A Semi-Automatic Optimization Design Method for SvcV-5 in DoDAF 2.0 Based on Service Identification

XIAOXUE ZHANG1, AIMIN LUO1, YI MAO2, MENGLONG LIN1, YUXIAO KOU1, AND JUNXIAN LIU1

1Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
2State Key Laboratory of Air Traffic Management System and Technology, Nanjing 210000, China

Corresponding author: Xiaoxue Zhang (snowing1124@126.com)

This work was supported in part by the State Key Laboratory of Air Traffic Management System and Technology, under Grant SKLATM201806, and in part by the National Natural Science Grant of China, under Grant 71171189.

ABSTRACT As one of the products in the US Department of Defense Architecture Framework version 2.0 (DoDAF 2.0), the Operational Activity to Services Traceability Matrix product (SvcV-5) links the operational viewpoint (OV) and service viewpoint (SvcV). SvcV-5 is an essential product for optimizing the design of other service view products; thus, it is necessary to study how to optimize the design of SvcV-5. This paper studies the semi-automatic design method of SvcV-5 based on the service identification method. The main idea of this paper is to extract relevant information from previous architecture design products, such as OV-6b, CV-4 and CV-6, and then to study the method of taking service identification and designing the SvcV-5 product. First, a message-based service initial identification method is used to generate the initial service set. Second, considering the measure of the cohesion and the coupling degree of the service, a service improving algorithm is used to update the initial service set. In this process, a dependency matrix between operational activities is established; this matrix includes data dependencies and capability requirement dependencies. Then, according to the mapping result of the service identification method, service-related information can be determined manually by designers. Finally, SvcV-5 can be generated through an automatic algorithm based on the service identification result and the OV-6b product. This article verifies the proposed method using the motivation example of the earthquake rescue information system (ERIS) architecture product design. Through conducting experimental analysis and comparing our method with other service identification methods, we illustrate the effectiveness of the proposed SvcV-5 optimization design method.

INDEX TERMS DoDAF 2.0, SvcV-5, architecture, optimization design, service identification, dependency matrix.

I. INTRODUCTION

Architecture is the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time [1]. In the past 20 years, a number of organizations and individual researchers have developed architectural techniques and processes and documented best practices. These studies focused on architecture description, analysis, validation and evaluation [2]–[4]. The ultimate goal of performing architecture analysis and evaluation is to optimize architecture design solutions. However, architecture is inherently multidimensional. When designing architecture, solutions should have an optimized design to save time in evaluation and then provide feedback to revise the solutions. In this way, when designers design architecture solutions, there should be an optimally aided method to help them formulate their design plans. Thus, the question of how to support designers with optimal plans is the key problem that we should study, and it is also the essential research focus in the architecture design area. Design solutions must be optimized using a comprehensive and dynamic method that is guided by various optimal goals. Therefore, studying the method of optimization design in the architecture development process remains a significant challenge.
In recent years, some organizations and groups have studied some architecture frameworks in command and control information system architecture design. The US Department of Defense Architecture Framework version 2.0 (DoDAF 2.0) is by far the most authoritative version in designing command and control information system architecture solutions, and DoDAF 2.0 is widely used in designing information systems architecture. DoDAF 2.0 is a framework for facilitating the Department of Defense managers in the US. It standardizes the specific methods and processes of architecture design [5]. According to different perspectives of stakeholders, the viewpoints in DoDAF 2.0 are divided into 8 viewpoints, named all view (AV), capability view (CV), data and information viewpoint (DIV), operational viewpoint (OV), project viewpoint (PV), service viewpoint (SvcV), technical viewpoint (TV) and system viewpoint (SV) [6]. Each of these eight viewpoints has several products that describe different points of interest for different stakeholders, and designers can choose different modeling methods for every product. In general, the manifestations of products include tabular type, behavioral type, mapping type, structural type, timeline type, and ontology type. As one kind of manifestation, the mapping type is represented in the form of a matrix or a table that describes the mapping relationship between two different types of data.

SvcV-5 is a typical mapping type product. DoDAF 2.0 defines SvcV-5 as an Operational Activity to Services Traceability Matrix; it expresses which services can support activities to execute in OV-6b. When developing an actual command and control information system architecture solution, the design of OV products focuses only on what to do, what kind of goals are to be achieved, and how to implement the business process to achieve the goal. The design of the SvcV products include: SvcV-5 Operational Activity to Services Traceability, SvcV-1 Service Context Description, SvcV-2 Service Resource Flow Description, SvcV-3a System-Services Matrix, SvcV-3b Services-Services Matrix, SvcV-4 Services Functionality Description, SvcV-6 Services Resource Flow Matrix, SvcV-7 Services Measures Matrix, SvcV-8 Services Evolution Description, SvcV-9 Services Technology & Skill Forecast, SvcV-10a Services Rules Model, SvcV-10b Services State Transition Model, and SvcV-10c Services Event-Trace Description. All of these SvcV products reflect who will perform these tasks and how they will be done. Among all these SvcV products, SvcV-5 is a significant model, as it ties together the operational activities in OV-6b with services to complete these operational activities. The operational activity is a specification of what is to be done, regardless of the mechanism used. A service specifies how resources carry out the activity. The distinction between an operational activity and a service is a question of what and how [5]. The intended usage of the SvcV-5 includes:

- Tracing service functional requirements to user requirements.
- Tracing solution options to requirements.
- Identification of overlaps or gaps.

In SvcV-5, the relationship between operational activities and services can be expected to be many-to-many (i.e., one activity may be supported by multiple services, and one service may support multiple activities). As the output of SvcV-5 is the input of other SvcV products, only after designing SvcV-5, the designers can start to design other SvcV products. In this way, the design of SvcV-5 is the premise in designing SvcV products, and it is a key product that ensures that the solution is reasonable, effective and optimal. If SvcV-5 is optimized designed, designers can design other SvcV products more effectively, and thus reduce the cost that would be incurred to revise the whole architecture design and evaluation. If the SvcV-5 is not optimized designed, the activities cannot trace the service requirements to users’ requirements correctly, and cannot identify the overlaps or gaps effectively. In a word, if SvcV-5 is not optimized, the design quality of the architecture solution cannot be guaranteed.

To help design SvcV-5, DoDAF 2.0 specifies the design content of each architecture product and proposes the use of the International Defense Enterprise Architecture Specification (IDEAS) and metamodels to define the relationship between data in different products or in the same product. However, DoDAF 2.0 does not answer the question of whether the products in the architecture are optimally designed. This problem is the issue of architecture optimization design. Recent research works related to DoDAF 2.0 data and viewpoint products design mainly emphasize the architecture design process, and they lack the specific and much more instructive methods to design important architecture data, such as SvcV5. Such instructive methods are called optimized design methods of architecture. To solve the problem of optimized design of architecture, we have studied architecture optimization design methods based on the DoDAF Metamodel (DM2) [7]. The research of this paper is also based on our previous relevant research [8] and explores how to optimize the design of SvcV-5 products.

The core idea of this paper is to realize the scientific and semi-automatic clustering of operational activities according to currently existing designed architecture products (CV-4, CV-6 and OV-6b). The clustering result of operational activities is a packaged service set. This process is also called service identification in service-oriented architecture (SOA)-related areas [9], [10]. The result of service identification directly supports the generation of SvcV-5 design and then supports the design of other products in DoDAF 2.0. The structure of this article is as follows: Part II analyzes the status of related research. Part III describes the SvcV-5 optimization design problem and details the motivation case. Part IV studies related concepts and the process of service identification. Part V proposes the initial determination method of the service set, and part VI studies the method for improving the service set based on cohesion and coupling indicators. Part VII is the process of SvcV-5 generation, and part VIII analyzes the performance and compares the method proposed in this paper with other methods. The method proposed in
The architecture design method is the technical means of helping or guiding the designers who are designing architecture solutions. When designing architecture solutions, designers need to follow some strategies or methods. Recent studies on architecture design strategies can be divided into three kinds: product-centric, activity-centric and data-centric. Regarding designing methods, structured designing methods and object-oriented designing methods have recently been proposed. A structured design method was proposed by L. Wagenhals in 2000 [11], who defined the steps when designing architecture solutions and defined what product to design at each step. Designers must obey the design steps when designing products. An object-oriented method was proposed by I. Shin in 2000 [12], who studied the architecture design method based on the Unified Modeling Language (UML).

Current architecture design tools include System Architect (SA) from IBM [13], Tau G2 [14], CORE [15], etc. Although these tools play important roles in developing architecture solutions, they do not support the generation of products either optimally or automatically.

Overall, recent design methods and related tools have mainly focused on what steps should be followed, what method should be used, and what modeling items should be adopted to design architecture data and products. They place a high emphasis on the contents of products, and the relationships among products and design rules [8]. However, they do not mention how to support the generation of products by already designed products either automatically or semi-automatically. When designing architecture solutions, designers still have no idea how to design the best products, and this problem seriously influences the quality of architecture solutions.

B. RESEARCH ON ARCHITECTURE OPTIMIZATION METHODS

Recent research on architecture optimization is mainly aimed at optimizing system structure design and is mostly based on architecture solutions. Existing architecture optimization methods can be divided into scene-based optimization method and simulation-based method. The scene-based optimization method is a qualitative method, such as method based on the architecture trade-off analysis (ATAM) [16], [17] and the software architecture analysis method (SAAM) [18]. The ATAM involves the following three concepts: the capability parameter description, the scenario and being capability parameter-based. The ATAM takes architecture as the core and obtains the compromise points, sensitive points, non-risk points and risk points of the architectural elements to achieve a multi-objective optimization of the architecture. The simulation-based approach mainly converts an executable model into a simulation model, thus analyzing the areas that can be improved and then optimizing the solutions. Current conversion simulation models mainly include the object Petri net model [19], the colored Petri net (CPN) model [4], the ExtendSim model [20], the discrete event systems specification (DEVS) model [21], [22], and the queued Petri net model [23]. In general, architecture optimization methods mostly use quantitative statistics to evaluate and optimize the architecture. The problem of these methods is that the time cost of building an executable model that performs architecture optimization is high.

C. RESEARCH ON SERVICE IDENTIFICATION METHODS

Service identification is the process of finding and extracting services from business requirements [24]. In recent years, service identification-related research is mainly based on business processes and SOA [25]–[28]. Researchers have proposed several methods to identify services; one classification of these methods is based on source of service identification, and related studies include: process-based service identification methods and data-based service identification methods [29].

The process-based identification method is a well-known strategy for service identification; it is also called a business process-driven strategy in some studies [30], [31]. In such a strategy, a process is defined as a set of tasks of activities that are performed in coordination to achieve specific goals. A major benefit of this approach is that the identified services satisfy functional needs. However, it does not take the capability requirement and data dependencies into account. Furthermore, process-based approaches mainly focus on structural relations between tasks and ignore the conceptual relations [32], [33]. As data indicates the main stable domain abstraction of an enterprise, and data in terms of an object or entity plays a major role in service identification, some researchers think data-based service identification can solve some challenges in process-based approaches [34], [35].

Another classification of service identification is based on approach of clustering of activities [36]. Related studies can
be divided into three kinds. The first kind is a top-down domain-based decomposition method, which mainly decomposes business elements into a certain granularity, called a service [37]–[40]. The second kind is the bottom-up method, which mainly analyzes the existing sets of functions in information systems and extracts certain functions as services; these services can be then identified [41]. The third kind is a combination of the top-down and bottom-up approaches. It comprehensively considers requirements and the functions that can be provided by the existing information system function set [9]. Among these three kinds of methods, the first favors business requirements, the second emphasizes the implementation of existing system functions, and the third method considers both existing application requirements and the relationships between services and activities.

In our study, we take architecture design data as the input of SvcV-5 optimization design, and the architecture data include both the activity process design data and capability requirement design data. Our SvcV-5 optimization design method is the combination of the top-down and bottom-up approach. On one hand, we consider relations with a capability requirement and activities, and the design process starts with the top-level requirements and then helps to determine the type of services, so our method is a top-down type. On the other hand, we also consider the information input and output relations among different activities, and the service generation is based on these underlying interaction relations, so our method is also a bottom-up type.

**D. RESEARCH RELATED TO DODAF 2.0 VIEWPOINT PRODUCTS DESIGN**

Although DoDAF 2.0 defined the principles of obtaining architecture data, it did not explain the specific and operational guidance to users. Regarding the applications of DoDAF 2.0, some researchers have studied the Space S&T design [42], Enterprise application [43], [44], mesh-networking waveform [45], etc. In terms of architecture analysis, J. D. Pilcher [46] studied the Monterey Phoenix Analyzer tool, and enabled the system architect to reduce design complexity while quickly and easily exposing architectural flaws prior to implementation. Following DoDAF 2.0, a set of Fit-for-Purpose views are constructed in the reference implementation of a system architecture. Such studies point out that DoDAF 2.0 has come a long way in helping to change the perception of architecture, by being data-centric and attempting to re-align system architecture to support the decision making processes within a development [45]. In summary, although the previous study proposed a purpose-driven methodology for specific architecture, they did not point out the specific steps or methods for designing these viewpoint products more correctly.

As the linkage between service functions and operational activities, DoDAF 2.0 defines SvcV-5 as depicting the mapping of services (and optionally, the capabilities and performers that provide them) to operational activities, and thus identifies the transformation of an operational need into a purposeful action performed by a service solution. To design SvcV-5 in DoDAF 2.0, designers first need to know how to obtain services-related data in the architecture. To obtain service-related data, DoDAF 2.0 refines some principles of obtaining service data [5]. The principles include the following: identify and capture the capabilities supported or provided by the services; identify and capture the operations, business functions and activities supported or automated by the service; identify and capture the organization responsible for providing the services; capture the information to be consumed by the service and the information that is being produced by the service; define and capture the logical and/or physical interfaces required by the services; define and capture the rules applied to the information consumed and produced by the service; and define and capture the rules governing or constraining the use of the service.

Service-related data in SvcV-5 include activity, service, and related parameters data associated with them. Such kinds of data are special and essential. When designing such data, we should obey the principles of obtaining service data in DoDAF 2.0, and most importantly, we need to determine the method of how to optimize the design of such kinds of data.

Specifically, current research related to DoDAF 2.0 data and viewpoint products design mainly emphasizes the design process, and it lacks the specific and more instructive methods or optimization method to design architecture data. As SvcV-5 is a very important product in the whole architecture design, in order to solve the problem of lacking an optimization method in designing the architecture of SvcV-5, it is necessary to study the SvcV-5 optimization problem.

**III. SVCV-5 OPTIMIZATION DESIGN PROBLEM DESCRIPTION AND MOTIVATION CASE**

**A. DESCRIPTION OF THE PROBLEM**

SvcV-5 is a service viewpoint product in DoDAF 2.0 that expresses the dependencies between operational activities and services; it demonstrates how services support the completion of operational activities. In DoDAF 2.0, the core concept layer can be divided into three levels: the capability layer, the operational activity layer, and the service layer. The relationship between these concept layers is connected by different viewpoint products. The capability layer mainly describes the relationship between different capabilities; the representative products include the CV-2 and CV-4 capability viewpoint products. The relationship between the capability layer and the operational activities layer describes the details between capabilities and operational activities and reflects which operational activities support which capabilities; the representative product is CV-6. The operational activity layer describes the analysis of the task and the execution process of the operational process. The representative products include operational viewpoint products such as OV-5a/OV-6b. The relationship between the operational activity layer and the service relationship layer describes which operational activities are completed by which service; the representative
product is SvcV-5. The service layer reflects the description of the service and the execution process of the services; the representative products include service viewpoint products such as SvcV-1. Specific introduction is shown in Fig. 1.

Fig. 1 also shows the optimization design process from the capability requirements to the operational activities. The realization of capabilities is supported by the execution of operational activities, which require the execution of services to be carried out.

When designers develop command information system architecture solutions, it is necessary to give designers the specific design steps to guide the design process. The design steps are referred to in the literature [7], [11]. Among them, the predecessor products of SvcV-5 products include CV-2, CV-4, CV-6, OV-6b, and OV-5a. The direct pre-continuation products include CV-4, CV-6, and OV-6b, as shown in Fig. 2. The OV-6b product defines a relationship between operational activities (data-based dependencies); CV-4 and CV-6 define the dependency relationship between operational activities based on the capability requirement. These two relationships guide the design process of SvcV-5. After the design of SvcV-5, designers can design SvcV-1 service list descriptions. From the definition and connotation of SvcV-5 products, the focus of the SvcV-5 product optimization design problem is how to find and confirm the service in the architecture design process and how to transfer the relationships into the SvcV-5 product. The design premise of SvcV-5 is that the designer has designed pre-order architecture products (including CV-2, CV-4, CV-6, OV-5a, and OV-6b).

**B. MOTIVATION CASE**

This paper takes the ERIS architecture design as a case to study how to optimize the design of SvcV-5 products. The ERIS is a command information system that integrates the reconnaissance system, command system, rescue system and logistics system after an earthquake. The operational node of the ERIS includes the reconnaissance center, rescue command center, seismic station, rescue team node and logistics node. Conceptually, the main operational process is as follows: After an earthquake happens, the reconnaissance center sends reconnaissance information (RI) to the rescue command center, and the seismic station sends the earthquake warning information (EWI) and monitoring information (MI) to the rescue command center. After receiving the RI, EWI and MI, the rescue command center undertakes a comprehensive assessment of the disaster situation and formulates a rescue plan. The rescue center starts to allocate rescue forces, releases rescue force allocation information (RFAI), and releases rescue orders (Res_Ord) to the rescue team node and surveillance orders (Sur_Ord) to reconnaissance center teams. After receiving the rescue command, the rescue team node formulates the team rescue plan, determines which rescue equipments will be used, and implements the rescue. Then, wounded individuals are rescued, the rescue needs are reported, and the demand for materials is reported. Subsequently, the rescue team node feeds back the rescue situation to the rescue command center to ensure that demands are met. After the logistics node receives the demand information and the need information from the rescue team node, it allocates logistics resources and transports them to the appropriate rescue location.
Suppose that designers design an ERIS architecture solution based on DoDAF 2.0. The viewpoints to be designed include the CV, OV, SvcV, and TV. Assume that the designer has designed CV-4, CV-6, CV-2, and OV-6b for the ERIS architecture, as shown in Table 1, Table 2, Fig. 3 and Fig. 4. Fig. 4 illustrates the state transition of the operational activities performed by the operational nodes using the Business Process Modeling Notation (BPMN) 2.0 lane map. The operational nodes include 5 kinds of nodes. The swim lane diagram represents the operational activities and interactions performed at different nodes. Table 5 summarizes and analyzes the information exchange relationship among the activities and nodes in Fig. 4.

OV-6b represents the basic information of an architecture development case: the operational activities, operational nodes, and operational information exchanged between nodes. Table 3 shows the related information in OV-6b of the motivation example. In Table 3, the create, read, update and delete (CRUD) relationship mean: the information can be created (C), read (R), updated (U) or deleted (D) in one activity at least once. In Table 3, Input(t_i) shows the input information of activity t_i, while Output(t_i) shows the output information of activity t_i.

IV. RELATED CONCEPTS AND PROCESS OF SsvcV-5 OPTIMIZATION DESIGN
To facilitate the discussion that follows, we first define some concepts and the formal representation:

- Capability: C. C_i represents the i-th capability in information systems. Capability is an abstract concept that illustrates the demand of designers when developing system solutions.
- Operational node: This node indicates where the operational activity is executed, i =1,2,..., num_of_activities.
- CRUD operations: CRUD refers to the actions on information when undertaking an activity. C represents creating information, R represents reading the information, U represents updating the information, and D represents deleting the information.
- Operational activities: t_i. t_i = {d_i, In_{t_i}, Out_{t_i}, r_i, d_i}. d_i is the description of t_i; In_{t_i} and Out_{t_i} show the input and output operational information of t_i; and r_i indicates the operational node where the operational activity is performed. A_i represents the CRUD operations related to t_i. Operational information can be described by text. An operational activity can have multiple CRUDs.
- Operational activities process: OAP. The OAP indicates the process of executing some operational activities.
- Service: Service is defined as a set encapsulation of system functions, and the functions are provided by service providers to service consumers through a service interface; alternatively, a service is a usable result for completing an actual operational mission or operational activities. The goal of a service is to achieve some activities. In this way, a service can be recognized as a cluster or a small set of activities during information.
We denote service as $s_j$, $s_j$ represents the $j$-th service in architecture, $j = 1, 2, \ldots, \text{num}\_\text{of}\_\text{services}$. $s_j = \{d_{sj}, In_{sj}, Out_{sj}, r_{sj}, A_j\}$. $d_{sj}$ represents the descriptions of $s_j$; $In_{sj}$ and $Out_{sj}$ show the input and output operational information of $s_j$, $r_{sj}$ indicates the operational node where the operational activity is performed, $A_j$ represents the CRUD operations related to $s_j$.

Based on the definition of service and operational activities, the process of service identification can be generated by the process of clustering operational activities. The input of the service formed by the cluster is the union of the inputs of all the operational activities in the cluster, the output of the service is the union of the outputs of all the operational activities in the cluster, and the CRUDs of the service are the union of the CRUD operations in all the operational activities in the cluster. In the process of service identification by clustering operational activities, two items need to be considered: the CRUD relationship between operational nodes and operational activities and the impact of capability requirement on service identification.

The idea of the SvcV-5 optimization design method is shown in Fig. 5. First, the initial service clustering generation based on messages is performed based on OV-6b; this step will be illustrated in part IV. Second, according to the cohesion degree and the coupling degree, the services clusters are improved. In this step, we need to calculate the capability requirement dependency matrix $RR$ based on CV-4 and CV-6, calculate the data dependency matrix $RD$ based on OV-6b, and calculate cohesion and the coupling degree and select the best pairs to combine and update the whole cluster set. In this way, services are combined and updated. The second step will be introduced in part V. Finally, the SvcV-5 product can be generated through a specific algorithm. Part VI will study this step.

The difference between the traditional SvcV-5 design method and our design method is as follows: Designers
design the SvcV-5 matrix based entirely on their experience with the traditional SvcV-5 design method, but in our method, the SvcV-5 matrix can be generated by applying some algorithms. Such algorithms are conducted by analyzing and calculating certain relationships of activities, capabilities and operational information. The whole process of our SvcV-5 optimization design method is realized by automatic algorithms, and the service name is manually determined. Therefore, we call this method a semi-automatic design method. The SvcV-5 optimization method includes two parts: the service identification method (includes step 1 and 2 in Fig. 5) and the SvcV-5 generation method (for step 3 in Fig. 5).

V. INITIAL SERVICE IDENTIFICATION

First, we define message in the SvcV-5 optimization design. A message is information that is created in one operational activity and used in another operational activity at another operational node in OV-6b. Use means that there is some type of CRUD action. Table 4 lists all the messages involved in the OV-6b in the ERIS architecture design.

The initial service identification method refers to the candidate service identification method in the literature [36]. The input data of the initial service identification method is based on the messages in OV-6b. The core idea of this method is to find the corresponding service for each activity. First, for each operational activity that generates a message, we create a new service. Then, we begin to search for the remaining operational activities and check the relationship between the remaining operational activities and the activities that have matched service. Under the following two conditions, we can combine these two activities into the same service to guarantee that the service is highly cohesive.

### TABLE 3. Information exchange relationship in OV-6b in the case.

| No. of activities | Name of activities | Input(t1) | Output(t1) | Operational nodes | Operations on information |
|-------------------|--------------------|-----------|------------|-------------------|--------------------------|
| t1                | Reconnaissance disaster | Reconnaissance order | RI | Reconnaissance command center | R (reconnaissance order), C (RI) |
| t2                | Reconnaissance information analysis | RI | Analysis result | Rescue command center | R (RI), C (analysis result) |
| t3                | Assess disaster | Analysis result, EWI, MI, RS, CS, medical needs | Assessment | Rescue command center | R (analysis result, EWI, MI, RS, CS, medical needs), C (assessment) |
| t4                | Making rescue plan | Analysis result, RI, assessment, rescue force information | Rescue plan | Rescue command Center | R (analysis result, RI, assessment, rescue force information), C (rescue plan) |
| t5                | Allocate rescue force | Rescue plan, rescue force information | RFAi | Rescue command center | R (rescue plan, rescue force information), C (RFAi) |
| t6                | Release rescue order | RFAi | Res, Ord, Sur, Ord, LSO | Rescue command center | R (RFAi), C (Res, Ord, Sur, Ord, LSO) |
| t7                | Surveillance disaster | Sur, Ord | Real-time RI | Reconnaissance center | R (Sur, Ord), C (real-time RI) |
| t8                | Earthquake warning | / | EWI | Earthquake prediction system | C (EWI) |
| t9                | Seismic monitoring | / | MI | Earthquake prediction system | C (MI) |
| t10               | Force assembly | Assembly command | / | Rescue team node | R (assembly command), D (assembly command) |
| t11               | Receive EWI | EWI, MI | / | Rescue team node | R (EWI, MI) |
| t12               | Receive rescue order | Res, Ord | / | Rescue team node | R (Res, Ord) |
| t13               | Making team rescue plan | EWI, Res, Ord, real-time RI, EI | TRP | Rescue team node | R (EWI, Res, Ord, real-time RI, EI), C (TRP) |
| t14               | Making team rescue plan | TRP | EAI, TRP | Rescue team node | R (TRP), C (EAI) |
| t15               | Implement rescue | TRP | RS | Rescue team node | R (TRP), C (RS) |
| t16               | Rescuing the wounded | RS | CS, RS, medical needs | Rescue team node | R (RS), C (CS, medical needs), U (RS) |
| t17               | Reporting the rescue needs | RS, EAI | EDI | Rescue team node | R (RS, EAI), C (EDI) |
| t18               | Reporting the demand for materials | RS, material information | MDI | Rescue team node | R (RS, material information), C (MDI) |
| t19               | Rescue situation feedback | CS, RS, medical needs | / | Rescue team node | R (CS, RS, medical needs) |
| t20               | Receive logistics needs | LSO, EDI, MDI | / | Logistics node | R (LSO, EDI, MDI) |
| t21               | Allocate material | EI, EDI, MDI | Allocate material information | Logistics node | R (EI, EDI, MDI), C (allocate material information) |
| t22               | Transporting materials to various places | Allocate material information | / | Logistics node | R (allocate material information) |
TABLE 4. CV-6 product in the ERIS architecture.

| Message | Operational node | Operational activities | Used by operational nodes | Used by operational activities |
|---------|------------------|------------------------|---------------------------|-----------------------------|
| RI      | Reconnaissance Center | t1 | Rescue Command Center | t2 |
| Res_Order | Rescue Command Center | t6 | Rescue Team Node | t12, t13 |
| Sur_Order | Rescue Command Center | t6 | Reconnaissance Center | t7 |
| LSO     | Rescue Command Center | t6 | Logistics Node | t20 |
| EWI     | Earthquake Prediction System | t8 | Rescue Team Node | t11, t3 |
| MI      | Earthquake Prediction System | t9 | Rescue Team Node | t3 |
| Real-Time RI | Reconnaissance Center | t7 | Rescue Team Node | t13 |
| EDI     | Rescue Team Node | t17 | Logistics Node | t20, t21 |
| MDI     | Rescue Team Node | t18 | Logistics Node | t20, t21 |

Condition 1: The input of one operational activity equals the output of the activity that has matched services, or the output of one operational activity equals the input of the activity that has matched services.

Condition 2: These two activities are performed by the same operational node.

Finally, after all searches are performed, we identify each of the remaining operational activities that have not matched with a single service.

Algorithm 1 in Fig. 6 is used to generate the initial service set in the motivation case, and the calculation result is shown in Fig. 7.

In Fig. 7, a total of 19 new services are created, which are $S_A$ to $S_{19}$. In Algorithm 1, if the input of one activity equals the output of another activity, these two activities can be clustered as one service. However, if the activity that has a corresponding service encounters branching and gathering (as shown in Fig. 8), the service cannot combine with another activity even if the output of the activity equals the input of this activity. For example, in Fig. 7(a), activity $t_i$ creates message $A$, and message $A$ is the whole input of $t_j$, but neither activity $t_i$ nor $t_j$ can be combined with service $S_B$ because of the branch. In Fig. 7(b), activity $t_i$ creates message $B$, and message $B$ is the whole input of $t_k$, but $t_k$ cannot be combined with service $S_c$ because of the gathering of $t_i$ and $t_j$.

VI. SERVICE IMPROVEMENT

The initial service generation scheme is a preliminary scheme. Fig. 7 shows that basically every operational activity in the scheme corresponds to one service because the input to the operational activity in the same node is another operational activity. The output is less. In practice, each service must be executed at the same operational node. Therefore, for the service improvement operation, we need to consider only the possibility of service merging for the service subsets in the same operational node. This section examines how to merge services based on the analysis of the dependence.

FIGURE 6. Algorithm 1 of the initial service identification method based on the information flow. Notes: $\Delta$ represents service set $\{S_A, S_B, \ldots\}$. In(t_i) and Out(t_i) represent the input and output information of activity $t_i$ in OV-6b in Table 3, respectively; r(t_i) represents the operational node that executes activity $t_i$. $\emptyset$ means the empty set.

FIGURE 7. Branching and gathering situation in Algorithm 1.
of operational activities, and proposes a measure of service cohesion that is occasionally based on dependencies. Based on the metrics of these two indicators, the service is merged to improve the service set.

A. ANALYSIS OF TWO DEPENDENCE RELATIONSHIPS OF OPERATIONAL ACTIVITIES

There are two kinds of dependencies between different operational activities: capability requirement dependencies and data dependencies. The relationship between operational activities and the capability requirement is based on the indirect relationship of the capability relationship, which is denoted as RR(\(t_i, t_j\)). Data dependency refers to the input and output relationship of the operational activities in the same operational node, denoted as RD(\(t_i, t_j\)).

1) CALCULATION OF THE CAPABILITY REQUIREMENT DEPENDENCY MATRIX RR

Capability requirement dependency is defined as follows: If the capabilities supported by different operational activities are related, then the two operational activities are also indirectly related. Since there are \(N\) activities, the capability requirement dependency can be represented by the \(N \times N\) dimension matrix, called RR.

The method of calculating the matrix RR is based on the mapping matrix of the operational activities and capabilities (CV-6 Matrix) and the capability correlation matrix (CV-4 Matrix). The operational activity correlation matrix is derived from Algorithm 2. In Algorithm 2, if operational activity \(t_i\) supports capability \(C_k\), then operational activity \(t_j\) supports capability \(C_m\), and if capabilities \(C_k\) and \(C_m\) are related, then \(t_i\) and \(t_j\) are related.

In the motivation example, we can obtain the capability requirement dependency matrix RR following Algorithm 2 based on CV-4 and CV-6 in Table 1 and Table 2. The results are shown in Table 5. The number 1 indicates the dependency relationship is existed between corresponding activities.

2) CALCULATION OF THE DATA DEPENDENCY MATRIX RD

The data dependence of activities is defined as follows: When two activities have relationships that are input and output relationships and these two activities are directly connected in

![FIGURE 8. Initial service-generated results in the motivation case after applying Algorithm 1. Notes: service under the activity means the support relation. For example: \(S_a\) supports the completion of activity \(t_1\), service \(S_b\) supports the completion of both activity \(t_5\) and \(t_6\).](image)

![FIGURE 9. Algorithm 2 for calculating Capability requirement dependency Matrix, RR. Notes: M1 and M2 represent the CV-4 and CV-6 Matrix, respectively.](image)

![TABLE 5. Capability requirement dependency matrix RR in the case.](table)
of the input and output relationships between operational activities. Information is shown by a separate representation in OV-6b, as shown before and after the activity rectangle.

To analyze the data dependency among activities, we divide the data dependencies of operational activities in OV-6b into 10 categories: $d_1(t_i, t_j), d_2(t_i, t_j), \ldots, d_{10}(t_i, t_j)$. These 10 relationships are shown in Fig. 10. Specifically:

- All the outputs of $t_i$ are also all the inputs of $t_j$, denoted as $d_1(t_i, t_j)$, such as reconnaissance activity $t_1$ and reconnaissance information analysis $t_2$ in Fig. 4. The relationship between $t_1$ and $t_2$ belongs to this type; that is, $RD(t_1, t_2) = d_1(t_1, t_2)$, as shown in Fig. 10(a).
- Part of the output of $t_i$ constitutes all of the inputs of $t_j$, denoted as $d_2(t_i, t_j)$. For example, the relationship between $t_{13}$ and $t_{15}$ in Fig. 4 belongs to this type of relationship; that is, $RD(t_{13}, t_{15}) = d_2(t_{13}, t_{15})$, as shown in Fig. 10(b).
- The total output of $t_i$ constitutes part of the input of $t_j$, which is denoted as $d_3(t_i, t_j)$; For example, the relationship between $t_{15}$ and $t_{17}$ in Fig. 4 belongs to this type of relationship; that is, $RD(t_{15}, t_{17}) = d_3(t_{15}, t_{17})$, as shown in Fig. 10(c).
- Part of the output of $t_i$ constitutes part of the input of $t_j$, which is denoted as $d_4(t_i, t_j)$; For example, the relationship between $t_{16}$ and $t_{18}$ in Fig. 4 belongs to this type of relationship; that is, $RD(t_{16}, t_{18}) = d_4(t_{16}, t_{18})$, as shown in Fig. 10(d).
- The input of $t_i$ is the input of $t_j$, which is denoted as $d_5(t_i, t_j)$, as shown in Fig. 10(e).
- The input of $t_i$ constitutes part of the input of $t_j$, which is uniformly denoted as $d_6(t_i, t_j)$, as shown in Fig. 10(f).
- Part of the input of $t_i$ constitutes part of the input of $t_j$, which is denoted as $d_7(t_i, t_j)$, as shown in Fig. 10(g).
- The output of $t_i$ is the output of $t_j$, which is denoted as $d_8(t_i, t_j)$, as shown in Fig. 10(h).

- The output of $t_i$ constitutes part of the output of $t_j$, which is denoted as $d_9(t_i, t_j)$, as shown in Fig. 10(i).
- Part of the output of $t_i$ constitutes part of the output of $t_j$, which is recorded as $d_{10}(t_i, t_j)$, as shown in Fig. 10(j).

To conclude, the data dependencies between two operational activities can be divided into three conditions:

**Condition 1:** There is one type in $d_1, d_2, \ldots, d_{10}$. For example, regarding the pair $(t_2, t_3)$ in Fig. 4, this pair has one data dependency type, $d_2$.

**Condition 2:** There are two types; one type is in $d_5, d_6, d_7$, and the other type is in $d_8, d_9$, and $d_{10}$.

**Condition 3:** There is no data dependency relationship.

According to the definition of the 10 kinds of data dependencies between operational activities, the types of data dependencies between all activities in the motivation case can be obtained; the results are shown in Table 6. In Table 6, ‘/’ indicates that there are no activity pairs that have a type of data dependency relationship.

According to OV-6b, a CRUD relationship matrix for the operational objects involved in each operational activity is automatically generated, as shown in Table 3.

The data dependency calculation formula for operational activities is shown in Equations (1)-(4). $RD1(t_i, t_j)$ is the calculated dependency value for cases $d_1, d_2, d_3$, and $d_4$. $RD2(t_i, t_j)$ is the calculated dependency value for cases $d_5, d_6$, and $d_7$. $RD3(t_i, t_j)$ is the calculated dependency value for cases $d_8, d_9$, and $d_{10}$. $RD(t_i, t_j)$ is the whole calculated dependency value of $t_i$ and $t_j$. $\|Out(t_i) + In(t_i)\|$ means the sum of the number of types of output information for activity $t_i$ and the number of types of input information for activity $t_j$, $\|both(t_i, t_j)\|$ means the number of types in output information for activity $t_i$ and the information is also available in input activity $t_j$; $\|In(t_i) + In(t_j)\|$ means the sum of the number of types of input information for activity $t_i$ and the number of types of input information for activity $t_j$ and the information is also available in input activity $t_j$; $\|Out(t_i) + Out(t_j)\|$ means the sum of the number of types of output information for activity $t_i$ and the number of types of output information for activity $t_j$.
Algorithm 3: Calculation of data dependency relationship value //input: Information flow dataset from OV-6b

//Output: RD Matrix
1: for all operational activity pairs \((t_i, t_j)\)
2: \(\text{set } RD_1(t_i, t_j)=0;\)
3: \(\text{set } RD_2(t_i, t_j)=0;\)
4: \(\text{set } RD_3(t_i, t_j)=0;\)
5: \(\text{if(type}(RD(t_i, t_j))=d_1|d_2|d_3|d_4)\)
6: \(\text{compute } RD_1;\)
7: \(\text{end if}\)
8: \(\text{if(type}(RD(t_i, t_j))=d_5|d_6|d_7)\)
9: \(\text{compute } RD_2;\)
10: \(\text{end if}\)
11: \(\text{if(type}(RD(t_i, t_j))=d_8|d_9|d_{10})\)
12: \(\text{compute } RD_3;\)
13: \(\text{compute } RD_4(t_i, t_j);\)
14: \(\text{return } RD(t_i, t_j);\)
15: \(\text{end for}\)

### FIGURE 11. Algorithm 3 for calculating the data dependency relationship value \(RD\).###

The information is also available in output activity \(t_j\).

\[
RD_1(t_i, t_j) = \begin{cases} 
2 \left\| \frac{\text{both}(t_i, t_j)}{\text{Out}(t_i) + \text{In}(t_j)} \right\|, & \text{if } \|\text{Out}(t_i) + \text{In}(t_j)\| > 0 \\
0, & \text{if } \|\text{Out}(t_i) + \text{In}(t_j)\| = 0
\end{cases}
\]

\[
RD_2(t_i, t_j) = \begin{cases} 
2 \left\| \frac{\text{bothIn}(t_i, t_j)}{\text{In}(t_i) + \text{In}(t_j)} \right\|, & \text{if } \|\text{In}(t_i) + \text{In}(t_j)\| > 0 \\
0, & \text{if } \|\text{In}(t_i) + \text{In}(t_j)\| = 0
\end{cases}
\]

\[
RD_3(t_i, t_j) = \begin{cases} 
2 \left\| \frac{\text{bothOut}(t_i, t_j)}{\text{Out}(t_i) + \text{Out}(t_j)} \right\|, & \text{if } \|\text{Out}(t_i) + \text{Out}(t_j)\| > 0 \\
0, & \text{if } \|\text{Out}(t_i) + \text{Out}(t_j)\| = 0
\end{cases}
\]

\[
RD(t_i, t_j) = RD_1(t_i, t_j) + RD_2(t_i, t_j) + RD_3(t_i, t_j)
\]

According to the calculation method of Equations (1)-(4), the value of RD between operational activities can be calculated. The calculation process is shown in Algorithm 3 of Fig. 11.

According to Algorithm 3 in Fig 11, the data dependency values in the motivation case are obtained, as shown in Table 7.

### TABLE 7. Date dependence matrix among activities in OV-6b.###

|   | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|----|----|----|----|----|----|----|----|----|----|----|----|
| 11 | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 12 | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 13 | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 14 | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 15 | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 16 | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  |
| 17 | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  |
| 18 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  |
| 19 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  |
| 20 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  |
| 21 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  |
| 22 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  |

The calculation of \(TT\) will be used in the calculation of cohesion and the coupling degree in the next subsection to improve the service set.

### B. METHOD FOR IMPROVING THE SERVICE SET###

In the initial service generation method, only the data dependencies and direct relationships of the service are considered, but some important indicators of the service, such as cohesion and the coupling degree, are not considered. In general, the search for the service is actually a search for the full data dependency of the service. However, that search finds only the operational activities that created the message and does not consider the operational activities that do not have a direct connection with the message. Therefore, in the method for improving the service set, it is necessary to check the dependency among activities to see whether we can merge two or more services into a service.

In the method for improving the service set, we should first compute the service indicators (cohesion and the coupling degree) and then perform service clustering to improve the indicators; this process is also called service composition.
To determine which services should be clustered, an objective cohesion of all services and the coupling among services. Our objective is to find a service set with high cohesion and the service clusters; it is the ratio of cohesion and coupling.

Function indicates both the degree of coupling and the cohesion of Equation (7).

The calculation of cohesion for single service is defined as the interconnection of activities executed by the service. The calculation of cohesion for single service \( S \) is illustrated in Equation (5). \(| S |\) means the number of activities that service \( S \) supports. \( \sum TT(t_i, t_j) \) indicates the sum of the integrated activity dependency for all activity pairs.

The coupling of two services is defined as the interconnection among activities executed by the service. The calculation of the coupling of services \( S_1 \) and \( S_2 \) is shown in Equation (7).

The combination of services across different operational nodes is not permitted. For example, \( S_B \) and \( S_I \) cannot be combined because of this condition.

After combination, we calculate the value of the new service set \( \Omega^2 \); if \( \Omega^2 < \Omega \), we accept the service combination and update the service set.

Third, we undertake a new round of service combination search until \( \Omega \) cannot be higher. Then, we obtain the final improved service set scheme \( \Delta \).

The service combination process of the motivation case is shown in Fig. 13. The combination of services across different operational nodes is not permitted. For example, \( S_B \) and \( S_I \) cannot be combined because of this condition.

According to Equations (5) and (7), we can calculate the cohesion of all services and the coupling among services. To determine which services should be clustered, an objective function \( \Omega \) of consolidation is defined in Equation (8). \( \Omega \) indicates both the degree of coupling and the cohesion of the service clusters; it is the ratio of cohesion and coupling. Our objective is to find a service set with high cohesion and low coupling; thus, the objective of the method for improving the service set is to minimize \( \Omega \). \( \text{coh}_\text{pro}(\text{OAP}) \) is the cohesion of the operational activity process, defined as the cohesion of all services divided by the amount of services \(| \Delta |\) in the operational activity process. \( \Delta \) is the service set; \(| \Delta |\) shows the numbers of services. \( \text{coup}_\text{pro}(\text{OAP}) \) is the degree of coupling of the operational activity processes in \( \Delta \). \( \text{coup}_\text{pro}(\text{OAP}) \) is defined as the sum of coupled services divided by the number of pairs of services in the operational activity process.

Algorithm 4 shows the method for improving the initial service scheme in Fig. 12. In Algorithm 4, we set service_updating as the mark of whether or not to improve the service set.

In Algorithm 4, First, we calculate the cohesion and coupling matrix of all services and calculate the initial objective function of \( \Omega \). We set service_updating as true.

Second, we choose two services to combine. Among the pairs of services that can be combined, services with the highest cohesion and lowest coupling should be selected first. At lines 7 and 8 in Algorithm 4, to select two effective services, we combine services with the highest cohesion and lowest coupling should be selected first.

Condition 1: Services \( S_i \) and \( S_j \) be valid subprocesses and in the right sequence.

Condition 2: With services \( S_i \) and \( S_j \), one service cannot come before the split, while the other service comes after split. For example, \( S_C \) and \( S_D \) cannot be combined because of this condition.

Condition 3: The combination of services across different operational nodes is not permitted. For example, \( S_B \) and \( S_I \) cannot be combined because of this condition.

Algorithm 4: Improving the service set

//Input:OV-6b information table, conditions for performing combination, initial service set \( \Delta \);
//Output: Service set \( \Delta \) after combination.
1: for all services 
2: begin 
3: calculate \( \Omega \);
4: \( \Delta' = \Delta \);
5: service_updating=true;
6: while (|\( \Delta | = 1 \) and service_updating) 
7: select two services \( S_I \) and \( S_J \) in \( \Delta \)
8: if \( S_I \) and \( S_J \) cannot meet conditions of combination 
9: break;
10: else 
11: calculate \( \Omega' \) for each service pairs; 
12: find the service pairs that have minimal value of \( \Omega' \);
13: if(\( \Omega' < \Omega \)) 
14: add composition \( S_I \) into \( \Delta' \);
15: delete \( S_I \) and \( S_J \) from \( \Delta' \);
16: \( \Delta = \Delta' \), \( \Omega = \Omega' \);
17: else 
18: break;
19: end if
20: end if
21: end while
22: return \( \Delta \);
22: end
FIGURE 13. Service combination process in the motivation case after applying Algorithm 4. Notes: Service \( S_D \) and \( S_E \) are combined (as a new \( S_D \)) to support the completion of activity \( t_4, t_5 \) and \( t_6 \); Service \( S_L \) and \( S_M \) are combined (as a new \( S_L \)) to support the completion of activity \( t_{13}, t_{14} \) and \( t_{15} \); Service \( S_Q \) and \( S_R \) are combined to support the completion of activity \( t_{20}, t_{21} \) and \( t_{22} \); and Service \( S_N \) and \( S_O \) are combined to support the completion of activity \( t_{16}, t_{19}, t_{17} \) and \( t_{18} \).

TABLE 8. Service information after algorithm 4.

| No. of services | Name of services                                      | Supporting activities |
|-----------------|-------------------------------------------------------|-----------------------|
| \( S_A \)       | Reconnaissance service                                | \( t_1 \)             |
| \( S_B \)       | Reconnaissance information analysis service           | \( t_2 \)             |
| \( S_C \)       | Assess disaster service                               | \( t_3 \)             |
| \( S_D \)       | Command and operation service                         | \( t_4, t_{15}, t_{16} \) |
| \( S_E \)       | Surveillance disaster service                         | \( t_7 \)             |
| \( S_F \)       | Earthquake warning service                            | \( t_8 \)             |
| \( S_G \)       | Seismic monitoring service                            | \( t_9 \)             |
| \( S_H \)       | Force assembly service                                | \( t_{10} \)           |
| \( S_I \)       | Receive EWI service                                   | \( t_{11} \)           |
| \( S_J \)       | Receive rescue order service                          | \( t_{12} \)           |
| \( S_K \)       | Team rescue plan formulation service                  | \( t_{13}, t_{14}, t_{15} \) |
| \( S_L \)       | Rescue service                                        | \( t_{16}, t_{17}, t_{18}, t_{19} \) |
| \( S_M \)       | Logistics service                                     | \( t_{20}, t_{21}, t_{22} \) |

service name and to map these services to existing services. Table 8 shows service names and their supporting activities based on the results of Fig. 13. For example, service \( S_D \) supports activities \( t_4, t_5, \) and \( t_6 \); \( t_4 \) is making the rescue plan, \( t_5 \) is allocating rescue forces, and \( t_6 \) is releasing rescue plans; thus, designers may decide that service \( S_D \) can be called command and operation service.

VII. SVCV-5 GENERATION

After the service set improving process, we obtain the final service set. In this section, we study how to generate the svcv-5 matrix. Based on the final service set and the methods for combining services, we can determine which services support the execution of which operational activities and finally obtain the mapping relation based on the service set result and the information in OV-6b.

The mapping relation between services and operational activities can be generated and should obey the following rules:

a) If services can support activities directly, the relationship in the svcv-5 matrix is strongly related, represented as “●”.

b) If services cannot support activities directly but the activity has the input and output information with activities in the service sets in OV-6b, the relationship in the svcv-5 matrix is a middle relation, represented as “○”.  

c) If services cannot support activities directly and the activity does not have the input and output information with activities in the service sets in OV-6b but the activity has a direct line with activities in the service sets in OV-6b, the relationship in the svcv-5 matrix is weakly related, represented as “□”.

Algorithm 5 in Fig. 14 illustrates the steps for generating the svcv-5 matrix from the results generated by Algorithm 4 and the data in OV-6b. According to the mapping result of the operational activities generated in Fig. 13 and Algorithm 5, the mapping of different services to service instances and the mapping of operational activities to service mapping matrix products (Svcv-5) can be generated, and the result is shown in Table 8.

In Algorithm 5, the priority of “○" is lower than that of “●", and the priority of “□" is lower than that of “○". Therefore, if the activity and the service both have “●” and “○", then the relationship should be “●”; if the relation between the activity and the service both have “○” and “□”, then the relationship should be “□". Svcv-5 can help design other products based on DoDAF 2.0 according to the motivation case in this paper, such as Svcv-4 and Svcv-1.

In Table 9, the strong relation (represented as “●") between the activity and service means the service is strongly needed for finishing the activity, such as \( S_D \) (command and operation service) strongly supports activity \( t_4 \) (making rescue plan), \( t_5 \) (allocate rescue force) and \( t_6 \) (release rescue order). The middle relation (represented as “○") means the service is moderately needed for finishing the activity, such as \( S_D \) (command and operation service) implies support
Algorithm-5: SvcV-5 generation algorithm

//Input: Service sets after aggregation, data in OV-6b
//Output: mapping information in SvcV-5_matrix
1: while(\text{true})
2:    for all operational activities $t_i$
3:        for all services $S_j$
4:            set SvcV_matrix($t_i$, $S_j$)=0;
5:        if ($S_j$ can support $t_i$) //check results of the service set
6:            set SvcV5_matrix($t_i$, $S_j$)=1;
7:        end if
8:    if($\exists$($t_i$.out($T(t_i)$.information in $S_j$)))
9:        set SvcV5_matrix($t_i$, $S_j$)=2;
10:   end if
11:    if(there is a line between $t_i$ and activities that are in $S_j$
12:        set SvcV5_matrix($t_i$, $S_j$)=3;
13:    end if
14:    end for
15: end for
16: return SvcV5_matrix
17:end while

**FIGURE 14.** Algorithm 5 for generating SvcV-5. The value of 1 represents a strong relation, denoted by “\text{1}”; the value of 2 represents a middle relation, denoted by “\text{2}”; and the value of 3 represents a weak relation, denoted by “\text{3}”.

activity $t_2$ (reconnaissance information analysis) and $t_3$ (assess disaster). The weak relation (represented as “\text{3}”) indicates the service is weakly needed for finishing the activity, such as $S_D$ (command and operation service) weakly supports activity $t_{12}$ (receive rescue order).

**VIII. EVALUATION AND COMPARISON WITH RELATED WORKS**

**A. EXPERIMENTAL SETUP AND ANALYSIS OF THE PERFORMANCE OF THE METHOD**

To run experiments, we set up a dataset containing different processes, different numbers of activities and different numbers of capabilities. The dataset is built by considering different aspects that can influence the performance of the method.

To analyze the influence of these five parameters on the SvcV-5 generation method, we set different levels of complexity of the parameters in the CV-4, CV-6, and OV-6b products. The experiment was performed on a Windows 7 system, and we used the MATLAB tool to realize the algorithms. Fig. 15 shows the results of how different parameters influence the performance of the proposed method.

The aspects include the following:

1) The complexity of the structure in OV-6b, which means the different numbers of operational nodes in OV-6b. The greater the number of branches (ranging from 2 to 7) in the processes, the greater the complexity of the structure complexity is, and the longer the execution of algorithms is, see Fig. 15(a).

2) The complexity of operational nodes. The greater the number of operational nodes (ranging from 2 to 8), the greater the complexity of the operational nodes is, and the shorter the service identification method runs, see Fig. 15(b). As the number of nodes increases, the number of activities of a single node or a lane decreases. Since the service identification algorithm is performed for the same node, the execution time of the total algorithm is reduced.

3) The complexity of operational capabilities. The greater the number of operational capabilities (from 2 to 24) in CV-4 and CV-6, the greater the operational capability is, but the execution time of algorithms are not affected by the complexity of capabilities significantly, see Fig. 15(c). This may occur because the computation time for the capability is very small compared to the execution time of other parts of the algorithm.

4) The complexity of operational information. The more kinds of operational information (from 4 to 24), the greater the operational information is, and the longer the execution of algorithms; however, the execution time is less affected, see Fig. 15(d).

**B. QUALITY OF SVCV-5 OPTIMIZATION DESIGN**

We invited architecture design experts to manually apply the SvcV-5 design to motivation case. We computed the value based on the obtained set of services and the matrix value in the SvcV-5 matrix. We monitored 5 users who performed service identification and the SvcV-5 generation method with and without our method. The optimization design result of the SvcV-5 product is recognized by all users. The users have different levels of skills in the service identification area and the architecture design area. Users 1 and 2 have high-level skills in the architecture design area, while users 3, 4, and 5 have middle-level skills in the architecture area. Users 2 and 3 have high-level skills in the service...
identification area, while users 1, 4, and 5 have middle-level skills in the service identification area.

Fig. 16 shows the correct rate results for the 5 users on the SvcV-5 design in the ERIS architecture. The correct rate of SvcV-5 products is the ratio of the number of correct relationships on the right place to the total place. The correct rate of the service designing result is the ratio of the number of clusters of activities to the total clusters of activities (one cluster is called a service).

For five users, we separately analyze how the method we proposed supports different kinds of users when they design the architecture product SvcV-5.

For user 1, since he has high-level skills in the architecture design area and middle-level skills in the service identification area, we can see that user 1 has a relatively moderate success rate (0.42 for designing SvcV-5 products without the SvcV-5 generation method and 0.38 for service designing results without the SvcV-5 optimization design method). However, once user 1 employs the help of the SvcV-5 generation method and SvcV-5 optimization design method, he can achieve high correct rates for designing both SvcV-5 products and service results (0.70 for designing SvcV-5 products with the SvcV-5 generation method and 0.92 for service designing results with the SvcV-5 optimization design method).

For user 2, since he has high-level skills in the architecture design area and high-level skills in the service identification area, we can see that user 2 has a relatively moderate success rate (0.48 for designing SvcV-5 products without the SvcV-5 generation method, which is higher than user 1). We can also see that user 2 has a success rate of 0.38 for service designing results without the SvcV-5 optimization design method. It is the same as user 1. This is because both user 2 and user 1 design SvcV-5 by experience, and they are both at the same level in architecture design. The main difference between user 1 and user 2 is that with the help of the SvcV-5 generation method, user 2 can design SvcV-5 more correctly than user 1 (0.70 vs. 0.56 in Fig. 16).

For user 3, user 4, and user 5, they have middle-level skills in the architecture design area, and user 3 has high-level skills in the service identification area, while users 4 and 5 have middle-level skills in the service identification area. Therefore, user 3 achieves a higher correct rate with the SvcV-5 optimization design method.

For all five, user 1, user 2, user 3, user 4 and user 5, with the SvcV-5 optimization design method, the SvcV-5 design quality can apparently be improved, which is shown in Fig. 16.
TABLE 10. Comparison of the service generation method with other service identification approaches.

| Development type | Whether SvcV-5 design is supported | Whether consider influence of requirement | Use of quantitative metrics | Availability of procedural guidelines | Availability of tool support | Validation |
|------------------|-----------------------------------|-----------------------------------------|-----------------------------|--------------------------------------|-----------------------------|------------|
| Our method       | Top-down and bottom-up            | Yes                                     | Yes (message-based initial service generation, calculate the cohesion and coupling of service set) | Yes                                 | No                          | Case study (Eris design case) |
| Devis Bianchini et al. [38] | Top-down                         | No                                      | No                           | Yes (service identification, refinement and reconciliation) | Yes[P2S tool] | Case study (sofa production case) |
| M.P. Papazoglou et al. [39] | Top-down                         | No                                      | No                           | No                                   | No                          | No         |
| Y. Kim et al. [40] | Top-down                         | No                                      | Yes (clustering algorithm)   | Yes (high-level development principles) | No                          | No         |
| T. Kleinert et al. [41] | Top-down                         | No                                      | Yes (tree-based structure)   | Yes (identify process regions that represent service candidates) | No                          | Case study |
| H. Leopold et al. [42] | Bottom-up                        | No                                      | No                           | Yes (logic-based matchmaking)        | No                          | No         |
| S. Ghosh et al. [43] | Top-down and bottom-up           | No                                      | No                           | Yes                                  | No                          | Case study |

Fig. 16 also shows that the adoption of the SvcV-5 generation method can improve the design correct rate for designers with high-level skills in the architecture design area, and can also mitigate the skill gap between different users who perform SvcV-5 design to a certain degree, such as user 2 and user 3, as they have a close success rate with the SvcV-5 optimization method (0.85 and 0.92, respectively).

C. COMPARATIVE ANALYSIS OF OUR WORK WITH RELATED METHODS

There is no quantitative SvcV-5 optimization design method or SvcV-5-aided design method in current studies. As the service identification method plays a significant role in the optimization design of SvcV-5, we compare the service identification method in the SvcV-5 optimization method with some representative service identification methods. From the perspectives of the development type, we consider whether there is support of the SvcV-5 design, whether the influence of the capability requirement is considered, the use of quantitative metrics, the availability of procedural guidelines, the availability of tool support, and validation, and thereby compare our method with available identification methods. Table 10 shows the comparison results. In Table 10, only our method can support the SvcV-5 design, and our method considers the influence of the requirement in SvcV-5 design, which is ignored by most studies. Furthermore, most of the studies either suggest guidelines for service identification without providing quantitative metrics to evaluate the quality of the identified services or use a limited set of metrics without supporting the SvcV-5 optimization design. However, our method considered the cohesion and coupling during the service identification process, and thus can support such an analysis. The shortcoming of our method is the lack of tool support, which will be addressed in our further study.

The advantage of the SvcV-5 optimization method includes four aspects: the methodological detail guidance, the coverage of the service identification phases and SvcV-5 generation phase, the part about automation, and systematically considering the data relationship among architecture viewpoint products in DoDAF 2.0.

The first advantage of the methodological detail guidance, refers to high-level descriptions of many service derivation approaches. Our method provides sufficient methodological details of how to practice. As a result, it is clear how our approach deals with specific characteristics of process models such as structural and linguistic information. The methodological detail makes it easy to apply our approach in practice and to re-implement the automated support they provide.

The second advantage is coverage of the service identification phases and SvcV-5 generation phase. Our approach provides a total solution to the service identification and SvcV-5 generation. As a top-down and bottom-up approach, we considered both the capability requirement and activity process issues (derived from CV-4, CV-6, and OV-6b),
and provide automated support for service identification by introducing two indicators (cohesion and coupling) and then perform service clustering to improve the indicators. In this way, we provide the technique that supports the user in automatically prioritizing the identified services and generating the best selection.

The third advantage refers to the part about automation. In our method, nearly all phases consider the potential of automating. Very little manual operation is required here to decide the service name and to map these services to existing services. Hence, compared with the traditional empirical design of SvcV-5, our method can efficiently and effectively contend with the situation in which large process model (OV-6b) repositories are available.

The fourth advantage refers to systematically considering the data relationship among architecture viewpoint products in DoDAF 2.0. During the process of our method, we analyzed the optimization design process of SvcV-5 by considering the capability viewpoint (such as CV-4 and CV-6), and the operational viewpoint (such as OV-6b). Such an optimization designing process of SvcV-5 is good for designers to master the overall design of the architecture, and thus help to improve the quality of the entire architecture design.

IX. CONCLUSION

In this work, we propose a semi-automatic optimization design method for SvcV-5. The method is mainly based on service identification. The relevant information is extracted from OV-6b, CV-2, CV-4 and CV-6. First, the initial service identification method based on messages is proposed. Second, the dependency matrix of operational activities is established, and the service set is improved by applying the service improving method that measures both the cohesion and coupling degree of the service set. Then, according to the mapping between the clustering results and the experience of designers, the service design in the architecture design is realized. Finally, we proposed the SvcV-5 generation method based on the service identification result and data input and output relationship. The SvcV-5 generation method can support the semi-automatic generation of SvcV-5. This paper validates the correctness and performance of the proposed method by designing the SvcV-5 in the ERIS architecture, and we compare our method with other service identification methods to illustrate the effectiveness of the proposed method. The main contribution of our method is the combination of the methodological perspective of service identification and SvcV-5 generation. Future work will include integrating our method with architecture design tools and exploring optimization design methods for other products in DoDAF 2.0.

REFERENCES

[1] M. W. Maier, D. Emery, and R. Hilliard, “Software architecture: Introducing IEEE standard 1471,” Computer, vol. 34, no. 4, pp. 107–109, Apr. 2001.
[2] S. J. Ring and D. Nicholson, “Activity-based methodology for development and analysis of integrated DoD architectures,” in Handbook of Enterprise Systems Architecture in Practice, P. Saha, Ed. Hershey, PA, USA: Information Science Reference, 2007, ch. 5, pp. 85–113.
[3] Telelogic Company. (2008). Telelogic System Architect for DoDAF-Enterprise Software Solution for the Department of Defense Architecture Framework. [Online]. Available: http://www.telelogic.com/products/systemarchitect/systemarchitectfordodaf/index.cfm
[4] R. Wang and C. H. Dagli, “Executable system architecture using systems modeling language in conjunction with colored Petri nets in a model-driven systems development process,” Syst. Eng., vol. 14, no. 4, pp. 383–409, Dec. 2011.
[5] DoD Architecture Framework Version 2.0 Volume II: Architectural Data and Models. DoD Archit. Framework Working Group, United States Dept. Defense., Arlington, VA, USA, 2009.
[6] DoD Architecture Framework Version 2.0 Volume III: DoD Meta-Model Physical Exchange Specification. DoD Archit. Framework Work. Group, United States Dept. Defense., Arlington, VA, USA, 2009.
[7] Z. Xiaoxue, Z. Mengmeng, and L. Gang, “Research on service optimization design of military information system architecture,” in Proc. 8th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS), Nov. 2017, pp. 828–831.
[8] X. X. Zhang, G. Liu, A. M. Luo, and J. X. LIU, “Design method of SvcV5 in DoDAF2.0 based on service discovery,” in Proc. 8th Int. Workshop Comput. Sci. Eng., Bangkok, Thailand, 2018, pp. 672–677.
[9] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, and K. Holley, “SOMA: A method for developing service-oriented solutions,” IBM Syst. J., vol. 47, no. 3, pp. 377–396, 2008.
[10] F. Kramer, C. Goring, and S. Wind, “Service identification—An explorative evaluation of recent methods,” in Proc. 47th Hawaii Int. Conf. Syst. Sci., Jan. 2014, pp. 1285–1295.
[11] L. W. Wagenhals, I. Shin, D. Kim, and A. H. Lewis, “C4ISR architectures: II. A structured analysis approach for architecture design,” Syst. Engin., vol. 3, no. 4, pp. 248–287, 2000.
[12] M. P. Bienvenu, I. Shin, and A. H. Lewis, “C4ISR architectures: III. An object-oriented approach for architecture design,” Syst. Engin., vol. 3, no. 4, pp. 288–312, 2000.
[13] System Architect USRPROPS Extensibility Guide. Popkin Softw., New York, NY, USA, 2008.
[14] Telelogic Company. (2008). Telelogic System Architect for DoDAF-Enterprise Software Solution for the Department of Defense Architecture Framework. [Online]. Available: http://www.telelogic.com/products/systemarchitect/systemarchitectfordodaf/index.cfm
[15] Vitech Corporation. (2007). Core 5.15 New Features Guide. [Online]. Available: http://www.vitechcorp.com/support/docs/515/WhatsNew.pdf
[16] R. Nord, M. Barbacci, P. Clements, R. Kazman, and M. Klein, “Integrating the architecture tradeoff analysis method (ATAM) with the cost benefit analysis method (CBAM),” Softw. Eng. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU/SEI-2003-TN-038, 2003.
[17] R. Kazman, M. Klein, and P. Clements, “ATAM: Method for architecture evaluation,” Softw. Eng. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU/SEI-2000-TR-004, 2000.
[18] R. Kazman, L. Bass, G. Abowd, and M. Webb, “SAAM: A method for analyzing the properties of software architectures,” in Proc. IEEE 16th Int. Conf. Softw. Eng. Los Alamitos, CA, USA: IEEE Computer Society Press, May 1994, pp. 81–90.
[19] X. S. Luo, A. M. Luo, and Y. H. Zhang, Application of Petri-Net in C4ISR System Modeling, Simulation and Analysis. Changsha, China: National Univ. of Defense Technology Press, 2007.
[20] B. Ge, K. W. Hipel, K. Yang, and Y. Chen, “A novel executable modeling approach for system-of-systems architecture,” IEEE Syst. J., vol. 8, no. 1, pp. 4–13, Mar. 2014.
[21] S. Mittal, “Extending DoDAF to allow integrated DEVS-based modeling and simulation,” J. Defense Model. Simul., vol. 3, no. 2, pp. 95–123, Apr. 2006.
[22] S. Mittal, A. Gupta, A. Mitra, and B. Zaieger, “Strengthening OV-6a semantics with rule-based meta-models in DEVS/DoDAF based life-cycle architectures development,” in Proc. IEEE Inf. Reuse Integr., Special Sect. (DoDAF), Waikoloa Village, HI, USA, Sep. 2006, pp. 80–85.
[23] H. Koziolek and R. Reussner, “A model transformation from the paladioc component model to layered queuing networks,” in Performance Evaluation: Metrics, Models and Benchmarks. SPECT (Lecture Notes in Computer Science), vol. 5119, S. Kounev, I. Gorion, and K. Sachs, Eds. Berlin, Germany: Springer, 2008, pp. 58–78.
[24] S. Cai, Y. Liu, and X. Wang, “A survey of service identification strategies,” in Proc. IEEE Asia-Pacific Services Comput. Conf., Dec. 2011, pp. 464–470.

[25] A. Shatnawi, H. Mili, M. Abdellatif, G. El Boussaidi, Y.-G. Guéhéneuc, N. Moha, and J. Privat, “What should you know before developing a service identification approach?” LATICE, Univ. du Québec à Montréal, Montreal, QC, Canada, Tech. Rep. 2017-2, 2017.

[26] M. Amiri, S. Parsa, and A. Lajevardi Mohammadzade, “Multifaceted service identification: Process, requirement and data,” Comput. Sci. Inf. Syst., vol. 13, no. 2, pp. 335–358, 2016.

[27] H. M. Shirazi, N. Fareghzadeh, and A. Seyyedi, “A combinational approach to service identification in SOA,” J. Appl. Sci. Res., vol. 5, no. 10, pp. 1390–1397, 2009.

[28] M. Daghaghzadeh, A. B. Dastjerdi, and H. Daghaghzadeh, “A metric for measuring degree of service cohesion in service oriented designs,” Int. J. Comput. Sci., vol. 8, no. 2, pp. 83–89, 2011.

[29] R. S. Huergo, P. F. Pires, F. C. Delicato, B. Costa, E. Cavalcante, and T. Batista, “A systematic survey of service identification methods,” Service Oriented Comput. Appl., vol. 8, no. 3, pp. 199–219, Sep. 2014.

[30] S. Mansour, P. Jamshidi, K. Sedighi, and S. Jamshidi, “An automated service identification method,” Automated Softw. Eng. Res. Group, Shahid Beheshti Univ., Tehran, Iran, Tech. Rep. TR-ASER-2012-01. 2012.

[31] T. Kohlborn, A. Korthaus, T. Chan, and M. Rosenmann, “Identification and analysis of business and software services—A consolidated approach,” IEEE Trans. Serv. Comput., vol. 2, no. 1, pp. 50–64, Jan. 2009.

[32] A. T. Zadeh, S. Sahari, and M. Mukhtar, “Automated service identification methods: A review,” Int. J. Adv. Sci., Eng. Inf. Technol., vol. 6, no. 6, p. 1053, Dec. 2016.

[33] A. T. Zadeh, M. Mukhtar, and S. Saharian, “Inheritance phases of automated service identification framework (ASIF),” J. Theor. Appl. Inf. Technol., vol. 83, no. 3, pp. 451–464, 2016.

[34] M. Reichert, “Process and Data: Two Sides of the Same Coin?”, in On the Move to Meaningful Internet Systems: OTM, Berlin, Germany: Springer, 2012, pp. 2–19.

[35] R. S. Huergo, P. F. Pires, and F. C. Delicato, “A method to identify services using master data and artifact-centric modeling approach,” in Proc. 29th Annu. ACM Symp. Appl. Comput. (SAC), 2014, pp. 1225–1230.

[36] Q. Gu and P. Lago, “Service identification methods: A systematic literature review,” in Towards a Service-Based Internet. ServiceWave (Lecture Notes in Computer Science), vol. 6481, E. Di Nitto and R. Yahyapour, Eds. Berlin, Germany: Springer, 2010, pp. 37–50.

[37] D. Bianchini, C. Cappiello, V. De Antonellis, and B. Pernici, “Service identification in interorganizational process design,” IEEE Trans. Serv. Comput., vol. 7, no. 2, pp. 265–278, Apr. 2014.

[38] M. Papazoglou and W-J. Van Den Heuvel, “Service adaptation: Business process development life cycle methodology,” Commun. ACM, vol. 50, no. 10, pp. 79–85, Oct. 2007.

[39] Y. Kim and K. Doh, “Formal identification of right-grained services for service-oriented modeling,” in Proc. WISE, 2009, pp. 261–273.

[40] T. Kleiner, S. Balzert, P. Fettke, and P. Loos, “Systematic identification of service-blueprints for service-processes—A method and exemplary application,” in Business Process Management Workshops (Lecture Notes in Business Information Processing), vol. 132, 2013, pp. 598–610.

[41] H. Leopold, F. Pittke, and J. Mendling, “Automatic service derivation from business process model repositories via semantic technology,” J. Syst. Softw., vol. 108, pp. 134–147, Oct. 2015.

[42] N. Merski and J. Colombi, “Tailored system architecture for design of S&T missions using DoDAF2.0,” in Proc. IEEE Aerosp. Conf. Proc., Mar. 2010, pp. 1–17.

[43] Z. G. Tao and M. Z. Wang, “Study on system of systems capability architecture development based on DoDAF V2.0,” Int. J. Syst. Syst. Eng., vol. 3, no. 1, p. 47, 2012.

[44] Z.-G. Tao, Y.-F. Luo, C.-X. Chen, M.-Z. Wang, and F. Ni, “Enterprise application architecture development based on DoDAF and TOGAF,” Enterprise Inf. Syst., vol. 11, no. 5, pp. 627–651, May 2017.

[45] A. Hunton, R. Frye, C. Hammer, T. Haynes, P. Khuu, S. Mosteiro, R. Nogay, and M. Weber, “Use of DoDAF and M&S for the design requirements and optimization of a GIG-enabled wideband mesh-networking waveform,” in Proc. IEEE Conf. Mil. Commun., Oct. 2006, pp. 1–7.

[46] J. D. Pilcher, “Generation of department of defense architecture framework (DoDAF) models using the Monterey Phoenix behavior modeling approach,” M.S. thesis, Naval Postgraduate School, Monterey, CA, USA, 2015.

XIAOXUE ZHANG was born in China, in 1986. She received the B.S. degree in system engineering, the M.S. degree in management science and engineering, and the Ph.D. degree in military operation research from the National University of Defense Technology, in 2008, 2010, and 2014, respectively.

From 2014 to 2019, she was a Lecturer with the Science and Technology on Information Systems Engineering Laboratory. She has authored more than 15 articles. She has taken participated in the design of multiple architecture under different background. Her research interests include information system architecture design, analysis, and optimization.

AIMEI LUO was born in China, in 1970. She received Ph.D. degree from the School of Information Systems and Management, National University of Defense Technology, in 2003.

She was a Professor and a Doctor/Master’s Supervisor with the School of Systems Engineering. She has authored three books, more than 70 articles, and more than 5 inventions. She has been engaged in teaching and researching on information theory and information systems architecture. Her research interests include basic theory of information systems and information system architecture, modeling, evaluation, and evolution.

YI MAO was born in China, in 1985. He received the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics. He is currently an Engineer with the State Key Laboratory of Air Traffic Management System and Technology, China. His research interests include transportation planning and management, and information system architecture.

MENGLONG LIN was born in China, in 1993. He received the M.S. degree in control science and engineering from the National University of Defense Technology, in 2018, where he is currently pursuing the Ph.D. degree with the College of Systems Engineering. His research interests are information system architecture design, analysis, and validation.

YUXIAO KOU was born in China, in 1996. He received the B.E. degree in command information system engineering from the National University of Defense Technology, Changsha, China, in 2017, where she is currently pursuing the master’s degree in management science and engineering. Her research is focused on analyzing and evaluating information systems.

JUNXIAN LIU was born in China, in 1973. He received Ph.D. degree from the School of Information Systems and Management, National University of Defense Technology, in 2003.

He was a Professor and Master’s Supervisor with the School of Systems Engineering. He has authored 3 books and more than 40 articles. He has been engaged in teaching and researching on design and evaluation of information systems. His research interests include information systems theory, and information system architecture design and evolution.

* * *