Object Management Group

109 Highland Avenue
Needham, MA 02494
USA

Telephone: +1-781-444-0404
Facsimile: +1-781-444-0320
rfp@omg.org

Ontology, Model and Specification Integration and Interoperability (OntoIOp) Request For Proposal

OMG Document: ad/2013-11-11

Letters of Intent due: 24 February 2014
Submissions due: 18 December 2014

Objective of this RFP

This RFP solicits proposals for the following:

- A specification for an abstract metalanguage with an associated metamodel targeted at cross-language interoperability among a class of concrete languages used to record logical expressions found in ontologies, models and specifications.
- A list of concrete languages and translations to be recognized and correctly processed by implementations of this specification.
- A description of constraints and conformance criteria for additional concrete languages and translations between concrete languages that are not explicitly supported, but nonetheless have equivalent uses that could be recognized and correctly processed by implementations.

For further details see Section 6 of this document.

1 Introduction

1.1 Goals of OMG

The Object Management Group (OMG) is a 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 published many widely-used specifications such as UML [UML], BPMN [BPMN], MOF [MOF], XMI [XMI], DDS [DDS] and CORBA [CORBA], to name but a few significant ones.

1.2 Organization of this document

The remainder of this document is organized as follows:

Section 2 – *Architectural Context*. Background information on OMG's Model Driven Architecture.

Section 3 – *Adoption Process*. Background information on the OMG specification adoption process.

Section 4 – *Instructions for Submitters*. Explanation of how to make a submission to this RFP.

Section 5 – *General Requirements on Proposals*. Requirements and evaluation criteria that apply to all proposals submitted to OMG.

Section 6 – *Specific Requirements on Proposals*. Problem statement, scope of proposals sought, mandatory requirements, non-mandatory 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 **"shall"**, **"shall not"**, **"should"**, **"should not"**, **"may"** and **"need not"** in this document should be interpreted as described in Part 2 of the ISO/IEC Directives [ISO2]. These ISO terms are compatible with the same terms in IETF RFC 2119 [RFC2119].

1.4 Contact Information

Questions related to OMG's technology adoption process and any questions about this RFP should be directed to *rfp@omg.org*.

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

2 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 (non-mandatory) rules of inference define what unstated properties can be deduced from 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 or templates with parameters that to be assigned during the mapping, or other forms yet to be determined.

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 [IDL], UML for model specification [UML], BPMN for Business Process specification [BPMN], 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], DDS [DDS]

5. *Protocols* – e.g. GIOP/IIOP [CORBA] (both structure and exchange protocol), DDS Interoperability Protocol [DDSI].

6. *Domain Specific Standards* – e.g. Model for Performance-Driven Government [MPG], Single Nucleotide Polymorphisms specification [SNP], TACSIT Controller Interface specification [TACSIT].

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 Adoption Process

3.1 Introduction

OMG decides which specifications to adopt via votes of its Membership. The specifications selected should satisfy the architectural vision of MDA. OMG bases its decisions on both business and technical considerations. Once a specification is adopted by OMG, it is made available for use by both OMG members and non-members alike, at no charge.

This section 3 provides an extended summary of the RFP process. For more detailed information, see the *Policies and Procedures of the OMG Technical Process* [P&P], specifically Section 4.2, and the *OMG Hitchhiker's Guide* [Guide]. In case of any inconsistency between this document or the Hitchhiker's Guide and the Policies and Procedures, the P&P is always authoritative. All IPR-related matters are governed by OMG's *Intellectual Property Rights Policy* [IPR].

3.2 The Adoption Process in detail

3.2.1 Development and Issuance of RFP

RFPs, such as this one, are drafted by OMG Members who are interested in the adoption of an OMG specification in a particular area. The draft RFP is presented to the appropriate TF, discussed and refined, and when ready is recommended for issuance. If endorsed by the Architecture Board, the RFP may then be issued as an OMG RFP by a TC vote.

Under the terms of OMG's Intellectual Property Rights Policy [IPR], every RFP shall include a statement of the IPR Mode under which any resulting specification will be published. To achieve this, RFP authors choose one of the three allowable IPR modes specified in [IPR] and include it in the RFP – see section 6.10.

3.2.2 Letter of Intent (LOI)

Each OMG Member organisation that intends to make a Submission in response to any RFP (including this one) shall submit a Letter of Intent (LOI) signed by an officer on or before the deadline specified in the RFP's timetable (see section 6.11). The LOI provides public notice that the organisation may make a submission, but does not oblige it to do so.

3.2.3 Voter Registration

Any interested OMG Members, other than Trial, Press and Analyst members, may participate in Task Force voting related to this RFP. If the RFP timetable includes a date for closing the voting list (see section 6.11), or if the Task Force separately decides to close the voting list, then only OMG Member that have registered by the given date and those that have made an Initial Submission may vote on Task Force motions related to this RFP.

Member organizations that have submitted an LOI are automatically registered to vote in the Task Force. Technical Committee votes are not affected by the Task Force voting list – all Contributing and Domain Members are eligible to vote in DTC polls relating to DTC RFPs, and all Contributing and Platform Members are eligible to vote in PTC polls on PTC RFPs.

3.2.4 Initial Submissions

Initial Submissions shall be made electronically on or before the Initial Submission deadline, which is specified in the RFP timetable (see section 6.11), or may later be adjusted by the Task Force. Submissions shall use the OMG specification template [TMPL], with the structure set out in section 4.9. Initial Submissions shall be written specifications capable of full evaluation, and not just a summary or outline. Submitters normally present their proposals to the Task Force at the first TF meeting after the submission deadline. Making a submission incurs obligations under OMG's IPR policy – see [IPR] for details.

An Initial Submission shall not be altered once the Initial Submission deadline has passed. The Task Force may choose to recommend an Initial Submission, unchanged, for adoption by OMG; however, instead Task Force members usually offer comments and feedback on the Initial Submissions, which submitters can address (if they choose) by making a later Revised Submission.

The goals of the Task Force's Submission evaluation are:

- Provide a fair and open process
- Facilitate critical review of the submissions by OMG Members
- 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.

3.2.5 Revised Submissions

Revised Submissions are due by the specified deadline. Revised Submissions cannot be altered once their submission deadline has passed. Submitters again normally present their proposals at the next meeting of the TF after the deadline. If necessary, the Task Force may set a succession of Revised Submission deadlines. Submitters choose whether or not to make Revised Submissions – if they decide not to, their most recent Submission is carried forward, unless the Submitter explicitly withdraws from the RFP process.

The evaluation of Revised Submissions has the same goals listed above.

3.2.6 Selection Votes

When the Task Force's voters believe that they sufficiently understand the relative merits of the available Submissions, a vote is taken to recommend a submission to the Task Force's parent Technical Committee. The Architecture Board reviews the recommended Submission for MDA compliance and technical merit. Once the AB has endorsed it, members of the relevant TC vote on the recommended Submission by email. Successful completion of this vote moves the recommendation to OMG's Board of Directors (BoD).

3.2.7 Business Committee Questionnaire

Before the BoD makes its final decision on turning a Technical Committee recommendation into an OMG published specification, it asks its Business Committee to evaluate whether implementations of the specification will be publicly available. To do this, the Business Committee will send a Questionnaire

[BCQ] to every OMG Member listed as a Submitter on the recommended Submission. Members that are not Submitters can also complete a Business Committee Questionnaire for the Submission if they choose.

If no organization commits to make use of the specification, then the BoD will typically not act on the recommendation to adopt it – so it is very important that submitters respond to the BCQ.

Once the Business Committee has received satisfactory BCQ responses, the Board takes the final publication vote. A Submission that has been adopted by the Board is termed an *Alpha Specification*.

At this point the RFP process is complete.

3.2.8 Finalization & Revision

Any specification adopted by OMG by any mechanism, whether RFP or otherwise, is subject to Finalisation. A Finalization Task Force (FTF) is chartered by the TC that recommended the Specification; its task is to correct any problems reported by early users of the published specification. The FTF first collaborates with OMG's Technical Editor to prepare a cleaned-up version of the Alpha Specification with submission-specific material removed. This is the Beta1 specification, and is made publicly available via OMG's web site. The FTF then works through the list of bug reports ("issues") reported by users of the Beta1 specification, to produce a Finalisation Report and another Beta specification (usually Beta2), which is a candidate for Formal publication. Once endorsed by the AB and adopted by the relevant TC and BoD, this is published as the final, Formal Specification.

Long-term maintenance of OMG specifications is handled by a sequence of Revision Task Forces (RTFs), each one chartered to rectify any residual problems in the most-recently published specification version. For full details, see P&P section 4.4 [P&P].

4 Instructions for Submitters

4.1 OMG Membership

To submit to an RFP issued by the Platform Technology Committee an organisation shall maintain either Platform or Contributing OMG Membership from the date of the initial submission deadline, while to submit to a Domain RFP an organisation shall maintain either a Contributing or Domain membership.

4.2 Intellectual Property Rights

By making a Submission, an organisation is deemed to have granted to OMG a perpetual, nonexclusive, irrevocable, royalty-free, paid up, worldwide license to copy and distribute the document and to modify the document and distribute copies of the modified version, and to allow others to do the same. Submitter(s) shall be the copyright owners of the text they submit, or have sufficient copyright and patent rights from the copyright owners to make the Submission under the terms of OMG's IPR Policy. Each Submitter shall disclose the identities of all copyright owners in its Submission.

Each OMG Member that makes a written Submission in response to this RFP shall identify patents containing Essential Claims that it believes will be infringed if that Submission is included in an OMG Formal Specification and implemented.

By making a written Submission to this RFP, an OMG Member also agrees to comply with the Patent Licensing terms set out in section 6.10.

This section 4.2 is neither a complete nor an authoritative statement of a submitter's IPR obligations – see [IPR] for the governing document for all OMG's IPR policies.

4.3 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. OMG is unable to reimburse submitters for any costs in conjunction with their submissions to this RFP.

4.4 Letter of Intent

Every organisation intending to make a Submission against this RFP shall submit a Letter of Intent (LOI) signed by an officer on or before the deadline listed in section 6.11, or as later varied by the issuing Task Force.

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. LOIs shall be sent by email, fax or paper mail to the “RFP Submissions Desk” at the OMG address shown on the first page of this RFP.

A suggested template for the Letter of Intent is available at http://doc.omg.org/loi [LOI].

4.5 Business Committee terms

This section contains the text of the Business Committee RFP attachment concerning commercial availability requirements placed on submissions. This attachment is available separately as OMG document omg/12-12-03.

4.5.1 *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.

4.5.2 Business Committee evaluation criteria

4.5.2.1 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 dependent on any one platform, cross-platform availability and interoperability of implementations should be also be demonstrated.

4.5.2.2 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.

4.5.2.3 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 an appropriate license to organizations (whether OMG members or not) which wish to make use of the specification. 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.

The governing document for all intellectual property rights (“IPR”) policies of Object Management Group is the Intellectual Property Rights statement, available at: http://doc.omg.org/ipr. It should be consulted for the authoritative statement of the submitter's patent disclosure and licensing obligations.

4.5.2.4 Publication of the specification

Should the submission be adopted, the submitter must grant OMG (and its sublicensees) a worldwide, 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. Please consult the Intellectual Property Rights statement (http://doc.omg.org/ipr) for the authoritative statement of the submitter's copyright licensing obligations.

4.5.2.5 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.

4.6 Responding to RFP items

4.6.1 Complete proposals

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

Submitters are encouraged to include any non-mandatory features listed in Section 6.

4.6.2 Additional specifications

Submissions may include additional specifications for items not covered by the RFP and which they believe to be necessary. Information on these additional items should be clearly distinguished. Submitters shall give a detailed rationale for why any such additional specifications should also be considered for adoption. 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.

4.6.3 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.

4.7 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.

4.8 Proof of Concept

Submissions shall 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 Submission Format

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.
- To the greatest extent possible, the submission should follow the document structure set out in "ISO/IEC Directives, Part 2 – Rules for the structure and drafting of International Standards" [ISO2]. An OMG specification template is available to make it easier to follow these guidelines.
- The key words **"shall"**, **"shall not"**, **"should"**, **"should not"**, **"may"** and **"need not"** shall be used as described in Part 2 of the ISO/IEC Directives [ISO2]. These ISO terms are compatible with the same terms in IETF RFC 2119 [RFC2119]. However, the RFC 2119 terms **"must"**, **"must not"**, **"optional"**, **"required"**, **"recommended"** and **"not recommended"** shall not be used (even though they are permitted under RFC2119).

4.9.2 Mandatory Outline

All submissions shall use the following structure, based on the OMG Specification template [TEMPL]:

Section 0 of the submission shall be used to provide all non-normative supporting material relevant to the evaluation of the proposed specification, including:

- The full name of the submission
- A complete list of all OMG Member(s) making the submission, with a named contact individual for each
- The acronym proposed for the specification (e.g. UML, CORBA)

- The name and OMG document number of the RFP to which this is a response
- The OMG document number of the main submission document
- Overview or guide to the material in the submission
- Statement of proof of concept (see 4.8)
- If the proposal does not satisfy any of the general requirements stated in Section 5, a detailed rationale explaining why
- Discussion of each of the “Issues To Be Discussed” identified in Section 6.
- An explanation of how the proposal satisfies the specific requirements and (if applicable) requests stated in Section 6.

Section 1 and subsequent sections of the submission shall contain the normative specification that the Submitter(s) is/are proposing for adoption by OMG, including:

- Scope of the proposed specification
- Overall design rationale
- Conformance criteria for implementations of the proposed specification, clearly stating the features that all conformant implementations shall support, and any features that implementations may support, but which are not mandatory.
- A list of the normative references that are used by the proposed specification
- A list of terms that are used in the proposed specification, with their definitions
- A list of any special symbols that are used in the proposed specification, together with their significance
- The proposed specification itself

Section 0 will be deleted from any specification that OMG adopts and publishes. Therefore Section 0 of the submission shall contain no normative material, and any non-normative material outside section 0 shall be explicitly identified.

The main submission document and any models or other machine-interpretable files accompanying it shall be listed in an inventory file conforming to the inventory template [INVENT].

The submission shall include a copyright waiver in a form acceptable to 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."

Other forms of copyright waiver may only be used if approved by OMG legal counsel beforehand.

4.10 How to Submit

Submitters should send an electronic version of their submission to the *RFP Submissions Desk (rfp@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, ISO/IEC 26300:2006 (OpenDoc 1.1), OASIS DocBook 4.x (or later) and ISO/IEC 29500:2008 (OOXML, .docx).

Submitters should ensure that they receive confirmation of receipt of their submission.

5 General Requirements on Proposals

5.1 Requirements

5.1.1 Use of modelling languages

Submitters are encouraged to express models using OMG modelling languages such as UML, MOF, CWM and SPEM (subject to any further constraints on the types of the models and modeling technologies specified in Section 6 of this RFP). Submissions containing models expressed using 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 PIMs & PSMs

Section 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 it's the mapping technique or the resulting PSM(s) that shall be considered normative.

5.1.3 Complete submissions

Proposals shall be *precise* and *functionally complete*. Any relevant assumptions and context necessary to implement the specification shall be provided.

5.1.4 Reuse

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

5.1.5 Changes to existing specifications

Each proposal shall justify and fully specify any *changes or extensions* to existing OMG specifications necessitated by adopting that proposal. In general, OMG favors proposals that are *upwards compatible* with existing standards and that minimize changes and extensions to existing specifications.

5.1.6 Minimalism

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

5.1.7 Independence

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

5.1.8 Compatibility

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.

5.1.9 Implementation flexibility

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.

5.1.10 Encapsulation

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.

5.1.11 Security

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

- What, if any, security-sensitive elements are introduced by the proposal?
- Which accesses to security-sensitive elements should 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 should 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. [SEC] [RAD].

5.1.12 Internationalization

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 5.1.11 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, OMG standard XMI metadata [XMI] representations should be provided.

6 Specific Requirements on Proposals

6.1 Problem Statement

Logical languages are used in several fields of computing for the development of formal, machine-processable texts that carry a formal semantics. Among those fields are 1) ontologies formalizing domain knowledge, 2) formal models of systems, and 3) the formal specification of systems.

- the ontology languages OWL [OWL2], RDF [RDF, RDF-Semantics], RDFS [RDFS],
- the modeling language UML [UML] (fUML [FUML] equips part of UML with a formal semantics)
- general-purpose first-order languages: TPTP FOF, TPTP TFF [TPTP], F-logic [FLogic], Common Logic [CL]
- more specialized specification logics like modal logics, temporal logics, higher-order logics, TPTP THF [TPTP]
- more complex fully-fledged specification languages like VDM [VDM], B [B], Z [Z], CASL [CASL]
- the rule languages in the RIF [RIF] (Rule Interchange Format) and RuleML [RuleML] families of languages, as well as in OMG PRR (at least as far they are based on monotonic logics; for non-monotonic logics, see the non-mandatory requirements section)
- further languages listed in the discussion section

This great diversity of languages is partly justified by different application areas and by different technical properties of the languages. However, often the diversity makes interoperability of ontologies, models, specifications and systems more difficult. Moreover, it is not possible to find a single logical language into which all others can be mapped; rather, it is necessary to adopt a heterogeneous approach to interoperability. Ontologies, specifications and models will (for the purpose of this document) henceforth be abbreviated as *OSM*, if all three can be treated in the same way. Note that the underlying logical notion is that of a *logical theory*.

Related to the diversity of languages, there also is a diversity of various operations and relations on OSMs that are in use:

- matching and alignment of different ontologies covering one domain. Note that the task of finding or constructing matching and alignments is outside the scope of this RFP; proposals will only provide a metalanguage for writing these down.
- interpretation and refinement of OSMs
- module extraction – get relevant information out of large OSMs
- approximation – model in an expressive language, reason fast in a lightweight one
- querying

- ontology-based database access/data management
- bridges between different axiomatizations, e.g. distributed description logics, E-connections
- translations of OSMs to other languages
- combinations of OSMs

Heterogeneity can be seen at both the level of the logical languages as well as the OSMs themselves. There are many domains in which multiple OSMs exist, in some cases axiomatized in the same language, and in other cases axiomatized in different logical languages. This leads to several challenges in the design and deployment of OSMs which have been addressed by current research in ontological engineering, formal software specification and formal modeling:

How can we support sharability and reusability of OSMs within the same domain?

How can we merge OSMs in different domains, particularly in the cases in which the OSMs are axiomatized in different logical languages?

What notions of modularity play a role when only part of an OSM is being shared or reused?

What are the relationships between versions of an OSM axiomatized in different logical languages?

These challenges can be illustrated by the following set of use cases:

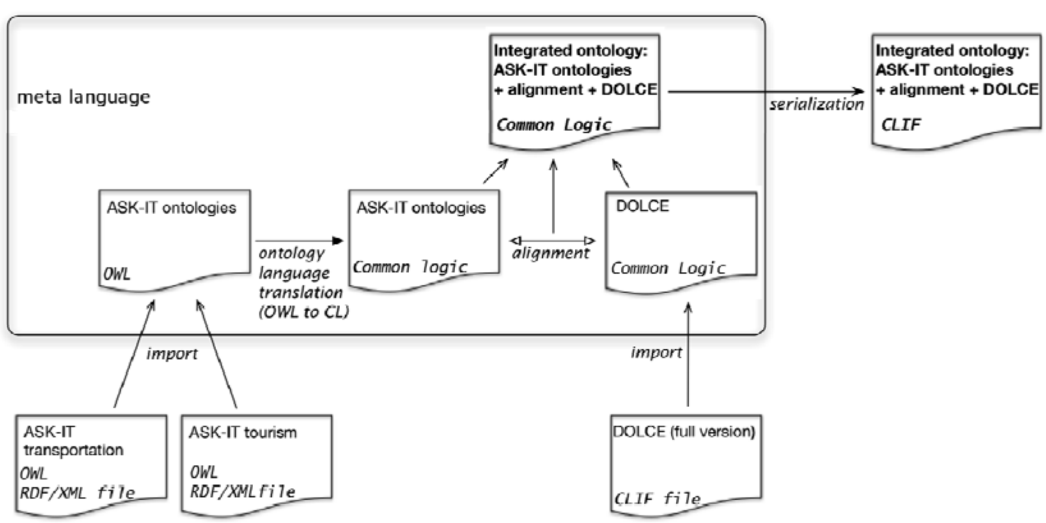
Use case Onto-1: Interoperability between OWL and FOL ontologies
In order to achieve interoperability, during ontology development it is often needed to describe concepts in a language more expressive than OWL. Therefore, it is common practice to informally annotate OWL ontologies with FOL axioms (e.g. Keet's mereo-topological ontology [Part-Whole], Dolce Lite [Dolce-lite], BFO-OWL[3]). OWL is used because of better tool support, FOL because of greater expressiveness. However, banning FOL axioms into informal annotations means that these are not available for machine processing. Another example of this problem is the following: For formally representing concept schemes (including taxonomies, thesauri and classification schemes) and provenance information there are the two W3C standards SKOS (Simple Knowledge Organization System) and PROV. The SKOS and PROV languages both have in common that their semantics is largely specified by an OWL ontology; however, as OWL cannot capture the full semantics, the rest is specified by informally first-order logic rule pseudo code[1]. I.e. valid models of SKOS or PROV are required to satisfy both some OWL and some FOL axioms.

[1] For SKOS, see http://www.w3.org/TR/2009/REC-skos-reference-20090818/ (look for “Integrity Conditions”); for PROV see http://www.w3.org/TR/2013/REC-prov-constraints-20130430/ and http://www.w3.org/TR/2013/NOTE-prov-sem-20130430/.

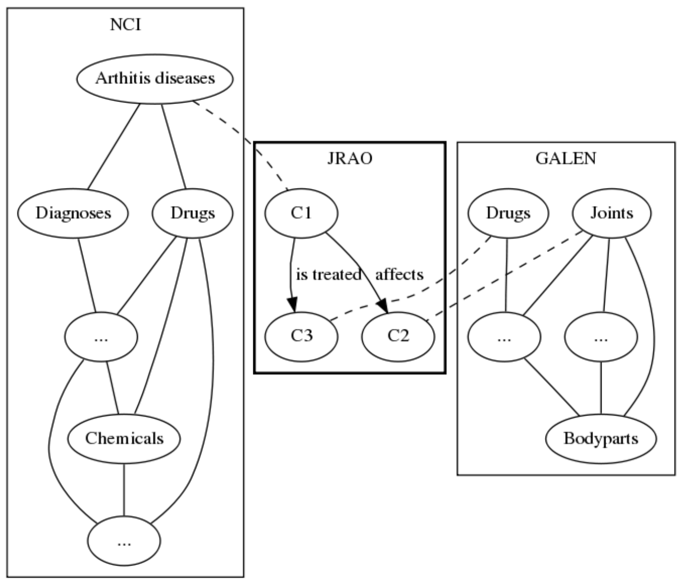
When solving reasoning tasks over either SKOS or PROV ontologies, OWL reasoners are not able to consider the FOL axioms. Hence, the information contained in these axioms is lost.
The OntoIOp metalanguage will allow the user to replace such informal annotations by formal axioms in a suitable ontology language. The relation between the OWL ontology and the FOL axioms is that of a heterogeneous import. In the result, both the OWL and the FOL axioms are amenable to e.g. automated consistency checks and theorem proving. Hence, all available information can be used in the reasoning process.

Use Case Onto-2: Ontology integration with foundational ontologies
One major use case for ontologies in industry is to achieve interoperability and data integration. However, if ontologies are developed independently and used within the same domain, the differences between the ontologies may actually cause lack of interoperability. One strategy to avoid this problem is the use of shared foundational ontologies (like DOLCE or BFO), which can be used to harmonize different domain ontologies. One challenge for this approach is that foundational ontologies typically rely on expressive ontology languages (e.g., Common Logic), while domain ontologies are represented in languages that are optimized for performance requirements (e.g., OWL EL). For this reason, currently the role of the foundational ontology is mainly to provide a conceptual framework that may be reused by the domain ontologies; further, watered-down versions of the foundational ontologies in OWL (like DOLCE-lite or the OWL version of BFO) are used as basis for domain ontologies.

The OntoIOp metalanguage will provide the framework for integrating different domain ontologies, aligning these to foundational ontologies [Alignment1-2] and combining the aligned ontologies into a coherent integrated ontology – even across different ontology languages. Thus, the OntoIOp metalanguge will enable ontology developers to utilize the complete content of foundational ontologies for ontology integration and validation.



Use Case Onto-3: Module extraction from large ontologies
Especially in the biomedical domain, ontologies tend to become very large (e.g. Snomed CT, FMA). Yet, none of these ontologies covers all aspects of a domain. Often, for a given knowledge representation problem in industry, knowledge form two such large reference ontologies needs to be integrated, but an integration of the whole large ontologies would be both unfeasible and unwieldy. Hence, parts (modules) of these ontologies are obtained by selecting the concepts and roles relevant for the intended application. An integrated version will then be based on these parts (modules). For example, the ontology JRAO has been built using modules from NCI and GALEN.
The OntoIOp metalanguage will support the description of such modules of ontologies, as well as their alignment and integration.



Use case Onto-4: Interoperability between closed-world data and open-world metadata
Current approaches to metadata of datasets that are closer to the raw sensor readings (e.g. geographical datasets derived from satellite data) have made efforts to develop standard templates that metadata authors could fill in. However, because the more complex metadata, beyond Dublin Core, can have great variety, the attempts to standardize such information has been largely a failure, with the default being to provide a link to a natural language document. Metadata for datasets of simulation results are even less standardizable. However, there is a lot of natural language source out there to serve as input.

The data is usually represented in some database, or some RDF representation assuming a closed world on the observed dataset. These languages are however not expressive enough to represent metadata about the simulation model used to generate the dataset, including the conditions for validity of the physical laws in the model of the object of observation. These metadata need some more expressive first-order language like Common Logic, with an open-world semantics. It would however be extremely redundant to translate a large dataset expressed in a compact form (e.g. flat tabular or stored in a database) into a first-order sentence for each datapoint. However, bridge rules describing how a generic datapoint has to be translated can be given.

The utility for a machine-readable representation of such bridge rules is for web applications that use online datasets for technical policy and business decision support. Current implementations used in ecological economics require manually annotating datasets with the information relevant for their processes. Taking a global view, this demand-side approach is redundant – every consuming application has to make their own annotations – and laborious to maintain as datasets get updated. Using bridge rules, a supply-side approach can be realized where the metadata are expressed in a way that consumers can derive from them the information they need.

The OntoIOp metalanguage will provide the framework for expressing such bridge rules in a systematic way, maintaining these, and building tools for them, see also [OBDA].

Use Case Onto-5: Verification of rules translating Dublin Core into PROV
The Dublin Core Metadata terms, which have been formalized as an RDFS ontology, are less comprehensive but more widely used than PROV. The rules for translating Dublin Core to the OWL subset of PROV (and, with restrictions, vice versa) are not known to yield valid instances of the PROV data model, i.e. they are not known to yield OWL ontologies consistent w.r.t. the OWL axioms that capture part of the PROV data model. This may disrupt systems that would like to reason about the provenance of an entity, and thus the assessment of the entity's quality, reliability or trustworthiness.
The Dublin Core to PROV ontology translation[2] is expressed partly by a symbol mapping and partly by FOL rules. These FOL rules are implemented by CONSTRUCT patterns in the SPARQL RDF query language (e.g. http://www.w3.org/TR/2013/NOTE-prov-dc-20130430/#dct-creator). SPARQL has a formal specification of the evaluation semantics of its algebraic expressions, which is different from the model-theoretic semantics of the OWL and RDFS logics; nevertheless SPARQL CONSTRUCT is a popular and immediately executable syntax for expressing translation rules between ontologies in RDF-based languages in a subset of FOL.
The OntoIOp metalanguage will not only support the reuse of the existing Dublin Core RDFS and PROV OWL ontologies as modules of a distributed ontology, but it will also be able to support the description of the FOL translation rules in a sufficiently expressive ontology language, e.g. Common Logic, and thus enable formal verification of the translation from Dublin Core to PROV.

[2] http://www.w3.org/TR/2013/NOTE-prov-dc-20130430/

Use case Spec-1: Specification Refinements

Especially in safety-critical areas such as medical systems, the automotive industry, avionics and the space industry, but also for microprocessor design, often a formal software and hardware development process is used in order to ensure the correct functioning of systems. Typically, a requirement specification is refined into a design specification and then an implementation, often involving several intermediate steps (see e.g. the V-model [V-model], although this does not require *formal* specification).

There is a multitude of specification formalisms in use; moreover, often during a formal development, the formalism needs to be changed (e.g. from a specification to a programming language, of from a temporal logic to a state machine). For each of these formalisms, notions of refinement have been defined and implemented. However, the lack of a standardized notion of refinement hinder the interoperability among different formal developments and the reuse of refinements.

The OntoIOp metalanguage will provide a notion of refinement that is equally applicable to all the OntoIOp conformant logical languages, and that will cover the at least the most relevant of the industrial use cases of specification refinement.

Use case Spec-2: Modularity of Specifications

In the context of use case Spec-1, often specifications become so large that it is necessary to structure them in a modular way, both for human readability, maintainability, and for more efficient tool support. The lack of a standard for such modular structuring hinders the interoperability among different formal developments and the reuse of specifications.

The OntoIOp metalanguage will provide a notion of structured modular specification that is equally applicable to all the OntoIOp conformant logical languages.

Use case Model-1: Coherent semantics for multi-language models

One and the same matter often has to be described using different formalisms, because each formalism has different user communities, expressiveness, tool support. A challenge is that typically the different formalizations are written by different people, and thus their overall consistency is hard to maintain.

The need for the use of multiple ontology languages is also reflected by the OMG Ontology Definition Metamodel (ODM), which provides a variety of transformations between such languages.

One example is the OMG Date-Time Vocabulary (DTV). DTV has been formulated in different languages, each of which addresses different audiences:

- SBVR: business users
- UML (class diagrams and OCL): software implementers
- OWL: ontology developers and users
- Common Logic: (foundational) ontology developers and users

With the OntoIOp metalanguage, one can e.g.

- formally relate the different formalizations,

- automatically translate among the different formalizations,
- check consistency across the different formalizations,
- extract submodules covering specific aspects, and
- specify the OWL version to be an approximation of the Common Logic version (using a heterogeneous interpretation of OSMs).

Note that the last point does not specify what is lost in the approximation. It may or may not be useful to state that the OWL approximation is the logically strongest OWL ontology logically implied by the Common Logic version.

Use case Model-2: Consistency among UML diagrams of different types
A typical UML model involves diagrams of different types. Such UML models may have intrinsic errors because diagrams of different types specify conflicting requirements. Typical questions that arise in this context are e.g.

- whether the sequential composition of actions in an interaction diagram is justified by an accompanying OCL specification,
- whether cooperating state machines comply with pre-/post-conditions and invariants
- if the behaviour prescribed in an interaction diagram is realisable by several state machines cooperating according to a composite structure diagram.

Such questions are currently hard to answer in a systematic manner. One method to answer these questions and finding such errors is a check for semantic consistency. Under some restrictions, in particular requiring the model to be finite state, the sub-problem whether cooperating state machines realise an interaction can be answered using model checking tools like Hugo/RT [HugoRT]. Once a formal semantics for the different diagram types has been chosen (see e.g. [HetUML]), it is possible to use the OntoIOp metalanguage to specify in which sense the diagrams need to be consistent, and check this by suitable tools.

Use case Model-3: Refinements between UML diagrams of different types, and their reuse
Lack of reuseability of refinements: Consider a controller for an elevator, which is specified with a UML protocol state machine, enriched with UML sequence diagrams and OCL constraints. Assume further that this model is not directly implemented, but first refined to a UML behaviour state machine (which then can be automatically or semi-automatically transformed into some implementation using standard UML tools). However, there is no standardized language to express, document and maintain the refinement relation itself (UML only allows very simple refinements, namely between state machines). This hinders both the reuse of such refinements in different contexts, as well as the interoperability of tools proving such refinements to be correct. The OntoIOp metalanguage will address these problems by providing a standardized notation with formal semantics for such refinements. Refinements expressed in this language could e.g. be parameterized and reused in different contexts.

Summary of use cases:
These use cases illustrate that in ontology design, in formal specification, and in model-driven development, the same problem arises: *the use of heterogeneous formal representations leads to interoperability challenges*. There are ad-hoc solutions for these challenges, and specialized languages and tools are used in practice. However, there is no standardized approach.

The OntoIOp standard will provide a standardized metalanguage for dealing with this variety of formalisms, with a well-defined semantics and model theory. The metalanguage will distill best practices of modularity and metarelations (like refinement and alignment) across the three areas of ontology design, formal specification, and model-driven development. It will lead to a formal interoperability even among heterogeneous OSMs. The metalanguage will help to solve the problems described in the use cases above. It also will foster the development of OSM libraries, tools and workflows that enable a better exchange and reuse of OSMs. Eventually, this will also lead to better, easier developed and maintained systems based on these OSMs.

6.2 Scope of Proposals Sought

Proposals shall face the diversity of languages, and not add to it by proposing yet another language that would subsume all the others. Instead, proposals shall accept the diverse reality and formulate means (on a sound and formal semantic basis) to compare and integrate OSMs (representing ontologies, specifications, or models) that are written in different formalisms.

Proposals shall specify a metalanguage which shall be able to handle OSMs formulated in specific languages (as listed in 6.1), as well as provide means for expressing s*tructuring operations* and relations between OSMs, even if these are formalized in different logical languages.
Thus, the metalanguage shall enable interoperability with a formal grounding and make heterogeneous OSMs based on them amenable to checking of coherence (e.g. consistency, conservativity, intended consequences, and compliance).

Within the OntoIOp framework, existing OSMs in conforming established languages shall remain as they are, acknowledging the wide tool support these languages enjoy. The proposed metalanguage shall enhance their modularity facilities to a superset of the modularity and annotation facilities they provide themselves. The metalanguage's modularity constructs shall be semantically well-founded within a library of formal relationships between the logics underlying the different supported logical languages.
Proposals shall specify a metalanguage providing constructs for

a) heterogeneous OSMs (ontologies, models and specifications) that combine parts written in different languages

b) links between *distributed and heterogeneous (possibly structured) OSMs* associating globally unique identifiers [URI, IRI] to any symbol, sentence, ontology and ontology link to allow for reference and annotation by means other than the metalanguage itself
c) translations between different logical languages
d) a formal semantics of (a)–(c)
e) criteria for existing or future logical languages and translations to conform with OntoIOp

Proposals shall focus on the metalanguage, while establishing OntoIOp-conformance of a small number of logical languages and translations to demonstrate the viability of the approach, as specified in 6.5.5. Establishing conformance of a larger set of logical languages and translations should be possible in principle, but is out of scope and will be left to the communities using OntoIOp.

The proliferation of specific ontologies and links between ontologies is outside the scope of OntoIOp. However, an informative annex may provide sample ontologies and links for the purpose of illustrating the constructs of the metalanguage.

6.3 Relationship to other OMG Specifications and activities

6.3.1 Relationship to OMG specifications

- Ontology Definition Metamodel (ODM) provides a graph of ontology languages and translations. Note that it captures abstract syntax only (using MOF meta models), not model theory. Proposals shall build on and may extend this graph, and have to consider model theory.
- Model Driven Architecture (MDA) – development methodology behind MOF, UML and other OMG standards
- Meta Object Facility (MOF) – metalanguage that will be used for the specification of abstract syntaxes of languages
- MOF Support for Semantic Structures (SMOF) – extension of MOF for multiple classifications and instantiations
- XML Metadata Interchange (XMI) – standard for exchanging metadata information via Extensible Markup Language (XML)
- MOF 2 XMI Mapping – mapping allowing the storage of MOF models as XMI/XML data
- MOF Model to Text Transformation Language (Mof2Text) – useful for specifying the transformation from the MOF model of the abstract syntax to the concrete syntax
- Unified Modeling Language (UML) – one specific language whose conformance with OntoIOp shall be established. Submitters shall use UML 2.4.1 or later

- Production Rule Representation (PRR) – one specific language whose conformance with OntoIOp may be established
- Semantics of Business Vocabulary and Business Rules (SBVR) – one specific language whose conformance with OntoIOp may be established (see discussion)

Date-Time Vocabulary (DTV) – use case for OntoIOp, as it has been implemented in UML, OCL, SVBR, Common Logic and OWL

6.3.2 Relationship to other OMG Documents and work in progress

- Application Programming Interfaces to Knowledge Bases (API4KB) – API for heterogeneous knowledge bases, for which OntoIOp can provide a language and a semantic basis. Vice versa, API4KB will be of importance when implementing OntoIOp, and developing OntoIOp-related APIs (see discussion).
- Semantic Information Modeling for Federation (SIMF) requests a modeling language for supporting information modeling. Responses to the OntoIOp RFP will provide a metalanguage for the relationships among modeling languages and the OSMs that are specified in the metalanguage.

6.4 Related non-OMG Activities, Documents and Standards

ISO

- WD (Working Draft) 17347 OntoIOp (ontoiop.org) developed within ISO TC 37/SC 3/WG 3 – initiative similar to the present one that has been cancelled in the meantime; the aim is to have a liaison with ISO
- Metadata Repository (ISO 19763, ISO 11179), Terminology, Metamodeling – standards for metadata. In particular, these standards' practices for allocating identifiers, and for associating downloadable human- and machine-readable encodings of descriptions of logical languages with such identifiers will be of interest of OntoIOp.
- Common Logic (ISO 24707) – family of languages that may be shown to be conformant with OntoIOp
- data description language, SQL – individual database-related language that may be shown to be conformant with OntoIOp

W3C

- OWL, RDF, RDFS, RIF, SKOS – these are W3C standards defining individual languages that may (or, in the case of OWL and RDF; shall) be shown to be conformant with OntoIOp

Other

- Open Ontology Repository Initiative (OOR) – aims at ontology repositories covering multiple ontology languages
- NeOn project – defines a number of modularity operations
- Future Internet Enterprise Systems (FInES)
- Software Platform for Integration of Engineering and Things (SPRINT)
- schema.org – RDFS-like schema developed by big search engines with the goal of structuring meta data for web pages

6.5 Mandatory Requirements

6.5.1 Metalogical Relationships

Proposals shall provide a specification of a metalanguage for the following:

- relationships between the components of logically heterogeneous OSMs, particularly the application T(Th) of a language translation T : L1 → L2 to an OSM Th written in language L1 (this will be needed by almost all use cases),
- importation in modular OSMs (cf. use cases Onto-1, Onto-3, Onto-5, Spec-2, Model-1 and Model-2),
- relationships between OSMs and their extracted modules e.g. the whole theory is a conservative extension of the module (cf. use cases Onto-3 and Model-1),
- relationships between OSMs and their approximation in less expressive languages such that the approximation is logically implied by the original theory, where the approximation generally has to be maximal in some suitable sense (cf. use case Model-1),
- links such as imports (cf. use case under "importation in modular OSMs" above), interpretations (cf. use case Model-1), refinements (cf. use cases Spec-1, Model-3), and alignments (cf. use cases Onto-2 and Onto-3) between OSMs/modules,
- combination of OSMs along links (cf. use cases Onto-2 and Onto-3).

.

6.5.2 Applicability to Multiple Logics

Proposals shall satisfy the following conditions:

- The constructs of the metalanguage shall be applicable to different logics (cf. all use cases).
- The metalanguage shall neither be restricted to OSMs in a specific domain, nor to OSMs represented in a specific logical language.
- The metalanguage shall not replace the object language constructs of the conforming logical languages.
- The metalanguage shall provide syntactic constructs for

- structuring OSMs regardless of the logic in which their sentences are formalized (cf. use case Onto-1, Onto-2, Onto-3, Onto-4, Spec-2);
- basic and structured OSMs and facilities to identify them in a globally unique way (cf. all use cases);

6.5.3 Specification of the Metalanguage

Proposals shall provide the following specifications for the metalanguage (cf. all use cases):

- an abstract syntax specified as an SMOF compliant meta model;
- a human-readable lexical concrete syntax in EBNF [EBNF] and serialization in XML [XML];
- complete round-trip mappings from the human-readable concrete syntax to the abstract syntax and vice versa;
- a formal semantics for the abstract syntax, including the relationships in 6.5.1 and the constructs in 6.5.2.

.

6.5.4 Scope of Conformant Logical Languages and Translations

Proposals shall be applicable to any logical language which either has a formal, logic-based semantics with notions of satisfiability and entailment, or which has a semantics defined by translation to another logical language with such a formal semantics.

- Existing OSMs in existing serializations (e.g. the XML-based XCL serialization of Common Logic, or the text-based OWL Manchester Syntax) shall validate as OSMs in the metalanguage with a minimum amount of syntactic adaptation.
- It shall be possible to refer to existing files/documents from an OSM implemented in the metalanguage without the need for modifying these files/documents.
- Translations between logical languages shall preserve (possibly to different degrees) the semantics of the logical languages. Between a given pair of logical languages, several translations are possible (one of them may be marked as default).

6.5.5 Conformance Criteria

Proposals shall specify formal criteria for establishing the conformance of a logical language and/or translation as required in 6.5.4.
Conformance criteria must be specified in enough detail such that they are testable. See [Institution] for an example meta notion that could be used.

- Informative annexes shall establish the conformance of a number of relevant logical languages. An initial set of language translations may be part of an informative annex,
- Conformance of the following subset of logical languages (see 6.1 above) shall be established: OWL2 (with profiles EL, RL, QL, see [OWL2-Profiles]), CLIF [CL], RDF, UML class diagrams, as well as a set of suitable translations among these.

6.5.6 Registry of Logical Languages

Proposals shall specify the technology and the rules and procedures for maintaining a *registry* of all conforming logical languages.

- The registry shall be freely readable for humans and machines (see below for serializations), using globally unique identifiers for logical languages (so that for any OSM that occurs or that is referred to in the context of the metalanguage one can explicitly refer to the language it is expressed in), and it shall offer the possibility to add further (also non-standardized) languages.
- The registry shall assign globally unique identifiers to OSMs, their parts, and links between them.

Existing standards and best practices for allocating identifiers, and for associating downloadable human- and machine-readable encodings of descriptions of logical languages with such identifiers, shall be reused, e.g. ISO/IEC 11179 or Linked Data.

6.6 Non-mandatory features

6.6.1 Tools

It is expected that suitable tools will support parsing and analysis of the OntoIOp metalanguage. Other tools will connect the metalanguage to model checkers, model finders, theorem provers etc., such that the involved verification conditions and proof obligations can be discharged. It cannot be expected, however, that there be tools for *all* such conditions and obligations. The standard can foster the development of such tools, and their interoperability.

6.6.2 Languages without a standardized model theory

There are a number of languages the inclusion of which into OntoIOp would be beneficial, and which fit in principle into the OntoIOp framework, because they have been equipped with at least one (often several) model theoretic semantics in the research literature. The problem is that there generally is no agreement on a standardized semantics that could be used in an off-the-shelf way for OntoIOp. Submissions may include such logics together with a proposal of a particular semantics; they may also propose the use of several alternative semantics. These languages include:

- the database description languages EER (Enhanced Entity-Relationship Diagrams), Datalog, ORM (object role modelling)

- the ontology language SKOS, insofar it is not regarded as just a specific ontology, but a framework that is axiomatized (e.g. in first-order logic)
- the ontology language OBO, insofar it has a formal semantics (e.g. through translation to OWL 2)
- OMG SBVR, insofar it has a formal semantics (e.g. through first-order deontic-alethic logic)
- the Common Logic extension IKL
- the meta model of schema.org
- rule languages such as Rulelog, Prova, and Flora-2

6.6.3 Non-monotonic logics and rule languages

Proposals may provide constructs for non-monotonic logics. Support for non-monotonic logics may be provided at the level of individual languages (e.g. negation as failure), as well as at the level of the metalanguage structuring constructs (e.g. circumscription). Note that rule language families like RIF, RuleML, Rulelog, Prova, and Flora-2 include non-monotonic languages.
A particular challenge for the inclusion of non-monotonic logics is the fact that logical entailment cannot be defined in the standard model-theoretic sense (all models of the premises must be models of the conclusion), because this definition always leads to a monotonic entailment. Hence, a suitable meta framework for non-monotonic logics may be provided.

6.6.4 Trade-offs among different translations

A characterization of the trade-offs among different translations may ease the user's choice. For example, a translation from OWL2 to Common Logic can either define all the OWL concepts (like existential restrictions) once and for all, leading to an elegant translation which however uses a quite large sublanguage of Common Logic. Or it can refrain from definition the OWL concepts and instead using an ad-hoc circumscription in each case, leading to a less elegant translation that however ends in a smaller sublanguage of Common Logic.

6.7 Issues to be discussed

Proposals should discuss whether existing language standards need to be extended or adapted in order to make them OntoIOp conformant

If existing language standards need to be extended or adapted in order to make them OntoIOp conformant, this shall be discussed. Preference is that languages can be used as-is without any modifications; however, it may not always possible to meet this goal when creating a unified meta framework.

Proposals should discuss whether the semantics of the metalanguage shall be included into the standard

Proposals definitely shall provide a metalanguage that is equipped with a formal semantics. However, it may be debatable whether this formal semantics (which can be quite long and technical) should be included into the standard, or whether it should be provided as a separate technical document.

Proposals should discuss the chosen list of logics and translations

Section 6.1 lists a number of logics that are in use. In principle, it should be possible to establish OntoIOp conformance for all of these logics. However, it may be the case that only part of these conformances will be actually part of the standard. Moreover, we can refer to the ODM for a number of logics. Proposals should discuss the rationale for the list of logics and logic translations whose conformance they establish

A similar remark holds for logic translations.

Proposals should discuss a meta ontology of logical languages and theories

Proposals should discuss the role of a meta ontology for describing languages, including their semantic and syntactic features, as well as OSMs. They also may provide such a meta ontology. It would be useful to coordinate this meta ontology with related meta ontologies (OMV, API4KB).

Proposals should discuss a concrete syntax for describing registry entries about logical languages

Any proposal shall specify a registry of logical languages (see section 5). Proposals should discuss a concrete syntax for describing logical languages in such registry entries. This syntax should support an extensible description vocabulary, which should be based on the meta ontology mentioned in the previous paragraph. It should be optimized for machine-comprehensibility.

Proposals should discuss the use of QVT for expressing logic translations

Since the syntax of logical languages will be expressed in MOF, it is natural to express translations between logical languages using QVT. Of course, QVT can only cover the syntactic aspect of translations. Proposals should discuss whether they want to use QVT or some other formalism here. Proposals should also discuss how semantic correctness of the translations is ensured.

Proposals should discuss the role of APIs

Proposals may include a set of APIs used to deliver OntoIOp services. An OntoIOp registry and repository could provide information about languages, translations, OSMs, their links, and suitable metadata. It would be useful to coordinate these APIs with related APIs (API4KB, OOR).

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. Place your responses to these Issues in Section 0 of your submission.

6.8 Evaluation Criteria

Proposals covering a broader range of features and of use cases will be favored.

Proposals covering existing language standards without (or with fewer) modifications will be favored.

Proposals establishing actually (or making this at least possible in theory) OntoIOp conformance of more logical languages and translations will be favored. Preference is given to logical languages that have been standardized by standardization bodies such as OMG, ISO or W3C. Recall that the minimum set of covered languages is OWL2 (with profiles EL, RL, QL), CLIF, RDF, UML class diagrams.

6.9 Other information unique to this RFP

6.10 IPR Mode

Every OMG Member that makes any written Submission in response to this RFP shall provide the Non-Assertion Covenant found in Appendix A of the OMG IPR Policy [IPR].

6.11 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.

Event or Activity	Date
Letter of Intent (LOI) deadline	*24 February 2014*
Initial Submission deadline	*18 November 2014*
Voter registration closes	*1 December 2014*
Initial Submission presentations	*15 December 2014*
Revised Submission deadline	*23 May 2015*

Revised Submission presentations	23 June 2015

Appendix A References & Glossary Specific to this RFP

A.1 References Specific to this RFP

[Alignment1] Alignment API and Alignment Server
http://alignapi.gforge.inria.fr/

[Alignment2] Jérôme Euzenat, Pavel Shvaiko: Ontology matching. Springer 2007

[Approx1] Jeff Z. Pan and Edward Thomas. Approximating OWL-DL ontologies. In AAAI, pages 1434–1439, 2007.

[Approx2] Yuan Ren, Jeff Z. Pan, and Yuting Zhao. Soundness preserving approximation for TBox reasoning. In AAAI, 2010.

[Approx3] Carsten Lutz, Inanç Seylan, and Frank Wolter. An automata-theoretic approach to uniform interpolation and approximation in the description logic EL. In Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012, 2012.
http://www.informatik.uni-bremen.de/~clu/papers/archive/KR12b.pdf

[Approx4] B. Selman and H. A. Kautz. Knowledge compilation and theory approximation. J. ACM, 43(2):193–224, 1996.

[Approx5] A. del Val. An analysis of approximate knowledge compilation. In IJCAI'95, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pages 830–836, 1995.

[Approx6] A. del Val. First order LUB approximations: characterization and algorithms. Artificial Intelligence, 162(1-2):7–48, 2005.

[Approx7] Klaus Lüttich: Approximation of Ontologies in CASL. FOIS 2006: 335-346

[B] The B method
http://www.methode-b.com/

[BFO-OWL]
http://bfo.googlecode.com/svn/releases/2012-07-20-graz/owl-group/bfo.owl

[CategoricalManifesto] Joseph A. Goguen: A Categorical Manifesto. Mathematical Structures in Computer Science 1(1): 49-67 (1991)

[CASL] The Common Algebraic Specification Language
http://www.cofi.info

[CL] ISO/IEC 24707:2007, Information technology – Common Logic (CL): a framework for a family of logic-based languages[EBNF] ISO/IEC 14977:1996, Information technology – Syntactic metalanguage – Extended BNF

[Colimits] Mihai Codescu, Till Mossakowski (2008). Heterogeneous colimits. In Frédéric Boulanger, Christophe Gaston, Pierre-Yves Schobbens (Eds.), MoVaH'08 Workshop on Modeling, Validation and Heterogeneity. IEEE press.

[Dolce-lite] *http://www.loa.istc.cnr.it/DOLCE.html*

[FLogic] F-Logic
http://en.wikipedia.org/wiki/F-logic

[FUML] Semantics of a Foundational Subset for Executable UML Models
http://www.omg.org/spec/FUML/

[GrothendieckInst] Razvan Diaconescu: Grothendieck Institutions. Applied Categorical Structures 10(4): 383-402 (2002)

[Heterogeneous1] A. Tarlecki. Towards heterogeneous specifications. In D. Gabbay, M. de Rijke, eds., Frontiers of Combining Systems 2, 1998, Studies in Logic and Computation, 337–360. Research Studies Press, 2000.

[Heterogeneous2] Till Mossakowski (2005). Heterogeneous specification and the heterogeneous tool set. University of Bremen. Habilitation thesis.

[HetUML] María Victoria Cengarle, Alexander Knapp, Andrzej Tarlecki, Martin Wirsing: A Heterogeneous Approach to UML Semantics. In Pierpaolo Degano, Rocco De Nicola, José Meseguer (Eds.): Concurrency, Graphs and Models, Essays Dedicated to Ugo Montanari on the Occasion of His 65th Birthday. Springer 2008 Lecture Notes in Computer Science.
http://www.informatik.uni-trier.de/~ley/db/conf/birthday/montanari2008.html#CengarleKTW08

[HugoRT]
http://www.informatik.uni-augsburg.de/en/chairs/swt/sse/hugort/

[HyperOnto] Oliver Kutz, Till Mossakowski, Dominik Lücke (2010). Carnap, Goguen, and the Hyperontologies – Logical Pluralism and Heterogeneous Structuring in Ontology Design. In Logica Univeralis, 4 (2), pp. 255–333.
http://www.informatik.uni-bremen.de/~till/papers/Hyperontology.pdf

[Institution] Joseph A. Goguen, Rod M. Burstall: Institutions: Abstract Model Theory for Specification and Programming. J. ACM 39(1): 95-146 (1992)

[IRI] IETF/RFC 3987, Internationalized Resource Identifiers (IRIs). January 2005.
http://tools.ietf.org/html/rfc3987

[Logic] Enderton, Herbert B. (1972). *A Mathematical Introduction to Logic* (1 ed.). Academic Press Second edition, 2001

[Modules1] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, Ulrike Sattler: Extracting Modules from Ontologies: A Logic-Based Approach. Modular Ontologies 2009: 159-186

[Modules2] Roman Kontchakov, Frank Wolter, Michael Zakharyaschev: Logic-based ontology comparison and module extraction, with an application to DL-Lite. Artif. Intell. 174(15): 1093-1141 (2010)

[Modules3] Ulrike Sattler, Thomas Schneider, Michael Zakharyaschev: Which Kind of Module Should I Extract? Description Logics 2009

[Modules4] Chiara Del Vescovo, Pavel Klinov, Bijan Parsia, Ulrike Sattler, Thomas Schneider, Dmitry Tsarkov: Empirical Study of Logic-Based Modules: Cheap Is Cheerful. Description Logics 2013: 144-155

[Morphisms] Joseph A. Goguen, Grigore Rosu: Institution Morphisms. Formal Asp. Comput. 13(3-5): 274-307 (2002)

[OBDA] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, Domenico Fabio Savo: The MASTRO system for ontology-based data access. Semantic Web 2(1): 43-53 (2011)

[OWL2-Profiles] W3C/TR REC-owl2-profiles:2009, OWL 2 Web Ontology Language: Profiles. W3C Recommendation, 27 October 2009.
http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/

[OWL2] W3C/TR REC-owl2-syntax:2009, OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. W3C Recommendation, 27 October 2009.
http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/

[Part-Whole] Keet, C. M., and Artale, A. Representing and reasoning over a taxonomy of part–whole relations. Applied Ontology 3, 1 (2008), 91–110.

[RDF-Semantics] W3C/TR REC-rdf-mt:2004, RDF Semantics. W3C Recommendation, 02 February 2004.
http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

[RDF] W3C/TR REC-rdf-concepts:2004, Resource Description Framework (RDF): Concepts and Ab-stract Syntax. W3C Recommendation, 02 February 2004.
http://www.w3.org/TR/2004/REC-rdf-concepts-20040

[RDFS] W3C/TR REC-rdf-schema:2004, RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 February 2004.
http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[RIF] Rule Interchange Format
http://www.w3.org/2005/rules/wiki/RIF_Working_Group

[RuleML] The Rule Markup Initiative
http://ruleml.org/

[Structuring1] Donald Sannella, Andrzej Tarlecki: Specifications in an Arbitrary Institution. Inf. Comput. 76(2/3): 165-210 (1988)

[Structuring2] Donald Sannella, Andrzej Tarlecki: Foundations of Algebraic Specification and Formal Software Development. EATCS Monographs on theoretical computer science, Springer 2012, ISBN 978-3-642-17335-6

[Structuring3] Răzvan Diaconescu, Joseph Goguen, Petros Stefaneas: Logical support for modularisation. Papers presented at the second annual Workshop on Logical environments. Pages 83-130. Cambridge University Press New York, NY, USA 1993.

[TPTP] The TPTP Problem Library for Automated Theorem Proving
http://www.tptp.org

[URI] IETF/RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. January 2005.
http://tools.ietf.org/html/rfc3986

[VDM] The Vienna Development Method
http://www.vdmportal.org

[XML] W3C/TR REC-xml:2008, Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation, 26 November 2008.
http://www.w3.org/TR/2008/REC-xml-20081126/

[Z] The Z notation
http://www.cs.york.ac.uk/hise/Zstandard/

A.2 Glossary Specific to this RFP

Alignment – flexible, relational link that does not always have a formal, logic-based semantics. See [Alignment1-2]

Approximation – reduction of a theory to a less expressive logical language, such that the original theory implies the approximation. See [Approx1-7]

Axiom – sentence postulated to be valid (i.e. true in every model), part of an OSM. See [Logic]

Basic OSM – set of non-logical symbols, sentences, annotations about them, which is used as a building block for a larger OSM. See [Logic]

Conservativity – property of an extension of theories, ensuring that the extension does not add new logical content. See [Logic]

Combination – aggregation of several OSMs along links to a new OSM where (only) the linked non-logical symbols of the involved OSMs are identified. See [CategoricalManifesto], [Colimits]

Heterogeneous OSM – OSM that involves several logical languages (mediated by translations). See [GrothendieckInst], [HyperOnto], [Heterogeneous1-2], [Structuring2]

Interpretation – logical link that postulates a relation between two OSMs. See [Logic]

Language translation – mapping from constructs in the source logical language to their equivalents in the target logical language. See [Morphisms]

Link – relationship between two OSMs, relating their non-logical symbols. Can be either an alignment or an interpretation

Logical theory – set of expressions (like non-logical symbols, sentences and structuring elements) in a given logical language. See [Logic]

Logical language – language that is used for writing down OSMs (e.g. formal ontologies, models and specification), equipped with a formal, declarative, logic-based semantics, plus non-logical annotations. See [Logic], [Institution], [HyperOnto]

Matching – algorithmic procedure that generates an alignment for two given OSMs. See [Alignment1-2]

Model – semantic interpretation of all non-logical symbols of an OSM, satisfying the theory's axioms. See [Logic]

Module – subtheory that conservatively extends to the whole OSM. See [Modules1-4]

Module extraction – activity of obtaining from an OSM concrete modules to be used for a particular purpose (e.g. to contain a particular sub-signature of the original OSM). See [Modules1-4]

Modularity – see *Structured OSM*

Non-logical symbol – atomic expression or syntactic constituent of an OSM that requires an interpretation through a model. See [Logic]

OSM – Ontology, specifcation or model. The logical content is a logical theory.

Satisfaction relation – relation between models and sentences indicating which sentences hold true in the model. See [Logic]

Sentence – term that is either true or false in a given model, i.e. which is assigned a truth value in this model. See [Logic]

Structured OSM – OSM that results from other OSMs by *structuring operations*. See [Structuring1-3]

Structuring operation – Operation for building a *structured OSM*, like import, union, combination, renaming.

Theorem – sentence that has been proven (in some OSM) from other axioms and theorem. See [Logic]

Appendix B General Reference and Glossary

B.1 General References

The following documents are referenced in this document:

[BCQ] OMG Board of Directors Business Committee Questionnaire, *http://doc.omg.org/bcq*

[CCM] CORBA Core Components Specification *http://www.omg.org/spec/CCM/*

[CORBA] Common Object Request Broker Architecture (CORBA)
http://www.omg.org/spec/CORBA/

[CORP] UML Profile for CORBA,
http://www.omg.org/spec/CORP

[CWM] Common Warehouse Metamodel Specification
http://www.omg.org/spec/CWM

[EDOC] UML Profile for EDOC Specification
http://www.omg.org/spec/EDOC/

[Guide] The OMG Hitchhiker's Guide
http://doc.omg.org/hh

[IDL] Interface Definition Language Specification
http://www.omg.org/spec/IDL35

[INVENT] Inventory of Files for a Submission/Revision/Finalization
http://doc.omg.org/inventory

[IPR] IPR Policy
http://doc.omg.org/ipr

[ISO2] ISO/IEC Directives, Part 2 – Rules for the structure and drafting of International Standards
http://isotc.iso.org/livelink/livelink?func=ll&objId=4230456

[LOI] OMG RFP Letter of Intent template
http://doc.omg.org/loi

[MDAa] OMG Architecture Board, "Model Driven Architecture – A Technical Perspective"
http://www.omg.org/mda/papers.htm

[MDAb] Developing in OMG's Model Driven Architecture (MDA)
http://www.omg.org/mda/papers.htm

[MDAc] MDA Guide
http://www.omg.org/docs/omg/03-06-01.pdf

[MDAd] MDA "The Architecture of Choice for a Changing World
http://www.omg.org/mda

[MOF] Meta Object Facility Specification
http://www.omg.org/spec/MOF/

[NS] Naming Service
http://www.omg.org/spec/NAM

[OMA] Object Management Architecture
http://www.omg.org/oma/

[OTS] Transaction Service
http://www.omg.org/spec/OTS

[P&P] Policies and Procedures of the OMG Technical Process
http://doc.omg.org/pp

[RAD] Resource Access Decision Facility
http://www.omg.org/spec/RAD

[ISO2] ISO/IEC Directives, Part 2 – Rules for the structure and drafting of International Standards
http://isotc.iso.org/livelink/livelink?func=ll&objId=4230456

[RM-ODP]
ISO/IEC 10746

[SEC] CORBA Security Service
http://www.omg.org/spec/SEC

[TEMPL] Specification Template
http://doc.omg.org/submission-template

[TOS] Trading Object Service
hptp://www.omg.org/spec/TRADE

[UML] Unified Modeling Language Specification,
http://www.omg.org/spec/UML

[XMI] XML Metadata Interchange Specification,
http://www.omg.org/spec/XMI

B.2 General Glossary

Architecture Board (AB) – The OMG plenary that is responsible for ensuring the technical merit and MDA-compliance of RFPs and their submissions.

Board of Directors (BoD) – The OMG body that is responsible for adopting technology.

Common Object Request Broker Architecture (CORBA) – An OMG distributed computing platform specification that is independent of implementation languages.

Common Warehouse Metamodel (CWM) – An OMG specification for data repository integration.

CORBA Component Model (CCM) – An OMG specification for an implementation language independent distributed component model.

Interface Definition Language (IDL) – An OMG and ISO standard language for specifying interfaces and associated data structures.

Letter of Intent (LOI) – A letter submitted to the OMG BoD's Business Committee signed by an officer of an 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.

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.

Metadata – Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel – A model of models.

Meta Object Facility (MOF) – An OMG standard, closely related to UML, that enables metadata management and language definition.

Model – A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA) – An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Normative – Provisions to which an implementation shall conform to in order to claim compliance with the standard (as opposed to non-normative or informative material, included only to assist in understanding the standard).

Normative Reference – References to documents that contain provisions to which an implementation shall conform to in order to claim compliance with the standard.

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.

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.

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

Request for Information (RFI) – A general request to industry, academia, and any other interested parties to submit information about a particular technology area to one of the OMG's Technology Committee subgroups.

Request for Proposal (RFP) – A document requesting OMG members to submit proposals to an OMG Technology Committee.

Task Force (TF) – The OMG Technology Committee subgroup responsible for issuing a RFP and evaluating submission(s).

Technology Committee (TC) – The body responsible for recommending technologies for adoption to the BoD. There are two TCs in OMG – the *Platform TC* (PTC) focuses on IT and modeling infrastructure related standards; while the *Domain TC* (DTC) focuses on domain specific standards.

Unified Modeling Language (UML) – An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile – A standardized set of extensions and constraints that tailors UML to particular use.

XML Metadata Interchange (XMI) – An OMG standard that facilitates interchange of models via XML documents.