Object Management Group

Super Distributed Objects

A White Paper

- Ver.1.00-

Super Distributed Objects DSIG

Co-chair

Seiichi Shin, University of Tokyo

Katsumi Kawano, Hitachi Ltd.

Workgroup members:

Stefan Arbanowski, GMD FOKUS

Tatsuya Suda, Junichi Suzuki, and Mike Wang, UC Irvine

Christophe Gransart, LIFL

Michiharu Takemoto, NTT

Shigetoshi Sameshima, Hitachi Ltd.

Editor:

Shigetoshi Sameshima, Hitachi Ltd.

Stefan Arbanowski, GMD Fokus

Junichi Suzuki, UC Irvine

DOCUMENT HISTORY

Ver. 0.1	December 2000
Initial version as input for the OMG December 2000 meeting. Based on SDO-RFI and corresponding responses.	
Ver. 0.2	**February 2001**
Based on discussions at the Orlando meeting.	
Ver. 0.3	**February 2001**
Document structure reorganization. Appendix is completed.	
Ver. 0.4	**April 2001**
Refined architecture and corresponding parts based on the discussion at the Irvine Meeting.	
Ver. 0.5	**April 2001**
Refined other parts based on the discussion at the Irvine Meeting	
Ver. 0.6	**April 2001**
Introduction and format update	
Ver. 0.7	**April 2001**
Refined terminologies and other parts based on the discussion at the Paris Meeting	
Ver. 0.8	**May 2001**
Refined based on the discussion at the Paris Meeting	
Ver. 1.0	**July 2001**
Finalized first version at the Danvers Meeting	

ABSTRACT

The progress of high-performance and low-cost processor technology is enabling computer power to be embedded densely in devices like mobile phones and PDAs and facilities Internet appliances as well as desktop computers. They are most likely to become present everywhere: at home and in the office, on the street, in cars, in factories, and in various other locations. The computer power and the network could lead to the ubiquitous availability of services – anytime, anywhere, and for anyone – and we can enhance the safety, security, and quality/comfort of our lives and open the way to a wide range of other beneficial effects. Much technical and business attention is being paid to new computing paradigms like mobile computing, ubiquitous computing and pervasive computing driven by these technical trends. Although there are standards and technologies for interconnecting devices in each application domain, no common standards exist to handle these devices for various applications, across these domains, in a unified manner.

This white paper on Super Distributed Objects (SDO) by OMG SDO DSIG is the result of responses to an RFI on “Super Distributed Objects: Initial Survey” (sdo/00-06-05). The idea for Super Distributed Objects is to provide a standard computing infrastructure that models real world entities (e.g. devices) as objects, deploys them in a highly distributed environment, allows them to seamlessly interwork with each other, and ubiquitously aids users in accomplishing their tasks. Super distribution means incorporating massive numbers of objects beyond centralized control, each of which performs its task autonomously or cooperatively with other objects. Incorporating these characteristics in a distributed object system would enable new applications and expand the business field of CORBA-based business systems.

The OMG Super Distributed Objects Domain Special Interest Group (SDO DSIG) focuses on (1) exploring the characteristics of super distribution (i.e. massive numbers of objects, decentralization, autonomy of objects, and cooperation between objects) in distributed object systems in terms of the OMG technologies (CORBA, UML, etc.), and (2) specifying the standards that allow us to develop super distributed systems by leveraging existing OMG specifications. This white paper defines a common terminology for describing super distributed systems and provides a reference architecture for creating, deploying, interoperating and managing such systems.

TABLE OF CONTENTS

1. Introduction

1.1. Motivation

The progress of high-performance and low-cost processor technology is enabling computer power to be embedded densely in devices like mobile phones and PDAs and facilities like Internet appliances as well as desktop computers. They are most likely to become present everywhere: at home and in the office, on the street, in cars, in factories, and in various other locations. Emerging wireless networks like Bluetooth are enabling these devices to be connected to each other without new wires, and IPv6 also supports their interconnection by accommodating a massive number of devices. This computer power and the network could lead to the ubiquitous availability of services – anytime, anywhere, and for anyone – and we can enhance the safety, security, and quality/comfort of our lives and open the way to a wide range of other beneficial effects.

Much technical and business attention is being paid to new computing paradigms like mobile computing, ubiquitous computing and pervasive computing driven by these technical trends. A service system making use of these devices is being constructed through the connection to already existing networks, namely the Internet. Although there are standards for interconnecting devices like HAVi, BACnet, and Jini in each application domain, no common standards exist to handle these devices for various service applications, across these domains, in a unified manner.

The idea for Super Distributed Objects (SDO) is to provide a standard computing infrastructure that models real world entities (e.g. devices) as objects, deploys them in a highly distributed environment, allows them to seamlessly interwork with each other, and ubiquitously aids users in accomplishing their tasks. Super distribution means incorporating massive numbers of objects beyond centralized control, each of which performs its task autonomously or cooperatively with other objects. Incorporating these characteristics in a distributed object system requires to address the issues, such as ad-hoc interaction between objects and temporary unavailability of objects, and the idea like peer-to-peer computing. It also would enable new applications and expand the business field of CORBA-based business systems.

1.2. Objectives

The OMG Super Distributed Objects Domain Special Interest Group (SDO DSIG) focuses on (1) exploring the characteristics of super distribution (i.e. massive numbers of objects, decentralization, autonomy of objects, and cooperation between objects) in distributed object systems in terms of the OMG technologies (CORBA, UML, etc.), and (2) specifying the standards that allow us to develop super distributed systems by leveraging existing OMG specifications.

This white paper defines a common terminology for describing super distributed systems and provides a reference architecture for creating, deploying, interoperating and managing such systems. It also analyzes technological extensions to existing OMG standards.

The remainder of this white paper is organized as follows: Section 1.3 shows application examples of super distributed system. Section 2 defines a common terminology for describing super distributed systems. Section 3 describes several requirements for developing super distributed systems. Section 4 overviews a reference architecture of super distributed systems. Section 5 and 6 address RFP candidates and a roadmap of the SDO DSIG. Other related technologies and standards are listed in Section 7. Section 8 overviews the responses for our first RFI, “Super Distributed Objects: Initial Survey” (sdo/00-06-05).

1.3. Application examples

Many applications have been considered for various situations in our daily lives at home, in the office, in the car, or in industry. Some examples are given below.

- A smart card keeps its owner information and the owner’s own data such as presentation files so that the owner can make a presentation, print it, and share with others everywhere he/she goes.

- Home appliances cooperate for energy-conservation service, health care, home security (theft and fire prevention), home theater, etc. in accordance with the preferences of each person living in the house. Intelligent appliances hold their usage conditions and history for their lifecycle management and better recycling.

- Smart car supports a comfortable driving for human by controlling itself as well as providing information on traffic and surrounding cars in cooperation with smart highways that embed smart sensors in a spatially distributed manner for safe and comfortable driving.

- Every vehicle in a large city has one or more attached sensor devises. These devises are capable of detecting their location, vehicle sizes, speeds and densities, road conditions and so on. As vehicles pass each other they exchange information summaries. These summaries eventually diffuse across sections of the city. In-car devises plan alternative routes, estimate trip time and warn dangerous driving conditions. Unlike the centralized systems sometimes seen today, decentralized architecture based on local communication would scale as the number of vehicles grows and provide much greater local details.

- Billboards, posters, catalogs or price tags in a store communicate with customers’ handheld devices or membership cards to dynamically change their contents according to each customer’s preference, purchase history or location.

- People receive information service on their location (e.g. “Which is a nearest restaurant?”), make reservations, and purchase tickets as they go through the entrance on arrival by a mobile phone or PDA.

- Software agents containing information contents like movie ticket advertise themselves to persons around their temporal residential devices in the restaurant.

- Thousands of disposable sensor devises are densely scattered over a disaster area. Some of them fall into regions affected by the disaster (e.g. fire) and are destroyed. The remaining devises collectively map these affected regions, direct the nearest emergency response teams to affected sites, or find safe evacuation paths. Disaster recovery today is very human intensive.

2. Terminology

Super distribution	A property of distributed systems incorporating massive numbers of objects in a decentralized manner. In the sense of SDO DSIG, independently developed objects scatter on highly distributed and ad-hoc environment. They behave autonomously or cooperatively, and form application services dynamically.
SDO Super Distributed Object	A logical representation of a certain piece of hardware or software that provides well-known functionality. A SDO should have the characteristics of super distribution (i.e. the massive numbers of objects, decentralization, autonomy of objects, and cooperation between objects). Examples include abstractions of devices such as mobile phones, PDAs and home appliances, but are not limited to hardware abstractions. SDOs may represent web services such as content distribution, content indexing, user profiling and billing. A SDO may also act as a peer in a peer-to-peer networking system. A SDO may be mobile or stationary. It behaves autonomously or cooperatively with others.
Service Logic	Service Logic is an object that defines what, when and how one or more SDOs perform in a given condition (e.g. event reactions, behavior policies, user preferences, device specifications). A Service Logic object can be defined for each SDO or for multiple SDOs. It can be hosted by SDO itself, or maintained as a separate component from SDO in distributed or centralized manner.
Application service	An application service is provided by one or more SDOs through interactions between SDOs and/or between SDO and Service Logic object. One or more SDOs and Service Logic objectcan participate to an application service. What/Who initiates to form an application service varies between implementations. Some implementations allow users to initiate application services manually through any user interface, and other implementations allow Service Logic object to initiate them automatically. An application service may be formed as an emergence of ad hoc interactions among SDOs and Service Logic object.
User	The human for whom application services are provided.
Owner	The human who has authority to make use of SDOs or to start/stop an application service.
Environment	The physical existence that may be sensed and/or actuated by SDOs.
Group	A set of SDOs that mutually cooperate to form an application service.

3. Requirements

This section describes a series of requirements for developing super distributed systems and what the SDO DSIG will try to specify.

3.1. System structure

3.1.1. Decentralization

Decentralization means that there is no centralized entity for controlling, coordinating and managing SDOs. A super distributed system requires a decentralized organization of SDOs because it is difficult to handle massive numbers of SDOs, which are highly distributed geometrically, in a centralized point of control (e.g. centralized directory). Decentralization allows SDOs to be loosely-coupled in that a SDO does not have to depend on others through a centralized entity that knows system's global information. It also allows SDOs to behave in an autonomous, dynamic and fault tolerant manner. These characteristics are described in the following sections. The SDO DSIG will not mandate any specific degree of decentralization because the degree varies between implementations. Some implementations support complete decentralization without any central point of control, others support a centralized control. There can be intermediate implementations that use a replicated group of a centralized Naming services or a federation of Trading services.

3.1.2. Autonomy

SDOs are loosely coupled with each other and can behave autonomously. They autonomously join network, discover other SDOs, provide their own functions, form an application service, and leave network without any intervention of users and other SDOs. They act based on local information (e.g. conditions in their local environment) and local interactions with other SDOs, while remaining decentralized. Here, "local" is defined as both spatially nearby and numerically limited. For example, an SDO may obtain information regarding the SDOs running within its communication scope (spatially nearby locality). It may also interact with a limited number of other distant SDOs (numerically limited locality). In general, autonomy increases scalability of the SDO population. Higher autonomy realizes higher degree of decentralization. The SDO DSIG does not mandate any specific level of autonomy.

3.1.3. Dynamic Relationship

An SDO establishes and maintains its relationships dynamically depending on its context (e.g. location). Each SDO can use its relationships to represent its acquaintances, discover other SDOs, communicate with other SDOs, or form a group of SDOs. For example, when an SDO, e.g. representing a PDA or cell phone, enters a room, it may discover SDOs representing devices in the room, establish new relationships with them, and eliminate some existing relationships. A topology of SDOs can be hierarchical or flat. Super distributed systems require a general mechanism to establish, query, evaluate and eliminate relationships. The SDO DSIG will evaluate applicability of the CoS Relationship Management Service, or develop a new appropriate service to meet our requirements.

3.1.4. Ad hoc System Boundary

The boundary of super distributed system often changes as SDOs frequently enter/leave/roam environment or go down. There is no fixed gateway or access point for SDOs to communicate with each other within a system or application service: each SDO acts as an access point or gateway. Therefore, communication path between SDOs always changes. A SDO will communicate with other SDOs in an ad-hoc and peer-to-peer manner. The SDO DSIG will extend existing OMG specifications or specify new services so that

DSIG will extend existing OMG specifications or specify new services so that SDOs and application services can deal with ad-hoc system boundary.

3.1.5. Interoperability between heterogeneous systems

In super distributed systems, network and system platforms will be heterogeneous. The SDO DSIG will specify standards that allow SDOs to interwork in platform independent manner. SDOs' functionalities are also heterogeneous. Independently developed SDOs have different sets of functionalities. Even if SDOs have the same functionality, they can provide it in different way or different level of quality. For example, SDOs providing the same function of 'printing' may provide different printing speeds and different resolutions. These heterogeneities should be addressed in future SDO specifications.

3.2. Communication

3.2.1. Human-Object Interaction

User interface (UI) allows users to access and manipulate SDOs. Different UI components can be attached to a single SDO. For example, a SDO that represents a sensor may have different user interfaces for the sensor's users and administrators. The UI components attached to a SDO can be changed by users or the SDO itself. The SDO DSIG will provide a standard interface to find UI components that interwork with a SDO, connect the SDO with feasible UI components and change UI components. The interface will not address how to implement UI components.

Not all SDOs have necessarily to provide own UI components. For instance, an SDO representing a VCR can be controlled via a user interface, which is provided by a PDA. In this case the PDA automatically adapts to the application service provided by the VCR and offers this functions via its user interface component to the user.

3.2.2. Inter-Object Interaction

The SDO DSIG will specify the interface with which an SDO communicates with others. The interface will allow several communication semantics including broadcast, multicast and point-to-point (object-to-object) communication. Beside this, the aspects of object naming and addressing have to be analyzed. Existing CoSs (e.g. Naming Service) are good starting points. The SDO DSIG will investigate whether they are feasible to fit the requirements of SDOs. Another example is the communication based on the physical location of SDOs.

3.2.3. Discovery

The dynamic and ad-hoc environment produces the need for a method to search for SDOs. The SDO DSIG will specify a service for SDO discovery. The discovery service is targeted at providing a method to locate them in a network, while remaining decentralized and adaptive to ad-hoc environment. Each SDO may maintain relationships to several other SDOs, and use them effectively for discovery. Searches originate from a SDO and may federate from SDO to SDO based upon relationships. Each SDO also contains descriptive information about itself, such as unique ID, functionality, location and owner name, in order to allow searches to distinguish SDOs. A SDO may establish a relationship with another SDO through actively searching for other SDOs, through introduction via other SDOs, and through discovery-related interactions.

3.3. Service

3.3.1. Service logic

In super distributed systems, massive numbers of SDOs exist to provide various application services. Service logic is used to form an application service by combining multiple SDOs dynamically. It parameterizes SDOs by defining what, when and how one or more SDOs performs in a given condition. A service logic object can be defined for a single or multiple SDOs. It can be hosted by SDO itself, or maintained as a separate component from SDO in distributed or centralized manner (see Figure 1). This influences the level of SDO's autonomy. A SDO hosting its own service logic inside can behave autonomously, whereas a SDO attached with external service logic is less autonomous. The SDO DSIG will not mandate how to maintain service logic and leave it to implementations.

An application service is formed through interactions between SDOs and/or between SDO and service logic. A SDO can participate to different application services simultaneously. Figure 1 shows possible deployments of SDO, service logic and application service.

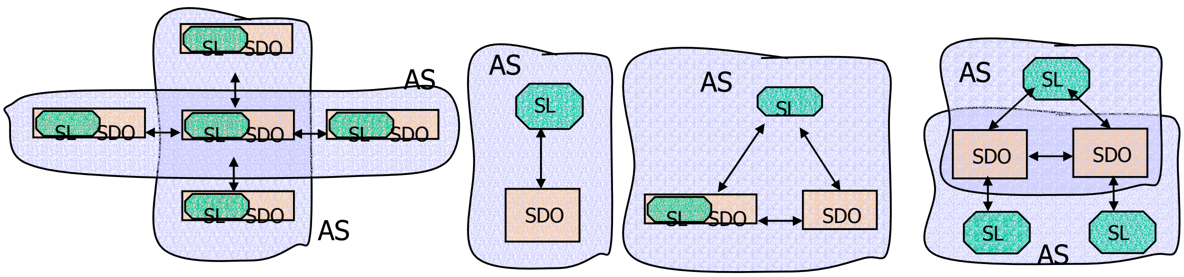

Figure 1: Possible deployments of SDO, service logic (SL) and application service (AS)

Service Logic decides, based on profiles and on the status of the SDO itself, how that SDO has to behave in a certain situation. The profiles and the status will be evaluated by an implementation specific Service Logic that afterwards control the SDO(s). Several different types of profiles have to be evaluated by Service Logic (e.g. User Profile, Service Profile, SDO Profile). See also section 4.1.

The way, how and where Service Logic is realized depends on the application requirements. Figure 1 only shows possible configurations. An implementer can decide what kind of distribution (topology) he prefers and what is suitable for its implementation.

3.3.2. Adaptation and Evolution

In order to provide self-adaptability and context-aware services, a SDO and application service customize their structure and behavior according to the current environmental condition and user's individual situation. For example, a SDO will dynamically change its behavior according to the current location, the kinds of neighboring SDOs, the number of relationships and distance from users. An application service (i.e. a group of SDOs) will change its structure and behavior dynamically according to network failure, processing load and user's preference. The customization may short term process (adaptation) or long term process (evolution) by self-learning capabilities.

3.3.3. SDO Interface

A SDO will be used differently in different contexts or different application services. Each SDO should provide a diverse interface for multiple usage as an application service to be provided. The same application program uses different SDOs in a different context (e.g., using the nearest printer), and the same SDO plays different roles in different application services (e.g., a mobile phone as a communication MMI (machine-machine interface) and remote controller for appliances). It should have an interpretation facility or multiple interfaces. And a mechanism for managing these uses is also necessary (e.g., monitoring for what application services each SDO is used for the purpose of profiling or accounting, etc.).

The SDO DSIG will evaluate applicability of the interface qualification scheme in CORBA and the interface traversal mechanism in CORBA Component Model.

3.4. Others

3.4.1. Security

Security is a very general issue to all distributed systems. SDO functional components have to be protected against illegal access. The followings are essential security requirements.

- identification,
- authentication, and
- access control.

The problems of authentication, identification, and access control will be analyzed in the context of SDO. The huge number of possible SDOs must be secured by a security framework that focuses on the special capabilities of SDOs. All aspects of security will be analyzed from two different viewpoints, regarding user-object interaction and inter-object interaction.

An other security aspect is the protection of the data transmitted among SDOs and their environment. For that purpose, the SDO DSIG will evaluate existing security mechanisms like SSL.

3.4.2. Wrapping legacy technologies

The basic idea of SDO, to model real-world devices and applications as CORBA objects, has to be reflected by the SDO DSIG. General procedures for wrapping legacy technology will be developed facing the fact that environmental constraints can effect the design of a CORBA-based system.

3.4.3. Analysis of enabling technologies

Several other institutions, standardization bodies, and enterprises are currently working on concepts that could influence or improve the work of the SDO DSIG. Therefore developments like JINI, UPnP, Bluetooth, HAVI, EJB and peer-to-peer systems should be evaluated for applicability to SDO.

3.4.4. QoS provision

An application service should not disturb other application services that are already working. And a system should assure an appropriate quality of service (QoS) for an application service to be provided. A system should degrade gracefully if it cannot keep these properties by using redundant SDOs.

3.4.5. Scalability and Fault tolerance

A super distributed system should be tolerant of a massive number of SDOs and failures of SDOs. The population of SDOs always changes because they spontaneously enter/leave/roam environment or go down. The already standardized mechanisms for naming, lifecycle, monitoring, fault tolerance etc. will be evaluated to see whether they are suitable for the requirements of super distributed system.

3.4.6. Performance

The performance of a system should be maintained even when available resources are limited. Communication between SDOs or processing by devices should assure real-time requirements for an application service. At least, the performance should be monitored and evaluated to see whether or not it is being met. Power should be supplied to devices represented by SDO by appropriate means according to their purpose and mobility. For example, by a conventional AC connector, field network, or radio waves like those supplying power to an IC tag. Memory limitations of devices represented by SDOs and bandwidth limitations of the network connecting SDOs necessary to execute application services are also examples of resource limitations.

4. Reference architecture

4.1. Structure

This section introduces the reference architecture, which follows the idea of modeling any certain piece of hardware or software as CORBA objects, defining their ways of interworking, and specifying standardized mechanisms for controlling them. The architecture contains three functional blocks, as shown in Figure 2, which are responsible for building the infrastructure of SDO. The main task of the SDO DSIG is to define the interfaces between these functional blocks.

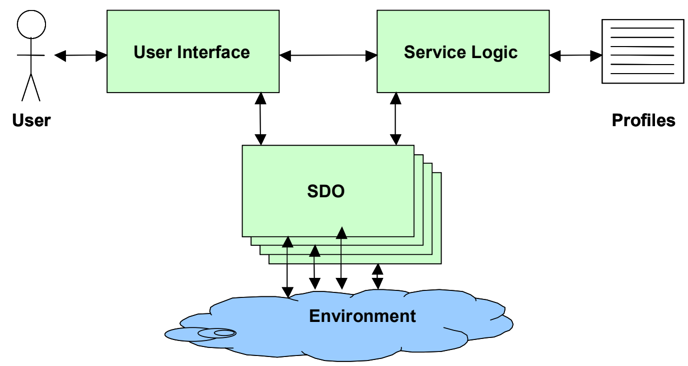

Figure 2: Functional components in the SDO Reference Architecture

The underlying environment of the SDO Reference Architecture can consist of several kinds of software and hardware. The actual application domain determines what specific technologies are available in the environment.

An SDO is a logical representation of a certain piece of hardware or software that provides well-known functionality. Using a standard interface, it wraps the proprietary implementation of hardware or software running in the underlying environment. The SDO DSIG will define this standard interface. A lot of companies and standard bodies are trying to build such unified interfaces for specific application domains (e.g., JINI, UPnP, and HAVI). The SDO DSIG will try to establish an abstract interface that can cope with different application domains (e.g., telecommunication and home automation).

There are two functional components that make use of SDOs: User Interface and Service Logic. The User Interface is responsible for human-machine-interaction (HMI) by adapting it selves on one hand to user preferences and terminals and on the other hand to different application services provided by SDOs . The HMI can be processed in many different ways (e.g., graphical, tactile, or speech-based). Therefore, this block must provide generic user interaction that can be mapped to interactions that are well known to the SDOs. A User Interface can adapt automatically to different kinds of SDOs. From its perspective, accessing a specific SDO means exploring its functionality and providing it in a self-adapting user interface. To provide self-adapting user interfaces, SDOs must provide a generic description of their interaction capabilities.

Service Logic is the functional component that is responsible for defining the behavior of SDOs. Using profiles, where user preferences and service parameter can be stored, this block decides how an SDO has to behave in a certain condition. The service logic can be parameterized during service setup as well as in an interactive process via the User Interface. A Service Logic object accesses an SDO via a well-defined interface. That makes it possible to control SDOs from a Service Logic object.

The profiles which have to be taken into account by Service Logic describe different kinds of information. Therefore, several types of profiles are necessary: e.g. User Profile, Service Profile, and SDO Profile.

A User Profile is responsible for the management of personal preferences and service subscriptions and permission for a certain user of an SDO environment. A Service Profile stores the technical Application Service parameter that can be used by the Service Logic to instruct SDOs for an Application Service. A SDO Profile stores the technical parameters a certain SDO has to be parameterized with. The kind of information is very implementation specific and depends on the services that has to be provided by an application scenario. Nevertheless the SDO DSIG will investigate how such opaque Profiles can be exchanged or accessed in a flexible way.

Beside the information that has to be stored in the profiles, also the location, where profiles have to be stored depends on the implementation. It is possible to host the complete set of profiles inside an SDO. On the other hand there can be other distributed objects that manage the profiles and provide necessary information on demand to Service Logic objects.

Service Logic is further responsible for:

- Application Service management: plug-n-play of SDO, context recognition, and contention resolution
- Application Service execution

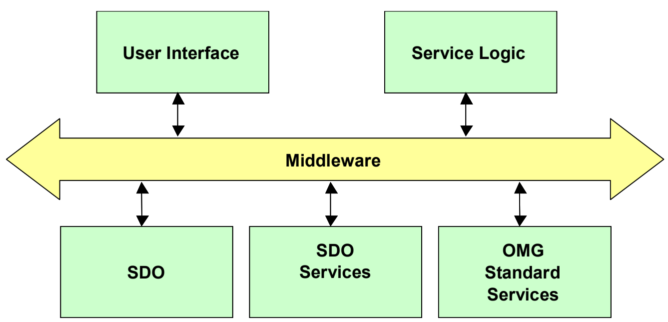

Figure 2: Functional structure.

The SDO DSIG will investigate how CORBA services can support SDO. On the other hand, SDO services will be defined that cope with the special functionalities of SDO like:

- Resource management: Logical resources(e.g. SDO directory) and physical resources(e.g. state of charge of devices, QoS of network connecting SDOs).
- SDO discovery
- Device alternation according to resource usability, performance, etc. by continuous monitoring of device status

4.2. Information model

As described in section 3, an application service is provided by multiple SDOs and an SDO is used for multiple application services. A group of SDOs, each providing application services, can be composed to provide one application service. Because Service Logic represents abstract logic, SDOs, lifecycle, and ownership should be resolved and managed based on the application service instance.

The information model for SDO provides a meta-model for SDO resolution and application service management. It contains two elements:

- SDO – a piece of software to provide an application service (i.e., a smart sensor, PDA, cellular phone, information appliance, etc.)
- Group – a set of SDOs that mutually cooperate for a specific purpose. A group of SDOs can act as one SDO, offering the sum of application services the contained SDOs provide. As group can act as a SDO, a group can contain other groups.

Figure 3 shows the SDO information model.

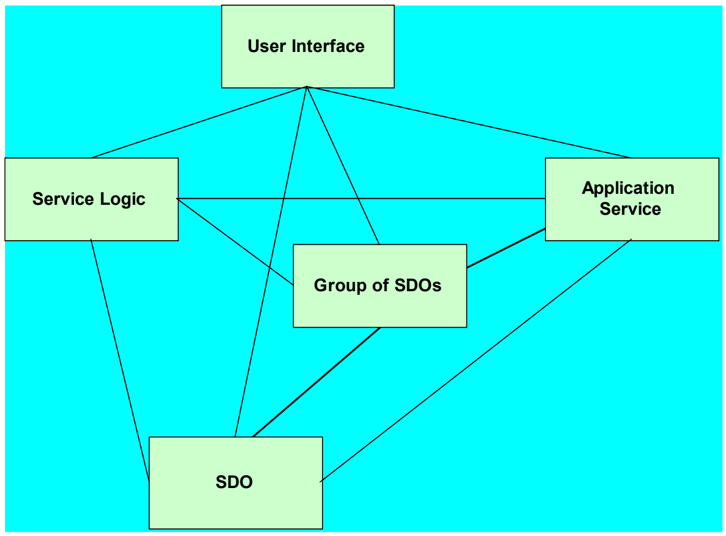

Figure 3: Information model.

The super distributed aspect is directly reflected by the information model as nearly all information objects are connected and the cardinality is in n in the most cases.

5. Requirements for CORBA – RFP candidates –

5.1. SDO interface

The SDO interface provides standard access interfaces to Super Distributed Objects (SDO). Super distributed systems incorporate massive numbers of objects (SDOs) that interact with each other in an ad hoc manner. SDOs represent heterogeneous resources including hardware devices and software components.

Specifically, this RFP solicits proposals for the following:

- Interface to enable unified access to massive heterogeneous resources
- Resource data model representing the heterogeneous resources

5.2. Other candidates

- SDO Discovery
- Conflict resolution and synchronization of SDO access
- Security framework for SDOs
- Description of SDO capabilities and examination interface
- Group formation of SDOs
- User interaction with SDOs
- Ad hoc relationship management of SDO topology

6. Roadmap

The first RFP should concentrate on the SDO interface..

Based on the SDO interface, follow-up RFPs should be targeted on SDO service and Service Logic as described in Section 4 and 5.

The SDO interface RFP is assumed to be scheduled for issue in September 2001. The initial submissions are expected to be received in March 2002. The issuing of follow-up RFPs can start in 2002.

7. Other related standards and efforts

There are a number of ongoing efforts to develop embedded systems around the world. The proposed framework comprises, is implemented on, or can cooperate with these technologies. Related works are listed below.

7.1. Open Communication Platform

- Bluetooth: Ref. http://www.bluetooth.com
- Radio Frequency Identification (RF-ID): Ref. ISO/IEC10536, 14443, 15693
- Wireless Access Protocol (WAP): Ref. http://www.wapforum.org
- Home Network, HAVi, Echonet, etc.: Ref. http://www.havi.org, www.homerf.org

7.2. Discovery and Self-organizing protocols

- Jini: Ref. http://www.sun.com/jini/.
- Universal Plug and Play: Ref. http://www.upnp.org.

7.3. Common Interfaces

- OSGi: Open Service Gateway Initiatives, http://www.osgi.org
- JTRS: Joint Tactical Radio Systems, SRA:http://www.sdrforum.org/, SCA: http://www.jtrs.sarda.army.mil/docs/

8. RFI Responses

8.1. RFI overview

A Request For Information(RFI) entitled “Super Distributed Objects: initial survey” (sdo/00-06-05; http:www.omg.org/pub/docs?sdo/00-06-05) was issued in June 2000. This RFI solicited information on requirements and infrastructure technologies for super distributed objects. The information provided will be used by the OMG to develop:

- This white paper
- A technology adoption roadmap
- A series of Requests For Proposal (RFPs) in this area

The goal of this RFI was to gather information in this area, rough out a framework, and seek technologies and interfaces required for the coming new service system utilizing the computer power pervading the physical environment around us. Its scope included, but was not limited to, the following areas:

- System Requirements
    - Trends: Market and technologies
    - System structure
    - System features
- Information model
    - Service model
    - Resource model
- Infrastructure & Environment capabilities
    - Plug-n-Play
    - Social networking of SDOs
    - Service context resolution: Criteria for service matching
    - Ownership/privacy management
    - Application protocol

8.2. Response lists

No.	Title	Responder	Doc. Number
1	What is ‘super’?	Univ. of Tokyo	sdo/00-09-02
2	Bio-network and its social directory service	UCI	sdo/00-09-03, 00-09-06
3	I-centric Communication	GMD Fokus	sdo/00-09-04, 00-11-01
4	AYA: An architecture for environment-aware service	Hitachi Ltd.	sdo/00-09-05
5	The Communicating Mobile Object (CMO) Project	LIFL, GemPlus	sdo/00-11-02
6	Bionet System Architecture: An Approach to Leverage Super Distributed Object Environment using Biological Concepts and Mecha-	UCI	sdo/00-12-06

	nism		
7	Adaptive Networking Architecture for Service Emergence (ANA-SE)	NTT	sdo/00-12-01
8	An Overview of the Software Defined Radio Architecture	Rockwell Collins, BAE systems, Motorola, Raytheon	sdo/00-12-05

8.3. Abstracts

8.3.1. What is 'super'?

http://cgi.omg.org/cgi-bin/doc?sdo/00-09-02

The down-sizing of computers with high-performance and low-cost as well as increasing network performance results in the need for a many-to-many communication model for massive computers. 'Super' means an incomplete, and hence, extensible feature enabling growth and plasticity. A server does not know all object members, so if a client is in a network that includes a server and the server cannot answer a request, the client should ask around the network. An object presents its own information or ability dependent on a request or a requested object, because in the super distributed objects environment, each object may have many functions and there are too many objects that have the same function. There are many interpretations of IDL, which depend on who wants to use the object.

8.3.2. Bio-network and its social directory service

http://cgi.omg.org/cgi-bin/doc?sdo/00-09-03, 00-09-06

The Bio-networking Architecture is based on the idea that by adopting biological concepts and mechanisms, network services and applications can also be scalable, evolving, secure, survivable, and simple. In the Bio-Networking Architecture, a service or application is implemented by a distributed, collective entity called the super-entity. A super entity is composed of multiple autonomous entities, each of which is called a cyber-entity. This is analogous to the concept of a bee colony consisting of multiple bees. This response presents Social Directory Service, in which cyber-entities establish relationships with each other, forming a social network on which queries for resources are routed and resolved.

8.3.3. I-centric Communication

http://cgi.omg.org/cgi-bin/doc?sdo/00-09-04, 00-11-01

The answer from GMD-FOKUS to the OMG Request for Information: "SDO DSIG Super Distributed Object" contains three parts: basic statements, I-centric communication, and possible tasks for SDO.

The basic statements reflect GMD FOKUS' ideas on Super Distributed Objects. Those statements describe the need for SDO regarding ongoing developments and significant trends in the area of telecommunication, home automation, and car control.

I-centric communication is introduced to outline an architectural framework that models personal communication spaces for different individuals. That architecture may profit from specifications that the SDO DSIG is working on, because of a similar approach to have all entities of the real world modeled as CORBA objects. A common approach to specify and manage super distributed objects may improve I-centric communications.

The task sections, outlines problems the SDO DSIG could be interested in. This section could be used to set up the roadmap for SDO in terms of further RFIs and RFPs.

8.3.4. AYA: An architecture for environment-aware service

http://cgi.omg.org/cgi-bin/doc?sdo/00-09-05

The response from Hitachi describes the trend of Super Distributed Objects from the view-points of both technology and applications in home/office, e-life, industry, transportation. The pervading computer power in the social infrastructure around humans is enabling environment-aware services and standards are needed for them. AYA is a basic concept for Super Distributed Objects to twill a service by weaving artifacts in the physical world. It also gives a reference architecture and typical functions to handle a massive number of and scalable objects in devices densely embedded in the environment, and weak dependent and plastic relationships among them. This response also describes system requirements in SDO, an information model, and some technical extensions to CORBA to support the properties of SDO. This response was and will be used to describe the roadmap of SDO SIG.

8.3.5. Communicating Mobile Objects (CMO) Project

http://cgi.omg.org/cgi-bin/doc?sdo/00-11-02

The LIFL/Gemplus response describes the Communicating Mobile Objects (CMO) project. This aims to design and test a communication platform dedicated to mobile objects. Targeted entities are objects such as electronic tags and PDAs or laptops with a wireless communication layer. This response presents the CMO architecture, the organization of objects using roles, and some ideas about security and ownership/privacy management.

8.3.6. Bionet System Architecture: An Approach to Leverage Super Distributed Object Environment using Biological Concepts and Mechanism

This talk overviews the Bio-Networking Architecture project that has been conducted at UC Irvine. It presents the current design and implementation of a Bionet system platform, which is a deliverable of this project.

This platform is a biologically-inspired system infrastructure designed to support highly distributed applications. It incorporates biological concepts and mechanisms such as species diversity, replication, reproduction, mutation, crossover, social networking, energy gain/consumption/exchange, and pheromone emission. This talk shows how these metaphors work well in designing highly distributed systems.

8.3.7. Adaptive Networking Architecture for Service Emergence

This response introduces the radically new paradigm of the Adaptive Networking Architecture for Service Emergence (ANA-SE). In this architecture, services are implemented by a collection of multiple autonomous entities called cyber-entities, which are fully distributed over a network. They have functionality related to their service and follow simple behavior rules (e.g., migration, reproduction, energy exchange, mutation, death) similar to biological entities. In the Adaptive Networking Architecture, useful emergent behaviors (e.g., scalability, adaptation, evolution, security, survivability, and simplicity) result when individual cyber-entities interact.

8.3.8. An Overview of the Software Defined Radio Architecture

The SRA is the commercial adaptation of the Software Communications Architecture (SCA) specification sponsored by the Joint Tactical Radio System (JTRS) program under contract with members of the Modular Software-programmable Radio Consortium (MSRC). SRA adaptation is an ongoing process within the Software Defined Radio Forum (SDRF).

Focus of the SRA:

- Specifies a common framework to build-up, configure, connect, and tear-down distributed, embedded radio applications

Specifies software interfaces to support the installation and use of distributed applications to support flexible, re-programmable communication capabilities.