SOFTWARE COMMUNICATIONS ARCHITECTURE SPECIFICATION

APPENDIX A: GLOSSARY

FINAL / 15 May 2006

Version 2.2.2

Prepared by:

REVISION SUMMARY

Version	Revisions
1.0	Initial Release
1.1	no changes
2.0	no changes
2.1	no changes
2.2	no changes
2.2.1	no changes
2.2.2	Updated

APPENDIX A GLOSSARY

A.1 ABBREVIATIONS AND ACRONYMS

Abbreviation	Definition
AEP	Application Environment Profile
API	Application Program Interface
CF	Core Framework
CORBA	Common Object Request Broker Architecture
DCD	Device Configuration Descriptor
DMD	DomainManager Configuration Descriptor
DPD	Device Package Descriptor
DSP	Digital Signal Processor
DTD	Document Type Definition
FPGA	Field Programmable Gate Array
GPP	General Purpose Processor
HCI	Human-Computer Interface
ID	Identification, Identifier
IDL	Interface Definition Language
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IOR	Interoperable Object Reference
ISO	International Standards Organization
N/A	Not Applicable
OE	Operating Environment
OMG	Object Management Group
ORB	Object Request Broker
OS	Operating System

Abbreviation	Definition
OSI	Open System Interconnection
POSIX®	Portable Operating System Interface
SAD	Software Assembly Descriptor
SCA	Software Communications Architecture
SCD	Software Component Descriptor
SPD	Software Package Descriptor
SRD	Support and Rationale Document (for the SCA)
SW	Software
UML	Unified Modeling Language
UUID	Universally Unique Identifier
XML	eXtensible Markup Language

A.2 DEFINITIONS

Application

The SCA defines an *Application* interface class that provides the interface for the control, configuration, and status of an instantiated application. An *Application* controls its components and establishes connections to other applications.

application

Generically, an executable software program which may contain one or more modules. Within the SCA, an application consists of one or more software modules which implement the Base Application Interfaces and which are identified within a Software Assembly Descriptor file. When loaded and executed, these modules create one or more components which comprise the application.

Software designed to fulfill the needs of a user.[1]

ApplicationFactory

An instantiation of the *ApplicationFactory* interface is used to create an instance of an application. The domain manager creates an application factory for each Software Assembly Descriptor that is installed.

® POSIX is a registered trademark of the Institute of Electrical and Electronics Engineers, Inc.

[1] IEEE Standard Glossary of Software Engineering Terminology, Std. 610.12-1990

Application Program Interface

An Application Program Interface (API) is the definition of operations and attributes contained in a set of related interfaces that provide a coherent functional capability.

assemblycontroller

The *assemblycontroller* element of the Software Assembly Descriptor indicates the component that is the main resource controller for an application.

Attribute (IDL)

An IDL attribute is a variable that contains a value of a specific type. Attributes may be declared with read-write or read-only access, and the appropriate get and set operations are generated when the IDL is compiled.

Child Device

A device intended to be strongly associated to a related parent device. See Parent Device.

Client

A component that invokes an operation of another component.

Commercial Standard

A commercial standard is a set of requirements maintained for common use by industry. As used in this specification, commercial standards are available for use without restrictive licensing and are supported by commercially available hardware or software.

Component

A software module or element that conforms to and implements an set of interfaces.

Consumer

A software component that can receive user data traffic.

CORBA Component

A software component that implements one ore more CORBA interfaces. A CORBA component within this specification is described by a Software Component Descriptor.

Core Application

A software implementation of one or more of the Framework Control or Service Interfaces.

Core Framework (CF)

The Core Framework is the set of open application-layer interfaces and services defined within this specification. The CF is to provide the essential ("core") set of interfaces needed to provide an abstraction of the underlying software and hardware layers for software application designers.

Destroy

The act of releasing / terminating a software object or component.

Device

1. Hardware device refers to a physical hardware element (typically a module performing a function or set of functions).

2. The SCA defines a *Device* interface class. This interface is an abstraction of a hardware device that defines the capabilities, attributes, and interfaces for that device.

Device Configuration Description (DCD)

A Device Configuration Descriptor is an element of the Domain Profile that contains information regarding a software component implementing the *Device* interface. It provides information about the children *Devices* when implementing the *AggregateDevice* interface, how to find the domain manager, and the device-specific configuration information.

Device Driver

The low-level software, at the physical layer, that controls the physical interface a device uses for communication, e.g. to a hardware bus.

Device Package Descriptor (DPD)

A Device Package Descriptor is an element of the Domain Profile that contains information about a hardware device. The DPD has properties that define specific information (manufacturer, model number, serial number, etc.) about the device.

Device Profile

The Device Profile is the set of XML files within the Domain Profile which fully describe a hardware device. The Device Profile contains a Device Package Descriptor, a Device Configuration Descriptor, and an optional Properties File. Information about the software associated with this hardware device is found in the associated Software Profile.

Document Type Definition (DTD)

"XML provides a mechanism, the Document Type Declaration, to define constraints on the logical structure and to support the use of predefined storage units. An XML document is valid if it has an associated document type declaration and if the document complies with the constraints expressed in it…The XML document type declaration contains or points to [a] markup declaration that provide a grammar for a class of documents."[2]

---

[2] XML: Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation, 04 February 2004.

Domain

A Domain defines a set of hardware devices and available applications under the control of a single domain manager component.

Domain Manager

An implementation of the *DomainManager* interface, a domain manager manages the complete set of available hardware devices and applications. It is responsible for the set-up and shut-down of applications and for allocating resources, devices, and non-CORBA components to hardware devices.

Domain Profile

The hardware devices and software components that make up an SCA system domain are described by a set of XML files that are collectively referred to as a Domain Profile. The domain manager uses the Domain Profile to build its internal information base from the descriptions of the individual hardware devices, software components, and application assemblies under its control.

Event Service

The Event Service is a CORBA service that decouples the communication between components. The CORBA Event Service defines two roles for components: the producer role (produce event data) and the consumer role (process event data). Event data are communicated between producers and consumers by issuing standard CORBA requests.

Event Channel

An Event Channel is an intervening component that allows multiple producers to communicate with multiple consumers asynchronously. An event channel is both a consumer and a supplier of events. Event Channel is the intermediary between the components (producers) being changed and components (consumers) interested in knowing about changes. Event Channels that provide change notification can be general purpose, well-known components (Incoming and Outgoing Domain Management Event Channels) that are run as part of a domain-wide framework or specific-to-task components (e.g., temporary Event Channels that are created at application deployment).

Host

A host is a computer/processor and/or software application that provides services to one or more elements connected to it. These services may include, but are not limited to, network access, program loading, database storage, and HMI. The element or elements connected to a host may be hardware elements (e.g. FPGAs), processing elements (e.g. DSPs), or a combination of elements (e.g. a JTRS radio).

Incoming Domain Management Event Channel

Incoming Domain Management Event Channel is an event channel that is internal to the domain and is used by domain's components to send events to the domain management components (*Application*, *ApplicationFactory*, *DomainManager*).

Initialize

The operation of setting a component to a known initial state.

Name

A user-friendly label such as the name used in DTDs of the Domain Profile.

Outgoing Domain Management Event Channel

Outgoing Domain Management Event Channel is an event channel that is external to the domain and is used by external domain's components (e.g., HCI) to receive events by domain management components (*Application*, *ApplicationFactory*, *DomainManager*).

Parent Device

A parent device uses the *AggregateDevice* interface and is composed of one or more child devices. The parent device and its children are strongly associated and have the same lifetime (i.e. removal of the parent device removes all child devices).

Port

A *Port* interface identifies a source /consumer (Provides Port) or a sink /producer (Uses Port) for data and/or commands.

Primitive

An abstract, implementation-independent representation of the interactions between service users and service providers.

Private

As used in the SCA, a proprietary interface definition.

Producer

A software component that can supply user data traffic.

Profile Descriptor

A Profile Descriptor is an element of the Domain Profile that contains an absolute pathname for either a Software Package Descriptor, Software Assembly Descriptor, *DomainManager* Configuration Descriptor, or a Device Configuration Descriptor.

Properties Descriptor

A Properties Descriptor is an element of the Domain Profile that contains information about the properties applicable to a software package or a device package such as configuration, test, execute, and allocation types.

Property

An SCA Property is a variable that contains a value of a specific type. Configuration Properties are parameters to the *configure* and *query* operations of the *PropertySet* interface. Allocation Properties define the capabilities required of a *Device* by a *Resource*.

Public

As used in the SCA, an open, publicly defined, non-license bearing interface definition.

Release (from the CORBA Environment)

When a CORBA object is released, it is no longer able to process object requests; its CORBA object reference unavailable to other objects. A release is analogous to the POA concept of deactivation. When a server object is deactivated, the association between the CORBA object and its implementation is removed. In the SCA, a component is removed from the OE and OE resources consumed by a component are returned back to the OE. For CORBA components, this includes deactivation. After a component is removed from the OE, a client is unable to communicate with the component.

Resource

A software component that implements the SCA defined *Resource* base application interface. All visible SCA-conformant components of a user application must implement the *Resource* interface.

Service Applications

Service applications (or services) are software programs running in the system that provide functionality available for use by other applications. Services are not defined by a particular interface but are recognized by within a domain by use of the SCA-defined *registerService* operations.

Software Assembly Descriptor (SAD)

A Software Assembly Descriptor is an element of the Domain Profile that contains information about the components that make up an application.

Software Component Descriptor (SCD)

A Software Component Descriptor is an element of the Domain Profile that contains information about a specific SCA software component (*Resource, ResourceFactory,* or *Device*).

Software Package Descriptor (SPD)

A Software Package Descriptor is an element of the Domain Profile that identifies a software component implementation(s). General information about a software package, such as the name, author, property file, and implementation code information and hardware and/or software dependencies are contained in a Software Package Descriptor file.

Software Profile

A Software Profile is a set of Domain Profile files which pertain to a specific SCA component. All software profiles for CORBA components include a Software Package Descriptor and a Software Component Descriptor (as well as optional Properties Descriptor files), but the other files contained in the profile depend on the SCA component in question. The profile for an application will include a Software Assembly Descriptor, a domain manager profile includes a DomainManager Configuration Descriptor, and a profile for a device manager will include a Device Configuration Descriptor.

Waveform

A waveform is the set of transformations applied to information that is transmitted over the air and the corresponding set of transformations to convert received signals back to their information content.

Waveform Application

A waveform application is the collection of software elements (modules or components) which perform any or all of the transformations defined for a specific waveform. This may include (SCA) application components as well as Core Applications, Services, and (SCA) devices.