3GPP TR 32.824 V9.0.0 (2009-06)

Technical Report

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Telecommunication management;
Service Oriented Architecture (SOA)
Integration Reference Point (IRP) study
(Release 8)

Keywords

UMTS, management, service

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Contents

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Background and Scope

Service Oriented Architecture (SOA) is gaining acceptance in the IS/IT industry. It promises to manage change [1], automate and simplify IT processes [1], optimize implementation [2], maximize (implementation) flexibility and scalability [3], facilitate integration beyond the enterprise (between companies, between partners and customers) [4], simplify development [5] and maintenance; etc.

IRP (Interface Reference Point) is the predominant standard for wireless network management since 2000. 3GPP developed it with 3GPP2 close collaboration. IRP architecture follows closely with that defined by ITU-T TMN work [6]. Besides publishing the IRP specifications, 3GPP also publishes its IRP methodology (e.g., the guidelines, templates on how to develop, maintain and publish IRP specifications). Today, the IRP specification methodology is being shared and jointly evolved and maintained by consortium of SDOs, such as ITU-T.

We note that the principles of SOA are currently being applied to the field of network management [11,12,16,17].

The purpose of this document is to analyse the IRP architecture and to provide a "gap analysis" on what enhancement is needed for the current set of IRP specifications such that it can claim to have the full set of characteristics of SOA.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] SOA Management and Security

[2] IBM CICS Service Flow Feature enables composition of CICS applications to create CICS business services

[3] SOA/Web services-based applications

[4] Extending the Benefits of SOA beyond the Enterprise, TIBCO

[5] BEA Announces WebLogic 9.2; Award-Winning Family Raises the Bar on SOA Enablement

[6] ITU-T TMN

[7] ROI of SOA, the Network World

[8] Open Group, Service-Oriented Architecture

[9] OASIS SOA Reference Model TC

[10] 3GPP TR 32.809; Feasibility Study of XML-based (SOAP/HTTP) IRP Solution Sets

[11] ETSI TS 188 001: "NGN Management OSS Architecture".

[12] M.3060 Principles for the Management of Next Generation Networks, ITU-T

[13] 3GPP TS 32.111 Alarm IRP IS

[14] S5-080168 TR 32.822 Work on Resource Monitoring Requirement for Itf-N

[15] 32.363 Entry Point IRP

[16] TMF (http://www.tmforum.org/browse.aspx)

[17] IETF (http://www.ietf.org/)

[18] 3GPP TS 32.101 Telecommunication management; Principles and high level requirements.

[19] 3GPP TS 32.102 Telecommunication management; Architecture

[20] 3GPP TS 32.150 Telecommunication management; Integration Reference Point (IRP) Concept and definitions

[21] RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1 http://www.rfc-editor.org/rfc/rfc2616.txt

[22] SOAP Version 1.2 http://www.w3.org/TR/soap12-part0/

[23] WSDL 1.1 http://www.w3.org/TR/wsdl

[24] WSDL 2.0 http://www.w3.org/TR/wsdl20-primer/

[25] REST http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm chapters 5 and 6

[26] RESTful http://en.wikipedia.org/wiki/Representational_State_Transfer#RESTful_example:_the_World_Wide_Web

[27] UDDI Version 3.0.2 http://uddi.org/pubs/uddi_v3.htm

[28] UDDI tutorial http://www.tutorialspoint.com/uddi/uddi_data_model.htm

[29] Using WSDL in a UDDI Registry, Technical Note http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v202-20040631.pdf

[30] SOAP Version 1.1 http://www.w3.org/TR/soap/

[31] WS-I Basic Profile Version 1.1 http://www.ws-i.org/Profiles/BasicProfile-1.1.html

[32] WS-I Basic Security Profile Version 1.0 http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html

[33] WS-I Basic Profile Version 1.2 http://www.ws-i.org/Profiles/BasicProfile-1_2(WGAD).html

[34] WS-I Basic Profile Version 2.0 http://www.ws-i.org/Profiles/BasicProfile-2_0(WGD).html

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply.

None

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CMIP Common Management Information Protocol

COBRA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
FCAPS	Fault, Configuration, Accounting, Performance and Security
HTTP	Hypertext transfer Protocol
IRP	Interface Reference Point
IS	Information Service
ITU-T	International Telecommunication Union- Telecommunications Standardization Sector
NRM	Network Resource Model
OASIS	Organization for the Advancement of Structured Information Standard
OMG	Object Management Group
OOA	Object Oriented Architecture
RPC	Remote Procedure Call
SDO	Standards Development Organization
SNMP	Simple Network Management Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SS	Solution Set
WSDL	Web Services Description Language
XML	Extensible Markup Language

4 SOA

There are multiple definitions of SOA. The Open group SOA definition [8] and OASIS SOA Reference Model [9] each defines its own SOA terms.

NOTE 1: Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains." [9]

Both claim, and correctly so, that their definitions and terms can be applied to technology and business domains.

The Network World provides a crisp description [7] of SOA from a high level view point.

> "SOA is a development (and deployment) methodology encourages sharing of remotely invocable application functions throughout networks. It's a way of doing more with less, where applications can be built more quickly and incrementally, with fewer lines of original code."

NOTE 2: The two words in bracket are added by author of this paper

SOA is not unlike other existing development and deployment methodology such as 1980's (remote procedure call) RPC, 1990's OOA (Object Oriented Architecture), CORBA and the 2000's IRP (see section 6), etc,

One can characterize SOA as follows.

> "An architectural style that provides services which are effectively the boundary of an application and coordinate the necessary response to service clients. This promotes reuse at a coarse granularity thereby encapsulating unnecessary implementation details. Often used in conjunction with existing legacy applications to provide a facade to the functionality these applications provide. In "Patterns of Enterprise Application Architecture", Martin Fowler describes the Service Layer pattern which essentially provides a facade to the domain layer of an application. SOA is commonly implemented through web services, but it is a common misconception that web-services are the only implementation of SOA. RPC, DCOM, and many others also provide implementations of SOA."

One can also characterize SOA as follows.

> "The upside of SOA is that the marginal cost of building new applications will continue to drop as the service-reuse rate climbs. The catch is that there's a significant ramp-up cost, because adopting an SOA means you're going to need to rethink many of your traditional approaches to application modeling, development, integration, deployment and management."

5 IRP

3GPP developed the IRP set of specifications since 2000. The first set of network management services standardized are based on those defined by ITU-T (for the management of the wire-line networks), namely, the fault, configuration, performance and security of the ITU-T defined FCAPS network management domain services.

Major characteristics of IRP are:

5.1 Business logic independence

The IRP network management services are designed with business logic independence in mind. For example, Alarm IRP network management service is focused on how to provide network fault management services to its users (applications). How the applications would make use of such services is not part of the Alarm IRP design. Various applications, such as trouble ticketing system, network planning tool, call tracing monitor system, routing table generator, large network control room display system, of different functionalities and purposes can make use of the same Alarm IRP network management services.

5.2 Network management domains independence

The IRP network management services are based on that defined by ITU-T. Identical to ITU-T case, the IRP network management services are domain specific. For example, the Alarm IRP specifically deals with a particular network management domain, i.e. the fault management domain. The Configuration IRP specifically deals with the configuration management domain. They are independent of each other, even though both IRPs deal with the same set of managed nodes.

5.3 Protocol and Data independence

The IRP network management services are provided by IRPAgent and consumed by multiple IRPManager(s). This service provisioning and consumption are realized by invocation of sets of standardized operations, notifications collectively called protocols. 3GPP standardized this set of protocols. In addition, 3GPP standardized the data carried by these protocols as well. The set of IRP specifications regarding protocol semantics and syntax are called the Interface IRPs. The set of IRP specifications regarding data semantics and syntax are called the Network Resource Model (NRM) IRP. The NRM IRP contains information representing the 3GPP defined managed nodes. The Interface IRP (such as Basic Configuration Management IRP) and NRM IRP (such as GERAN NRM IRP) are independent of each other. This independence allows other SDOs to use 3GPP Interface IRP (unchanged) for managing their own managed nodes (not 3GPP defined managed nodes). In other words, the SDO can recommend the use of 3GPP Interface as is (unchanged) and design its own set of NRM IRPs representing the managed nodes defined by the SDO. Currently, 3GPP2 is such SDO using such approach.

5.4 Technologies independence

Within each Interface or NRM IRP, there are 3 levels of specification. The first level is the Requirement level. The second level is the Information Service (IS) level. This level specifies the semantics (the meaning) of the operations, notifications, operation parameters, notification parameters, class and class attributes. The third level is the Solution Set (SS) level. It specifies the syntax of the constructs (e.g. operation parameter) defined by the corresponding IS specification. Also, where appropriate, this level also specifies the call model that is dependent on specific technology. For example, in Alarm IRP, the alarm list is returned as the return parameter of the `get_alarm_list` method in CORBA SS while it is returned as notifications in CMIP SS. The three-level hierarchy reflects the relative stability of the Requirement level specifications, IS level specifications and SS level specifications. For example, the alarm management requirements, such as to retrieve a list of active alarm, have been stable for a long time. The operation to retrieve a list of active alarm (i.e. the IS level specification) is also relatively stable. However, comparatively speaking, the syntax or the technology to implement the retrieve operation have changed over time, e.g. from ITU-T CMIP syntax, to SNMP syntax, to CORBA technology, to JAVA and lately, to XML technology, etc.

It is noted that 3GPP has concluded a study [10] on XML-based (SOAP/HTTP) IRP Solution Sets. The conclusion of that study is in Annex A for ease of reference.

6 Relation of SOA and IRP

This clause presents a high level analysis of the 'gap' between attributes of SOA and IRP architecture.

6.1 Reuse, composability, componentization, and interoperability

IRP has this attribute. From specification writing viewpoint, IRP follows the above stated attributes. From implementation viewpoint, IRP does not mandate these attributes since IRP implementation is considered outside 3GPP standard.

6.2 Compliance to standards (both common and industry-specific)

IRP is a set of standard published by 3GPP. The IRP specifications themselves recommend use of available industrial standard where applicable, and will not re-specify them.

6.3 Services identification and categorization, provisioning and delivery, and monitoring and tracking

IRP provides set of network management services that is broadly categorized along network management domain (see 5.2). Each IRP network management service, e.g. service provided by Alarm IRP, service provided by Bulk Configuration Management IRP, is identified by an identifier called "IRP document version number string (IRPVersion)" (see section 3 of [13]) whose scope of uniqueness is within 3GPP. The IRPVersion also categorize the network management service since it unambiguously identified the 3GPP IRP specification title that bears the name of the network management service.

IRP today have not addressed the (network management service) provisioning aspect. For example, IRP does not provide a standard means for new network management service instances to register itself in a Registry so that a potential IRPManager can discover the newly provisioned network management service. For discussion on various aspects of Registry/Repository, see Annex C.

IRP today does not provide a standard means to monitor and track the performance of the deployed network management service, e.g. AlarmIRP, BulkCMIRP, etc. However, 3GPP SA5 have begun the study [14] on this area.

6.4 Service encapsulation

This attribute requires the service to hide internal implementation, internal reference to other services and logic from external service consumer views.

One of the intent of this paper is to promote and encourage 3GPP SA5 to start work on a new SS based on SOAP. See item 1 of Annex A. This new set of SS, if agreed to be designed and specify, should be SOA 'compliant'. We do not suggest spending effort to 'encapsulate' existing and future CORBA SS. For discussion on technology specific aspects, see Annex D.

6.5 Service loose coupling

This attribute requires services to maintain awareness of each other while minimizing dependences in the service relationship

IRP has this attribute. IRP maintains loose coupling in the following ways today.

The IRP notification system is one that provides the decoupling. Notification producers need not be aware the notification consumer identities. Notification producer produces whenever appropriate. Consumer decides when it wants to be notified. They both need not to be "on line" at the same time.

All IRP interactions, except that of notification reception, do not require state or session concept, e.g. an interaction can be implemented/described regardless of the outcome of the previous interaction.

6.6 Service contract

This attribute requires services offered to adhere to a well defined agreement describing among other things, the (kind of) services offered, the protocol to access those services and any related constraints The syntax of the description is also well defined so that the service contract can be read by service consumer processes.

IRP has this attribute.

The 'contract' for IRP today is specified by the IRP interactions in terms operation, operation parameters, notification, notification parameters, IOC and IOC attributes. The IS-level specifies the semantics of the constructs mentioned above. The SS-level specifies the syntax of constructs mentioned. In addition, it also specifies the bindings in that the CORAB SS implies the binding is CORBA technology while CMIP SS implies the binding is CMIP technology. Furthermore, each type of SS, e.g. CORBA SS, also specifies the transport technology used.

However, since the W3C WSDL is well established for describing web based services therefore, we will need to investigate if WSDL is useful/suitable for our proposed SOAP SS, see section 7 WID Proposal One), we would like to investigate if the IS level specification (see 5.4) can effectively somehow incorporate WSDL way of writing contract (i.e. service offerings). The mapping of the logical contract (bindings) to SOAP or to CORBA and the mapping of the physical contract (transport) to HTTP or IIOP, are matter of the SS-level. This is to avoid two ways (i.e. the WSDL way and the current IRP IS-level way) of describing a contract at the IS-level and SS-level.

It is noted that OMG has begun a process to determine if there is a need to define the so-called non-functional characteristics, such as cost, scalability, in SOA context. Once OMG has decided that it is relevant to specify the non-functional characteristics in SOA context, we will examine if specification of these non-functional characteristics are applicable in 3GPP SA5 SOA IRP context.

6.7 Service abstraction

This attribute requires the services offered to be described in the service contract and that the internal implementation and logic of the service be hidden from service consumers

IRP has this attribute. The implementation of the IRP network management services are not standardized (specified by IRP specification). Consumers of such service cannot tell how the service is implemented. The only things visible and observable are those defined by the 'contract', i.e. IRP operations, notifications and IOCs.

6.8 Service reusability

This attribute requires the service logic be divided into service components with the intention of promoting reuse of those components by other services.

IRP has this attribute.

6.9 Service composability

This attribute allows a collections of services be coordinated and assembled to form a composite service.

There is no IRP defined services today that is an assembly of some other IRP services. One IRP service, such as PM IRP, may use the Alarm IRP service to register a threshold crossing alarm. But such interaction or assembly is specified in the specification documentation level (and not at run-time). Whether a product implementation does assembly at run-time or at "factory" is outside the scope of 3GPP standard.

This attribute is at times called Service Orchestration.

6.10 Service autonomy

This attribute requires the service offered to be self-managing and that it has control over the logic it encapsulates

IRP has this attribute.

6.11 Service discoverability

Network management services are offered and consumed. This attribute allows the registration of service offered and also allows detection of service offered by consumers.

IRP does not satisfy this attribute fully today.

IRP today provides the Entry Point IRP [15]. The service allows IRPManager(s) (potential service consumers) to discover the "service identification and categorization", i.e. IRPVersion of the current available services (including the Interface IRP IRPVersion and NRM IRP IRPVersion) and the some aspects of the "contract", i.e. operation and notification signatures. The IRPManager, on reception of these IRPVersion, needs to relate them to specific 3GPP IRP specifications to "discover" the semantics of the services offered.

3GPP in the past has discussion on specifying a mechanism where the "contract", including the NRM schema, can be provided by IRPAgent to IRPManager. See 7.2.

The EP IRP [15] provides the service where IRPManager can discover what and where (the service attachment point) the IRP network services are provided. The EP IRP does not provide a "registration service" where newly deployed IRP network management service can register itself to announce its availability. See 7.2.

6.12 Secured Access

This attribute requires that the access to the various services offered be secured. Securing access to information is critical in SOA environment due to a) its loose coupling of service providers and service consumers and b) the possibilities that the service providers and the service consumers can be situated within their own trust zones or boundaries.

IRP has this attribute today. Operations across the Itf-N can be secured in that, for example, the service requesters' identity can be authenticated, the service provider's identity can be authenticated. Information across the Itf-N can be secured in that, for example, corrupt information can be detected, the information can be encrypted, etc.

7 Recommendations and Conclusions

SOA provides methods for systems development and integration where systems group functionality around business processes and packages these as interoperable services. A SOA infrastructure allows different applications to exchange data with one another as they participate in business processes.

The IRP approach is well suited for operating within a SOA environment (see Section 6). In operator's environment, the FCAPS types of service, supported by the various IRPs such as AlarmIRP, PMIRP, are one of many key inputs to the aforementioned business processes.

It is the conclusion and recommendation of this paper that the various IRPs be evolved further, modified in such that they can fit even better into an SOA infrastructure. Specifically, this paper calls for a new Work Item for Release 9 entitled "SOA for IRP". This Work Item would:

- Enhance 32.101 [18] to include the support of SOA infrastructure as part of its Principles and high level requirements.
- Enhance 32.102 [19] and 32.150 [20] to include the descriptions of a) the SOA infrastructure and b) the relationship between the SOA infrastructure and the IRP Architecture.
- Enhance the relevant Interface IRPs in areas that require amendments for its implementations to improve participation in an SOA infrastructure environment.

Annex A: Conclusion and recommendations of 3GPP TR 32.809; Feasibility Study of XML-based (SOAP/HTTP) IRP Solution Sets [3]

As a result of this feasibility study, the following points are agreed and accepted if 3GPP SA5 were to go ahead with an XML-Based Solution Set type:

- XML-Based Solution Set would not introduce any significant restrictions when compared to existing and used Solution Sets. However, the ability to handle large volume real-time notifications is still to be demonstrated,

- XML-based Solution Sets introduce fundamental and easy implementation concepts such as flexibility and extensibility, and independence on the transport layer. However, the study has reservation about extensibility in terms of backward compatibility and forward compatibility. Currently, W3C has recognized the importance of such capability and is studying this aspect pending conclusion,

- Moving to XML-Based Solution Set is clearly a vehicle to use towards harmonization in the Telecom Industry,

This Feasibility Study indicates that SA5 goes ahead with the use of XML-Based Solution Sets. The introduction of such technologies is beyond 3GPP Release 7. For a start a particular focus should be set on Service Management but this shouldn't be restrictive. Starting with Service Management implementation should allow a smooth transition and to gain a know-how experience.

Annex B: (informative) Table of Solution Sets

This section lists SA5 Release 8 XML related Solution Set. This information is for information only.

Annex C: Registry/Repository aspects

A registry/repository for a SOA can hold meta-data about services. It is conventionally called a registry when it contains links to information about the service. In a SOA, a registry is interesting at design time and runtime; at design time for discovery and description of the service, and at runtime to dynamically discover the service instances and bind to the contract and policies in order to achieve true loose coupling of applications and services. Repository functions where data regarding the service is stored may be desired for a mature SOA, and repository functions may include code versions and documentation for design time and functions such as message store and logging for runtime.

For standardization purposes, we have several aspects to consider regarding registry/repository:

- publication of the standard in a registry
- standardizing that services (i.e. IRP implementations with SOA interface) are to be published in a registry and which information to be published
- runtime/machine discovery mechanism
- runtime/machine binding to the found service

Publication

A standard can be published in a registry. The use of having a standard published in a registry could be for developers and integrators during design time to more easily check that own developed service contracts comply with the standard, as well as identifying compatible/compliant published implementations of those interfaces. Another usage is that services in runtime could be validated for conformance to the standard.

There are (at least) two options for publishing the standard: 1) specify the documentation to be published and specify a registry to where it shall be published, 2) specify the documentation but leave the registry "place" unspecified in the standard..

More importantly, the services implementing the standard can also be published. The usage of having a service published in a registry is apparent from the SOA principles: to be discovered and bound to.

The information about the standard-compliant services to be published could be standardized.

A publish mechanism is needed for runtime or a machine to publish that a service instance that complies with the standard is deployed.

An unpublish mechanism is needed for runtime or a machine to unpublish a service instance that complies with the standard.

Discovery

A discovery mechanism is needed for runtime or a machine to dynamically find deployed service instances that comply with the standard.

Bind to / invoke the service

A binding mechanism is needed for runtime or a machine to bind to a found service instance that complies with the standard. Information to be able to bind to the service instance must be available in the registry. The next step is to invoke the service, which may be done in the same operation as binding to the service. The binding/invocation mechanism itself is outside the concern of a registry.

NOTE: The binding we talk about here is different than a WSDL binding to for example SOAP.

Technology

The IRP methodology of documenting on a technology neutral level in terms of requirements and IS and on a technology specific level in terms of solution set, could be applied to the aspects of publishing, discovery and binding.

UDDI

A UDDI (Universal Description, Discovery, and Integration) [27] is the name of a group of web-based registries that expose information about a business, service or other entity and its technical interfaces (or API's).

Core components of the UDDI data model include the following, for which descriptions are based on [28] and [29]:

tModel data structure: A tModel is a way of describing the various business, service, and template structures stored within the UDDI registry. Any definition of an abstract concept can be registered within UDDI as a tModel (including for example the URI to a WSDL file or another document). When a particular specification is registered in the UDDI registry as a tModel, it is assigned a unique key, called a tModelKey. This key is used by other UDDI entities to reference the tModel, for example to indicate compliance with the specification.

bindingTemplate data structure: Binding templates are the technical descriptions of the web services and represents the actual implementation of the web service. Binding templates can refer tModels. Service vendors use binding templates for their services, and these can refer to standards represented in tModels.

businessService data structure: The business service structure represents an individual web service and uses binding templates.

To access a UDDI, there are two interfaces; the Publisher interface and the Inquiry interface. These interfaces provide for a publication mechanism and a discovery mechanism.

Reference [29] recommends an approach for mapping between WSDL and the UDDI data model. Creating such a mapping, the contract itself can be stored in the UDDI registry. Such a mapping uses also the businessService data structure in addition to the tModel data structure. Therefore, for the WSDLs of the IRPs, such a mapping does not seem suitable since businessService is not to be referenced from a binding template.

Entry Point IRP

The Entry Point IRP (EPIRP) provides some registry capabilities. It provides information about the IRPs that are supported by the IRPAgent that contains the EPIRP. Optionally, the EPIRP provides also information about IRPs that are supported by other IRPAgents.

The standard does not specify how the information gets into the EPIRP, it only specifies how to get information from the EPIRP and how to request information to be released (deleted) from the EPIRP. A notification of changes to the information is also provided.

Annex D: Technology specific aspects

D.1 Background technical information

HTTP

A HTTP [21] message is either a request or a response. A request has a method, either GET, PUT, POST, DELETE or another of the specified HTTP request methods. These methods operate on a resource identified with a URI supplied with the request, and this URI is called Request-URI.

GET requests to retrieve whatever information (in the form of an entity) is identified by the supplied Request-URI.

PUT requests that the enclosed entity be stored as a resource identified with the supplied Request-URI.

POST requests that enclosed entity be handled, and with the supplied Request-URI identifies the resource that will handle the enclosed entity.

DELETE requests that the resource identified with the supplied Request-URI be deleted.

SOAP

SOAP [22] wraps messages bound to SOAP in SOAP envelopes. A SOAP envelope consists of an optional header and a mandatory body. SOAP binds SOAP messages to an underlying protocol.

Binding to HTTP POST for the Request-Response message exchange pattern and HTTP GET for the SOAP Response message exchange pattern is specified as the binding called "SOAP 1.2 HTTP Binding". Other bindings are allowed although not specified by the SOAP specifications.

The "Web Method Feature" in SOAP 1.2 provides methods GET, PUT, POST and DELETE (within the SOAP layer). These are also bound to an underlying protocol, so that a PUT can be bound to an HTTP POST.

SOAP 1.1 [30] only defines binding to the HTTP POST method for requests over HTTP.

WSDL

A WSDL 1.1 [23] document consists of five parts:

types: Constructs specifying the data structures used by the service

message: A construct specifying the data structure that is used in a particular input, output, or fault message (each message references a structure specified in the types section)

portType: A construct defining a set of operations, and for each operation specifies the input, output, and fault messages that are exchanged for that operation. Each operation references the messages described in the messages section. A portType can be used by multiple services in different namespaces.

binding: A construct that binds a service to a portType and protocol. The protocol that can be bound to in a WSDL binding is SOAP 1.1, SOAP 1.2, HTTP Get, or HTTP Post. If bound to SOAP, a choice must be made between the two styles *document* and *rpc* for bindings and operations, and a choice between the outlines *literal* or *encoded* for the body parts. A binding can be used by multiple services in different namespaces.

service: A construct that specifies the physical endpoint that provides access to a particular binding for the Web service. A service *port* defined within the *service*, references a *binding*.

WSDL 2.0 [24] has some changes compared to WSDL 1.1. The *message* section is removed, and *portType* is renamed *interface*. An *interface* defines its input, output, and fault messages, and references the types directly (instead of referencing the messages as in WSDL 1.1). WSDL 2.0 can bind to also the other HTTP operations than HTTP Get and HTTP Post, and also HTTPS operations.

A single WSDL document may contain multiple services.

SOAP based Web services

The interface to a Web service can be represented by a WSDL document. We can say that a WSDL description provides all the information a client needs to use a Web service. This information includes the address, allowable communication mechanisms, interface/portType, and messages. Therefore, from an interface perspective, usage of WSDL for an Interface IRP would ensure that we provide for a Web service based interface.

RESTful Web services

REST (Representational State Transfer) [25] and so-called RESTful [26] Web services is commonly described as an alternative to using SOAP for Web services. One can consider SOA being at a higher level of abstraction than REST, so that also REST can be used as a technology in fulfilling a SOA, similar as SOAP can be used in fulfilling a SOA.

A RESTful architecture is resource oriented and data-centric. The resources are identified with URIs. A RESTful Web services is called so when it is represented with a resource identified with a URI and MIME-encoded data, together with either of the HTTP operations GET, PUT, POST, or DELETE.

For the specification of SOA services, one can specify contracts that consist of resources and data operated on with HTTP operations directly. An option is to specify the contract in a WSDL 2.0 document which binds the service to an interface that uses appropriate HTTP operation.

As an exercise for giving an example and exploring the possibility, let us consider the createMO operation of the Basic CM IRP as a service that we want to expose in a SOA and realize with a RESTful Web service. We can then write a contract for createMO, where we specify the Request-URI,

http://URL_of_IRPAgent/32.607/createMO

and in the contract specify usage of HTTP POST and that the DN of the MO shall be in the HTTP message body. When the IRPAgent receives this HTTP request, it invokes intelligence as addressed by the Request-URI and creates the MO.

But one could argue that the above representation of the createMO service is not particularly RESTful since it involves invocation of an operation in addition to HTTP. A more RESTful method could be the following:

We write a contract for createMO, where we specify the Request-URI,

http://URL_of_IRPAgent/32.607/<DN of the MO to be created encoded with URI syntax>

and in the contract specify usage of HTTP PUT. When the IRPAgent receives this HTTP request, it creates the MO given in the Request-URI.

For many operations of the IRPs, for example subscribe (of the Notification IRP), the mapping to a RESTful representation would be less straight forward.

RESTful services promise great scalability, and an example of a RESTful architecture is the World Wide Web where the capabilities of HTTP are utilized.

One limitation with RESTful services is that they are made for peer-to-peer communication without the use of intermediaries.

Discussion SOAP vs. RESTful for IRPs

For the following discussion, it is assumed that the operations on IS level are kept the same and represent services to be mapped to the solution set level. Then on the solution set level we map the IS level to either a SOAP based representation or a RESTful representation.

SOAP uses the Internet as transport. SOAP can be transported in HTTP or another protocol. One can argue that in the case that SOAP uses HTTP, in this case SOAP uses the Web. But it is only HTTP POST that is used (for requests), not for example DELETE or PUT. Over the HTTP layer, SOAP constitutes a messaging layer, so that instead of merely using HTTP operations, operations such as createMO are encoded in SOAP messages transported over HTTP.

In distinction, RESTful services use HTTP operations as the actual operations of the services. The service would then be represented by the combination of operation (verb) and resource (noun).

In this perspective, usage of SOAP involves an extra layer of operations in comparison to the RESTful approach.

Another perspective is that the semantics of a service is the same with both approaches. If we simplify the operation part in the sense of limiting the operation (verb) repertoire to GET, PUT, DELETE and POST, we will have to add complexity to the nouns for many services. Therefore, the complexity of using SOAP versus RESTful is debatable.

We can observe that there is nothing in SOAP based Web services that are really related to the Web, since the Web is merely used as underlying transport, or if SOAP binds to for example TCP directly then the Internet is used as transport similarly as for CORBA. Therefore, for the discussion of SOA, it seems the most constructive to consider Web services as just a name of the category of services when the service is expressed in form of a WSDL document, rather than presuming any characteristic of Web services that can be associated with the ones of the Web.

To note is that WSDL 2.0 provides the mean of specifying a contract in a standard XML format also for a RESTful Web service.

Given the operations already specified on IS level for the IRPs, it seems more straight forward to map these operations to WSDL with binding to SOAP, rather than representing them as RESTful services. However, if we in evolving to SOA change the IS level, then a possibility is of course to adapt the IS level to be more suitable to be represented as RESTful services.

Web Service Interoperability

The Web Services Interoperability Organization (WS-I) created profiles to solve interoperability issues with Web Services. WS-I Basic Profile 1.1 [31] is finalized and describes messaging based on SOAP 1.1 and service description based on WSDL 1.1. Additionally the WS-I Basic Security Profile 1.0 has been published, which describes interoperable secured communication between Web services [30].

Among other things WS-I Basic Profile 1.1 limits the WSDL binding style to Document/literal and RPC/literal.

WS-I Basic Profile 1.2 [33] is a working group approval draft and includes WS-Addressing that provides transport neutral communication between Web Services. WS-I Basic Profile 2.0 [34] is currently drafted and will include SOAP 1.2 messaging and service discovery based on UDDI 3.0.

In order to avoid interoperability issues and creation of another Web Service profile WS-I profiles should be considered.

D.2 Possible technology choices

The relevant choices are listed in the following table.

Choice #	Approach	Documentation	Binding	WSDL binding style	Underlying protocol
1	WSDL/SOAP	WSDL v1.1	SOAP 1.1	Document/literal	HTTP 1.1 (POST)
2					Unspecified
3				RPC/encoded *(not WS-I compliant)*	HTTP 1.1 (POST)
4					Unspecified
5				RPC/literal	HTTP 1.1 (POST)
6					Unspecified
7			SOAP 1.2	Document/literal	HTTP 1.1 (POST and GET)
8					Unspecified
9				RPC/encoded *(not WS-I compliant)*	HTTP 1.1 (POST and GET)
10					Unspecified
11				RPC/literal	HTTP 1.1 (POST and GET)
12					Unspecified
13		WSDL v2.0	SOAP 1.1	Document/literal	HTTP 1.1 (POST)
14					Unspecified
15				RPC/encoded *(not WS-I compliant)*	HTTP 1.1 (POST)
16					Unspecified
17				RPC/literal	HTTP 1.1 (POST)
18					Unspecified

19			SOAP 1.2	Document/literal	HTTP 1.1 (POST and GET)
20					Unspecified
21				RPC/encoded *(not WS-I compliant)*	HTTP 1.1 (POST and GET)
22					Unspecified
23				RPC/literal	HTTP 1.1 (POST and GET)
24					Unspecified
25	RESTful	WSDL 2.0	HTTP 1.1 (POST, GET, PUT, DELETE)	-	-
26		HTTP 1.1 (POST, GET, PUT, DELETE)	-	-	-

For information: Release 8 of the IRP SOAP solution sets; Generic (32.317), Notification (32.307), Kernel CM (32.667), Basic CM (32.607), Bulk CM (617), Alarm (32.111-7), Performance Management (32.417), File Transfer (32.347), and Entry Point (32.367) have choice 1.

Annex E:
Change history

Change history								
Date	**TSG #**	**TSG Doc.**	**CR**	**R**	**Subject/Comment**	**Cat**	**Old**	**New**
2009-06	SA#44	SP-090300	--	--	Presentation to SA for information and approval	-	1.0.0	9.0.0