

## **Object Management Group**

Framingham Corporate Center  
492 Old Connecticut Path  
Framingham, MA 01701-4568  
U.S.A.

Telephone: +1-508-820 4300

Facsimile: +1-508-820 4303

# **Business Object DTF**

## **Common Business Objects**

**OMG Document bom/97-12-04**

**12/02/97 7:52 AM**

**Version 1.5**

## **1. Introduction**

### **1.1. Industry Problem**

Increasingly all businesses are becoming highly dependant on software to operate and manage themselves. At the same time the rate of change within business environments is accelerating faster than current practices can accommodate. Businesses need a way to express business concepts, processes and events so that both business and information technology professionals can understand, so that the result of that understanding can be transformed quickly and correctly into information systems. The resulting systems must be scalable, reliable, secure and interoperable. And these systems must gracefully withstand chronic change without becoming un-maintainable.

Business object technology, based on OMG's CORBA and Business Object Framework specifications, addresses this situation. Business objects provide a mechanism to express business models, then aid in the transformation into software designs and implementations. The resulting software should still reflect the original business model in many important respects, thus helping to manage and minimize the impact of change.

### **1.2. How This White Paper addresses the Industry Problem**

This version of the white paper is a first step in providing the following with respect to the above problem statement:

- definition for business objects and common business objects
- criteria for deciding when something is or is not a business object
- taxonomy for organizing our understanding and discussion of business objects
- conceptual architecture for implementing business objects, such that various OMG adopted technologies and current activities can be related to one another

Work is continuing to develop and refine these concepts and subsequent white papers will be submitted for approval as progress is made. In particular, next steps will include looking at the relationship between CBOs and business components; and extending the taxonomy.

### **1.3. Intended Audience**

Consumers of CORBA-based technologies, such as professionals in Information Technology organizations, consultants or systems integrators

Suppliers and consumers of business components or applications

Parties engaged in business process engineering, business modeling, information systems strategy, planning and/or architecture

OMG Members engaged in Domain Technical Committee activities, and those of the Analysis & Design Task Force of the Platform Technical Committee

## 1.4. Writing Team

Editor: Robert Shelton of Open Engineering Inc., Chair of the Common Business Objects Working Group of the OMG Business Objects Domain Task Force.

Contributors:

Bruce Ambler, Lucent	Jeff Leitheiser, Rockwell-Allan-Bradley
Ed Barkmeyer, NIST US	Denise Lynch, United Technologies
Mike Bradley, BellSouth	Naohiko Mori, NCALS JP
Jeanette Breton, Shell	Tom Mowbray, MITRE
Roger Burkhart, John Deere	Peter Nicklin, NSH UK
David Chizmadia, NSA US	David Schramm, HBO & Company
Paul Clements, MSI UK	Francesca Smith, Open Engineering
David Depinet, Shell	Kevin Tyson, Enterprise Engineering
Bill Hatch, EDS	Stephen Tockey, Rockwell
Peter Herzum, SSA	Brian Wood, Open IT
Verlyn Johnson, IBM	David Zenie, IBM/NIIP
Rainer Kossmann, Nortel	

Substantial input was also provided by feedback from the OMG Manufacturing DTF as part of their commitment to identify and define industry and cross-industry CBOs.

## 1.5. Questions & Comments

Please address content-related questions or feedback to [cbo@omg.org](mailto:cbo@omg.org) and to the paper editor [rshelton@openeng.com](mailto:rshelton@openeng.com)

In the interests of mailing list efficiency, please address errata or omissions directly to the editor only.

## 2. Definition of Business Object

### 2.1. General Concept

It is possible and desirable to define both businesses and their software applications in terms of business objects. A business object captures information about a real world (business) concept, operations on that concept, constraints on those operations, and relationships between that concept and other business concepts. The business concept can then be transformed into a software design and implementation. And a business application can be specified in terms of interactions among a configuration of implemented business objects.

As one can infer from the above, the term *business object* is used in two distinct but related ways, with slightly different meanings for each usage:

- In a business model, business objects describe a business itself, and its business context. The business objects capture business concepts and express an abstract view of the business's "real world". The term "modeling business object" is used to designate this usage.
- In a design for a software system or in program code, business objects reflects how business concepts are represented in software. The abstraction here reflects the transformation of business ideas into a software realization. The term "systems business objects" is used to designate this usage.

A growing number of organizations use business object modeling to describe, evaluate and re-design their business processes and entities. Increasingly, businesses are asking for "standard" and "off the shelf" modeling business objects to facilitate such work. Vendors are emerging that sell business object models – generalized descriptions of business practices and templates for common constructs like *Customer*, *Order*, *Product*, *Procurement* or *Billing*. Such products can be used for business engineering, planning, and as input to systems development efforts – just as would internally developed business object models. But these offerings do not necessarily include software, as the vendor is selling business experience and know how.

Many systems developers are building applications with business objects. Increasingly they, too, are asking for libraries of "off the shelf" business objects. But they are looking for designs and code to speed up development and facilitate integration between systems or business that use the same systems business objects. Increasingly, these consumers are demanding that libraries of systems business objects also include the business model "behind" the systems business object implementations. Thus the market is looking for systems business objects that are based on and related to business object models. Systems developers are looking to business objects to provide a bridge between business concepts and their software realizations.

Hence the term business object conveys different meaning in these two contexts. These meanings are further explained in the next two sections, followed by a section that explains the relationship between the two.

## 2.2. Business Objects in a Business Model – First Usage

A business object describes a thing, concept, process or event in operation, management, planning or accounting of a business or other organization. It is a conceptual object that has been specified for the purpose of directly describing and representing, and thus serving, a business concept or purpose. The focus/subject is the business subject/concept being modeled.

A business object in this usage is a specification for a kind of object which may exist in one or more business domains. This specification of a business object may include attributes, relationships, and actions/events that apply to these objects. The form of this specification could be textual (e.g. CDL specification language), graphical (e.g. UML or other modeling language), or a natural language description.

These modeling business objects exist regardless of the existence of information systems, applications, software design or program code. They are independent of information systems because modeling business objects directly reflect and abstract business concepts. Thus modeling business objects are defined independently of application systems.

Unlike other objects historically considered at OMG, interfaces (in the IDL specification sense) are not the primary issue with modeling business objects. The primary issue is capturing common business semantics, having a common idea or concept that is usable by different parts of a business or by different independent businesses. Interfaces as currently defined by OMG do not supply common underlying semantics and interfaces are not required to define a business object.

## 2.3. Business Objects in a System Model or Implementation – Second Usage

A business object when used to describe a system represents something in the system that is itself an abstraction representing something in the real world.

Usually the real world concept would have first been represented in a business object model, using modeling business objects as defined in the previous section. Thus a business objects that describes a business concept is input to specifying the business objects in a system. A business object when used to describe a system has a correlation to a business object that has been used to describe the business, although this correlation may not be one-to-one. The business concepts provide constraints and context.

These systems business objects are a type of object as defined by the OMG Object Model. Thus these objects have the properties<sup>1</sup> a developer would expect from a software or design object. Additionally, these business objects have the following properties not defined in the OMG object model, but describable in CDL and recognized in the BODTF meta model:

- Behavior
- Business rules – specific constraints on behavior, relationships and/or attributes that reflect rules that govern business conduct

---

<sup>1</sup> The term *properties* as used here refers to the characteristics of the object's type or class; for example *inheritance*, *encapsulation*, *attributes* and *behaviors* are properties commonly ascribed to object types.

- Business identity – one or more attributes capture for every business object type the name-value pairs that uniquely identify in a business-meaningful way every instance of that type as apart from every other instance of that type

Systems business objects also have properties that are not mandatory under the OMG object model, but that are mandatory for all business objects; which properties may be provided by OMG CORBA services:

- Integrity of instances and inter-instance relationships through business rules
- Persistence – outlasting the application which created any given instance
- Security – protecting its instances from unauthorized use
- Interoperability with business objects defined by other vendors
- Transactability – ensuring the completion or rollback of changes

A business object when used to describe a design or software representation of a business concept must itself be describable by OMG Domain Object Model (Business) and implementable from an OMG IDL specification or in the OMG Business Object Framework. It is not conversely required that every possible software implementation that is possible in a BOF is automatically a business object.

Purchased systems business objects would usually contain both a software executable and a software specification. As such, a class library of purchased business objects could be viewed as a software framework – prefabricated, confinable software that solves a specific problem (i.e. capturing, operating on and maintaining *Customer* data). Additionally, it would be reasonable to expect business object products to combine the software design and implementation with a business-level model as described in the previous section.

## **2.4. Correspondence between the Business Model and the System Model**

Systems business objects correspond to modeling business objects because systems business objects represent in a system the information and dynamics of the business concept as captured in the business model.

There may be objects in a system model that are not business objects, and that are not derived from modeling business objects. The simple fact that a design or software implements a business application does not make all of the objects in that design or software into business objects. This occurs because objects can represent concepts that are technological or application-specific, rather than business, in nature.

The information and dynamics represented by the systems business objects are determined by the processing that must be carried out by the system in order to fulfill its role or roles in the business model. There may be business objects for which there are no information and dynamics in the system model. Consequently, it is also true that not every modeling business object in the business model will have a corresponding business object in the system model. This is a matter of scope and implementation decisions.

## **2.5. Common Business Objects**

A common business object (CBO) is a business object that is on the Roadmap or specified in an RFI or RFP of more than one OMG Domain Task Force or Working Group. A CBO need not be universal to all OMG domains.

For the purposes of defining CBOs, a domain is defined as an industry sector, a list of which may be found in the US Department Commerce list of Standard Industry Codes (SIC codes). SIC codes are 4 digit codes, the first two digits of which identify an industry sector. This is reasonable as OMG's Domain Task Forces are frequently formed around industry sectors – Finance, Insurance, Health Care, Manufacturing, Transportation, Telecommunications, Utilities. Those DTFs that are not “vertical” in focus address “cross industry” concerns, and thus may directly address CBOs.

The designation *common* can be applied to business objects of either usage described above. As with all business objects in general, the most important issue with CBOs is a common concept. Common interfaces, though important, are a secondary consideration to commonality of underlying business concepts. Thus one can speak of a CBO in a business model; or a CBO in a systems design or in software.

## **2.6. Domain Variations on Common Business Objects**

RFPs are in progress in various OMG DTFs to obtain domain-specific variations on business objects that CBO Working Group and the DTFs believe are common business objects. This paper and other work taking place in the Business Objects DTF are intended to facilitate and coordinate these domain-specific efforts.

Many domain variations are expected to be specializations of, and should be expressible in terms of, one or more underlying common business objects. Such domain business objects would be industry-specific extensions of the underlying name, definition, and properties of the common business object.

Some business objects that are identified by OMG Domain groups will not be variations on or specializations of common business objects. Rather they will be unique to one specific domain. They will express concepts that are material (at least at some point in time) to only one OMG Domain group. Some of these domain-specific business objects may, over time, be redefined as common. Or they may be recognized as having an underlying abstraction in common with business objects in another OMG domain. It is fully expected that other business objects will remain domain unique.

## **2.7. Regional Variations on Common Business Objects**

Business concepts may be defined differently by geo-political regions, economic organizations of nations, or nations. This variation may be as fundamental as naming, definition and existence (certain concepts may not have complete analogues in every area of the world); to differences in common business practices, laws and rules.

Regionalization raises two issues:

(1) CBOs must be customizable to a specific region or nation. For example, VAT rates vary from region to region and a VAT rate for a specific region may be raised in the future.

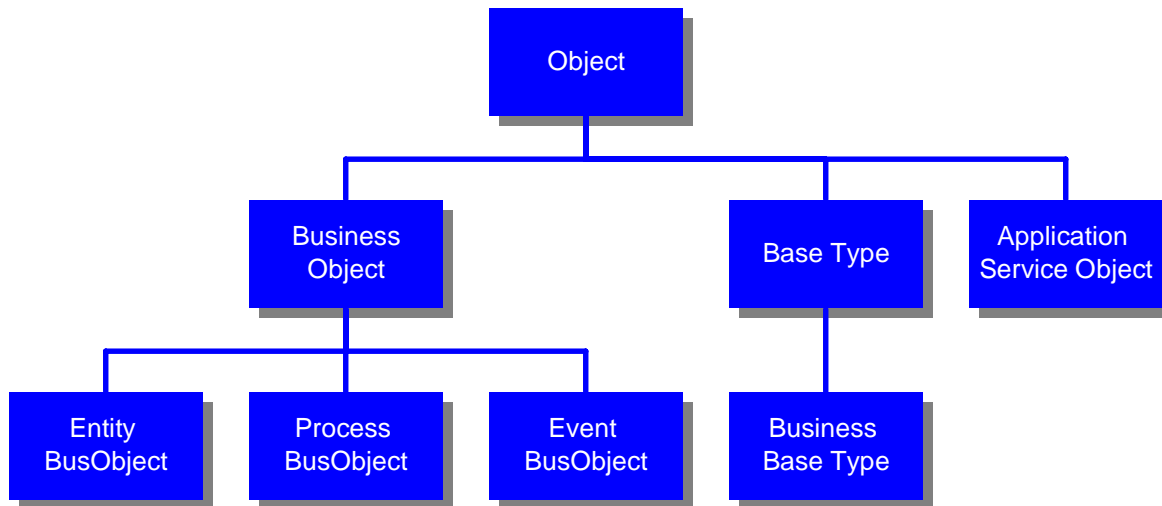
(2) CBOs customized for a specific region or nation must be interoperable with those customized for another. For example, a "dollar" object and a "yen" object must interoperate in the context of a currency exchange.

To be domain-specific (i.e. vertical industry) and region-specific are orthogonal concepts. Any domain-specific object may have to be regionalized. Likewise any CBO can be regionalized. This is an area that requires further exploration, as the basic issue of regionalization comes into play with respect to a wide variety of geo-political boundaries.



### 3. Taxonomy

OMG BODTF organizes business objects into a taxonomy to facilitate identification, communication, and synchronization across Domain Task Forces. The taxonomy should be read as a type hierarchy, and does not presume to be complete at this time. The groupings in this taxonomy will be explained below.



#### 3.1. Explanation of Constructs in the Taxonomy

##### 3.1.1. *Object*

A modeling or software package of attributes encapsulated in services. Capable of requesting services of other objects by sending messages. Capable of being specialized through a mechanism like inheritance or delegation.

##### 3.1.2. *Business Object*

A business object is either a modeling business object or a systems business object, as stated in Section 2.1 of this paper. To summarize that section, a modeling business object is an object that describes a thing in the business itself; while a systems business object is a software object that represents a business concept in software.

##### 3.1.3. *Entity Business Object*

A business object that describes a person, place, thing or concept. These are the basic concepts in any business that are engaged in the conduct of business processes.

Examples:

- Involved Party – person, organization, customer, supplier, employee, patient
- Location – address, phone number, email address, regional area

- Product – good, service, part, bill-of-materials
- Agreement – contract, policy
- Order – purchase order, sales order, work order, trade, shipment manifest
- Financial Instrument – currency

#### **3.1.4. *Process Business Object***

A business object that describes a business process or workflow, is comprised of a specified collection of entity business objects and a pattern of interactions and business events. A process business object type can be thought of as a configuration of objects, a template for process instances that capture the specific interactions among instances of the involved entity business object types. Interactions represent and implement work steps / activities. The entity business objects are the actors (i.e. roles) and subjects of action (i.e. resources consumed, products produced, equipment operated, orders placed, etc.)

Examples: order fulfillment, procurement, making or collecting a payment

#### **3.1.5. *Event Business Object***

A business object that describes a business event, which may be an occurrence or may be related to the passage of time or crossing of a boundary in time. Events trigger and result from interactions between entity business objects in the context of a process business object.

Examples: part inventory below threshold, tank pressure over limit, account overdrawn, bond mature, end of fiscal year

#### **3.1.6. *Base Type***

An OMG object that does not capture a business concept per-se, and that might be represented as a programming language construct except that it needs to be expressed as an OMG object to take advantage of properties like inheritance/specialization or methods.

Examples: decimal, description, date, time

#### **3.1.7. *Business Base Type***

A base type that has specific business meaning, yet it has more in common with the notion of data types than with business objects per-se. Instances of Base Business Type are derived from Base Types. The derivation adds general or domain-specific business meaning.

Examples: length, width, area, dimensional size

#### **3.1.8. *Application Service Object***

A design or software object that provides functions needed by many applications. These objects concepts that are typically are known to users of the applications, but are not thought of as business data and function.

Examples: Resource adapter, role, group, user (as proposed in the NIIP submission to BODTF RFP-1), sequential number generator.

### 3.2. Relationship of Taxonomy to RM-ODP

The RM-ODP is an OMG-adopted reference model. The various taxonomy constructs can be mapped to the viewpoints in RM-ODP as in the table below. For more information about RM-ODP, please see the Terms & References section of this paper.

	Entity Bus Object	Process Bus Object	Event Bus Object	Base Type	Business Base Type	Platform Technology
Enterprise	x	X	X			
Information	x	X	X			
Computation	x	X	X			
Engineering	x	X	X	x	X	
Technology						X

It is important to note that the business concept represented by any given business object, say a *Customer* entity or *Ordering* process, would have different representation in each viewpoint. Thus the Enterprise viewpoint would capture the system context view of *Customer*, while the Information viewpoint would represent the attributes and data relationships that the system (i.e. the business object server) would retain for use by multiple client applications across the business.

It is our preliminary belief that

- “modeling business objects” and their use in a business description are related to the ODP concept of an Enterprise (viewpoint) description – an object model of the business in which the IT system has one or more roles;
- “systems business objects” and their use in a design for a software system or in program code is related (depending upon the what is being described) either
  - to the ODP concepts of Information and Computational (viewpoint) descriptions – object models of the semantics and computational structure of business processing, or
  - to the ODP concept of an Engineering (viewpoint) description) – an object model of the infrastructure necessary to support the business processing.

This observation is preliminary, is intended to stimulate discussion, and requires further consideration.

### 3.3. Relationship of Taxonomy to Zachman Framework

The various taxonomy constructs can also be mapped to the Zachman Framework for Information Systems Architecture, another common reference model used by

commercial IT organizations. For more information on the Zachman Framework, please see the Terms & References section of this paper.

	Entity Bus Object	Process Bus Object	Event Bus Object	Base Type	Business Base Type	Platform Technology
Scope	x	x	x			
Enterprise	x	x	x			
System	x	x	x			
Technology	x	x	x	x	x	x
Component	x	x	x	x	x	x
Implementation						

As with the RM-ODP, the Zachman Framework specifies a number of independent but related viewpoints. The same business object would be represented in each viewpoint as it progressed in consideration from business-level planning model (Scope) to business model (Enterprise), ultimately to software components (Component). As with RM-ODP, the subject matter of consideration differs with each viewpoint, while the core business concept under discussion remains the common thread.

Furthermore, it is our preliminary belief that modeling business objects may play a role in the Scope, Enterprise and System viewpoints of the Zachman Framework; while system business objects may play a role in the System, Technology, Component and Implementation viewpoints. This observation is preliminary, intended to stimulate discussion, and requires further consideration.

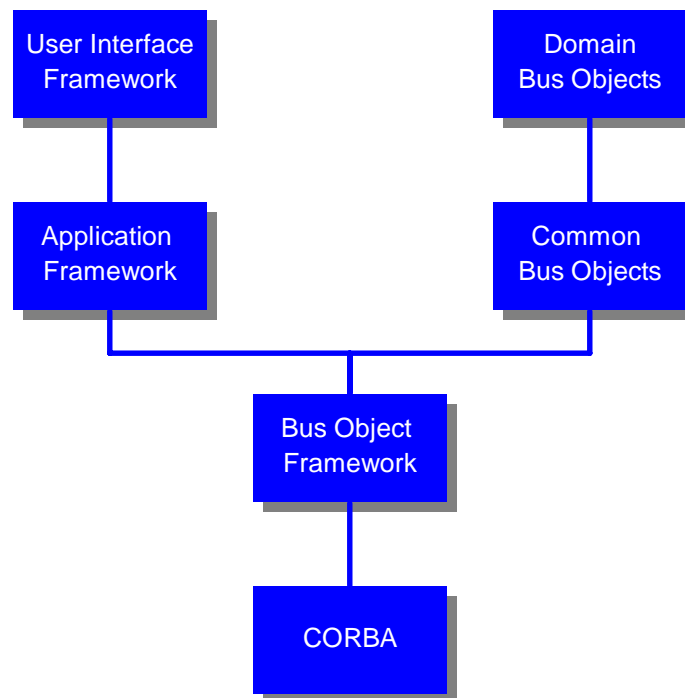
## 4. How Business Objects Fit the CORBA Picture

### 4.1. OMA View

This work fits into the current OMG Object Management Architecture in the Domain section. Representation of this fit will be resolved by the ORMSC Reference Model Working Group in cooperation with the OMG DTFs.

### 4.2. Runtime View

In the course of discussions, a picture of how business objects fit with CORBA, the BODTF Business Object Framework and other OMG activities has evolved. This section is provided to position aspects of work currently in progress within BODTF and other DTFs, and some relevant though out-of-scope industry work. It is not a formal statement of architecture. Rather it is provided because it has helped members of the CBO Working Group discuss and understand our own work in relation to that of others in OMG.



#### 4.2.1. CORBA

Platform technology for distributed object computing, including system services (i.e. Security, Transactions...). This is the basic plumbing upon which all OMG adopted technology rests, and is the foundation on top of which CBO Working Group proposes that business objects would operate.

#### **4.2.2. Business Object Framework**

A socket into which systems business objects can be designed and implemented. The BOF interface provides a design target. BOF enables interoperability among business objects provided by different vendors, makes the underlying ORB transparent to the business object developer, and provides a consistently integrated set of services. This technology is currently being adopted through OMG Business Objects DTF. CBO Working Group presumes that, while business objects will not be required to use BOF, it will be in the best interests of consumers and suppliers alike to have BOF compliant business objects.

#### **4.2.3. Common Business Objects**

The cross-domain core of business objects. Implementations may be targeted to BOF, directly to CORBA, or to some other framework. By definition, these are adopted or targeted for adoption in two or more OMG domains. Most likely, these objects will be base types from which industry-specific and consumer-specific business objects are specialized (i.e. via inheritance at the language level).

Examples in progress: Involved Party, Address, Currency

#### **4.2.4. Domain Business Objects**

Business objects that are specific to an industry (i.e. that have no generalization in other industries, or at least no generalization that is on the Roadmap of any other DTF) or industry specializations of common business objects. Implementations may be targeted to BOF or directly to CORBA. Implementations may or may not be based on underlying CBOs.

Examples in progress: Patient (Health Care DTF), Part (Manufacturing DTF), Currency (Finance DTF)

#### **4.2.5. Application Framework**

A framework of classes that facilitate programming user-interface-level applications based on BOF. These objects abstract and present underlying constructs that are needed to build applications.

Examples in progress: NIIP submission to BODTF RFP-1

#### **4.2.6. User Interface Framework**

This is out of OMG scope, but is placed on the diagram to distinguish between the application framework and the GUI development environments that are unrelated to BOF, but that likely would be used to develop applications based on BOF.

A number of clarifications should be made on the above diagram and explanation – specifically about what this section does not imply. First, the diagram is *not* intended to suggest that all domain business objects arise from common; rather the intent is to express visually the relationship between the two in light of the more thorough explanation above. Second, OMG does *not* believe that all business objects must be implemented in the Business Object Framework. Other frameworks could be used to

implement business objects. Thus one should ***not*** conclude from this diagram that business objects are only those things that are implemented in a business object framework, and none other. Indeed, one should ***not*** conclude that business objects can only be implemented on CORBA. Rather this diagram suggests how a specific set of issues and standards under consideration in the OMG Business Object Domain Task Force and other OMG Task Forces can be seen in relation to one another.

## 5. Considerations for CBO RFPs

### 5.1. General Sequencing

CBO Working Group plans to start coordination efforts with Entity Business Objects. Later we expect to engage on Process and Event Business Objects. This choice is partly tactical, as OMG DTFs are predominately focusing on Entity Business Objects at this time. Also from commercial experience this is a logical prerequisite to the proposed subsequent efforts.

### 5.2. Prioritization

The following common business objects will most likely be addressed starting in the order below, though these activities will not be totally sequential or independent. Note that this list reflects the order in which CBO Working Group anticipates structuring its work, and not necessarily the state of affairs in OMG DTFs.

- Involved Party
- Location (including Address)
- Agreement
- Product (including Part)

### 5.3. Structure of RFPs

#### 5.3.1. Domain vs. Generic

RFPs for business objects will need to address both the common and domain-specific portions. Several options are available:

- Separate RFPs for the common and each domain portion, where the domain RFPs reference the common RFP.
- Domain-driven RFP where the “first” domain addressing the common portion also requests the common portion; “later” domains reference the adopted common business object.
- Combined RFP which requests the common portion and multiple domain specializations.

CBO Working Group believes that the third option is preferable, and recommends that RFP submitters be strongly encouraged (and where appropriate, required) to respond to the generic portion along with one (or optionally more than one) domain specific specialization. Further, language needs to be developed to form a template for this process so that DTFs need not invent the language and process for such combined RFPs.

This area requires further work by CBO Working Group and the OMG DTFs.



### **5.3.2. *CDL or IDL Specification***

CBO Working Group believes that the Business Object Framework technology currently being evaluated for adoption is important to ensuring inter-domain interoperability in business objects. We also believe that a semantics-rich specification will be necessary for adopting business objects in both business model and software design/interface forms. Business Object DTF is proposing, as part of the RFP-1 process, a Component Definition Language to address this specification problem. CDL is a superset of OMG IDL. While it may be reasonable to specify some business objects and many base types in IDL alone, we expect that CDL (when and as adopted) will become a requirement for business object RFP responses. Presently submitters to a variety of Domain RFPs are using CDL, and believe that this approach is workable and effective.

### **5.3.3. *Business and Systems Model Specification***

Historically OMG has focused on RFPs for and responses in the form of OMG IDL – interface definitions which specify syntax and limited semantics. These efforts focused on delivered software technology.

With the advent of business objects, software remains a valid and desired response to many Domain RFPs. Business object models become, however, valid responses in many cases. Essentially, DTFs may “adopt technology” that is in fact a business-level specification, which itself becomes the basis for requesting “implementable” technology.

This is recognized as an issue. Questions have been raised as to whether or not this is possible within current OMG Policies & Procedures. With minor alterations that are in progress, CBO Working Group believes that Domain RFPs can be issued that ask for business models absent software implementations. More work is required in this area, however, to ensure that this happens productively.

## 6. Terms & Reference

### 6.1. Terms

- CDL – Component Definition Language – a proposed superset of OMG IDL intended to facilitate specification of business objects and the business object framework interface.
- DTFs – OMG Domain Task Forces – the technology adoption bodies of the OMG Domain Technical Committee.
- IDL – Interface Definition Language – OMG standard for technology specification.
- RM-ODP – Reference Model of Open Distributed Computing – an ISO reference model that has been adopted by OMG to facilitate communication and understanding in OMG efforts.
- UML – Unified Modeling Language – an adopted technology by OMG Analysis & Design PTF for graphical representation of software analysis and design models.

### 6.2. References

Documents on many topics addressed in this paper may be found on the OMG server, or obtained by request to OMG staff.

#### 6.2.1. *BODTF RFP-1*

The Business Objects Domain Task Force first request for proposal is OMG Document CF/96-01-04. This RFP requested both horizontal business object framework technology submissions and common business objects.

#### 6.2.2. *BODTF Meta Model*

The meta model describes the parts of business objects and relationships among those parts, initially from the viewpoint of the BOF; and later also from the viewpoint of a business model. The meta model is being specified starting with the target of the second usage of business object as defined in this paper, and will migrate to represent the first usage. A description can be found in OMG Document bom/97-09-01 or its successors.

#### 6.2.3. *Business Object Framework (BOF)*

- Multiple BOF technology proposals were submitted in response to BODTF RFP-1.

Convergence of these submissions has resulted in substantial change. Contact OMG staff for the most recent versions, or visit the BODTF web page via the OMG web site.

#### 6.2.4. *Common Business Objects*

Two submissions also were received in response to BODTF RFP-1 that addressed common business objects. Additionally, many responses to the BODTF RFI-1

proposed common business objects. Further information may be obtained from OMG staff, or by visiting the BODTF web page via the OMG web site.

#### **6.2.5. *Component Definition Language (CDL)***

Component definition language is an IDL superset that addresses issues of broader interface semantics specification and behavior specification. A description can be found in OMG Document bom/97-09-01 or its successors.

#### **6.2.6. *Interface Definition Language (IDL)***

IDL is the specification language used at OMG to define technology specifications and CORBA language mappings. See Formal/97-07-01 for IDL and Formal/97-02-25 for CORBA II and language mappings in IDL.

#### **6.2.7. *Object Management Architecture (OMA)***

The OMA is OMG's organizational architecture. The current statement of the OMA may be obtained from OMG staff.

#### **6.2.8. *RM-ODP***

Reference Model of Open Distributed Computing is an ISO standard and ITU-T Recommendation (ISO/IEC 10746 | ITU-T Rec. X.901-904) that has been adopted by OMG as a reference model. Extensive consideration is given to this model in the work of the Object Reference Model Subcommittee, specifically the Reference Model Working Group. A summary may be found in:

- ORMSC/97-06-06, "International Standards Define Semantics: RM-ODP & GRM", Hiam Kilov, Merrill Lynch
- ORMSC/97-06-07, "RM-ODP Part 3: An Introduction", Kevin Tyson, Enterprise Engineering Associates

The RM-ODP standard spans multiple documents, several of which are available to OMG members or the general public as of this writing:

- Parts 1 (Overview), 2 (Foundations) and 3 (Architecture) are available as OM/96-10-02, OM/96-10-03 & OM/96-10-04.
- Parts 2 (Foundations) and 3 (Architecture) can be viewed on-line at [www.iso.ch:8000/RM-ODP](http://www.iso.ch:8000/RM-ODP).

#### **6.2.9. *Zachman Framework for Information Systems Architecture***

The Zachman Framework was initially developed by John Zachman of IBM. Written background on the Zachman Framework may be found in:

- Sowa & Zachman, "Extending & Formalizing the Framework for Information Systems Architecture", IBM Systems Journal, Vol. 31, No 3, 1992
- Zachman, "A Framework for Information Systems Architecture", IBM Systems Journal, Vol. 26, No 3., 1987.
- ORMSC/97-06-08, "Introduction to Zachman Framework", Stephen Tockey, Rockwell International

- ORMSC/97-06-09, “Adapting Zachman Framework to Business Objects”, Robert Shelton, Open Engineering

More information may be found at <http://www.zifa.com/>, the home page of the Zachman Institute for Framework Advancement.