



**HAL**  
open science

# An Evolution Method of Service View Products Supported Dynamic Integration of Information Resources

Shu Zhen, Luo Aimin, Luo Xueshan

► **To cite this version:**

Shu Zhen, Luo Aimin, Luo Xueshan. An Evolution Method of Service View Products Supported Dynamic Integration of Information Resources. 16th International Conference on Informatics and Semiotics in Organisations (ICISO), Mar 2015, Toulouse, France. pp.102-111, 10.1007/978-3-319-16274-4\_11 . hal-01324968

**HAL Id: hal-01324968**

<https://inria.hal.science/hal-01324968v1>

Submitted on 1 Jun 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# An Evolution Method of Service View Products Supported Dynamic Integration of Information Resources

Shu Zhen, Luo Aimin, Luo Xueshan

Science and Technology on Information Systems Engineering Laboratory,  
National University of Defense Technology, Changsha, Hunan, P.R.China, 410073

sz\_1226@sina.com, {amluo, xsluo}@nudt.edu.cn

**Abstract.** The original information system architecture Framework provides an integration mechanism of information system resources to support a specific operational mission, but it is difficult to describe and analyze the dynamic integration process of the information system resources to support the dynamic adjustment of operational mission in the future complex, changeable environment. This paper presents the modeling approach of the service view product, and focus on the evolution method of service view products to support the dynamic integration process of the information system resources.

**Keywords:** Architecture · Operational activity · Service View · System Resource · Evolution · Dynamic Integration.

## 1 Introduction

Information system architecture is defined as the structure of components consisting of its system, relationships, and the disciplines and guidelines governing their design and evolution over time [1-3]. It has great importance on the information system design, development, operation and evolution. Architecture is one of the key technologies to guide the information system integration. At present, much research on information system architecture is generally conducted based on the research of DoDAF. The original information system architecture Framework (C4ISR/AF, DoDAF, i.e.) is based on many views, such as “Operational View, System View, Technical View,” etc. through the mapping between operational activities and system functions, provides an integration mechanism of information system resources to support a specific operational mission, while it is difficult to describe and analyze the dynamic integration process of information system resources to support the dynamic adjustment of operational mission in the future complex, changeable environment. Although DoDAF2.0, NATOAFv3.0, MoDAF1.2 and other latest architecture frameworks including the description of service view[4-6], they just only describe the definition, role and content of the service view products without analyzing the internal logical relationship between operational view, system view and service view, so that it is

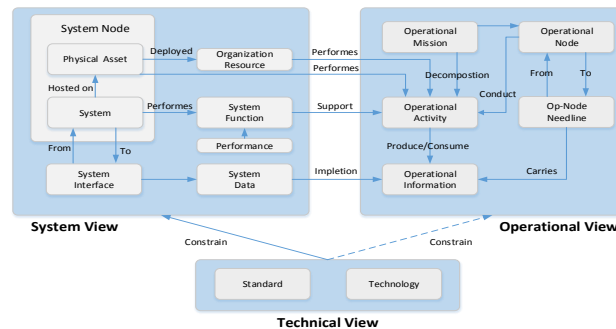
difficult to guide modeling and analyzing application of the service view products. Therefore, using the SOA and service-oriented thinking<sup>[7-8]</sup>, this paper extends the original information system architecture framework with increasing the service view products description in order to achieve the separation between the operational activities and the system resources. Then under the service view framework, this paper presents the modeling method of the service view products, and focus on the evolution method of service view products to support the dynamic integration process of the information system resources.

This paper is organized as follows. In section 2, we present a service view description framework of information system architecture. Section 3 discusses modeling methods of the service view products within information system architecture. In section 4, we propose an evolution method of service view products within information system architecture. Section 5 concludes the paper by noting the future works.

## 2 A Service View Description Framework of Information System Architecture

### 2.1 The Original Information System Architecture Framework and Data Elements Relationship

The original information system architecture framework mainly described the system from three perspectives: the operational requirement and application, system design and technical constraints, namely Operational View, System View and Technical View<sup>[1-3]</sup>. In particular, operational view describes the operational mission decomposition, operational activity model, operational process model, the rules of operational activities execution and information exchange etc. which is contributed to complete the operational mission. Through the traceability relationship between operational activities and system functions, it can determine the relationships of systems, system functions, system communication and data transmission to support to complete operational activities, and provides an integration mechanism of information system resources. The structure and data elements of the original information system framework are shown in figure 1.





spective of service-oriented. The services can support the exchange of information between operational activities through XML data of standard SOAP protocol.

### 2.3 A Service View Description Framework of Information System Architecture

Service view acted as the intermediate layer between operation view and system view. On the one hand, it can abstract and package the distributed, heterogeneous and inter-connection system resources of system view, on the other hand, it also need to support and adapt to the change of operational activities process in operational view. Therefore, the data elements described in service view include the description of service itself, such as service structure, service interface, service behavior, service property, etc. and the relationship between services, such as service connection structure, service interaction rules, service level, etc. Therefore the service view description framework can be divided into: the description of service itself, the description of relationship between services and the description of the relationship between services and the relation between services and other architectural elements, as shown in the figure 3.

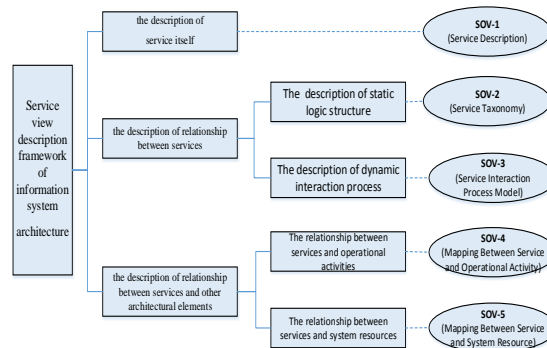


Fig. 3. The service view description framework of information system architecture

In order to avoid confused with the product name of systems View (SV), the name of Service View taken from the Service Oriented View, referred to as SOV.

## 3 The modeling and description method of service view products within information system architecture

### 3.1 The modeling and description method of each service view product

#### (1) Service Description (SOV-1)

This product is the foundation of the description of other service view products. This product mainly realizes the package and description of a single system resources,

namely the description of member service. Member Services(MS) can be represented by the following nine-tuple:

$$MS = (ID, Name, Description, Operation, Provider, Input, Output, Capability, QoS)$$

Among them, *ID* represent service identification, *Name* represent service name, *Description* represent the description of service function or the supported operational activity, *Operation* represent the operation provided by service, *Provider* represent the service provider's information, *Input* and *Output* respectively represent the input and output parameters of the service, *Capability* represent the service ability, *QoS* represent the service quality attributes.

#### (2) Service Taxonomy (SOV-2)

This product is mainly responsible for the classification and management of numerous member services involved in the information system, including hierarchical relationships, combination relations, derived relations and classification relations of member services. For this product, we can use the concept of the Service Group, to distinguish services from the similarity of service function. Service Group (SG) can be represented by the following seven-tuple:

$$SG = (ID, Name, Description, Operation, Input, Output, Capability)$$

The meaning of each element in this tuple is similar to the element in *MS*. Each service in the *SG* is corresponding to one member service. Any service in the same *SG* has the same function and interface, but the service provider's information and QoS are different.

#### (3) Service interaction process model (SOV-3)

This product is mainly used to describe the dynamic interaction between services, to support to complete operational activities process. We can use object-oriented Petri net to model this product<sup>[11]</sup>. Using the advantage of expression ability based on object-oriented Petri net, we can solve the defects of enough semantic expression based on ordinary Petri net. We can use the object of object-oriented Petri net to describe the sub-processes, and use the object Petri net hierarchical description mechanism reduces the complexity of modeling service interaction process, and can establish a mapping between service node and service group through elements of object-oriented Petri net, such as token, action function and predicate transfer function, etc. in order to solve the path selection of uncertainty activity and system resources dynamic change problem in the execution process of service interaction, to meet the needs of service interaction process model description.

#### (4) Mapping between service and operational activity (SOV-4)

This product is mainly used to describe the traceability relationships between operational view and service view in system architecture design, to ensure the accessibility between operational activities and services supported operational activities. It is a many-to-many relationship between services and operational activities. The traceability matrix can be used to describe this product. Traceability matrix can be represented by the following six tuple:

$$SerToAct = (ID, ReIDescription, SerID, SerName, ActID, ActName)$$

Among them, *ID* represent mapping relation identification, *ReIDDescription* represent mapping relationship description content, *SerID* and *SerName* respectively represent service identification and service name, *ActID* and *ActName* represent activity identification and activity name.

(5) *Mapping between service and system resource (SOV-5)*

This product describes the traceability relationships between services and system resources, to ensure services can be implemented and deployed by the specific system resources. It is a many-to-many relationship between services and system resources. The traceability matrix can also be used to describe this product. Traceability matrix can be represented by the following six tuple:

$$SerToSys = (ID, ReIDDescription, SerID, SerName, SysID, SysName)$$

The meaning of each element in this tuple is similar to the element in *SerToAct*.

### 3.2 The relationship between service view products

According to the above analysis of description and modeling method of service view products, the logical relationship between service view products is shown as figure 4.

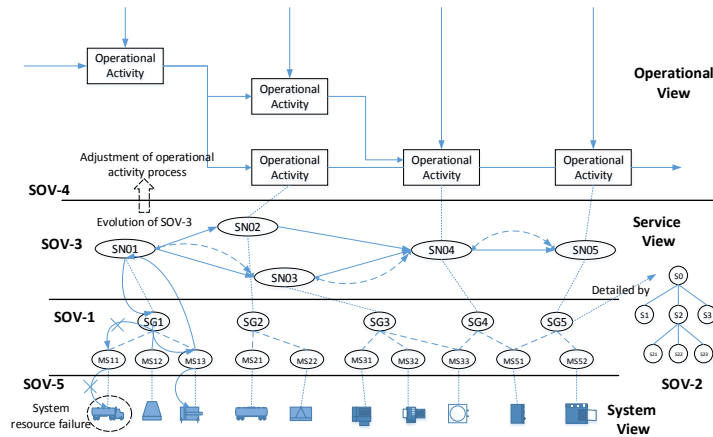


Fig. 4. The relationship between service view products

Among them, SOV-1 implement system resources packaged modeling and description, and set the detailed description of member service. With using the concept of service group, SOV-2 can realize classification management of member services described in SOV-1. Mapping relationship between services and operational activities (SOV-4) and mapping relationship between services and system resources (SOV-5) determine the traceability mapping relationship among service view, operational view and system view, and with other products of service view can realize the separation between system resources and operational activities.

## 4 Evolution process of service view products within information system architecture

In current complex, changeable environment, operational mission and state of system resources are easy to change, which requires decision makers adjust operational activities process and system resources timely according to the actual situation<sup>[10]</sup>. Because the heterogeneity of information system resources, it is difficult to adjust system resources to adapt to the adjustment of operational activities process. Therefore, in the information system architecture design level, describing and analyzing the possible evolution requirements of system execution process is convenient to guide information system executing to adapt to the change of information system construction demand in the future.

### 4.1 Evolution demand of information system architecture

#### (1) system resource failure

As in figure 5, through the analysis of the operational mission requirements, operational activities process model is established, including operational activity A, B, C and the information exchange among them. According to the mapping relationship between operational activities and system resources, operational activity A, B, C respectively supported by system resource 1, 2, 3, the demand of information exchange among operational activities determines connection relationship among system 1, 2, 3.

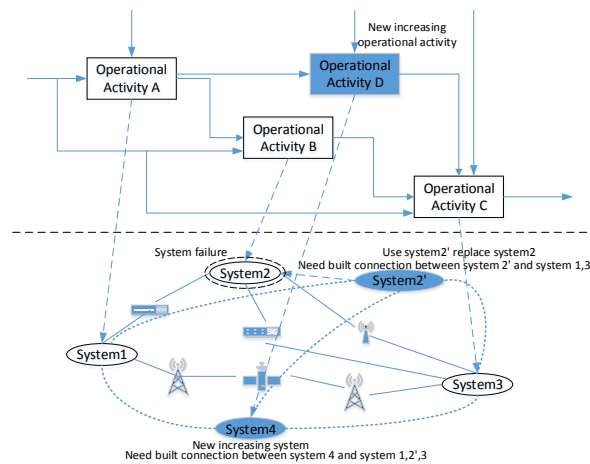


Fig. 5. Evolution demand of information system architecture

Assumes that system 2 suddenly becomes invalid, it is difficult to support to complete operational activity B. Then you need to use a system 2' which has the same function as system 2 to replace system 2, to enable the operational activities process to continue. Owing to the heterogeneity between the system resources, system 2 and system 2' may be developed by different manufacturers, used different techniques to realize,



which cause their interfaces are different and difficult to establish communication connection between system 2' and system 1(system 3), thus leading to the failure of the operational activity execution process, affecting the completion of operational mission.

(2) *Operational activities process adjustment*

Using the above example shown in figure 5, assume it is need to increase new operational activity D, at the same time the logic relations between activity D and original activity A, B, C, such as exchange of information and sequential process, are set up. In order to satisfy the adjustment of operational activities process, it need to increase system 4 in system resources, and need to increase information exchange relations between system 4 and original system 1, 2', 3 according to information exchange relations of operational activity D. Also, owing to the heterogeneity between the system resources, it is difficult to establish communication connection between system 4 and original systems in order to support or adapt to the adjustment of operational activities process quickly.

#### **4.2 The analysis of service view products evolution process**

Service view acted as the intermediate layer between operational view and system view can effectively realize the separation between operational activities and system resources, and do not need specify a specific service in the service interaction process model, in order to effectively adapt to the evolution requirement of system resources failure and adjustment of operational activities process.

(1) *The evolution process of service view products under system resource failure*

In service view, the execution resource in service interaction process model(SOV-3) choose service group, not specify the specific member service instance. Therefore we can use service group to deal with the failure situation of system resources. Service view products evolution process is shown in figure 6, assume that the failure system resource is the specific member service instance chosen in service interaction process model.

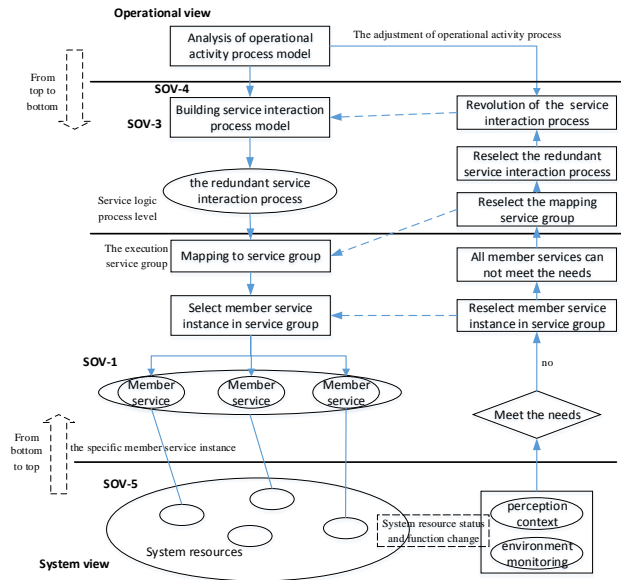
*Step 1:* Collect and manage member services information using service group, and learn member service status and function change through perception context and environment monitoring.

*Step 2:* When the service group learn the member service served as execution resource in the service interaction process model is difficult to meet the needs of the service interaction process execution, Service group reselect another member service served as the new execution resource in the service interaction process model, which does not affect the execution of service interaction process and does not need to adjust operational activity process.

*Step 3:* When all member services in service group are difficult to meet the functional requirements of service interaction process execution, we can choose other backup service group registered in SOV-2, and choose the suitable member service instance meeting the requirement of service interaction process execution through backup service group. If not find alternative service group, then turn to the fourth step.

*Step 4:* Reselect redundant service interaction process model registered in SOV-3, and need to pay attention to ensure the mapping effectiveness between service node and service group.

*Step 5:* If there still exist service group failure mapped by service node in redundant service interaction process model, it need to perform evolution operation on service interaction process model by architecture designers.



**Fig. 6.** The evolution process of service view products

(2) *The evolution process of service view products under operational activities process adjustment*

As operational activities execution process is supported by the service interaction process, it is need to analyze and adjust the service evolution process to deal with operational activities process adjustment.

*Step 1:* In the view of future uncertain operational activities process, we design the redundant service interaction process in SOV-3. If the redundant service interaction process model can satisfy the operational activities adjustment, the evolution process is over, otherwise turn to the second step.

*Step 2:* According to the adjustment of operational activity process, architecture designers perform evolution operation on original service interaction process model to ensure the new model can support and adapt to the adjustment of operational activities process.

Note that, the above two evolution processes are likely to be involved in the evolution of service interaction process model. The evolution refers to adjust the service interaction process according to the adjustment of operational activity process, to support or adapt to operational activities after the adjustment process. From 3.1, we know service interaction process model is described by object-orient Petri net model, therefore the evolution of the service interaction process model can be converted into the opera-

tion of increase, delete, replace in object-oriented Petri net model, at the same time also need to ensure the structure correctness and service resource effectiveness of service interaction process model after revolution. Due to the limitation of space, the specific algorithm is not introduced, and it can be found in [11].

## 5 Conclusions

This paper presents the modeling and analyzing of the service view within the information system architecture using service-oriented thinking and research on SOA, in order to describe and analysis the system resources dynamic integration and evolution demand of information system in the future complex, changeable environment. Our contributions make a base ground for the future research of information system architecture designing and analyzing. Future work will be perfect service view description framework of information system architecture and develop modeling tool based on this theory.

## References

1. C4ISR Architecture Working Group. C4ISR Architecture Framework Version 2.0 [R]. U.S.: Department of Defense, 1997.
2. DoD Architecture Framework Working Group. DoD Architecture Framework Version 1.0 Volume I: Definitions and Guidelines [R]. U.S.: Department of Defense, 2003.
3. DoD Architecture Framework Working Group, "DoD Architecture Framework Version 1.5 Volume I: Definitions and Guidelines" [R], U.S.: Department of Defense, 2007.
4. DoD Architecture Framework Working Group. DoD Architecture Framework Version 2.0: Architect's Guide-Architectural Data and Models [R].U.S.: Department of Defense, 2009.
5. The MODAF Development Team, "The MOD Architecture Framework Version 1.2" [R], The UK: Ministry of Defense, [www.modaf.org.uk](http://www.modaf.org.uk), 22 September 2008.
6. The NATO C3 Board, "The NATO Architecture Framework Version 3.0" [R], North Atlantic Treaty Organization: [http://www.nhqc3s.nato.int/Browser.asp?Target=\\_docs/NAF\\_v3](http://www.nhqc3s.nato.int/Browser.asp?Target=_docs/NAF_v3), November 2007.
7. H.-M. Chen, "Towards Service Engineering: Service Orientation and Business-IT Alignment," Proc. IEEE Hawaii Int'l Conf. System Science, 2008.
8. "Software as a Service (SaaS): An Enterprise Perspective," Microsoft Corp, <http://msdn.microsoft.com/en-us/library/aa905332.aspx>, 2010.
9. Nicholas J. Merski. Tailored Systems Architecture For Design of Space Science and Technology Missions Using DoDAF2.0 [D]. Wright-Patterson Air Force Base, Ohio Department of The Air Force Air University, 2010.
10. "Business Alignment and Agility," Hewlett Packard, video, [http://h30423.www3.hp.com/?fr\\_story=0e4d9cbbd23432b8a860c6c9dc456458debf36b3](http://h30423.www3.hp.com/?fr_story=0e4d9cbbd23432b8a860c6c9dc456458debf36b3), 2010.
11. Zhen Shu, Honghui Chen, Xueshan Luo. "Research on Military Information Service Composition and Validation Method Based on Object Petri Net". Journal of Systems Engineering and Electronics, 2011, 33(7):1558-1564.