1999]):

"I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A 'Semantic Web', which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The 'intelligent agents' people have touted for ages will finally materialize."

Case studies and use cases have been published by the W3C Consortium in [SWCS]

Going down to the technical building blocks, W3C ([SWBuildingBlocks]) lists the following: tools

- RDF ([RDF]) is at the core of the Semantic Web that gives a formal definition for relations among resources on the Web.
- Tools to query information.
- Tools to have a finer and more detailed classification and characterization of those relationships as well as the resources being characterized.

- Tools to define logical relationships among resources and their relationships.

In that context, again, the API4KB standard will provide a standard way to access and query ontologies to the tools building the Semantic Web.

… TBD…

6.1.3 Semantic Web Services

The Semantic Web Services extends the Web services technology – mainly the technologies based on the WSDL [WSDL], SOAP [SOAP] and WS-I [WSI] standards – by describing a range of characteristics of services using machine-readable semantics that leverage usual Web services tasks by providing domain-specific terminology and/or enables some automation for various service related tasks enabling seamless integration of services: discovering, executing and composing services.

The usually described characteristics of web services encompass: the goals of the services, of their interfaces, the content of the parameters of their methods, the service level agreements (SLA) offered by the services, the calling process needed to perform one their capabilities and so forth.

The commonly known approaches to the Semantic Web are WSMO [WSMO], OWL-S [OWLS], SWSF [SWSF] and SAWSDL [SAWSDL]).

The API4KB standard will provide the standard way to access to the machine-readable semantics needed by the Semantic Web Services.

… TBD…

6.1.4 Domain-specific applications

Application of Knowledge Base, and thus of API4KB, for domain-specific applications are manifold, encompassing:

- The energy sector knows large amounts of heterogeneous data being generated daily from multiple sources, and information search and access across the data sources and formats.
- Manufacturing and production chains involve complex process and rules, heterogeneous systems, and need to integrate seamlessly different subsystems as well as the systems from other players.

- eHealth is a early adopter of semantic technologies. Examples of applications of semantics for health, as studied in the W3C Health Care and Life Sciences group, are in areas such as drug safety and efficacy, adaptable clinical protocol and pathways, and clinical observations interoperability. Moreover, the health community is intensely active in creating ontologies.

  Examples (including these ones) may be found in the Semantic Technologies Roadmap ([STRWS3.0]) delivered by the support action project "Web Service 3.0" funded by the European Commission (http://www.serviceweb30.eu).

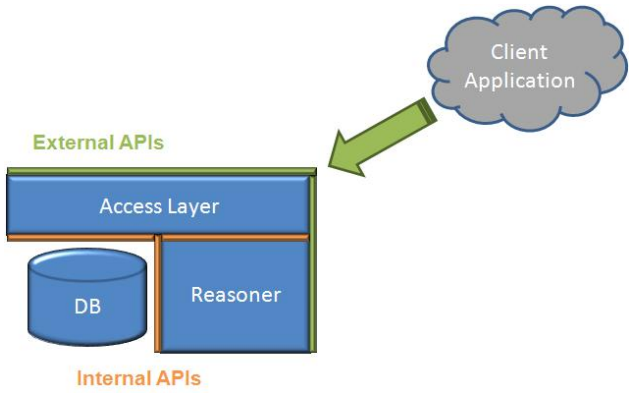
... TBD...

6.2 Scope of Proposals Sought

< Note to RFP Editors: Describe the composition and main characteristics of the solution for which proposals are being sought. >

6.2.1 KB overall view

The following picture captures roughly the main components of a Knowledge Base (these components are not mandatory for this RFP but are depicted here for understanding what API4KB is and is not).



The "Access Layer" component allows external applications to use the capabilities offered by the KB through the "external APIs". The "Reasoner" component (or Reasoning Engine or Semantic Reasoner) is "a piece of software able to infer logical consequences from a set of asserted facts or axioms" (http://en.wikipedia.org/wiki/Semantic_reasoner). The "DB" component is the component that copes with the storages and indexing of facts and axioms. All these components interact through the "Internal APIs".

The API4KB RFP focuses on the External (green colored) APIs enabling a standardization of today's Jena (http://jena.sourceforge.net/) and University of Manchester's OWL API (http://owlapi.sourceforge.net/).

6.2.2 Expressive power of KB

As already stated, a reasoner is “a piece of software able to infer logical consequences from a set of asserted facts or axioms”. There are multiple levels of reasoning, depending of the types of fact and axiom the reasoning algorithm can cope with. In the following, we will call this the expressive power of the KB.

This RFP solicits submissions that define a standardized specification of the different expressive powers that may be known by the reasoner of a KB and the APIs necessary to discover, check, or employ the expressive power of a KB.

As an example, here is a reminder of what [OWL2-profile] calls “OWL2 profile”; submitters are free to take these into account, extend them or not for their standardized set of expressive powers:

- “**OWL2 EL**~~:~~ is particularly useful in applications employing ontologies that contain very large numbers of properties and/or classes. This profile captures the expressive power used by many such ontologies and is a subset of OWL 2 for which the basic reasoning problems can be performed in time that is polynomial with respect to the size of the ontology [EL++] (…). Dedicated reasoning algorithms for this profile are available and have been demonstrated to be implementable in a highly scalable way. The EL acronym reflects the profile's basis in the EL family of description logics [EL++], logics that provide only Existential quantification.”~~TBD~~
- “**OWL 2 QL** is aimed at applications that use very large volumes of instance data, and where query answering is the most important reasoning task. In OWL 2 QL, conjunctive query answering can be implemented using conventional relational database systems. Using a suitable reasoning technique, sound and complete conjunctive query answering can be performed in LOGSPACE with respect to the size of the data (assertions). As in OWL 2 EL, polynomial time algorithms can be used to implement the ontology consistency and class expression subsumption reasoning problems. The expressive power of the profile is necessarily quite limited, although it does include most of the main features of conceptual models such as UML class diagrams and ER diagrams. The QL acronym reflects the fact that query answering in this profile can be implemented by rewriting queries into a standard relational Query Language.~~OWL2 QL: TBD~~
- ~~OWL2 RL: TBD~~”**OWL 2 RL** is aimed at applications that require scalable reasoning without sacrificing too much expressive power. It is designed to accommodate OWL 2 applications that can trade the full expressivity of the language for efficiency, as well as RDF(S)

applications that need some added expressivity. OWL 2 RL reasoning systems can be implemented using rule-based reasoning engines. The ontology consistency, class expression satisfiability, class expression subsumption, instance checking, and conjunctive query answering problems can be solved in time that is polynomial with respect to the size of the ontology. The RL acronym reflects the fact that reasoning in this profile can be implemented using a standard Rule Language"

OWL2 DL and OWL2 Full may also be worthwhile.

6.2.3 Reasoning tasks

In a KB, reasoning can perform much more difficult tasks than "just" querying. Considered tasks are (cfsee. [OWL2-Direct-Semantics] for a formal definition of them):

- Ontology Consistency,: TBD
- Ontology Entailment: TBD,
- Ontology Equivalence: TBD,
- Class Expression Satisfiability: TBD,
- Class Expression Subsumption: TBD,
- Instance Checking: TBD,
- Conjunctive Query Answering: TBD.

Other reasoning tasks may be added by submitters.

More?

Again, this RFP solicits submission to define a standardized specification of the different reasoning tasks that may be performed by the reasoner of a KB and the APIs necessary to get, check and perform these tasks.

6.2.4 PIM and PSMs

This RFP solicits submissions for a Platform Independent Model (PIM) of how applications can access to a KB but independently from the way these applications invoke these APIs, and for a set of Platform Specific Models to specific languages (Java) or middleware (WSDL, RESTful) that realizes the PIM in terms of actual implementation.

The PIM shall describe, in a platform-independent way, interfaces to retrieve, modify and query knowledge to/from a KB. Due to the influence of OWL in today's semantic field, this RFP uses the term introduced by OWL: ontology, class, properties and so forth must be understood in this context.

6.3 Relationship to other OMG Specifications and activities

6.3.1 Relationship to OMG specifications

< Note to RFP Editors: Describe the possible relationships that proposals may have to OMG Formal or Beta specifications in terms of potential reuse of models, mappings, interfaces, and potential dependencies on pervasive services and facilities. >

Ontology Definition Metamodel 1.0: http://www.omg.org/spec/ODM/1.0/.

6.3.2 Relationship to other OMG Documents and work in progress

< Note to RFP Editors: If proposals are expected to have any relationship to or dependencies on OMG documents that are not Formal or Beta specifications (such as Alpha specifications or discussion papers), describe those relationships here. If there are none, omit this section. >Common Terminology Services (CTS2) RFP: http://www.omg.org/cgi-bin/doc?ad/09-09-17.pdf

The CTS2 RFP is focused on standardizing a model and structure for terminology. In the context of CTS2, the word "terminology" refers to a generalized model that represents a collection of characteristics that span the range of simple code/value sets, hierarchical lists of names and descriptions, formal and semi-formal thesauri, coding and classification systems and, ontology. The goal of the RFP is to arrive at a standard meta-model of terminology systems that can be used to:

- Identify the minimal set of elements that must be present for a resource to be treated as a "terminology".
- Describe a single, standardized way that common idioms found throughout these resources may be expressed.
- Standardize a set of interfaces that allow these resources to be created, queried and exchanged that are based on the standardized structure and idioms.

The elements that comprise such a terminology are described via existing HL7, ISO (e.g., 11179), and other standards, such as the W3C Simple Knowledge Organization System (SKOS). These descriptions are defined via models,

schema, or ontologies that themselves are expressed in languages such as the Resource Description Framework (RDF), Web Ontology Language (OWL), and Unified Modeling Language (UML). The distinction between the languages and standards used to define CTS2 and the CTS2 model and related interfaces is crucial for understanding the difference between CTS2 and API4KBs – they co-exist at different levels of abstraction (M3/M2 for the modeling paradigms and related standards, including API4KBs, M2/M1 for the CTS2 model and services). There is a large (potentially infinite) set of vocabularies and schema that can be expressed in RDF that, while accessible through SPARQL, Jena, etc., would not be sensibly queried through the CTS2 interface. Similarly, while the standard components of some OWL ontologies should be accessible through CTS2 services, many OWL 2 ontologies, especially those including OWL annotations, will necessarily include features that are insufficiently defined for or beyond the scope of CTS2.

In parallel with and somewhat behind the CTS2 effort, this RFP requests proposals to develop one or more standard APIs for accessing knowledge bases that contain RDF vocabularies and OWL ontologies for a broad range of applications. These APIs, which will leverage existing de facto standard APIs and query languages, including but not limited to Jena, the Sesame SAIL API, various FIPA (Foundation for Intelligent Physical Agents) standards, DIG, and the University of Manchester's OWL API, will provide interfaces to knowledge bases containing RDF and OWL-based models, generally, without customization tailored to CTS2. Such a set of standardized and robust APIs could provide the platform-independent infrastructure on which CTS2 applications may depend, however. Because this effort is quite nascent, we anticipate that CTS2 implementers will develop services using some combination of the above-mentioned APIs, the SKOS API, and the SPARQL query language – until the API4KBs becomes available. The API4KBs will be developed with CTS2 services as one of a number of use cases that must be considered by submitters.

It should be noted that the CTS2 specification calls for services that will provide relationship traversal, subsumption testing and reasoning support. The intent of these requirements is to provide access to underlying reasoning capabilities that can be accessed in terms of the CTS2 idioms. *It is not* the intent of CTS2 to create an API that competes with the more abstract approaches currently expressed in the OWL API, Jena, DIG, etc. and envisioned for API4KBs.

6.4 Related non-OMG Activities, Documents and Standards

~~*< Note to RFP Editors: List documents, URLs, standards, etc. that are relevant to the problem and the proposals being sought. Also describe any known overlaps with specification activities or specifications, competing or complementary, from other standards bodies. >*~~

RDF – W3C Recommendation

OWL2 – W3C Recommendation

Jena (http://jena.sourceforge.net/) is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine.

OWL API (http://owlapi.sourceforge.net/) is a Java interface and implementation for the W3C Web Ontology Language OWL. The latest version of the API is focused towards OWL 2 which encompasses OWL-Lite, OWL-DL and some elements of OWL-Full. The OWL API is primarily maintained at the University of Manchester, but there have been significant contributions from Clark & Parsia LLC and the University of Ulm.

Sesame SAIL API

FIPA (Foundation for Intelligent Physical Agents) standards

DIG

6.5 Mandatory Requirements

6.5.1 Retrieval of raw information from KB

Proposals shall provide a PIM that defines the following capabilities:

- List the ontologies known in the KB;
- List the classes known in a given ontology;
- List the properties known in a given ontology;
- List the datatypes known in a given ontology;
- List the direct sub-classes of a given class;
- List the direct super-classes of a given class;
- List the direct equivalent classes to a given class;
- List the direct sub-properties of a given property;
- List the direct super-properties of given property;
- List the direct equivalent properties to a given property;
- Get the reference URI of a given ontology, class or property;

- Get the specification of a given class: enumerated, restriction, equivalent specification along with the parameters (constraints…) of this specification;
- Get the specification of a given property: datatype, object, equivalent, inverse property along with the parameters (constraints, classes at both ends…) of this specification;
- Get the anonymous class (equivalent to owl:Thing);
- Test if a class is equivalent to the anonymous class;

6.5.2 Modification & creation of raw information in KB

Proposals shall provide a PIM that defines the following capabilities:

- Create an ontology from its reference URI;
- Create a class in a given ontology from its reference URI and its specification: enumerated, restriction, equivalent…;
- Delete a class;
- Create a property in a given ontology from its reference URI and its specification: datatype, object, equivalent, inverse…:
- Delete a property.

6.5.3 Querying a KB

6.5.3.1 Proposals shall provide a PIM that specifies the set and standardized expressive powers that a KB may know. For each of these expressive powers, proposals shall clearly specify the semantics.

6.5.3.2 Proposals shall provide a PIM that specifies the set and standardized reasoning tasks that a KB may perform. For each of these reasoning tasks, proposals shall clearly specify the semantics.

6.5.3.3 Proposals shall provide a PIM that defines the following capabilities:

- Get the expressive power of the KB;
- Check if a given expressive power is supported by a KB;
- Get the reasoning tasks of the KB;
- Check if a given reasoning tasks is supported by a KB;

6.5.3.4 Proposals shall provide a PIM that allows the following reasoning tasks (on ontologies):

- Check the consistency of a given ontology;
- Check the entailment of a given ontology in another ontology;
- Check the equivalency between two given ontologies;

6.5.3.5 Proposals shall provide a PIM that allows the following reasoning tasks (on classes):

- Check satisfiability of a given class expression;
- Check subsumption of a given class expression by another class expression;
- List the classes that subsumes a given class expression;
- List the classes that are subsumed by a given class expression;
- Equivalency of a given class expression in another class expression;
- List the classes that are equivalent to a given class expression;

6.5.3.6 Proposals shall provide a PIM that allows the following reasoning tasks (on instances):

- Check if a given instance is an instance of a given class expression;
- List all known instances of a given class expression;
- List all answers to a given Conjunctive Query Answering;

6.5.4 Reification in a KB

Proposals shall provide a PIM that defines the following capabilities:

- ~~TBD~~Creation of an instance of a class.
- Deletion of an instance.
- Modification of the values of the properties of an instance (add, remove, get and set the values of).

6.5.5 Administration

Proposals shall provide a PIM that defines the following capabilities:

- Load a RDF or an OWL file in the KB;
- Unload (i.e. remove from the list of known ontologies) an ontology from a KB, if possible;
- Export an ontology in a serialized form (rdf/xml…).

6.5.6 PSMs

6.5.6.1 Proposals shall provide a PSM for the Java language that fully maps the PIM such that the service can be accessed from a Java application.

6.5.6.2 Proposals shall provide a PSM for WSDL that fully maps the PIM such that a KB can be accessed from a WS-I Basic Profile 1.1 compliant client.

6.5.6.3 Proposals shall provide a PSM for REST that defines how to access a KB can be accessed from a HTTP client.

Note that the "PSM for REST" is not required to be a full map for the PIM.

6.6 Optional Requirements

~~*< Note to RFP Editors: Make requests for optional features which proposals may satisfy. While the satisfaction of requests is desirable (and will be taken into account in evaluating the submissions), proposals are not required to satisfy them, i.e. specify an implementable solution.*~~

~~*Requests should be stated using phrases such as:*~~

~~*"Proposals may provide…", or*~~
~~*"Proposals may support the ability to…">*~~

6.6.1 Proposals may introduce versioning information in their APIs.

6.6.2 Proposals may provide a PSM for CORBA IDL that fully maps the PIM such that a KB can be accessed from a CORBA or a CCM compliant application.

6.7 Issues to be discussed

~~*< Note to RFP Editors: Describe the issues that proposals should discuss. Issues to be discussed shall be stated in terms of phrases such as:*~~

~~*"Proposals shall discuss how... ", or*~~
~~*"Proposals shall include information on...", or*~~
~~*"Proposals shall provide the design rationale for...".>*~~

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. (Place them in Part I of the submission.)

- Proposals should discuss the performance impact of using API4KB.
- Proposals should discuss any specific performance issues that a particular platform mapping may introduce.
- Proposals should discuss the security issues when using API4KB.

6.8 Evaluation Criteria

- ~~< Note to RFP Editors: Conformance to the mandatory requirements along with consideration of the optional requirements and issues to be discussed, are implied evaluation criteria. RFP authors should describe any additional criteria that submitters should be aware of that will be applied during the evaluation process. >~~Submissions dealing with more reasoning tasks will be favored.
- Submissions dealing with more expressive powers will be favored
- Submissions that result in higher performance or more scalable systems will be favored.

6.9 Other information unique to this RFP

None

~~*< Note to RFP Editors: Include any further information pertinent to this RFP that does not fit into the sections above, or which is intended to override statements in the Chapters 1 to 5. >*~~

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules under the item identified by the name of this RFP. Note that "<month>" and "<approximate month>" is the name of the month spelled out; e.g., January.

Event or Activity	Actual Date
Preparation of RFP by TF	
RFP placed on OMG document server	*"Four week rule"May 24th, 2010*
Approval of RFP by Architecture Board *Review by TC*	*June 25th, 2010*
TC votes to issue RFP	*<approximate month>June, 2010*
LOI to submit to RFP due	*<month> <day>, <year>January, 30th 2010*
Initial Submissions due and placed on OMG document server ("Four week rule")	<month> <day>, <year>
Voter registration closes	*<month> <day>, <year>February, 28th 2011*
Initial Submission presentations	*<month> <day>, <year>March, 2011*
Preliminary evaluation by TF	*September, 2011*
Revised Submissions due and placed on OMG document server ("Four week rule")	<month> <day>, <year>
Revised Submission presentations	*<month> <day>, <year>December, 2011*
Final evaluation and selection by TF *Recommendation to AB and TC*	*March, 2012*
Approval by Architecture Board *Review by TC*	*March, 2012*
TC votes to recommend specification	*March, 2012<approximate month>*
BoD votes to adopt specification	*March, 2012<approximate month>*

< Note to RFP Editors: Insert additional chapter if needed here and update the list and brief description of chapters in Chapter 1. >

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[OWL2-Direct-Semantics] *OWL 2 Web Ontology Language Direct Semantics* W3C Recommendation 27 October 2009 - http://www.w3.org/TR/owl2-direct-semantics/

[OWL2-Profiles] *OWL 2 Web Ontology Language Profiles* W3C Recommendation 27 October 2009 - http://www.w3.org/TR/owl2-profiles/

[RDF] *Resource Description Framework* - http://www.w3.org/RDF/

[WSDL] *Web Service Definition Language* - http://www.w3.org/TR/wsdl

[SOAP] *Simple Object Access Protocol* - http://www.w3.org/TR/soap/

[WSI] *Web service interoperability organization* - http://www.ws-i.org/

[WSMO] *Web Service Modeling Ontology* - http://www.wsmo.org/

[OWLS] *OWL for Services* - http://www.ai.sri.com/daml/services/owl-s/

[SWSF] *Semantic Web Services Framework* - http://www.daml.org/services/swsf/1.0/

[SAWSDL] *Semantic Annotations for WSDL and XML Schema* - http://www.w3.org/TR/sawsdl/

[Sauermann2005] Leo Sauermann, Ansgar Bernardi, Andreas Dengel: *Overview and Outlook on the Semantic Desktop*. In Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference - http://www.dfki.uni-kl.de/~sauermann/papers/Sauermann+2005d.pdf

[BernersLeeFischetti1999] Berners-Lee, Tim; Fischetti, Mark (1999). *Weaving the Web*. HarperSanFrancisco. chapter 12. ISBN 9780062515872.

[SWCS] *W3C's Semantic Web Case Studies and Use Cases*: http://www.w3.org/2001/sw/sweo/public/UseCases/

[SWBuildingBlocks] *What are the major building blocks of the Semantic Web?* http://www.w3.org/2001/sw/SW-FAQ#whatarebuildingblocks

[STRWS3.0] *D1.2.2 Special Purpose Roadmaps Semantic Technologies* - http://www.serviceweb30.eu/cms/index.php/resources/doc_download/97-d122-special-purpose-roadmaps-semantic-technologies

[EL++] *Pushing the EL Envelope*. Franz Baader, Sebastian Brandt, and Carsten Lutz. In Proc. of the 19th Joint Int. Conf. on Artificial Intelligence (IJCAI 2005), 2005 - http://lat.inf.tu-dresden.de/research/papers/2005/BaaderBrandtLutz-IJCAI-05.pdf

A.2 Glossary Specific to this RFP

< Note to RFP Editors: Insert any glossary items specific to this RFP that are used in Section 6 and any additional sections in the same format as in Section B.2 and in alphabetical order in this section. >None

Appendix B General Reference and Glossary

< Note to RFP Editors: Append additional appendices if needed here and update the list and brief description of appendices in Chapter 1. >

Appendix C Revision History

Date	Version	Changes
Nov. 2, 2009	V01	Creation
Nov. 12, 2009	V02	Some enhancements…
Dec. 08, 2009	V03	Version for first draft publication: ontology/2009-11-01
Mar. 10, 2010	V04	Filling in Sections 6.1, 6.2.2, 6.3 Final proposition for requirements, optional requirements, issues to be discussed and evaluation criteria. Proposition for Time Table. Second draft publication: ontology/2010-03-01