Object Management Group

First Needham Place
250 First Avenue, Suite 201
Needham, MA 02494

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

Request For Proposal

UML 2.0 Superstructure RFP

OMG Document: ad/00-09-02

Submissions due: August 20, 2001

Objective of this RFP

This Request for Proposal (RFP) solicits proposals related to developing the next major revision of the OMG Unified Modeling Language™ Specification, which is commonly referred to as UML™ 2.0. It addresses demonstrated user needs for improved modeling support. Another UML 2.0 RFP is expected to address a major revision to the infrastructure of UML upon which this modeling superstructure rests.

The Unified Modeling Language is a language for visualizing, specifying, constructing and documenting the artifacts of software systems. It is a general-purpose modeling language that can be used with all major object and component methods and applied to all application domains. The OMG adopted the UML 1.1 specification in November 1997. Since then UML Revision Task Forces have produced several minor revisions, the most recent being the UML 1.4 specification, which is scheduled to be adopted sometime in 2000.

Under the stewardship of the OMG, the UML has emerged as the software industry's dominant modeling language. It has been

successfully applied to a wide range of domains, ranging from health and finance to aerospace to e-commerce. As should be expected, its extensive use has raised numerous application and implementation issues by modelers and vendors. As of the time of this writing over 500 formal usage and implementation issues have been submitted to the OMG for consideration.

Although many of the issues have been resolved in minor revisions by Revision Task Forces, other issues require major changes to the language that are outside the scope of an RTF. The need for a major UML revision has been further substantiated by vendor and user feedback to the Object Analysis and Design Platform Task Force (AD PTF) UML architectural roadmap and its UML 2.0 Request for Information. Consequently, it is clear that there is widespread support for a major revision to UML that will address substantive usage and implementation issues.

This RFP solicits proposals for the following:

- Enable the modeling of structural patterns, such as component-based development and the specification of run-time architectures.
- Clarify the semantics of the generalization, dependency, and association relationships.
- Support encapsulation and scalability in behavioral modeling, in particular, for state machines and interactions.
- Remove restrictions on activity graph modeling due to the mapping to state machines.

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 a membership of over 800 vendors, developers, and end users. Established in 1989, its mission is to promote the theory and practice of Object Technology (OT) for the development of distributed computing systems.

A key goal of OMG is create a standardized object-oriented architectural framework for distributed applications based on specifications that enable and support distributed objects. Objectives include the *reusability*, *portability*, and *interoperability* of object-oriented software components in heterogeneous environments. To this end, the OMG adopts interface and protocol specifications, based on commercially available object technology, that together define an Object Management Architecture (OMA).

1.2 Organization of this document

The remainder of this document is organized as follows:

Chapter 2 - *Architectural Context* - background information on OMG's Object Management 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, mandatory and optional requirements, issues to be discussed, evaluation criteria, and timetable that apply specifically to this RFP.

Additional RFP-specific chapters may also be included following Chapter 6.

1.3 References

The following documents are referenced in this document:

Richard Soley (ed.), *Object Management Architecture Guide,* Third Edition, Wiley, June 1995. OMG Document ab/97-05-05, or successor.

The Common Object Request Broker: Architecture and Specification, Revision 2.1, August 1997. OMG Document formal/97-09-01, or successor.

CORBAservices: Common Object Services Specification, Revised Edition, July 1997. OMG Document formal/97-07-04, or successor.

CORBAfacilities Architecture, Revision 4.0, November 1995.

Business Committee RFP Attachment, OMG Document omg/97-10-01.

Policies and Procedures of the OMG Technical Process, OMG Document pp/97-06-01 or successor.

These documents can be obtained by contacting OMG at document@omg.org. Many OMG documents, including this document, are available electronically from OMG's document server. Send a message containing the single line "help" to server@omg.org for more information, or visit the OMG Web page (URL http://www.omg.org/), which also has more information about OMG in general. If you have general questions about this RFP send email to responses@omg.org.

2.0 Architectural Context

2.1 Object Management Architecture

The *Object Management Architecture Guide* (OMAG) describes OMG's technical objectives and terminology and provides the conceptual infrastructure upon which supporting specifications are based. The guide includes the *OMG Object Model,* which defines common semantics for specifying the externally visible characteristics of objects in a standard implementation-independent way, and the *OMA Reference Model.*

The Reference Model identifies and characterizes the components, interfaces, and protocols that compose the OMA. This includes the Object Request Broker (ORB) component that enables clients and objects to communicate in a distributed environment, and four categories of object interfaces:

- *Object Services* are interfaces for general services that are likely to be used in any program based on distributed objects.
- *Common Facilities* are interfaces for horizontal end-user-oriented facilities applicable to most application domains.
- *Domain Interfaces* are application domain-specific interfaces.
- *Application Interfaces* are non-standardized application-specific interfaces.

A second part of the Reference Model introduces the notion of domain-specific *Object Frameworks.* An Object Framework component is a collection of cooperating objects that provide an integrated solution within an application or technology domain and which is intended for customization by the developer or user.

Through a series of RFPs, OMG is populating the OMA with detailed specifications for each component and interface category in the Reference Model. Adopted specifications include the Common Object Request Broker Architecture (CORBA), CORBAservices, and CORBAfacilities.

The wide-scale industry adoption of OMG's OMA provides application developers and users with the means to build interoperable software

systems distributed across all major hardware, operating system, and programming language environments.

2.2 CORBA

The *Common Object Request Broker Architecture* defines the programming interfaces to the OMA ORB component. An ORB is the basic mechanism by which objects transparently make requests to - and receive responses from - each other on the same machine or across a network. A client need not be aware of the mechanisms used to communicate with or activate an object, how the object is implemented, nor where the object is located. The ORB thus forms the foundation for building applications constructed from distributed objects and for interoperability between applications in both homogeneous and heterogeneous environments.

The *OMG Interface Definition Language* (IDL) provides a standardized way to define the interfaces to CORBA objects. The IDL definition is the contract between the implementor of an object and the client. IDL is a strongly typed declarative language that is programming language-independent. Language mappings enable objects to be implemented and sent requests in the developer's programming language of choice in a style that is natural to that language.

CORBA 2.0 is an extension and restructuring of the earlier CORBA 1.2 specification. CORBA 2.0 is a family of specifications consisting of the following components:

- Core (including IDL syntax and semantics)
- Interoperability
- An expanding set of language mappings, including:

  C
  C++
  SmallTalk
  Ada95
  COBOL

Each component is a separate compliance point. The minimum required for a CORBA-compliant implementation is adherence to the core and one language mapping.

2.3 CORBA/Interoperability

Interoperability between CORBA-compliant ORBs is provided by OMG's *Internet Inter-ORB Protocol* (IIOP). Adopted in December 1994 as the mandatory CORBA 2.0 protocol for “out of the box” interoperability, IIOP is the TCP/IP transport mapping of a *General Inter-ORB Protocol* (GIOP). IIOP enables requests to be sent to networked objects managed by other ORBs in other domains.

The OMG interoperability architecture also accommodates communication using optional *Environment-Specific IOPs* (ESIOPs), the first of which is the DCE-CIOP.

2.4 CORBAservices

Object Services are general-purpose services that are either fundamental for developing useful CORBA-based applications composed of distributed objects, or that provide a universal - application domain-independent - basis for application interoperability.

Object Services are the basic building blocks for distributed object applications. Compliant objects can be combined in many different ways and put to many different uses in applications. They can be used to construct higher-level facilities and object frameworks that can interoperate across multiple platform environments.

Adopted OMG Object Services are collectively called CORBAservices and include Naming, Events, Lifecycle, Persistent Object, Relationships, Externalization, Transactions, Concurrency Control, Licensing, Query, Properties, Security, Time, Collections, and Trading Services.

2.5 CORBAfacilities

Common Facilities are interfaces for horizontal end-user-oriented facilities applicable to most domains. Adopted OMG Common Facilities are collectively called CORBAfacilities and include an OpenDoc-based Distributed Document Component Facility.

A specification of a Common Facility or Object Service typically includes the set of interface definitions - expressed in OMG IDL - that objects in various roles must support in order to *provide*, *use,* or *participate in* the facility or service. As with all specifications adopted by OMG, facilities and services are defined in terms of interfaces and their semantics, and not a particular implementation.

2.6 Object Frameworks and Domain Interfaces

Unlike the interfaces to individual parts of the OMA "plumbing" infrastructure, Object Frameworks are complete higher-level components that provide functionality of direct interest to end-users in particular application or technology domains. They are vertical slices down the OMG "interface stack".

Object Frameworks are collections of cooperating objects categorized into *Application, Domain, Facility,* and *Service Objects*. Each object in a framework supports (through interface inheritance) or makes use of (via client requests) some combination of Application, Domain, CORBAfacilities, and CORBAservices *interfaces*.

A specification of an Object Framework defines such things as the structure, interfaces, types, operation sequencing, and qualities of service of the objects that make up the framework. This includes requirements on implementations in order to guarantee application portability and interoperability across different platforms.

Domain Task Force RFPs are likely to focus on Object Framework specifications that include new Domain Interfaces for application domains such as Finance, Healthcare, Manufacturing, Telecom, Electronic Commerce, and Transportation.

3.0 Adoption Process

3.1 Introduction

OMG adopts specifications for interfaces and protocols by explicit vote on a technology-by-technology basis. The specifications selected each fill in a portion of the OMA Reference Model. 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.

For more detailed information on the adoption process see the *Policies and Procedures of the OMG Technical Process*.

3.2 Role of Board of Directors

The OMG Board of Directors votes to formally adopt specifications on behalf of OMG. The OMG Technology Committees (Domain and Platform TCs) and Architecture Board (AB) provide technical guidance to the Board of Directors. In addition, the Business Committee of the Board provides guidance to ensure that implementations of adopted specifications are made commercially available.

3.3 Role of Technology Committees and Architecture Board

Submissions to RFPs are evaluated by the TC Task Force (TF) that initiated the RFP. Selected specifications are recommended to the parent TC after being reviewed by the Architecture Board for consistency with the OMA. The full TC then votes to *recommend adoption* to the OMG Board.

3.4 Role of Task Forces

The role of the initiating TF is to technically evaluate submissions and select one or more specifications that satisfy the requirements of the RFP. The process typically takes the following form:

- Voter Registration

  Interested TF members may register to participate in specification selection votes for an 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. Companies who 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 next following meeting of the TF. Initial submissions are expected to be full and complete proposals and working implementations of the proposed specifications are expected to exist at the time of submission.

- Evaluation Phase

  A period of approximately 120 days follows during which the TF evaluates the submissions. During this time submitting companies have the opportunity to revise and / or merge their initial submissions, if they so choose.

- Revised Submissions

  Final revised submissions are due by a specified deadline. Submitters again normally present their proposals at the next following meeting of the TF. Finalists may be requested to demonstrate implementations of their proposal.

- Selection Vote

  When the registered voters of the TF believe that they sufficiently understand the relative merits of the revised submissions, a specification selection vote is taken.

3.5 Goals of the evaluation

The primary goals of the TF evaluation process are to:

- Provide a fair and open process
- Force a critical review of the submissions and discussion by all members of the TF
- Give feedback to allow submitters to address concerns in their revised submissions
- Build consensus on acceptable solutions
- Enable voting members to make an informed selection decision

Submitters are expected actively to contribute to the evaluation process.

4.0 Instructions for Submitters

4.1 OMG Membership

Submissions to this RFP may only be made by Platform, Domain or Contributing members of the OMG. To submit to an RFP issued by the Platform Technology Committee an organization must be a Platform or Contributing member at 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 concerned.

4.2 Submission Effort

Unlike a submission to an OMG Request For Information (RFI), an RFP submission may require significant effort in terms of document preparation, presentations to the initiating 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 your organization signifying your intent to respond to the RFP and confirming your 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 your organization for receipt of all subsequent information regarding this RFP and your 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/98-03-01.

<____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, available separately as document omg/98-03-01, was approved by the OMG Board in February 1998.

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 Committee; 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 organizations. 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.

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 organizations 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 of such specification, unless OMG believes that such IPR owner will grant a license to implementers (whether OMG members or not) on non-discriminatory and commercially reasonable terms which wish to implement 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 implementation "as is". If the submitter owns IPR to which an implementation 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 implementer on non-discriminatory and commercially reasonable terms, to permit development and commercialization of an implementation that includes such IPR.

It is the goal of the OMG to make all of its specifications 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 license 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 implementation of the specification.

A2.4 Publication of the specification

Should the submission be adopted, the submitter must grant OMG (and its sublicensees) a worldwide, royalty-free license 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 Separate proposals

Unless otherwise indicated in Chapter 6, independent proposals are solicited for each separate item in the RFP. Each item is considered a separate architectural entity for which a proposal may be made. A submitter may respond to any or all items. Each item will be evaluated independently by the initiating TF. Submissions that do not present clearly separable proposals for multiple items may therefore be at a disadvantage.

It should be noted that a given technology (e.g. software product) may support two or more RFP items. So long as the interfaces for each item are separable, this is not precluded.

4.5.2 Complete proposals

Proposals for each separate RFP item must be complete. A submission must propose full specifications for each item and address all the relevant general and specific requirements detailed in this RFP.

4.5.3 Additional specifications

Submissions may include additional specifications for items not covered by the RFP which 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.

4.5.4 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 within the OMA 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 confidentiality or proprietary information of any kind will be accepted in a submission to this RFP.

4.7 Copyright Waiver

If a submitted document is copyrighted, a waiver of copyright for unlimited duplication by the OMG is required to be stated in the document. In addition, a limited waiver of copyright is required that allows each OMG member to make up to fifty (50) copies of the document for review purposes only.

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 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 to the satisfaction of the TF the technical viability of their proposal. OMG will favor proposals based on technology for which sufficient relevant experience has been gained in CORBA-based or comparable environments.

4.9 Format of RFP Submissions

This section provides guidance on how to structure your RFP submission.

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 models and terminology in the *Object Management Architecture Guide* and *CORBA* should be used in your submission. Where you believe this is not appropriate, describe and provide a rationale for the models and terminology you believe OMG should use. Submitters are encouraged to document their object models and designs using OMG UML where appropriate, and to supply an OMG XMI representation of the design (including a machine-readable copy) for the

convenience of those wishing to import the UML model into design tools.

4.9.2 Suggested Outline

A three part structure for submissions is suggested:

PART I

- Copyright Waiver (see 4.5)
- Submission contact point (see 4.2)
- Overview or guide to the material in the submission
- Overall design rationale (if appropriate)
- Statement of proof of concept (see 4.6)
- Resolution of RFP mandatory and optional requirements

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

  In addition, if your 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

- Proposed specification

PART III

- Summary of optional versus mandatory interfaces

  Submissions must clearly distinguish interfaces that all implementations must support from those that may be optionally supported.

- Proposed compliance points

  Submissions should propose appropriate compliance points for implementations.

- Changes or extensions required to adopted 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.

- Complete IDL definitions

  For reference purposes and to facilitate electronic usage, submissions should reproduce in one place a complete listing in compilable form of the IDL definitions proposed for standardization.

4.10 How to Submit

Submitters should send an electronic version of their submission to the *RFP Submissions Desk* (rfp@omg.org) at OMG by 5:00 PM U.S. Eastern Standard Time (22:00 GMT) on the day of the submission deadline. Acceptable formats are Postscript, ASCII, PDF, FrameMaker, Word, and WordPerfect. However, it should be noted that a successful submission must be supplied to OMG's technical editors in Framemaker source format, using the most recent available OMG submission template (document ab/97-06-02 at the time of writing). The AB will not endorse adoption of any submission for which appropriately-formatted Framemaker sources are not available; it may therefore be convenient to prepare all stages of a submission using this template.

Submitters should make sure they receive electronic or voice confirmation of the successful receipt of their submission. Submitters should also send, within three (3) working days after the submission deadline, a single hardcopy version of their submission 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 Mandatory Requirements

5.1.1 Proposals shall express interfaces in OMG IDL. Proposals should follow accepted OMG IDL and CORBA programming style. The correctness of the IDL shall be verified using at least one IDL compiler (and preferably more then one). In addition to IDL quoted in the text of the submission, all the IDL associated with the proposal shall be supplied to OMG in compiler-readable form.

5.1.2 Proposals shall specify *operation behavior*, *sequencing*, and *side-effects* (if any).

5.1.3 Proposals shall be *precise* and *functionally complete*. There should be no implied or hidden interfaces, operations, or functions required to enable an implementation of the proposed specification.

5.1.4 Proposals shall clearly distinguish *mandatory* interfaces and other specification elements that all implementations must support from those that may be *optionally* supported.

5.1.5 Proposals shall *reuse* existing OMG specifications including CORBA, CORBAservices, and CORBAfacilities in preference to defining new interfaces to perform similar functions.

5.1.6 Proposals shall justify and fully specify any *changes or extensions* required to existing OMG specifications. This includes changes and extensions to CORBA inter-ORB protocols necessary to support interoperability. In general, OMG favors *upwards compatible* proposals that minimize changes and extensions to existing OMG specifications.

5.1.7 Proposals shall factor out functions that could be used in different contexts and specify their interfaces separately. Such *minimality* fosters re-use and avoids functional duplication.

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

5.1.9 Proposals shall specify interfaces that are *compatible* and can be used with existing OMG specifications. Separate functions doing separate jobs should be capable of being used together where it makes sense for them to do so.

5.1.10 Proposals shall preserve maximum *implementation flexibility*. Implementation descriptions should not be included, however proposals may specify constraints on object behavior that implementations need to take into account over and above those defined by the interface semantics.

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 / IEC 10746, Reference Model of Open Distributed Processing (ODP). Where such compatibility is not achieved, 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 service or facility 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 objects that are introduced by the proposal?
- Which accesses to security-sensitive objects must be subject to security policy control?
- Does the proposed service or facility need to be security aware?
- What CORBAsecurity level and options are required to protect an implementation of the proposal? In answer to this question, a reasonably complete description of how the facilities provided by the level and options (e.g. authentication, audit, authorization, message protection etc.) are used to protect access to the sensitive objects introduced by the proposal shall be provided.
- What default policies should be applied to the security sensitive objects introduced by the proposal?

- Of what security considerations must the implementers of your proposal be aware?

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.

5.2 Evaluation criteria

Although the OMG adopts interface 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 ORB 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 Compliance: Inspectability and Testability

The adequacy of proposed specifications for the purposes of compliance inspection and testing will be considered. Specifications should provide sufficient constraints on interfaces and implementation characteristics to ensure that compliance 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 representations will be considered, since this allows specifications to be easily interchanged between XMI compliant tools and applications. Since use of XML (including XMI, 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

Under the stewardship of the OMG, the UML has emerged as the software industry's dominant modeling language. It has been successfully applied to a wide range of domains, ranging from health and finance to aerospace to e-commerce. As should be expected, this extensive use has raised various application and implementation issues by modelers and vendors. As of the time of this writing over 500 formal usage and implementation issues have been submitted to the OMG for consideration.

Although many of the issues have been resolved in minor revisions by Revision Task Forces, other issues require major changes to the language that are outside the scope of an RTF. The need for a major UML revision has been further substantiated by vendor and user feedback to the AD PTF UML architectural roadmap (see [1]) and the UML 2.0 Request for Information (see [2]). Consequently, it is clear that there is widespread support for a major revision to UML to address substantive usage and implementation issues.

The requirements for the UML 2.0 major revision are divided into three categories:

- **General requirements**: These requirements apply to all major revisions, and consequently apply to both infrastructure and superstructure changes.
- **Infrastructure requirements**: These requirements are primarily concerned with architectural alignment and restructuring, and address how UML 2.0 will be defined and structured as a metamodel.
- **Superstructure requirements**: These requirements are primarily focused on the refinement and extension of UML 1.x semantics and notation.

Proposals to address these requirements are solicited in related Request for Proposals: one focused on the infrastructure requirements, and one or more focused on the superstructure requirements. These RFPs may be issued concurrently, but the infrastructure RFP is expected to be completed in advance of the superstructure RFP (or RFPs).

Submitters should note that is likely there will be some overlap between infrastructure and superstructure requirements, as when a refinement of existing semantics and notation requires an infrastructure change.

The separation of infrastructure requirements from superstructure requirements is intended to help submitters understand the architectural dependencies between the two. This understanding should help submitters maintain architectural integrity when they divide the specification work and collaborate in parallel.

The most significant issues concerning the superstructure of UML that proposals should address are summarized in the following subsections.

6.1.1 Structural Modeling

The UML supports modeling of patterns of interaction, that is, signal and operation invocation between roles. This allows classifiers to be defined independently of how they participate in any particular collaboration. The UML does not support the same for structural modeling.

6.1.1.1 Component-based Development

Although the UML contains basic constructs for component-based development, it was not designed to fully support component-based methods and architectural frameworks, such as Enterprise JavaBeans, CORBA Component Model and COM+.

Issues associated with modeling components include, but are not limited to, the following:

- Modelers cannot adequately specify components to be plug-substitutable. The current notion of Interface is too weak to capture the full semantics of interactions between components: in addition to simple message calls and receptions, it must include outputs and the definition of complex transactions. Modelers need to specify the requirements a component places on its environment, in addition to the services the component offers to its environment.
- Modelers find it difficult to specify component architectural frameworks and component application frameworks.

6.1.1.2 Run-time Architectures

Modelers commonly agree that the architecture of systems is best described by the hierarchical decomposition of its internal structure (see [1], [3]). This internal structure is one of layered or interconnected instances (see [4]). It describes the organization of the parts of a system through which they interact to carry out the functionality of the whole. For the modeling of a system it is important to support its specification in terms of its parts, how these parts are encapsulated, how these parts are connected, and what communication between the parts is possible.

- Currently, profiles solve this need through extensive imposition of additional constraints on the general semantics. However, the ability to model architectures is a common requirement for most software domains and, consequently, should be part of the core modeling capabilities of UML rather than being limited to a profile.

- Further, the way that objects and other data flow between parts of a system is crucial to understanding its architecure. The UML currently supports object / data flow only at the lowest level of granularity, between the steps in an activity graph. It is important for architects to be able to model object and data flows between entities at a higher level of granularity, such as classifiers and packages.

6.1.2 Relationships

Issues encountered in modeling relationships include, but are not limited to the following.

- It is not clear how to specify the inheritance mechanisms of various implementation languages using the generalization relationship. Further, the UML does not describe in detail what elements are inherited for each model element (most strikingly, StateMachines cannot be generalized).

- The «refine» and «trace» dependencies lack detailed semantics and usage guidelines.

- Composite aggregations cannot limit associations between aggregated parts to the context of the whole without constraining how instances of the aggregated classes are associated or attributed in other contexts.

6.1.3 Behavioral Modeling

Behavioral modeling has been hampered by a lack of support for encapsulation and scalability. In addition, the mapping of activity graphs to state machines has imposed undesired restrictions on activity graph modeling.

6.1.3.1 State machines

It is common for a state machine to contain large numbers of substates and transitions. State machines provide composite states for decomposition and can hierarchically structure their states to an arbitrary depth. However, the complexity of state machines cannot always be fully captured by hierarchical composition. There may be groups of states with identical behavior but where the same state participates in more than one such group.

6.1.3.2 Activity Graphs

UML defines an activity graph as a "special case of a state machine that defines a computational process in terms of the control-flow and object-flow among its constituent actions." This has led to much confusion and to many undesired restrictions on activity diagrams due to the mapping to state machines. For example, events are discarded after they are used once, so that multiple reactions to an event over time cannot be modeled. Also, it is not possible to model flows that do not return to the originating line of control, unnecessarily limiting the kinds of flows that can be supported. It also has made the notation difficult to understand in terms of the underlying semantics.

6.1.3.3 Interactions

Experience has shown that to maintain a large set of sequence diagrams is difficult. The current sequence diagram notation offers little help to structure specifications using sequence diagrams nor does it provide an overview of how these sequence diagrams are related to each other. Issues encountered include, but are not limited to the following:

- Modelers cannot decompose the complex behavior in sequence diagrams. This may entail both a decomposition of the roles involved in an interaction as well as a decomposition of the sequence itself.

- Modelers cannot reference one sequence diagram from another.

- Modelers cannot combine sequence diagrams by means other than sequential composition (e.g., parallel, optional, repetition).
- Modelers have limited ability to define variability within a single sequence diagram.

These issues also apply to collaborations.

6.1.4 Notation

The UML notation has a number of inconsistencies and shortcomings. These include, but are not limited to the following:

- The different diagrams are applied inconsistently: Some diagrams define the content of one specific element (e.g., a static structure diagram defines the namespace content of a package), other diagrams define an element and its properties (e.g., a statechart defines a StateMachine), still other diagrams define different views without any corresponding model element (e.g., a deployment diagram).
- The overloading of keywords for both kernel constructs (e.g., «model», «subsystem») and stereotypes (e.g., «framework», «file») is perplexing to many modelers.
- Some concepts have inconsistent representations in different diagrams. For example, composite aggregation for classes has the same notation as namespace containment for packages (consequentially, the static structure diagram cannot be used for specifying the namespace content of a class).

6.2 Scope of Proposals Sought

This RFP solicits proposals for a major revision of the UML that will address the various issues described in Section 6.1 and will satisfy the requirements described in Section 6.5. Although respondents have considerable latitude to propose improvements to the specification, they should keep the following things in mind:

- Any proposed changes should either maintain or improve the rigor and the integrity of the current specification.
- Any new concept that is introduced should take care to minimize semantic and notational overlap with existing concepts.
- Since UML has a large installed user base all proposed changes should consider backward incompatibility issues.

- Any proposed changes should consider the pragmatics of usage and implementation within a reasonable time frame.
- The reasons for choosing not to fulfill any mandatory requirement should be fully documented.

6.3 Relationship to Existing OMG Specifications

The UML 2.0 is a major revision to the UML 1.x version series, which includes OMG UML 1.1 and all of its subsequent minor revisions. In general, proposals should be consistent with, and use the terminology of the most current UML 1.x specification at the time of submission. If there is reason to deviate from UML 1.x terminology in order to make a major revision that reason should be clearly explained. Submitters are strongly encouraged to consider backward-compatibility issues when recommending major revisions; gratuitous changes to the current UML specification are strongly discouraged.

The UML 2.0 must be compliant with the most current Meta-Object Facility Specification (currently 1.3, OMG document formal/2000-04-01) at the time of the submission. Proposals for UML 2.0 may suggest revisions to the Meta Object Facility, but they should try to minimize the impact on existing MOF usage.

The UML 2.0 shall consider reconciling with the action semantics proposal under development (see the RFP, OMG document ad/98-11-01).

The UML 2.0 shall take into account existing profiles for components, if any (see, for example, the UML Profile for Enterprise Distributed Object Computing RFP, OMG document ad/99-03-10).

The UML 2.0 must be complementary to UML-related adopted technologies such as XMI (currently 1.1, OMG document formal/2000-06-01). Therefore the vocabulary and underlying models of these adopted technologies must be used whenever possible. Restrictions and extensions to these technologies must be called out explicitly.

6.4 Related Documents and Standards

[1] Analysis and Design PTF UML 2.0 Roadmap Recommendation, OMG document ad/00-06-01.
[2] UML 2.0 RFI Overview, OMG document ad/00-01-07.
[3] D. Garlan, R. Monroe, D. Wile, *Acme Reference Guide*. Available from http://www.cs.cmu.edu/~acme

[4] D. Garlan, J. Knapman, B. Møller-Pedersen, B. Selic, and T. Weigert, "Modelling Architectures with UML," Proc. 3rd Intl. Conf. on the UML, Springer, 2000.
[5] N. Medvidocic and R. Taylor, A Framework for Classifying and Comparing Architecture Description Languages, Proc. 6th European Software Engineering Conference, 1997.
[6] Open Distributed Processing -- Reference Model -- Foundations. ITU-T Recommendation X.902 (1995) and ISO / IEC 10746-2:1996.
[7] Open Distributed Processing - Reference Model – Architecture. ITU-T Recommendation X.903 (1995) | ISO / IEC 10746-3:1996.
[8] ISO 10303-11 Information Modelling Language EXPRESS, 1994.

6.5 Mandatory Requirements

6.5.1 General Requirements

- Proposals shall enforce a clear separation of concerns between the specification of the metamodel semantics and notation, including precise bi-directional mappings between them.

- Proposals shall minimize the impact on users of the current UML 1.x, XMI 1.x and MOF 1.x specifications, and shall provide a precise mapping between the current UML 1.x and the UML 2.0 metamodels. Proposals shall ensure that there is a well-defined upgrade path from the XMI DTD for UML 1.x to the XMI DTD for UML 2.0. Wherever changes have adversely impacted backward compatibility with previous specifications, submissions shall provide rationales and change summaries along with their precise mappings.

- Proposals shall identify language elements to be retired from the language for reasons such as being vague, gratuitous, too specific, or not used.

- Proposals shall specify an XMI DTD for the UML metamodel.

6.5.2 Structural Modeling

Proposals shall refine the existing structural modeling capabilities of the UML to support the following.

6.5.2.1 Component-Based Development

- Proposals shall support component assembly and plug-substitutability by providing for the specification of both what a component makes available to other components and its connection requirements (e.g., which operations and signals it will require from its connected components).

- Proposals shall support the specification of common interaction patterns that might occur between two or more components as reusable and generalizable modeling elements.

- Proposals shall support modeling of component execution contexts (e.g., EJB and CCM containers), and communication between an execution context and the components that it contains.

- Proposals shall enable definition of profiles for models based on, at least, the following component architectures: EJB, CORBA components, and COM+.

6.5.2.2 Run-Time Architectures

- Proposals shall support the modeling of the internal structureof a classifier in terms of its hierarchical decomposition. The internal structure shall be allowed to contain instances of classifiers and links between these instances, without affecting the usage of these classifiers elsewhere. The connections between instances shall, at a minimum, specify possible communication.

- Proposals shall support the specification of the dynamic behavior of the internal structure of a classifier, including its connection to the statemachine of the classifier, if any, its initial instantiation, as well as the dynamic addition and removal of parts and connections to / from the internal structure.

6.5.3 Relationships

- Proposals shall specify how the features and behavior of all generalizable model elements are affected by specialization. They shall also address the replacement of features and behavior that are specialized from an ancestor.

- Proposals shall specify what «refine» and «trace» mean and provide usage guidelines.

- Proposals shall specify the scope that is covered by an association. It shall be possible that associations or initial attribute values specified within a scope are valid only within the context of that scope. In particular, proposals shall clarify the semantics of composite aggregation with respect to scope.

6.5.4 Behavioral modeling

6.5.4.1 State machines

- Proposals shall provide for an encapsulation mechanism for states and state machines, so that the internal details of a composite state can be defined

independently of its use in the enclosing state. Proposals shall support reuse of behavioral specifications across multiple classes.

- Proposals shall clarify the semantics of protocol state machines.
- Proposals shall clarify the application of state machines to Behavioral Features and to Classifiers other than Classes.
- Proposals shall specify how StateMachines can be specialized.

6.5.4.2 Activity Graphs

- Proposals shall provide for improved control and data flow modeling in activity graphs. For example, more permissive concurrency or separate semantics for control and data flow.
- Proposals shall improve the management of events in activity graphs. Examples are keeping and referring to event history or Boolean combination of events in triggers.

6.5.4.3 Interactions

- Proposals shall define mechanisms to describe the decomposition of a role in an interaction into an interaction of its constituent parts.
- Proposals shall provide mechanisms to refer from one interaction to other interactions to support composition of interactions. It shall be possible to define, at least, sequences of interactions, alternative interactions based on conditions, parallel execution of interactions, and repetition of an interaction.

6.5.5 Notation

No mandatory requirements.

6.5.6 Other

No mandatory requirements.

6.6 Optional Requirements

6.6.1 General Requirements

No optional requirements.

6.6.2 Structural Modeling

- Proposals may provide for data flow modeling at a high level of abstraction. For example, it may be possible to show data or object flow between packages and classifiers.

6.6.3 Relationships

No optional requirements.

6.6.4 Behavioral modeling

- Proposals may support the grouping of states into possibly overlapping sets of states, such that it is possible to share behavior across such sets.
- Proposals may support specification of the particular events that an instance can receive and from which objects it can receive them.
- Proposals may provide for the capability to establish the target object of a communication by various means with differing levels of coupling between the communicating objects.

6.6.5 Notation

- Proposals may review and improve the consistency of how symbols and icons are used in the various kinds of diagrams.
- Proposals may provide an improved notation for defining patterns.
- Proposals may define notation for applying constraints on the instantiation of templates (e.g., constraints on template parameters that are checked at instantiation time).

6.6.6 Other

- Proposals may consider semantic alignment with other specification language standards, such as ISO EXPRESS, ITU-T SDL / MSC, ISO GRM, or ITU-T RM-ODP.

6.7 Issues to be discussed

Proposals should stipulate the mechanisms by which compliance to the specification will be determined, recognizing that determination of conformance is on a subset (of the language specification) basis and that

not all parts of a package are always needed. For example, proposals might submit XMI DTDs to test the compliance of a tool to the specification in a respective subset area.

Proposals should discuss the impact of any changes to the UML metamodel on adopted profiles.

6.8 Evaluation Criteria

Proposals shall be evaluated on the effectiveness with which they address the requirements put forward in this Request for Proposal.

Any changes to the MOF required to achieve the goals stated in this Request for Proposals should be clearly specified.

Proposals that unify mechanisms will be preferred over proposals that introduce several, but similar mechanisms. This includes notational proposals. For example,

- A solution that supports both the modeling of large and complex systems and the needs of component-based development will be preferred. It is desirable for any solution to integrate well into existing concepts of classes and subsystems.
- Simple proposals that unify generalization mechanisms are preferred over those that treat different model elements differently. E.g., it is preferable if attributes and associations are treated in a consistent manner with respect to generalization.

Proposals that unify existing concepts are preferable. For example, proposals may seek to merge the notion of instantiable subsystems and class composition. Instantiable subsystems have many of the features of (singleton) classes, while classes as part of Composition have other properties that are desired for the specification of system architectures.

Proposals describing mechanisms at an abstract level are preferable over concrete or implementation-oriented mechanisms. For example, a proposal that suggests a general mechanism of method dispatch is preferable over one, which determines precisely which method will be invoked in situations where superseding and replacement of an inherited method had taken place. The more concrete mechanism should be selectable by establishing a profile.

Proposals that allow notations to be applied to several model elements that have identical semantics are preferable over proposals that introduce different notation, or do not provide for a notation for all such model elements. For example, notation that is currently defined only for

activity diagrams but which could apply equally well to statecharts should be available for statecharts as well. (This includes notations for actions states, decisions, control icons for signal sending, signal receipt, and deferred events, omitted synch state notation, and conditional forks.)

Any semantics proposed for model elements with behavior must be unambiguous regarding execution.

Proposals that provide simple mappings between the Notation Guide and the Semantics are preferred. Proposals should avoid mappings that impose undesirable restrictions on a modeling concept. Overly complicated mappings will also interfere with model interchange.

Proposals that address requirements using concepts or notations already deployed in the UML are preferred over proposals introducing new concepts or notations. For example, a notation for structuring of sequence diagrams that reuses the notation for activity diagrams is preferred over a proposal introducing a new and different notation.

For new mechanisms introduced in responses to this RFP, the notation should be consistent with the existing notation (modulo any improvements made to the existing notation). Concepts in the Semantics should be supported by an accompanying notation.

Proposals shall demonstrate that proposed solutions solve common needs of UML users. For example, proposals addressing component-based development shall demonstrate that they enable definition of profiles for models based on, at least, the following component architectures: EJB, CORBA components, and COM+. They should show that they specify logical (analysis), physical (design), and implementation models for each of these component architectures.

Every modification to structural aspects of the UML should be clearly accounted for in their ramification on behavior. For example, changes to composite aggregation should be accompanied by an explanation of the effect of creating and destroying composite objects, and adding and removing parts of a composite object.

Precise mappings for backward compatibility should be accompanied by example model translations.

Proposals that maximize the degree of internationalization supported will be preferred over proposals that rely on customs of particular regions.

6.9 Other information unique to this RFP

There is no other information requested that is considered unique to this RFP.

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF 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 in the Member Services section of OMG's Web page (URL http://www.omg.org/)

Approx Day	Event or Activity	Actual Date
	Preparation of RFP by TF	
	Approval of RFP by Architecture Board *Review by TC ("Three week rule")*	
0	*TC votes to issue RFP*	*Sep 13, 2000*
60	*LOI to submit to RFP due*	*Nov 12, 2000*
341	*Initial submissions due*	*Aug 20, 2001*
355	*Voter registration closes*	*Sep 3, 2001*
364	*Initial submission presentations*	*Sep 12, 2001*
	Preliminary evaluation by TF	
504	*Revised submissions due*	*Feb 1, 2002*
527	*Revised submission presentations*	*Feb 22, 2002*
	Final evaluation and selection by TF *Recommendation to AB and TC*	
	Approval by Architecture Board *Review by TC ("Three week rule")*	
587	*TC votes to recommend specifications*	*Apr 23, 2002*
617	*BOD votes to adopt specifications*	*May 23, 2002*