DESCRIBING AND SELECTING COMMUNICATION SERVICES IN A SERVICE ORIENTED NETWORK ARCHITECTURE

Rahamatullah Khondoker, Bernd Reuther, Dennis Schwerdel, Abbas Siddiqui, Paul Müller
Integrated Communication Systems, University of Kaiserslautern, Germany
{khondoker, reuther, schwerdel, siddiqui, pmueller}@cs.uni-kl.de

ABSTRACT

Today networks offer communication services ranging from a rather simple and unsecure one to secure and reliable data transmission for communicating on the network. In the future, it is expected that networks will offer a large number of different communication services. With so many services available, determining which service to select and use becomes much more difficult. Here we propose a description schema including an ontology for describing communication services. For service selection a decision making process called Analytic Hierarchy Process (AHP) is utilized which is specially adapted and extended for automatic processing.

Keywords—Ontology, Communication Services, Service Oriented Network Architecture, Future Internet, Next Generation Internet, Analytic Hierarchy Process (AHP)

1. INTRODUCTION

The term 'Service' is widely used but the definition varies from domain to domain. The Organization for the Advancement of Structured Information Standards (OASIS) defines a service as "a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description" [1]. Other definitions exist as well [2, 3].

Services can be either atomic or composite [4]. An atomic service is a service of unbreakable functionality. On the other hand, a composite service consists of a number of atomic services to constitute a complex service. In this paper, service means either atomic service or composite service.

Services are the central design elements of a service-oriented architecture (SOA) which can be defined as "[an enabling] framework for integrating business processes and supporting information technology infrastructure as [loosely coupled and] secure, standardized components — services — that can be reused and combined to address changing business priorities [4]." Three fundamental roles of a service-oriented architecture are Service Providers, Service Consumers and Service Brokers [5] as shown in Figure 1.

The Service Provider produces and supplies the service, the Service Consumer utilizes the service and the Service Broker provides the facility to advertise, search and discover the desired services.

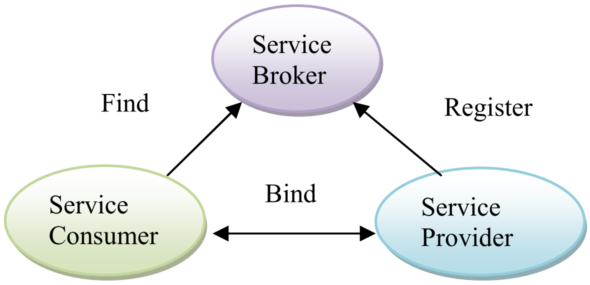

Figure 1. Roles and Operations of SOA

It has been proposed to build networks in the future following the SOA paradigm in order to achieve a flexible and well maintainable network architecture. For example, (SONATE) [6, 8] is a prototype architecture ensuring the SOA paradigm. In this architecture, the authors did not provide a facility for the users/application providers to define their preferences of one effect over another (for instance, success rate over delay, delay over cost) and did not provide a model for selecting the best service from a set of available services based on preferences specified by users/applications.

Searching and selecting a suitable service is not a problem today as there are not many communication services. In other words, the number of communication services in the Internet is limited. For example, an internet-service provider (ISP) offers an infrastructure for using certain services such as connection management, reliability, and routing by utilizing the TCP/IP protocol. Other Service Providers offer their services using various methods such as SOAP, HTTP, SIP, etc.

The future internet will have an even wider selection of communication services to choose from. With such a variety of services, each network user needs to have a way to select the communication service most appropriate for his needs.

Today application requirements are specified by application providers and end-users cannot directly influence those fixed requirements. For example, a user might want to communicate with his friends without worrying about security, but wants to do so affordably. On the other hand, other users might be willing to pay more to have secure communications. Future internet should provide a facility for users to influence their application requirements and thus the requirements will not always be same.

The ability to describe services is crucial. Here it is presented how to describe service from different "services providers", common protocol stacks as well as service of highly dynamic future networks. The central element for service descriptions is a common ontology and the ability to express precedence. It will be shown how the Analytic hierarchy process (AHP) [13] algorithm from decision theory can be adapted to an automatic process of service selection.

The outline of the paper is as follows. Section 2 provides a model for selecting services in the service oriented future network architecture. Use of ontology in this architecture is proposed in section 3. How to describe services in a generic way is shown in section 4. Section 5 illustrates the automatic service selection process by using and extending the Analytic Hierarchy Process (AHP). At last, section 6 concludes the paper.

2. A MODEL FOR SELECTING SERVICES IN THE SERVICE ORIENTED NETWORK ARCHITECTURE

By using the Service-oriented Architecture (SOA) paradigm, a service oriented network architecture (SONATE) has been proposed. As with SOA paradigm, the main three components of SONATE are Service Provider, Service Consumer and Service Broker as shown in Figure 2.

The numbers of providers for communication services are expected to increase in the future. Different providers will offer different types of communication services ranging from services provided by current protocols to fully dynamic composition services.

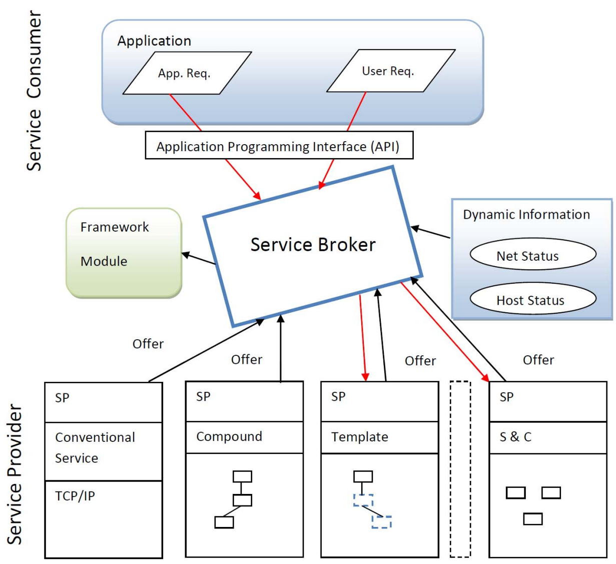

Figure 2. Components of a service-oriented network architecture

2.1. Service Provider

The main functionality of the Service Provider is to offer communication services. Different Service Providers can provide different types of services ranging from a conventional TCP/IP services towards fully dynamic services for the future. For example, one Service Provider might provide already existing communication services, for instance, services provided by UDP/IP which is required for backward compatibility.

Another Service Provider might provide new services which are pre-defined, pre-compiled and pre-composed. Several methods have already been developed for doing the pre-composition. For example, the Netlets approach [14].

A Service Provider might also provide a template for constructing a service which is slightly more flexible compared to the pre-composed approach. Here, certain functionality can be added or deleted based on the needs of the user.

A fully automatic selection and composition (S&C) can be provided by one of the Service Providers which take the requests from the user and compose a service on-the-fly based on the requirements specified by the user.

Taking the number of services and their composition time into account, different types of Service Providers are considered here. The number of services in the conventional Service Provider is limited and can be accessed very quickly whenever necessary. Compound Service Providers which store a lot of pre-composed services might consume time to search and select the appropriate one. The template Service Provider requires computation to compose the appropriate service after getting the requirements. A S&C Service Provider might take very long time to compose a service.

2.2. Service Consumer

Most of the communication requirements are usually specified by application programs. As a result, they are the primary consumer of a communication service. Another important consumer of a service is human. But, human users cannot provide their requirements to the Service Broker directly. They need an application program by which they can specify their demands. Users can specify their requirements by using the Graphical User Interface (GUI) which is usually associated with the application.

The Consumer needs a common Application Programming Interface (API) to make it transparent to him which Service Provider will be selected, and of course a large number of different APIs cannot be supported. It is notable that there may be different APIs for different classes of services, but the API supported by a service could be just one mandatory requirement. Definition of such an API is beyond the scope of this paper.

2.3. Service Broker

Residing in the middle of the Service Provider and the Service Consumer, the Service Broker accepts the user and

Figure 3. An ontology for effects to describe communication services

application requirements from the Service Consumer, receives dynamic information from the host and the network, gets services from the Service Providers and then selects the appropriate service based on the application requirements and dynamic information. For achieving the brokering task, the Service Broker might play a role as a negotiator.

In addition, the broker takes into account that

- There may be a large number of services,
- Some Service Providers need requirements to perform composition and
- Some of them may need a non-neglectable amount of time for composing a service (i.e., add delay to the decision making process)

The different types of Service Providers in the SONATE architecture facilitate consumers (users/applications/nodes) by providing services considering their time and demand. This is handled by using an API between Application and Broker which must provide limits for the selection time. Since the process of "rating" a service, which will be discussed in section 5, uses vector multiplication and can be done quickly, it is possible to promptly handle a large number of services.

After selecting the service, the Service Broker provides the workflow to the Framework Module for execution.

3. ONTOLOGY IN SONATE

In case when the requested service is not available to the Service Broker or template and/or S&C can offer a better service, the service needs to be composed partially or completely on-the-fly. It is impossible to compose a service without its clear and meaningful description. In addition, for matching a requested service with the offered services require explicit description as well.

When more than one Service Provider offers the same service but with different quality attributes (loss ratio, delay), the Broker must decide which provider to select for service provisioning which requires clear description

Mere works [6] have been accomplished concentrating on service description providing semantics for requested and offered services. Comparing the semantics of requests and offerings is performed only by matching strings which reference terms of a common ontology.

We defined an ontology [10] for a list of communication services as shown in Figure 3. Provided that each service in the ontology has a specific clear meaning, all of the services can be described by the Service Provider and can, in turn, be consumed by the Service Consumer.

Using the ontology, we described the name of communication services and their properties. Communication services can have two types of properties: mandatory properties, and optional properties. Each of those types of properties can contain qualitative properties and/or quantitative properties which can be defined by using an ontology. So, it can be concluded that an ontology is essential for describing services in the service oriented network architecture.

4. SERVICE DESCRIPTION

Having different types of properties and their units make the service complex in structure which should be described in a generic way so that it can be handled easily.

The description must not be limited to a fixed ontology; it must be extendable so that new features can also be described in the future without changing the Service Broker.

A service can be described by a set of visible effects [6]. So, a service, S, is

$$S = \{VE_1, VE_2, \dots\dots\dots VE_n\} \quad \dots\dots \text{eq(1)}$$

Services can be either requested or offered. The service that is requested by a user or an application is called requested service, can be denoted by S_R and the service that is offered by the Service Provider is called offered service and can be denoted by S_O.

For getting the optimal service, it is required to compare S_R and S_O and the result can be one of the followings:

Conditions	Action
$S_O < S_R$	User requirements cannot be fulfilled
$S_O \geq S_R$	User requirements can be fulfilled
$S_O = S_R$	Perfect match according to user expectation
$S_O \neq S_R$	No match and user cannot proceed

According to eq (1), comparing of services can be done by comparing visual effects (VEi, where i = 1 to n). The requirements of the Service Consumer can be fulfilled when the following equation is satisfied

$$S_O = \{VE_i\} \supseteq \{VE_j\} = S_R$$

Every individual visual effect VEi can either be a mandatory property, or an optional property. A mandatory property is a property that should be compared at first and must be fulfilled. As the name indicates, an optional property is not required to match exactly but can be negotiated between the application and the provider which is done by the broker. This property is optional and can be omitted as well when fulfilling the property takes an unacceptable amount of time. Both mandatory and optional properties can have qualitative and quantitative properties. Examples of qualitative properties (low/medium/high) are cost and security whereas examples of quantitative properties are success rate and throughput. These properties can be used for the following purposes

- find a service that is appropriate (has all mandatory properties)
- find the best service by optimization of optional properties

When more than one Service Provider offers the same service fulfilling mandatory properties but with different optional property, it is required to calculate or measure the weight of these properties (measured values) to choose the optimal service. The service description helps in the process of service selection by describing those properties in a generic way.

5. SERVICE SELECTION

A client can ask for a service from a set of alternatives specifying the criteria for selection. Analytic Hierarchy Process (AHP) [13] is a mathematical model used to choose one alternative from a given set of alternatives, usually when multiple decision criteria are involved. How this process can be adapted to an automatic process of service selection is shown here.

5.1. Analytic Hierarchy Process (AHP)

AHP, originally proposed by Saaty [13], is a process designed for human decision making. Basically, AHP is used for determining priorities of different alternatives. The flowchart of the process is shown in Figure 4.

The first step of the process is to define a hierarchy. The first and last levels in the hierarchy contain the goal and the alternatives to choose from respectively. One or more hierarchies in the middle contain evaluation criteria.

The second step is to assign pairwise priority to the criteria. The pairwise priority is the preference or satisfaction feelings of one evaluation criteria over another. For defining pairwise priority, a scale between -9 to +9 is used as shown in Figure 5.

The next step is to calculate the overall priority value or priority vector which provides the relative weight among the things or criteria we compare. The calculation detail is beyond the scope of the text.

The subsequent step is to check the consistency of the priority vector by using the method proposed by Saaty. If the vector is not consistent, the next step is to change the pairwise priority of the criteria and repeat the process from the second steps. When the vector is proved to be consistent, the priority vector of the next levels in the hierarchy is calculated.

Except for the first criteria level (i.e., the start level having criteria) in the hierarchy, the priority vector of subsequent hierarchy levels is calculated by multiplying the priority vector calculated from the nearest upper hierarchy which is consistent and the priority vector of the hierarchy.

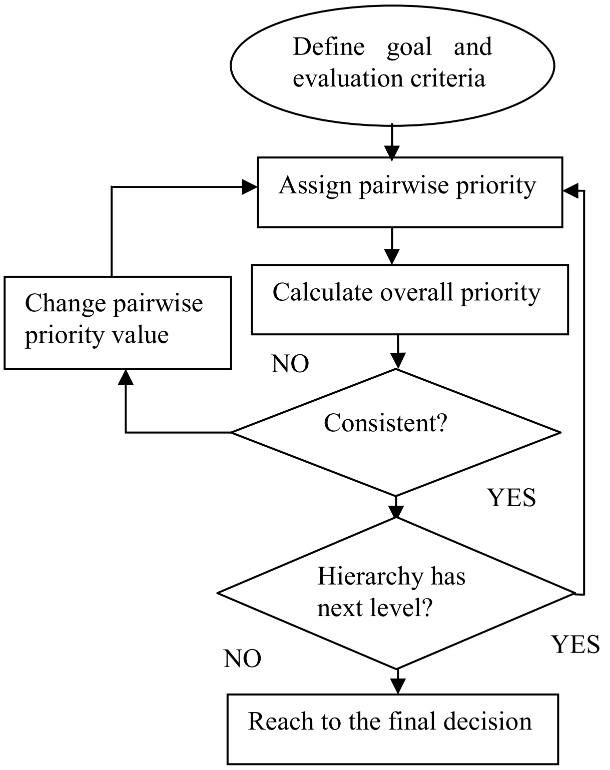

Figure 4. Analytic Hierarchy Process (AHP)

The priority vector of the last hierarchy provides the final decision.

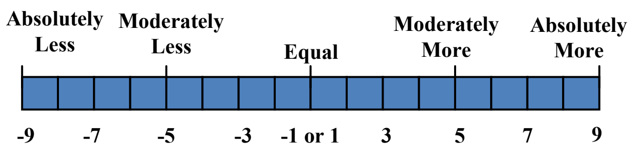

Figure 5. Pairwise comparison scale

The existing AHP process is described in this section. The next section will show how the process can be extended to select a service automatically.

5.2. AHP in service description and selection

AHP can be used for describing services and selecting the best one from available services. The procedures for selecting the best service are

1. Define the effects for selecting a service
2. Assign pairwise priority to the effects
3. Calculate a priority vector of the effects not violating consistency
4. Pass the priority vector of the effects to the broker along with the requirements specified by the user/application.
5. Calculate pairwise priority among the offered services based on the requirements specified by the user and effects provided by the services. This requires a mapping mechanism which cannot be done by AHP. That is why, we propose a mapping mechanism in the section 4.2.1.
6. Calculate priority vector of the offered services preserving consistency
7. Calculate the overall priority vector of the offered services by multiplying the priority vector coming from the application and the priority vector calculated in the offers.
8. Select the service with the highest priority.

The hierarchy for selecting the best service is shown in Figure 7. The top of the hierarchy exposes the goal and the next hierarchy defines the effects that are necessary for achieving the goal. The effects can be either qualitative properties or quantitative properties or both.

Defining a priority is not necessary for mandatory properties because these properties need to match exactly. Instead, the priority needs to be defined for optional properties for achieving the optimal result/service.

The priority is usually assigned by the application (application developer). But, users might influence to define such priorities.

Besides the priority, users/applications need to specify some more values of the effects that are required for the broker to calculate which services in a certain effect should be given more priority than others. These values can be expressed as <effect-name,lb,ub>. Examples of effects are success rate, delay, and cost. The lower bound (lb) and upper bound (ub) are the quantitative limits of an effect where lb is the minimum value of the effect, ub is the maximum value of the effect. <Success rate, 80%, 0> means that the minimum value of the success rate for the application/user is 80%. <Delay, 0, 100ms> means that the maximum required delay is 100ms. <Cost, 5, 5> means that the desired cost is 5.

5.2.1. Pairwise prioritization of services per effect

Defining pairwise prioritization on services for every effect is a critical task, and cannot be provided by the Service Provider which has no idea of the application needs. The question is who then can provide the prioritization of services and how can this be done. In general, this crucial task can be accomplished by using the following process

1. Offerings (given by the Service Provider) describe measured values for every effect.
2. Requirements contain hints for mapping measured values into prioritization as shown in Figure 6.
3. The Service Broker uses the hints of the requirements to parameterize a mapping algorithm.
4. Use the resulting mapping to calculate the prioritization of services with the "measured values" provided by the offerings.

The mapping should have certain properties. Firstly, the mapping must be generic, i.e. not specific to effects or units of measured values. Secondly, the mapping must be monotonic. Thirdly, a liner mapping of measured values to prioritization is not adequate (for instance: delay of 10ms = +9 50 ms = +1 500ms = -9)

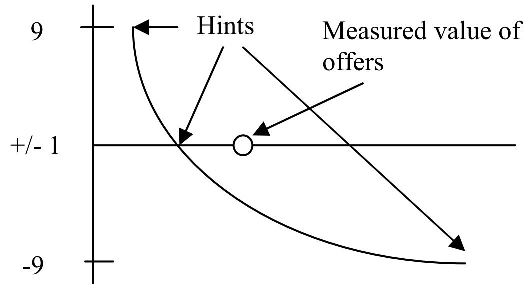

Figure 6. Values measured in terms of hints

An approach for mapping is proposed here to use a monotonic interpolation/extrapolation. In this case, requirements provide value points for interpolation/ extrapolation (must be monotonic). A monotonic interpolation/ extrapolation of these points are used to define mapping. In addition, the specific "measured values" of the offerings are then mapped to its priority.

Let us assume that f () is a function used to define mapping. As an example, considering interpolation, the requirements must contain at least the following two points

- x_0 where f (x_0)= - 9
- x_n where f (x_n)= + 9

If there are measurement values y not within the interval [x_0,x_n], we can extrapolate

- if y < x_0, then f (y) = - 9
- if y > x_n, then f (y) = + 9

Using inter-/extrapolation, a requestor must provide two points but can specify as many parameters as he wants to be more precise.

The aforementioned mapping mechanism facilitates the broker to assign a priority of one service over another for every single effect.

5.2.2. Example Scenario of AHP in service selection

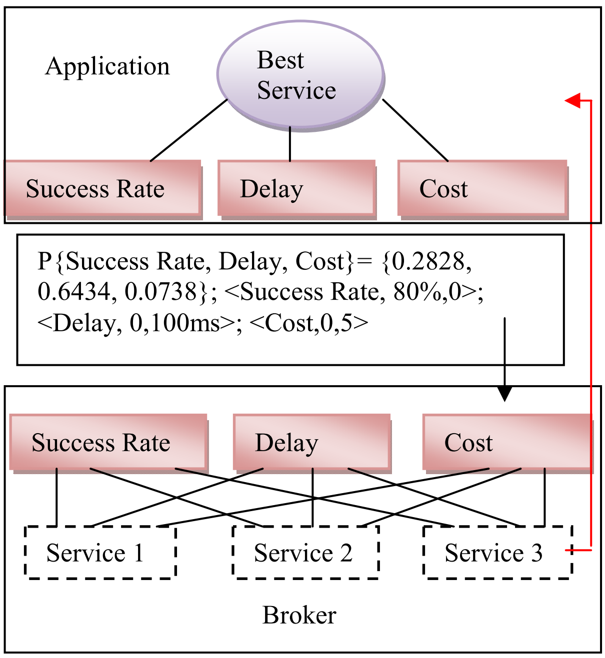

Figure 7. Example scenario for getting the best service

An example scenario for selecting the best service is shown in Figure 7. Here, the application developer defines the pairwise priority as {Success Rate, Delay} = {1,-3}, {Success Rate, Cost} = {5,1} and {Delay, Cost} = {7,1}. Moreover, the expected values of Success Rate, Delay and Cost are specified as shown in the Figure. After calculating the priority matrix, the application sends these values (priority matrix and user/application defined value which is called requirements) to the broker. The broker compares the values of the effects of the offered services with the requirements and then assigns the pairwise priority to the services based on effects. In this case, based on success rate, delay and cost, the assigned pairwise priorities are shown in Table 1 where Service 1, Service 2 and Service 3 are denoted as S1, S2 and S3 respectively.

Table 1. Pairwise priority for different effects

Effects	{S1, S2}	{S1, S3}	{S2,S3}
Success Rate	{-3,1}	{1, -4}	{-2,1}
Delay	{-4,6}	{1, -6}	{-3,1}
Cost	{-3,1}	{1, 4}	{7,1}

Then, the priority vectors are calculated using the values which are shown in Table 2. Finally, the overall priority vector is calculated by multiplying the priority vector received from the application and the priority vector calculated by the broker providing the final result of the priority {Service1, Service 2, Service 3} = {0.11, 0.315, 0.574}. Now, the broker returns the best service (Service 3) to the application.

Table 2. Calculated priority vector for different effects

	{S1, S2, S3}
Success Rate	{0.1226, 0.3202, 0.5571}
Delay	{0.0869, 0.2737, 0.6393}
Cost	{0.2684, 0.6555, 0.0796}

6. CONCLUSION

The number of communication services is expected to increase in the future. Each provider will provide specific types of communication services ranging from services provided by current protocols to fully dynamic composition services. As the number and variety of communication services increase, it is essential to describe them in a generic way so that the Service Consumer can select the most appropriate one. For solving this, the description schema was extended to include an ontology for describing services and their properties meaningfully. Furthermore, the Analytic Hierarchy Process (AHP) was extended to select between similar services according to the client's requirements.

A Service Broker, residing in the middle, takes the requests from the consumer and offered services as inputs and selects the best offered service based on the requests. A schema including an ontology has been described to provide the meanings of the offered services so that client requirements are met.

For selecting the best service by using AHP, it is required to calculate the priority of one service over another. Defining pairwise prioritization on offered services for every effect is a critical task, and cannot be provided by the Service Provider which has no clue of the application needs. Here, we proposed a mapping mechanism for doing this task.

By extending the Service Oriented Network Architecture (SONATE) with an ontology and AHP, the problem of describing services and selecting the best one is solved. To implement this type of solution, a standard description scheme and ontology need to be developed and agreed upon. The properties upon which AHP or other similar methods relies on also need to be standardized. A standard organization might play a role in the standardization process.

Appendix A. Examples of Communication Services

Some examples of communication services are given below:

Security: Security is one of the most popular and necessary communication services which mean that the data is kept safe from intruders/middle-man during communication. This service is necessary for online banking transactions, military communication, medical communication, emergency needs and much more. There are several security services: *integrity*, *data-origin authentication* and *confidentiality*. *Data origin authentication* is a security service that verifies the identity of the claimed source of data. This ensures that the information is sent to or from a trusted partner. *Integrity* is a security service that ensures that modifications to the data are detectable. Even if the intruder obtains the information, *confidentiality* ensures that the man-in-the-middle cannot understand the information by changing the information into an unintelligible form.

Users can ask either request 'Security' in general or one or more of those security services: data-origin authentication, integrity, or confidentiality. Here, it is assumed that, when users request the 'Security' service, all of those security services will be provided.

Routing: This service, in general, routes the packet from source to destination.

RTT_Information: User or application can get Round Trip Time information by using this service

Hop_Information: Using this service, users or applications can get the number of hops between the sender and the receiver

Addressing_Conversion: Using this service, the addresses can be converted from one type to another. For example, from IPv4 to IPv6 address.

Prioritization: When the user or the application needs to give priority of one class of traffic to another, this service can be used.

Signal_Conversion: In case the application needs to convert from one signal type to another, this service can be used, for instance, conversion from analog to digital signal or the other way around.

Size_Reduction: If the application cannot send a file because of its size, a Size Reduction service can be taken to do the task. Compression is one type of size reduction service. The user can request for either of the services to get the desired task done.

Availability: Availability covers different services. The most common one is monitoring which observes whether the peer host is still up and the connection is still alive. Employing the monitoring service as a foundation, a path management service can be created. These can have two different types: the basic one is multihoming. In this case, there are multiple available paths. If for certain circumstances, one path fails, it switches to another path that is not erroneous. A drawback of this service is that always only one path is active. There is a service called *Load_Sharing* which uses different paths simultaneously. Another availability service is Denial-of-Service availability (*DoS_Availability*) which ensures that the authorized users are still able to get served even when the system is under attack. Users can ask explicitly for one or more of the availability services or for the 'Availability' service in general when only the Monitoring service will be provided.

Addressing: This is a common communication service that identifies the source and destination process and its devices. Users or applications can request for one or more of the addressing services explicitly.

Connection_Management: This service provides connection management including connection establishment and connection termination. Users or applications can explicitly request either of the services or in general 'Connection Management' where both of the services will be provided.

Reliability: The Reliability service ensures that the data must reach the destination without any corruption. There are several reliability services:- error detection, data flow limiting, order preservation and error control. As the name indicates, error detection service detects errors that have been happened on the way. Data flow limiting is an important service in a shared network which is used to avoid source, destination and network congestion by limiting data flow. The order preservation service ensures that the data arrives at the destination in the same order as the data has been sent. When 'Reliability' service in general is requested, all of the reliability services will be provided. But, the user can clearly ask for one or more of the reliability services for decreasing the cost of communication.

Packet_Boundaries_Preservation: In case of user or application needs, this service ensures that the packet will not be segmented or fragmented.

Path_MTU: This service provides the size of the maximum transfer unit between the source and the destination.

Loop_Avoidance: This service avoids loop during routing data.

REFERENCES

[1] OASIS Reference Model for Service Oriented Architecture 1.0, Official OASIS Standard (Normative PDF), Oct. 12, 2006, http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf, accessed on 08th April 2010.

[2] A. Kabzeva, M. Hillenbrand, P. Müller, and Ralf Steinmetzy, Towards an Architecture for the Internet of Services, 35th Euromicro SEAA conference, 27-29 AUgust 2009, Patras, Greece.

[3] E. A. Marks & M. Bell, Executive's guide to service-oriented architecture, John Wiley & Sons, 2006

[4] J. Lawler and H. Howell-Barber, Service-oriented architecture : SOA strategy, methodology, and technology, Auerbach Publications, 2008

[5] T. Erl, Service-Oriented Architecture – Concepts, Technology, and Design, Prentice Hall, 2005, ISBN: 0-13-185858-0

[6] B. Reuther & D. Henrici, A model for service-oriented communication systems, Journal of Systems Architecture, 2008

[7] B. Reuther, D. Schwerdel, A. Siddiqui, Z. Dimitrova, P. Müller, A Protocol Framework for a Service Oriented Future Internet Architecture, GI/ITG KuVS Fachgespräch Future Internet, Munich, 2009

[8] B. Reuther and P. Müller, Future Internet Architecture - A Service Oriented Approach. In Oldenbourg Verlag, München, 2008.

[9] D. Schwerdel, Z. Dimitrova, A. Siddiqui, B. Reuther, P. Müller, Paul, Composition of Self Descriptive Protocols for Future Network Architectures, EuroMicro 2009, 27-29 August, Patras, Greece

[10] T. R. Gruber, Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum. Comput. Stud., 43 (5–6):907–928, 1995.

[11] L. Liu and M. T. Özsu, Encyclopedia of Database Systems, Springer-Verlag, 2009.

[12] Protégé ontology editor, http://protege.stanford.edu/

[13] T. L. Saaty. Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors - The Analytic Hierarchy/Network Process, RACSAM (Review of the Royal Spanish Academy of Sciences, Series A, Mathematics), , (2008-06), 102 (2): 251–318.

[14] L. Voelker, D. Martin, I. E. Khayat, C. Werle & M. Zitterbart, An Architecture for Concurrent Future Networks, 2nd GI/ITG KuVS Workshop on The Future Internet, GI/ITG Kommunikation und Verteilte Systeme, Karlsruhe, Germany, Nov 2008.

[15] G. Klyne, J.J. Carroll (Eds.), Resource Description Framework: Concept and Abstract Syntax, W3C 2004