Object Management Group

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

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

API4KB

Request For Proposal

OMG Document: ad/2010-06-09

Letters of Intent due: 30 January 2010
Submissions due: 21 February, 2011

Objective of this RFP

The intent of this RFP is to request proposals for **Application Programming Interfaces** (API) **to Knowledge Bases** (KB). According to Wikipedia ([WIK-KB]), "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 basic administration services as well as the retrieval and the modification of knowledge in a KB** designed specifically for use with knowledge representation languages supported by OMG standards, including but not limited to those supported by the Ontology Definition Metamodel (ODM - http://www.omg.org/spec/ODM/1.0/). Proposals are sought that create a common set of APIs and interfaces for accessing namespaces and IRIs (internationalized uniform resource identifiers), documents, and other common infrastructure, some of which is modeled, for example, in the ODM RDFWeb package of the RDF metamodel, regardless of the target representation language, for knowledge representation languages that can be serialized in XML. The primary target is for accessing such KBs in the context of a tool, such as a parser, ontology editor, inference engine, or other applications where a uniform set of APIs and related services is desirable. APIs and service interfaces supporting basic queries, such as those that can be represented in SPARQL, are also desired.

While this API shall be independent from the way the KB was populated and how the KB is designed and organized, the resultant API(s) and services shall be designed to complement and work in concert with ODM. It is understood that this effort may necessitate modifications to the ODM, including collaboration with ODM revisions underway to support OWL 2.

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 create instances:
- API to administrate the KB: add, import or remove ontology…

For further details see Chapter 6 of this document.

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.

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

Appendix B – General References and Glossary

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

MDA provides a set of guidelines for structuring specifications expressed as models and the mappings between those models. The MDA initiative and the standards that support it allow the same model specifying business system or application functionality and behavior to be realized on multiple platforms.

MDA enables different applications to be integrated by explicitly relating their models; this facilitates integration and interoperability and supports system evolution (deployment choices) as platform technologies change. The three primary goals of MDA are portability, interoperability and reusability.

Portability of any subsystem is relative to the subsystems on which it depends. The collection of subsystems that a given subsystem depends upon is often loosely called the *platform*, which supports that subsystem. Portability – and reusability - of such a subsystem is enabled if all the subsystems that it depends upon use standardized interfaces (APIs) and usage patterns.

MDA provides a pattern comprising a portable subsystem that is able to use any one of multiple specific implementations of a platform. This pattern is repeatedly usable in the specification of systems. The five important concepts related to this pattern are:

1. *Model* – A model is a representation of a part of the function, structure and/or behavior of an application or system. A representation is said to be formal when it is based on a language that has a well-defined form ("syntax"), meaning ("semantics"), and possibly rules of analysis, inference, or proof for its constructs. The syntax may be graphical or textual. The semantics might be defined, more or less formally, in terms of things observed in the world being described (e.g. message sends and replies, object states and state changes, etc.), or by translating higher-level language constructs into other constructs that have a well-defined meaning. The optional rules of inference define what unstated properties you can deduce from the explicit statements in the model. In MDA, a representation that is not formal in this sense is not a model. Thus, a diagram with boxes and lines and arrows that is not supported by a definition of the meaning of a box, and the meaning of a line and of an arrow is not a model—it is just an informal diagram.

2. *Platform* – A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

3. *Platform Independent Model (PIM)* – A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

4. *Platform Specific Model (PSM)* – A model of a subsystem that includes information about the specific technology that is used in the realization of that subsystem on a specific platform, and hence possibly contains elements that are specific to the platform.

5. *Mapping* – Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel. A mapping may be expressed as associations, constraints, rules, templates with parameters that must be assigned during the mapping, or other forms yet to be determined.

For example, in case of CORBA the platform is specified by a set of interfaces and usage patterns that constitute the CORBA Core Specification [CORBA]. The CORBA platform is independent of operating systems and programming languages. The OMG Trading Object Service specification [TOS] (consisting of interface specifications in OMG Interface Definition Language (OMG IDL)) can be considered to be a PIM from the viewpoint of CORBA, because it is independent of operating systems and programming languages. When the IDL to C++ Language Mapping specification is applied to the Trading Service PIM, the C++-specific result can be considered to be a PSM for the Trading Service, where the platform is the C++ language and the C++ ORB implementation. Thus the IDL to C++ Language Mapping specification [IDLC++] determines the mapping from the Trading Service PIM to the Trading Service PSM.

Note that the Trading Service model expressed in IDL is a PSM relative to the CORBA platform too. This highlights the fact that platform-independence and platform-specificity are relative concepts.

The UML Profile for EDOC specification [EDOC] is another example of the application of various aspects of MDA. It defines a set of modeling constructs that are independent of middleware platforms such as EJB [EJB], CCM [CCM], MQSeries [MQS], etc. A PIM based on the EDOC profile uses the middleware-independent constructs defined by the profile and thus is middleware-independent. In addition, the specification defines formal metamodels for some specific middleware platforms such as EJB, supplementing the already-existing OMG metamodel of CCM (CORBA Component Model). The specification also defines mappings from the EDOC profile to the middleware metamodels. For example, it defines a mapping from the EDOC profile to EJB. The mapping specifications facilitate the transformation of any EDOC-based PIM into a corresponding PSM for any of the specific platforms for which a mapping is specified.

Continuing with this example, one of the PSMs corresponding to the EDOC PIM could be for the CORBA platform. This PSM then potentially constitutes a PIM, corresponding to which there would be implementation language specific PSMs derived via the CORBA language mappings, thus illustrating recursive use of the Platform-PIM-PSM-Mapping pattern.

Note that the EDOC profile can also be considered to be a platform in its own right. Thus, a model expressed via the profile is a PSM relative to the EDOC platform.

An analogous set of concepts apply to Interoperability Protocols wherein there is a PIM of the payload data and a PIM of the interactions that cause the data to find its way from one place to another. These then are realized in specific ways for specific platforms in the corresponding PSMs.

Analogously, in case of databases there could be a PIM of the data (say using the Relational Data Model), and corresponding PSMs specifying how the data is actually represented on a storage medium based on some particular data storage paradigm etc., and a mapping from the PIM to each PSM.

OMG adopts standard specifications of models that exploit the MDA pattern to facilitate portability, interoperability and reusability, either through ab initio development of standards or by reference to existing standards. Some examples of OMG adopted specifications are:

1. *Languages* – e.g. IDL for interface specification, UML for model specification, OCL for constraint specification, etc.

2. Mappings – e.g. Mapping of OMG IDL to specific implementation languages (CORBA PIM to Implementation Language PSMs), UML Profile for EDOC (PIM) to CCM (CORBA PSM) and EJB (Java PSM), CORBA (PSM) to COM (PSM) etc.

3. *Services* – e.g. Naming Service [NS], Transaction Service [OTS], Security Service [SEC], Trading Object Service [TOS] etc.

4. *Platforms* – e.g. CORBA [CORBA].

5. *Protocols* – e.g. GIOP/IIOP [CORBA] (both structure and exchange protocol), XML Metadata Interchange [XMI] (structure specification usable as payload on multiple exchange protocols).

6. *Domain Specific Standards* – e.g. Data Acquisition from Industrial Systems (Manufacturing) [DAIS], General Ledger Specification (Finance) [GLS], Air Traffic Control (Transportation) [ATC], Gene Expression (Life Science Research) [GE], Personal Identification Service (Healthcare) [PIDS], etc.

For an introduction to MDA, see [MDAa]. For a discourse on the details of MDA please refer to [MDAc]. To see an example of the application of MDA see [MDAb]. For general information on MDA, see [MDAd].

Object Management Architecture (OMA) is a distributed object computing platform architecture within MDA that is related to ISO's Reference Model of Open Distributed Processing RM-ODP[RM-ODP]. CORBA and any extensions to it are based on OMA. For information on OMA see [OMA].

3.0 Adoption Process

3.1 Introduction

OMG adopts specifications by explicit vote on a technology-by-technology basis. The specifications selected each satisfy the architectural vision of MDA. OMG bases its decisions on both business and technical considerations. Once a specification adoption is finalized by OMG, it is made available for use by both OMG members and non-members alike.

Request for Proposals (RFP) are issued by a *Technology Committee* (TC), typically upon the recommendation of a *Task Force* (TF) and duly endorsed by the *Architecture Board* (AB).

Submissions to RFPs are evaluated by the TF that initiated the RFP. Selected specifications are *recommended* to the parent TC after being *reviewed* for technical merit and consistency with MDA and other adopted specifications and *endorsed* by the AB. The parent TC of the initiating TF then votes to *recommend adoption* to the OMG Board of Directors (BoD). The BoD acts on the recommendation to complete the adoption process.

For more detailed information on the adoption process see the *Policies and Procedures of the OMG Technical Process* [P&P] and the *OMG Hitchhiker's Guide* [Guide]. In case of any inconsistency between this document and the [P&P] in all cases the [P&P] shall prevail.

3.2 Steps in the Adoption Process

A TF, its parent TC, the AB and the Board of Directors participate in a collaborative process, which typically takes the following form:

- *Development and Issuance of RFP*

RFPs are drafted by one or more OMG members who are interested in the adoption of a standard in some specific area. The draft RFP is presented to an appropriate TF, based on its subject area, for approval and recommendation to issue. The TF and the AB provide guidance to the drafters of the RFP. When the TF and the AB are satisfied that the RFP is appropriate and ready for issuance, the TF recommends issuance to its parent TC, and the AB endorses the recommendation. The TC then acts on the recommendation and issues the RFP.

- *Letter of Intent (LOI)*

A Letter of Intent (LOI) must be submitted to the OMG signed by an officer of the member organization which intends to respond to the RFP, confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements. (See section 4.3 for more information.). In order to respond to an RFP the organization must be a member of the TC that issued the RFP.

- *Voter Registration*

Interested OMG members, other than Trial, Press and Analyst members, may participate in specification selection votes in the TF for an RFP. They may need to register to do so, if so stated in the RFP. Registration ends on a specified date, 6 or more weeks after the announcement of the registration period. The registration closure date is typically around the time of initial submissions. Member organizations that have submitted an LOI are automatically registered to vote.

- *Initial Submissions*

Initial Submissions are due by a specified deadline. Submitters normally present their proposals at the first meeting of the TF after the deadline. Initial Submissions are expected to be complete enough to provide insight on the technical directions and content of the proposals.

- *Revision Phase*

During this time submitters have the opportunity to revise their Submissions, if they so choose.

- *Revised Submissions*

Revised Submissions are due by a specified deadline. Submitters again normally present their proposals at the next meeting of the TF after the deadline. (Note that there may be more than one Revised Submission deadline. The decision to set new Revised Submission deadlines is made by the registered voters for that RFP.)

- *Selection Votes*

When the registered voters for the RFP believe that they sufficiently understand the relative merits of the Revised Submissions, a selection vote is taken. The result of this selection vote is a recommendation for adoption to the TC. The AB reviews the proposal for MDA compliance and technical merit. An endorsement from the AB moves the voting process into the issuing Technology Committee. An eight-week voting period ensues in which the TC votes to recommend adoption to the OMG Board of Directors (BoD). The final vote, the vote to adopt, is taken by the BoD and is based on technical merit as well as business qualifications. The resulting draft standard is called the *Alpha Specification*.

- Business Committee Questionnaire

The submitting members whose proposal is recommended for adoption need to submit their response to the BoD Business Committee Questionnaire [BCQ] detailing how they plan to make use of and/or make the resulting standard available in products. If no organization commits to make use of the standard, then the BoD will typically not act on the recommendation to adopt the standard - so it is very important to fulfill this requirement.

- Finalization

A Finalization Task Force (FTF) is chartered by the TC that issued the RFP, to prepare an Alpha submission for publishing as a Formal (i.e. publicly available) specification, by fixing any problems that are reported by early users of the specification. Upon completion of its activity the FTF recommends adoption of the resulting Beta (draft) specification. The parent TC acts on the recommendation and recommends adoption to the BoD. OMG Technical Editors produce the Formal Specification document based on this Beta Specification.

- Revision

A Revision Task Force (RTF) is normally chartered by a TC, after the FTF completes its work, to manage issues filed against the Formal Specification by implementers and users. The output of the RTF is a Beta specification reflecting minor technical changes, which the TC and Board will usually approve for adoption as the next version of the Formal Specification.

3.3 Goals of the evaluation

The primary goals of the TF evaluation are to:

- Provide a fair and open process
- Facilitate critical review of the submissions by members of OMG
- Provide feedback to submitters enabling them to address concerns in their revised submissions
- Build consensus on acceptable solutions

- Enable voting members to make an informed selection decision

Submitters are expected to actively contribute to the evaluation process.

4.0 Instructions for Submitters

4.1 OMG Membership

To submit to an RFP issued by the Platform Technology Committee the submitter or submitters must be either Platform or Contributing members on the date of the submission deadline, while for Domain Technology RFPs the submitter or submitters must be either Contributing or Domain members. Submitters sometimes choose to name other organizations that support a submission in some way; however, this has no formal status within the OMG process, and for OMG's purposes confers neither duties nor privileges on the organizations thus named.

4.2 Submission Effort

An RFP submission may require significant effort in terms of document preparation, presentations to the issuing TF, and participation in the TF evaluation process. Several staff months of effort might be necessary. OMG is unable to reimburse submitters for any costs in conjunction with their submissions to this RFP.

4.3 Letter of Intent

A Letter of Intent (LOI) must be submitted to the OMG Business Committee signed by an officer of the submitting organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements. These terms, conditions, and requirements are defined in the *Business Committee RFP Attachment* and are reproduced verbatim in section 4.4 below.

The LOI should designate a single contact point within the submitting organization for receipt of all subsequent information regarding this RFP and the submission. The name of this contact will be made available to all OMG members. The LOI is typically due 60 days before the deadline for initial submissions. LOIs must be sent by fax or paper mail to the "RFP Submissions Desk" at the main OMG address shown on the first page of this RFP.

Here is a suggested template for the Letter of Intent:

This letter confirms the intent of <organization required> (the organization) to submit a response to the OMG <RFP name required> RFP. We will grant OMG and its members the right to copy our response for review purposes as specified in section 4.7 of the RFP. Should our response be adopted by OMG we will comply with the OMG Business Committee terms set out in section 4.4 of the RFP and in document omg/06-03-02.

<contact name and details required> will be responsible for liaison with OMG regarding this RFP response.

The signatory below is an officer of the organization and has the approval and authority to make this commitment on behalf of the organization.

<signature required>

4.4 Business Committee RFP Attachment

This section contains the text of the Business Committee RFP attachment concerning commercial availability requirements placed on submissions. This attachment is available separately as an OMG document omg/06-03-02.

Commercial considerations in OMG technology adoption

A1 Introduction

OMG wishes to encourage rapid commercial adoption of the specifications it publishes. To this end, there must be neither technical, legal nor commercial obstacles to their implementation. Freedom from the first is largely judged through technical review by the relevant OMG Technology Committees; the second two are the responsibility of the OMG Business Committee. The BC also looks for evidence of a commitment by a submitter to the commercial success of products based on the submission.

A2 Business Committee evaluation criteria

A2.1 Viable to implement across platforms

While it is understood that final candidate OMG submissions often combine technologies before they have all been implemented in one system, the Business Committee nevertheless wishes to see evidence that each major feature has been implemented, preferably more than once, and by separate organisations. Pre-product implementations are acceptable. Since use of OMG specifications *should not be dependant on any one platform, cross-platform availability and interoperability of implementations should be also be demonstrated.*

A2.2 Commercial availability

In addition to demonstrating the existence of implementations of the specification, the submitter must also show that products based on the specification are commercially available, or will be within 12 months of the date when the specification was recommended for adoption by the appropriate Task Force. Proof of intent to ship product within 12 months might include:

- *A public product announcement with a shipping date within the time limit.*
- *Demonstration of a prototype implementation and accompanying draft user documentation.*

Alternatively, and at the Business Committee's discretion, submissions may be adopted where the submitter is not a commercial software provider, and therefore will not make implementations commercially available. However, in this case the BC will require concrete evidence of two or more independent implementations of the specification being used by end- user organisations as part of their businesses. Regardless of which requirement is in use, the submitter must inform the OMG of completion of the implementations when commercially available.

A2.3 Access to Intellectual Property Rights

OMG will not adopt a specification if OMG is aware of any submitter, member or third party which holds a patent, copyright or other intellectual property right (collectively referred to in this policy statement as "IPR") which might be infringed by implementation or recommendation of such specification, unless OMG believes that such IPR owner will grant a license to organisations (whether OMG members or not) on non-discriminatory and commercially reasonable terms which wish to make use of the specification. Accordingly, the submitter must certify that it is not aware of any claim that the specification infringes any IPR of a third party or that it is aware and believes that an appropriate non-discriminatory license is available from that third party. Except for this certification, the submitter will not be required to make any other warranty, and specifications will be offered by OMG for use "as is". If the submitter owns IPR to which an use of a specification based upon its submission would necessarily be subject, it must certify to the Business Committee that it will make a suitable license available to any user on non- discriminatory and commercially reasonable terms, to permit development and commercialisation of an implementation that includes such IPR.

It is the goal of the OMG to make all of its technology available with as few impediments and disincentives to adoption as possible, and therefore OMG strongly encourages the submission of technology as to which royalty-free licenses will be available. However, in all events, the submitter shall also certify that any necessary licence will be made available on commercially reasonable, non-discriminatory terms. The submitter is responsible for disclosing in detail all known restrictions, placed either by the submitter or, if known, others, on technology necessary for any use of the specification.

A2.4 Publication of the specification

Should the submission be adopted, the submitter must grant OMG (and its sublicensees) a world- wide, royalty-free licence to edit, store, duplicate and distribute both the specification and works derived from it (such as revisions and teaching materials). This requirement applies only to the written specification, not to any implementation of it.

A2.5 Continuing support

The submitter must show a commitment to continue supporting the technology underlying the specification after OMG adoption, for instance by showing the BC development plans for future revisions, enhancement or maintenance.

4.5 Responding to RFP items

4.5.1 Complete proposals

A submission must propose full specifications for all of the relevant requirements detailed in Chapter 6 of this RFP. Submissions that do not present complete proposals may be at a disadvantage.

Submitters are highly encouraged to propose solutions to any optional requirements enumerated in Chapter 6.

4.5.2 Additional specifications

Submissions may include additional specifications for items not covered by the RFP that they believe to be necessary and integral to their proposal. Information on these additional items should be clearly distinguished.

Submitters must give a detailed rationale as to why these specifications should also be considered for adoption. However submitters should note that a TF is unlikely to consider additional items that are already on the roadmap of an OMG TF, since this would pre-empt the normal adoption process.

4.5.3 Alternative approaches

Submitters may provide alternative RFP item definitions, categorizations, and groupings so long as the rationale for doing so is clearly stated. Equally, submitters may provide alternative models for how items are provided if there are compelling technological reasons for a different approach.

4.6 Confidential and Proprietary Information

The OMG specification adoption process is an open process. Responses to this RFP become public documents of the OMG and are available to members and non-members alike for perusal. No confidential or proprietary information of any kind will be accepted in a submission to this RFP.

4.7 Copyright Waiver

Every submission document must contain: (i) a waiver of copyright for unlimited duplication by the OMG, and (ii) a limited waiver of copyright that allows each OMG member to make up to fifty (50) copies of the document for review purposes only. See Section 4.9.2 for recommended language.

4.8 Proof of Concept

Submissions must include a “proof of concept” statement, explaining how the submitted specifications have been demonstrated to be technically viable. The technical viability has to do with the state of development and maturity of the technology on which a submission is based. This is not the same as commercial availability. Proof of concept statements can contain any information deemed relevant by the submitter; for example:

> “This specification has completed the design phase and is in the process of being prototyped.”
>
> “An implementation of this specification has been in beta-test for 4 months.”
>
> “A named product (with a specified customer base) is a realization of this specification.”

It is incumbent upon submitters to demonstrate the technical viability of their proposal to the satisfaction of the TF managing the evaluation process. OMG will favor proposals based on technology for which sufficient relevant experience has been gained.

4.9 Format of RFP Submissions

This section presents the structure of a submission in response to an RFP. *All submissions* must contain the elements itemized in section 4.9.2 below before they can be accepted as a valid response for evaluation or a vote can be taken to recommend for adoption.

4.9.1 General

- Submissions that are concise and easy to read will inevitably receive more consideration.
- Submitted documentation should be confined to that directly relevant to the items requested in the RFP. If this is not practical, submitters must make clear what portion of the documentation pertains directly to the RFP and what portion does not.
- The key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" shall be used in the submissions with the meanings as described in RFC 2119 [RFC2119].

4.9.2 Required Outline

A three-part structure for submissions is required. Part I is non-normative, providing information relevant to the evaluation of the proposed specification. Part II is normative, representing the proposed specification. Specific sections like Appendices may be explicitly identified as non-normative in Part II. Part III is normative specifying changes that must be made to previously adopted specifications in order to be able to implement the specification proposed in Part II.

PART I

- •A cover page carrying the following information (a template for this is available [Inventory]):

- The full name of the submission
- The primary contact for the submission
- The acronym proposed for the specification (e.g. UML, CORBA)
- The name and document number of the RFP to which this is a response
- The document number of the main submission document

- An inventory of all accompanying documents, with OMG document number, short description, a URL where appropriate, and whether they are normative.

- List of OMG members making the submission (see 4.1) listing exactly which members are making the submission, so that submitters can be matched with LOI responders and their current eligibility can be verified.

- Copyright waiver (see 4.7), in a form acceptable to the OMG.

One acceptable form is:

“Each of the entities listed above: (i) grants to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version, and (ii) grants to each member of the OMG a nonexclusive, royalty-free, paid up, worldwide license to make up to fifty (50) copies of this document for internal review purposes only and not for distribution, and (iii) has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used any OMG specification that may be based hereon or having conformed any computer software to such specification.”

If you wish to use some other form you must get it approved by the OMG legal counsel before using it in a submission.

- For each member making the submission, an individual contact point who is authorized by the member to officially state the member’s position relative to the submission, including matters related to copyright ownership, etc. (see 4.3)

- Overview or guide to the material in the submission

- Overall design rationale (if appropriate)

- Statement of proof of concept (see 4.8)

- Resolution of RFP requirements and requests

Explain how the proposal satisfies the specific requirements and (if applicable) requests stated in Chapter 6. References to supporting material in Part II should be given.

In addition, if the proposal does not satisfy any of the general requirements stated in Chapter 5, provide a detailed rationale.

- Responses to RFP issues to be discussed

Discuss each of the “Issues To Be Discussed” identified in Chapter 6.

PART II

The contents of this part should be structured based on the template found in [FORMS] and should contain the following elements as per the instructions in the template document cited above:

- Scope of the proposed specification
- Proposed conformance criteria

Submissions should propose appropriate conformance criteria for implementations.

- Proposed normative references

Submissions should provide a list of the normative references that are used by the proposed specification

- Proposed list of terms and definitions

Submissions should provide a list of terms that are used in the proposed specification with their definitions.

- Proposed list of symbols

Submissions should provide a list of special symbols that are used in the proposed specification together with their significance

- Proposed specification

PART III

- Changes or extensions required to existing OMG specifications

Submissions must include a full specification of any changes or extensions required to existing OMG specifications. This should be in a form that enables “mechanical” section-by-section revision of the existing specification.

4.10 How to Submit

Submitters should send an electronic version of their submission to the *RFP Submissions Desk* (omg-documents@omg.org) at OMG Headquarters by 5:00 PM U.S. Eastern Standard Time (22:00 GMT) on the day of the Initial and Revised Submission deadlines. Acceptable formats are Adobe FrameMaker source, ODF (ISO/IEC 26300), OASIS Darwin Information Typing Architecture (DITA) or OASIS DocBook 4.x (or later).

Submitters should make sure they receive electronic or voice confirmation of the successful receipt of their submission. Submitters should be prepared to send a single hardcopy version of their submission, if requested by OMG staff, to the attention of the “RFP Submissions Desk” at the main OMG address shown on the first page of this RFP.

5.0 General Requirements on Proposals

5.1 Requirements

5.1.1 Submitters are encouraged to express models using OMG modeling languages such as UML, MOF, CWM and SPEM (subject to any further constraints on the types of the models and modeling technologies specified in Chapter 6 of this RFP). Submissions containing models expressed via OMG modeling languages shall be accompanied by an OMG XMI [XMI] representation of the models (including a machine-readable copy). A best effort should be made to provide an OMG XMI representation even in those cases where models are expressed via non-OMG modeling languages.

5.1.2 Chapter 6 of this RFP specifies whether PIM(s), PSM(s), or both are being solicited. If proposals specify a PIM and corresponding PSM(s), then the rules specifying the mapping(s) between the PIM and PSM(s) shall either be identified by reference to a standard mapping or specified in the proposal. In order to allow possible inconsistencies in a proposal to be resolved later, proposals shall identify whether the mapping technique or the resulting PSM(s) are to be considered normative.

5.1.3 Proposals shall be *precise* and *functionally complete*. All relevant assumptions and context required for implementing the specification shall be provided.

5.1.4 Proposals shall specify *conformance criteria* that clearly state what features all implementations must support and which features (if any) may *optionally* be supported.

5.1.5 Proposals shall *reuse* existing OMG and other standard specifications in preference to defining new models to specify similar functionality.

5.1.6 Proposals shall justify and fully specify any *changes or extensions* required to existing OMG specifications. In general, OMG favors proposals that are *upwards compatible* with existing standards and that minimize changes and extensions to existing specifications.

5.1.7 Proposals shall factor out functionality that could be used in different contexts and specify their models, interfaces, etc. separately. Such *minimalism* fosters re-use and avoids functional duplication.

5.1.8 Proposals shall use or depend on other specifications only where it is actually necessary. While re-use of existing specifications to avoid duplication will be encouraged, proposals should avoid gratuitous use.

5.1.9 Proposals shall be *compatible* with and *usable* with existing specifications from OMG and other standards bodies, as appropriate. Separate specifications offering distinct functionality should be usable together where it makes sense to do so.

5.1.10 Proposals shall preserve maximum *implementation flexibility*. Implementation descriptions should not be included and proposals shall not constrain implementations any more than is necessary to promote interoperability.

5.1.11 Proposals shall allow *independent implementations* that are *substitutable* and *interoperable*. An implementation should be replaceable by an alternative implementation without requiring changes to any client.

5.1.12 Proposals shall be compatible with the architecture for system distribution defined in ISO's Reference Model of Open Distributed Processing [RM-ODP]. Where such compatibility is not achieved, or is not appropriate, the response to the RFP must include reasons why compatibility is not appropriate and an outline of any plans to achieve such compatibility in the future.

5.1.13 In order to demonstrate that the specification proposed in response to this RFP can be made secure in environments requiring security, answers to the following questions shall be provided:

- What, if any, are the security sensitive elements that are introduced by the proposal?
- Which accesses to security-sensitive elements must be subject to security policy control?
- Does the proposed service or facility need to be security aware?

- What default policies (e.g., for authentication, audit, authorization, message protection etc.) should be applied to the security sensitive elements introduced by the proposal? Of what security considerations must the implementers of your proposal be aware?

The OMG has adopted several specifications, which cover different aspects of security and provide useful resources in formulating responses. [CSIV2] [SEC] [RAD].

5.1.14 Proposals shall specify the degree of internationalization support that they provide. The degrees of support are as follows:

a) Uncategorized: Internationalization has not been considered.

b) Specific to <region name>: The proposal supports the customs of the specified region only, and is not guaranteed to support the customs of any other region. Any fault or error caused by requesting the services outside of a context in which the customs of the specified region are being consistently followed is the responsibility of the requester.

c) Specific to <multiple region names>: The proposal supports the customs of the specified regions only, and is not guaranteed to support the customs of any other regions. Any fault or error caused by requesting the services outside of a context in which the customs of at least one of the specified regions are being consistently followed is the responsibility of the requester.

d) Explicitly not specific to <region(s) name>: The proposal does not support the customs of the specified region(s). Any fault or error caused by requesting the services in a context in which the customs of the specified region(s) are being followed is the responsibility of the requester.

5.2 Evaluation criteria

Although the OMG adopts model-based specifications and not implementations of those specifications, the technical viability of implementations will be taken into account during the evaluation process. The following criteria will be used:

5.2.1 Performance

Potential implementation trade-offs for performance will be considered.

5.2.2 Portability

The ease of implementation on a variety of systems and software platforms will be considered.

5.2.3 Securability

The answer to questions in section 5.1.13 shall be taken into consideration to ascertain that an implementation of the proposal is securable in an environment requiring security.

5.2.4 Conformance: Inspectability and Testability

The adequacy of proposed specifications for the purposes of conformance inspection and testing will be considered. Specifications should provide sufficient constraints on interfaces and implementation characteristics to ensure that conformance can be unambiguously assessed through both manual inspection and automated testing.

5.2.5 Standardized Metadata

Where proposals incorporate metadata specifications, usage of OMG standard XMI metadata [XMI] representations must be provided as this allows specifications to be easily interchanged between XMI compliant tools and applications. Since use of XML (including XMI and XML/Value [XML/Value]) is evolving rapidly, the use of industry specific XML vocabularies (which may not be XMI compliant) is acceptable where justified.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Information Architecture (IA) is an emerging, important discipline that bridges a range of standards and best practices from communities as diverse as the Object Management Group (OMG's) Model Driven Architecture (MDA) methodology and related software engineering standards, to knowledge representation and reasoning including the World Wide Web Consortium (W3C)'s Semantic Web technologies, to ISO "metadata registry and repository management" standards, and beyond. IA is focused on defining, developing, and managing information, and in particular information models, in order to separate content and context-specific concerns from other aspects of the software, process, or service architecture of a system or business. The relevant standards cover terminology specification, master data management and governance, information analysis and definition, information modeling, and specification of the declarative knowledge underlying knowledge-based (or intelligent) systems. IA facilitates structural and semantic interoperability for large-scale systems and across domain and organizational boundaries, through language, terminology, and nomenclature standardization and the establishment of canonical information definitions. Canonical definitions developed through application of information architecture methods are platform-independent, and can be domain-independent or domain-specific, as appropriate, suitable for use across systems and software engineering disciplines at any level of abstraction.

Knowledge bases, particularly those defined using the standard schema languages OWL and RDFS, are now a significant part of many information architectures. There are, however, no standard mechanisms for accessing either the knowledge base schemas expressed in these languages or the fact populations of these knowledge bases. Semantic Web technologies are relatively mature by some measures – triple stores for managing RDF data are widely available, but APIs providing support for richer knowledge-based accessibility remain research-oriented rather than sufficiently robust for commercial adoption. At present, the most commonly used API for accessing OWL knowledge bases is an open source project under development (primarily) at the University of Manchester, supported by a small team with a much larger community of interest.[1] The primary APIs for more general RDF accessibility include the Sail API, developed for the Sesame project[2], and the Jena Semantic Web Framework for Java[3]. These APIs are brittle, have minimal support for exception handling or explanation services, and documentation is limited. From a robustness perspective, the best of these, the Jena API, is now evolving towards greater commercialization, but does not support OWL DL reasoning or OWL 2 constructs. The OWL API, which is well used in reference implementations, such as by reasoning engines including Pellet and FaCT++, and by ontology editors, such as Protégé[4] and Sandpiper's Visual Ontology Modeler[5], a UML tool plug-in for modeling ontologies using the ODM profiles for RDF and OWL, is limited to OWL DL support and thus requires tools to use multiple APIs if coverage for both RDF Schema and OWL is desired, let alone additional languages such as CL/IKL.

This RFP solicits proposals for a set of standard interfaces to knowledge bases described in OWL and RDF Schema, although they may extend to other description logic languages, more general uses of RDF, and possibly other knowledge engineering technologies. The proposed interfaces should support access to, and modification of, both the knowledge base information and the knowledge base schemas. The proposed interfaces should also support access to relevant features of the knowledge management system itself, such as reasoner capabilities. This is further described in section 6.2 below.

[1] http://owlapi.sourceforge.net/
[2] http://www.openrdf.org/doc/sesame2/system/ch05.html
[3] http://jena.sourceforge.net/
[4] Hhttp://protege.stanford.edu/H
[5] Hhttp://www.sandsoft.com/products.htmlH

Proposals should also specify a reference information model for the concepts and structures of the schema languages that are manipulated, exported and imported by the proposed services. The use of the OMG standard Ontology Definition Metamodel for this purpose is preferred, although proposals may adapt it or modify it.

Important usage scenarios for these APIs include semantically-enabled desktop services, applications of linked open data (LOD), semantic web services, and numerous domain-specific applications.

6.1.1 Semantically Enabled Desktop Services

The semantic desktop is an answer to the following problems:

- Firstly, the lack of data about all documents stored in a computer as well as the heterogeneity in the way to encode these 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 provides an integrated view of 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 import ontologies, create classes and properties and so forth. These API may for instance be used by administration tools as well as helpers to automatically populate/enhance KB from the content of the texts.
- To access and query knowledge stored on a semantic desktop as Semantic Web resources. These API may be used by the user GUI to annotate their documents.

6.1.2 Applications of Linked Open Data (LOD) and 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]

The API4KB standard will provide a standard way to access and query knowledge bases in tools building the Semantic Web.

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.

Commonly known approaches to Semantic Web Services encompass OWL-S [OWLS] and SAWSDL [SAWSDL] which leverage OWL and RDF languages in the field of Semantic Web Services.

The API4KB standard will provide the standard way to access to the machine-readable semantics needed by the Semantic Web Services as well as the way to perform some of the basic administration tasks of Semantic Web Services (importing and exporting ontologies…).

6.1.4 Domain-specific applications

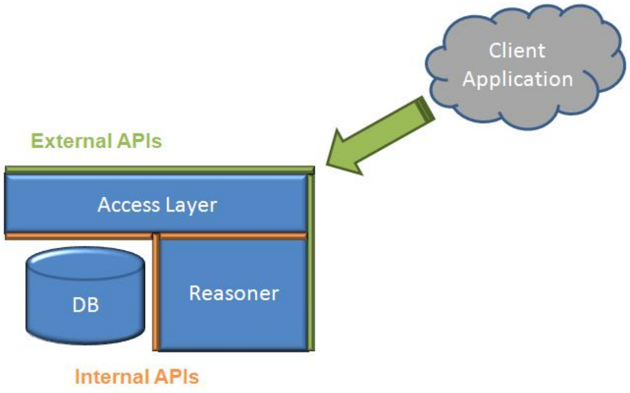
Application of Knowledge Base, and thus of API4KB, for domain-specific applications are manifold.

For instance, eHealth is an 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.

6.2 Scope of Proposals Sought

6.2.1 KB overview

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 APIs" (top-most and right-most layers in the preceding picture) enabling a standardization of today's Jena (http://jena.sourceforge.net/) and University of Manchester's OWL API (http://owlapi.sourceforge.net/), among others.

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 the types of fact and axiom that 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."

- "**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.

- "**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".

The ability to support OWL2 DL, OWL2 Full, and RDF entailment is required. Support for other representation paradigms with respect to the level of expressivity of a particular vocabulary, ontology or knowledge base, for example for the IKL extension to Common Logic, which has been discussed as a potential candidate for logical foundation work proposed by the Architecture Ecosystem SIG, may be desirable as well. Useful references with regards to vocabulary and ontology characterization include (1) the Ontology Summit 2007

Communiqué[6], (2) the OMV vocabulary[7], (3) the Proof Markup Language[8], and (4) metadata used by the BioPortal for similar purposes[9].

Here is a reminder of how [OWL2-primer] conceptually differentiates between OWL2 DL and OW2L Full:

- "One can see OWL 2 DL as a syntactically restricted version of OWL 2 Full where the restrictions are designed to make life easier for implementors. In fact, since OWL 2 Full (…) is undecidable, OWL 2 DL (…) makes writing a reasoner that, in principle, can return all "yes or no" answers (subject to resource constraints) possible. As a consequence of its design, there are several production quality reasoners that cover the entire OWL 2 DL language under the direct model-theoretic semantics. There are no such reasoners for OWL 2 Full …

- One can see OWL 2 Full as the most straightforward extension of RDFS. As such, the RDF-Based Semantics for OWL 2 Full follows the RDFS semantics and general syntactic philosophy (i.e., everything is a triple and the language is fully reflective)".

6.2.3 Reasoning tasks

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

- Ontology Consistency,
- Ontology Entailment,
- Ontology Equivalence,
- Class Expression Satisfiability,
- Class Expression Subsumption,
- Instance Checking,
- Conjunctive Query Answering (see [CQA]).

Other reasoning tasks may be added by submitters.

[6] Hhttp://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2007_CommuniqueH

[7] Hhttp://sourceforge.net/projects/omv2/H

[8] Hhttp://tw.rpi.edu/wiki/Proof_Markup_LanguageH

[9] Hhttp://bioportal.bioontology.org/H

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

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

Unified Modeling Language™ (UML™) 2.3: http://www.omg.org/spec/UML/2.3/

Meta Object Facility™ (MOF™) 2.0: http://www.omg.org/spec/MOF/2.0/

6.3.2 Relationship to other OMG Documents and work in progress

Common Terminology Services (CTS2) RFP: http://www.omg.org/cgi-bin/doc?ad/09-09-17.pdf

CTS2 is a standardization effort (the initial PIM/PSM submissions were recently published) related to API4KB. Its aim is to standardize various aspects of computational representations of terminologies and ontologies including metadata to describe such terminologies/ontologies and their elements, mechanisms to define and describe terminology and ontology subsets ("value sets") and mappings as well as SOA service interfaces and interface operations to access, reason against and manage terminologies and ontologies. Therefore, API4KB could be used as a major part of the realization of CTS2 Service implementations.

MOF Support for Semantic Structures (SMOF): http://www.omg.org/cgi-bin/doc?ad/06-06-08.pdf

This RFP requests to propose incremental changes to several existing specifications in order to allow i/ MOF Objects to have multiple types and to change types, ii/ the modeling of exclusive features in MOF, iii/ less class-based MOF structures and iv/ the modeling of unary associations. All these enhancements are aimed at reconciling MOF and Semantic Web approach.

6.4 Related non-OMG Activities, Documents and Standards

RDF – W3C Recommendation

RDF Schema – W3C Recommendation

OWL2 – W3C Recommendation

SPARQL – 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.

SAIL (Storage and Inference Layer) is an API for the Sesame open source Java framework for storing and querying RDF data. It is one of the primary open source alternatives to the Jena framework, and is also widely used in research, particularly in the EU. See http://www.openrdf.org/ for documentation on both the Sesame 2 project and SAIL API.

DIG - from http://dl.kr.org/dig/index.html: “The DIG interface (often just known as DIG) provides uniform access to Description Logic Reasoners. The interface defines a simple protocol (based on HTTP PUT/GET) along with an XML Schema that describes a concept language and accompanying operations. The interface is not intended as a heavyweight specification of a reasoning service. Rather, it provides a minimal set of operations (e.g. satisfiability and subsumption checking and classification reasoning) that have been shown to be useful in applications”.

OWLlink – from http://www.owllink.org/: “The OWLlink protocol facilitates client applications to configure a reasoner, to transmit OWL 2 ontologies or fractions thereof, and to access reasoning services via a set of basic queries. Furthermore, OWLlink is flexible in that it allows to add any desired functionality by defining a corresponding extension”.

6.5 Mandatory Requirements

6.5.1 Reference information model

6.5.1.1 Submitters shall specify a reference information model for the terms used in the descriptions of the service functions and the interface parameters. This model should be a PIM in the UML or MOF form. This model shall be closely aligned with the W3C specification for OWL and RDF.

6.5.2 Retrieval of raw information from KB

Proposals shall provide a PIM that defines at least 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 explicit sub-classes of a given class;
- List the explicit super-classes of a given class;
- List the explicit equivalent classes to a given class;
- List the explicit sub-properties of a given property;
- List the explicit super-properties of given property;
- List the explicit 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 ‘Top’ (DL terminology) class (equivalent to owl:Thing);
- Test if two classes are “structurally equivalent” (see [SS]).

6.5.3 Modification & creation of raw information in KB

Proposals shall provide a PIM that defines at least the following capabilities:

- 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.

6.5.4 Querying a KB

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

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

6.5.4.3 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 a list of reasoning tasks supported by the KB;
- Check if a given list of reasoning tasks are supported by a KB;

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

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

6.5.4.5 Proposals shall provide a PIM that allows at least 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;
- Check equivalency of a given class expression in another class expression;
- List the classes that are equivalent to a given class expression;

6.5.4.6 Proposals shall provide a PIM that allows at least 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 (see [CQA]);

6.5.5 Instances in a KB

Proposals shall provide a PIM that defines at least the following capabilities:

- 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).

6.5.6 Administration

Proposals shall provide a PIM that defines at least the following capabilities:

- Load a RDF or an OWL file in the KB (from the web, and/or from a local file store, …);
- 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…).

6.5.7 PSMs

6.5.7.1 Proposals shall specify a platform mapping that fully maps the PIM into Java interface such that the service can be accessed from a Java application.

6.5.7.2 Proposals shall specify a platform mapping that fully maps the PIM into WSDL [WSDL] such that the service can be accessed from a WS-I Basic Profile 1.1 compliant client.

6.5.7.3 Proposals shall define the RESTful platform [RESTful] to which they map and provide a mapping that defines how to access a service implementing the PIM from a RESTful client.

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

6.5.7.4 Proposals shall specify one or more PSMs (e.g., OWL notation, XML Schema) for the serialization of knowledge base schemas and knowledge base populations. This PSM should correspond to the reference information model PIM that is required by 6.5.1.The proposal shall use one of the existing interchange formats already defined for OWL and RDF.

6.6 Optional Requirements

6.6.1 If appropriate, submitters may propose modifications to the ODM specification that would generally improve its usefulness in specifying interfaces to knowledge bases. In particular, submitters may propose changes that enable consistency between the ODM metamodel and the concepts and forms used in the proposed service operations, including import/export of ontologies as specified in 6.5.7.4.

6.6.2 Submitters may define a mechanism to allow non-(yet)-standardized reasoning tasks in the PIM.

6.6.3 Submitters may define a mechanism to allow non-(yet)-standardized capabilities in the PIM.

6.6.4 Proposals may introduce versioning information in their APIs.

6.6.5 Proposals shall provide a platform mapping that fully maps the PIM into IDL such that the service can be accessed from a CORBA or CCM compliant client.

6.6.6 Proposals may specify a PSM providing a full mapping of the PIM to a domain-specific technology not listed in previous requirements.

6.6.7 Proposals may introduce into their APIs the ability to maintain & operate against specific "configurations" of resources where a "configuration" enumerates the knowledge resources to be considered, rules and inference to be applied, versions to be used, the vocabulary and language of interest and where updates are to be stored. Such a configuration will be defined by an ontology specified by the proposal.

6.6.8 Proposals may use a formal language to specify the semantics of the possible expressive powers (see 6.5.4.1) and/or of the possible reasoning tasks (see 6.5.4.2).

6.7 Issues to be discussed

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.)

6.7.1 Proposals should discuss exception handling (API and service exceptions) as well as explanation support, meaning, explanation of the proof steps taken by a reasoner to arrive at a specific conclusion.

6.7.2 Proposals should discuss the performance impact of using API4KB.

6.7.3 Proposals should discuss any specific performance issues that a particular platform mapping may introduce.

6.7.4 Proposals should discuss the security issues when using API4KB.

6.7.5 If the proposed reference information model differs from the ODM specification, or the proposal proposes changes to the ODM specification, submitters shall explain the reasons for the differences or the proposed changes.

6.7.6 Proposal should discuss how they position themselves to other related efforts at OMG such as CTS2.

6.7.7 Proposal should discuss the alignment with the standard metamodels for OWL and RDF that are contained in the OMG Ontology Definition Metamodel specification [ODM].

6.8 Evaluation Criteria

- 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.
- Submissions that result in higher fidelity explanation support and greater flexibility as well as coverage for exception handling will be favored.

6.9 Other information unique to this RFP

None

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	*May 24th, 2010*
Approval of RFP by Architecture Board *Review by TC*	*June 25th, 2010*
TC votes to issue RFP	*June, 2010*
LOI to submit to RFP due	*January, 30th 2011*
Initial Submissions due and placed on OMG document server (“Four week rule”)	*February 21st, 2011*
Voter registration closes	*February, 28th 2011*
Initial Submission presentations	*March, 2011*
Preliminary evaluation by TF	*September, 2011*
Revised Submissions due and placed on OMG document server (“Four week rule”)	*November, 14th 2011*
Revised Submission presentations	*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*
BoD votes to adopt specification	*March, 2012*

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[WIK-KB] http://en.wikipedia.org/wiki/Knowledge_base

[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/

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

[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/

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

[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/

[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

[SS] *Structural Specification* in *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax* - http://www.w3.org/TR/owl2-syntax/#Structural_Specification

[CQA] *Conjunctive Query*: http://www.w3.org/TR/2009/WD-owl2-semantics-20090421/#Inference_Problems

[RESTful] - As described in a dissertation by Roy Fielding, REST is an "architectural style" that basically exploits the existing technology and protocols of the Web, including HTTP (Hypertext Transfer Protocol) and XML. REST is simpler to use than the well-known SOAP (Simple Object Access Protocol) approach, which requires writing or using a provided server program (to serve data) and a client program (to request data). SOAP, however, offers potentially more capability. For example, a syndicator that wanted to include up-to-date stock prices to subscribing Web sites might need to use SOAP, which allows a greater amount of program interaction between client and server.

[WSDL] - As described by the W3C, WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings standardized by the W3C describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME. For the purposes of this RFP the term "WDSL platform" shall refer to the set of standard specifications defined by the WS-I Basic Profile 1.1 specification.

A.2 Glossary Specific to this RFP

None

Appendix B General Reference and Glossary

None

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
Mar. 31, 2010	V05	Reviewed by E. Kendall and E. Wallace. Include problem statement (Section 6.1) by E. Kendall Third draft publication: ontology/2010-04-01
May 21, 2010	V06	References to Sesame SAIL API, FIPA and DIG removed (not enough time) Published as ad/2010-05-08
May 28, 2010	V07	Fixes after remarks from Ed Barkmeyer (NIST), Cory Casanave (Model Driven Solutions Inc.), Jobst Landgrebe (II4SM), Jeff Pan (University of Aberdeen) and Nicolas Rouquette (NASA/JPL). Published as ad/2010-05-13
June 18, 2010	V08	Fixes after remarks from Manfred R. Koethe (88 Solutions – see http://www.omg.org/archives/ab/msg04100.html), Ed Barkmeyer (NIST – see http://www.omg.org/archives/adtf/msg04672.html), Elisa Kendall (SandPiper) and Sumeet Malhotra (Unisys - . OWL-DL and OWL-Full defined more fully. References to UML and MOF added. SMOF added in Section 6.3.2. Requirement 6.5.1.2 moved to Section 6.6 (Optional requirements). Unclosed

		list requirements (“at least”). Optional domain-specific technology PSM added. Discussing the positioning wrt other related efforts added. Added that API4KB also deals with basic administrative APIs (aka knowledge base management services). Intent of collaboration with future ODMv2 for OWL2 added. Discussion point on exception handling added. Evaluation criteria on fidelity explanation support, flexibility and coverage for exception handling added. Allow users of tools that implement this API4KB spec to be able to add new "reasoning tasks" and “capabilities”. Published as ontology/2010-06-01
June 21, 2010	V09	Fixes after remarks from Sumeet Malhotra (Unisys), Tom Rutt (CoastIn) and Peter Patel-Schneider (Bell Labs). References to DIG and OWLlink added. Use of a formal language to specify the semantics added as an optional requirement. PSMs requirements fixed to be clearer wrt what are REST and WSDL platforms. Required use of one of the existing interchange formats already defined for OWL and RDF added. Alignment with the standard metamodels for OWL and RDF that are contained in ODM moved from mandatory requirements to issue to be discussed. Published as ad/2010-06-09