Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Enterprise-level business component identification in business architecture integration*

Jiong FU‡, Xue-shan LUO, Ai-min LUO, Jun-xian LIU

(*Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China*)

E-mail: jiongfu@nudt.edu.cn; xsluo@nudt.edu.cn; amluo@nudt.edu.cn; 18674864900@163.com

Received Dec. 20, 2016; Revision accepted Apr. 10, 2017; Crosschecked Sept. 30, 2017

Abstract: The component-based business architecture integration of military information systems is a popular research topic in the field of military operational research. Identifying enterprise-level business components is an important issue in business architecture integration. Currently used methodologies for business component identification tend to focus on software-level business components, and ignore such enterprise concerns in business architectures as organizations and resources. Moreover, approaches to enterprise-level business component identification have proven laborious. In this study, we propose a novel approach to enterprise-level business component identification by considering overall cohesion, coupling, granularity, maintainability, and reusability. We first define and formulate enterprise-level business components based on the component business model and the Department of Defense Architecture Framework (DoDAF) models. To quantify the indices of business components, we formulate a create, read, update, and delete (CRUD) matrix and use six metrics as criteria. We then formulate business component identification as a multi-objective optimization problem and solve it by a novel meta-heuristic optimization algorithm called the 'simulated annealing hybrid genetic algorithm (SHGA)'. Case studies showed that our approach is more practical and efficient for enterprise-level business component identification than prevalent approaches.

Key words: Business architecture integration; Business component; Component identification; Create, read, update, and delete (CRUD) matrix; Heuristic
https://doi.org/10.1631/FITEE.1601836 **CLC number:** TP303

1 Introduction

With the rapid development of information technology and its wide application to the military, integrated joint operation (IJO) has emerged as a major style of combat for the future (Fu *et al.*, 2016). To meet the requirements of IJO, operational functions and resources need to be integrated. From the perspective of architecture, business architectures should be integrated by analyzing business models to obtain a set of business components with high reuse value, since business architecture concerns mainly business functions and resources. In component-based business architecture integration, enterprise-level business component identification is a primary research problem (Meng *et al.*, 2005), and is our main concern in this study.

Enterprise-level business components, in contrast with software-level business components that ignore enterprise concerns (Scheer, 2000), are the basic units for the integration of business architecture. Prevalent methodologies for business component identification tend to focus on software-level business components and are not applicable at the scale of enterprises. There are two typical categories of such methods: cohesion coupling based cluster-

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. 71571189)
ORCID: Jiong FU, http://orcid.org/0000-0001-8298-141X

ing methods (Jain *et al.*, 2001; Lee *et al.*, 2001), and the create, read, update, and delete (CRUD) matrix based methods (Lee *et al.*, 1999; Ganesan and Sengupta, 2001). Cohesion coupling based clustering methods convert business models into weighted directional graphs by analyzing the semantic relationships among business elements, and cluster business components by dividing the graphs. CRUD matrix based methods convert business models into CRUD matrices and identify business components through a CRUD matrix clustering algorithm. However, both types of methods aim at specific business model types, e.g., unified modeling language (UML) use case and UML activity model, which are micro-level abstractions, and ignore other performance metrics during identification (Wang *et al.*, 2008), e.g., granularity, reusability, and maintainability. Moreover, methods for enterprise-level business component identification have proven qualitatively poor, laborious, and less productive, e.g., the business system planning method (Yang *et al.*, 2002) and the computer-integrated manufacturing openness system architecture (CIMOSA) based method (Qin *et al.*, 2003).

We propose a novel approach to enterprise-level business component identification where the overall cohesion, coupling, granularity, maintainability, and reusability are considered. We define enterprise-level business components in business architecture integration, and map their relationships to the Department of Defense Architecture Framework (DoDAF) models from business architecture to gather architecture data for the business components. To quantify the indices of the business components, we convert DoDAF models into a CRUD matrix and use six technical metrics as criteria. We then formulate business component identification as a multi-objective optimization problem by maximizing cohesion, granularity, maintainability, and reusability, and by minimizing coupling. To solve the multi-objective optimization problem, we then use a meta-heuristic optimization algorithm called the 'simulated annealing hybrid genetic algorithm (SHGA)'.

The contributions of this work are three-fold: (1) We define the formal model of enterprise-level business components and map their relationships with DoDAF models. This creates the possibility for transforming DoDAF models into business components. (2) We consider the overall cohesion, coupling, granularity, maintainability, and reusability of the components during identification, and quantify these indices by using six technical metrics. (3) We formulate business component identification as a multi-objective optimization problem and solve it by a novel meta-heuristic optimization algorithm.

2 Related work

According to different sources and types of objective components, the related research can be categorized into two branches: top-down and bottom-up identification approaches. In top-down identification approaches, researchers identify business components using requirement models, where the business components belong to the enterprise level. In bottom-up identification approaches, researchers identify business components from legacy systems, where the business components belong to the software level.

2.1 Top-down identification approaches

Yang *et al.* (2002) proposed a business system planning (BSP) method to identify business activities during the system planning program. Although this method has been applied at enterprise-level scale, it lacks sufficiently rigorous principles and guidelines for identification, which leads to rough results and a laborious identification process.

Qin *et al.* (2003) developed a CIMOSA-based method to identify business components. This method uses the CIMOSA function view as the principle to identify enterprise-level business components from business functions. However, there are no measurable metrics during identification and the method is qualitative.

Levi and Arsanjani (2002) proposed a goal-driven approach to enterprise component identification and specification. This approach centers on enterprise business processes, business rule models, and so on, to goal service graphs (GSGs), and identifies components by decomposing the GSG. However, the method provides only a process for designing the GSG, whereas the decomposition algorithm of the GSG is limited.

2.2 Bottom-up identification approaches

Bottom-up identification approaches can be classified into cohesion coupling based clustering analysis methods and CRUD-based identification methods.

The former class of methods, such as those proposed by Meng *et al.* (2005), Yuan and Qin (2009), and Cai *et al.* (2011), convert business models into weighted directional graphs where the nodes represent business elements and the weights of edges between nodes represent the strength of the semantic dependence. By using graph clustering or matrix analysis techniques, the graph can then be clustered into several parts, which are the business components.

Meng *et al.* (2005) proposed an approach to identify business components from a domain business model by using a hierarchical clustering technique based on a graph. The method substitutes edge strength for edge weight, and considers cohesion, coupling, and granularity as technical metrics.

A business component identification approach based on decoupling analysis was presented by Yuan and Qin (2009). The method uses graph partitioning based on a business process graph (BPG) to identify the subsets of the coupling business process. Cohesion and coupling are the main technical metrics.

Cai *et al.* (2011) dealt with component identification from business models using fuzzy formal concept analysis. By converting memberships of business elements and their properties into a fuzzy formal concept graph, the components can be identified according to concept dispersion and distance. The method focuses on cohesion and coupling throughout the identification process using an automatic algorithm with high precision.

CRUD-based identification methods use four semantic relationships (Create-C, Read-R, Update-U, and Delete-D) between behavioral and static elements to calculate the association weight, and merge use cases and entities with C or D relationship into one business component (Wang *et al.*, 2008).

Lee *et al.* (1999) proposed a UML model based object-oriented component development method (COMO) to identify components from use cases and classes by creating a CRUD matrix between the use cases and the classes. Although the method designs clustering algorithms for identification, only cohesion and coupling are considered. Moreover, the method focuses on UML use cases and classes, which are not suitable for the enterprise scale.

Ganesan and Sengupta (2001) proposed an objects-to-business-components (O2BC) method for identifying business components based on an object-oriented specification by constructing an entity–event interaction CRUD matrix and clustering the matrix into blocks as components. However, the method lacks qualitative metrics and algorithms for identification.

Arch-int and Batanov (2003) proposed a methodology to develop business software components as basic building elements of industrial information systems by using a CRUD matrix between business activities and business objects. However, formal and accurate techniques to measure coupling and cohesion are lacking in the proposal.

2.3 Concluding remarks

We find that many identification approaches, especially top-down ones, have a low degree of automation and are usually executed manually. Most contemporary methods consider only part of the technical metrics and ignore the remainder during the identification process, which leads to incompleteness of component performance (Wang *et al.*, 2008). Moreover, many methods focus on software-level business components and are not practically applicable to the enterprise scale.

Our method aims at enterprise-level business component identification. We consider technical metrics on the whole, including cohesion, coupling, granularity, maintainability, and reusability. Moreover, we turn business component identification into a multi-objective optimization problem and use a novel meta-heuristic optimization algorithm to generate results.

3 Business component identification problem analysis

3.1 Problem delineation

Service-oriented architecture (SOA) technologies provide a flexible way for application integration, which motivates business architecture integration, i.e., the business component based business architecture integration method. The basic process of this method consists of two phases: business component identification and business component integration (Fu *et al.*, 2016). Business component identification is considered a difficult task. Its purpose is to cluster and partition business processes, organizations, and resources from the business architecture into a series of business components, which form the basic units for business architecture integration (Fu *et al.*, 2016).

In our study, enterprise-level business component identification is considerably different from component identification in software engineering. First, the scales of conception of these two business components are different. Enterprise-level business components concern enterprise-level business elements, including activities, organizations, and resources in business architecture, whereas software engineering is concerned mainly with activities or processes at micro-level abstractions, such as UML case models. Hence, current business component identification methods in software engineering cannot satisfy all technical metrics of enterprise-level business components, since these methods consider cohesion and coupling while ignoring other metrics (Wang *et al.*, 2008).

In this study, we consider the overall cohesion, coupling, granularity, maintainability, and reusability of the component during identification. Since these metrics are mutually restrained and cannot be optimized at the same time (Wang *et al.*, 2008), business component identification is a multi-objective optimization problem that involves minimizing coupling while maximizing coarse-grained granularity, cohesion, maintainability, and reusability. This multi-objective optimization problem can be expressed as

$$F(C)=\begin{cases}\max \mathrm{Coh}(C_{\mathrm{com}}),\\ \min \mathrm{Cou}(C_{\mathrm{com}}),\\ \max \mathrm{Mai}(C_{\mathrm{com}}),\\ \max \mathrm{Reu}(C_{\mathrm{com}}),\\ \max \mathrm{Gra}(C_{\mathrm{com}}),\end{cases} \tag{1}$$

where $\mathrm{Coh}(C_{\mathrm{com}})$, $\mathrm{Mai}(C_{\mathrm{com}})$, $\mathrm{Reu}(C_{\mathrm{com}})$, and $\mathrm{Gra}(C_{\mathrm{com}})$ represent cohesion, maintainability, reusability, and granularity, respectively, and $\mathrm{Cou}(C_{\mathrm{com}})$ represents coupling.

Since the formulation of the above metrics is dependent on the conception and formulation of business components, we present the definition and formulation of business components in the next subsection.

3.2 Business components

We use the definition of the business component from the component business model (IBM Business Consulting Services, 2006), since this conception of business components covers the main business architecture models in DoDAF. We first define a business component, and then analyze the mapping relationship between DoDAF models and business components according to the different dimensions of the latter to gather data from the former.

IBM developed the component business model. According to IBM Business Consulting Services (2006), a business component has five dimensions: business purpose, activities, resources, governance, and services.

1. Business purpose: value or capabilities that are provided by a business component to other components.

2. Activities: a set of simple and cohesive activities performed by a business component to achieve its business purpose.

3. Resources: tangible assets and human resources required by a business component to support their activities.

4. Governance: rules or a governance model on which each component is based in order to manage as an independent entity.

5. Business services: business components provide and receive business services so that they can be linked.

DoDAF 2.0 serves as the comprehensive architecture framework to facilitate the ability of Department of Defense (DoD) managers at all levels, so that they can make key decisions more effectively across the joint capability areas, mission, component, and program boundaries (DoD Architecture Framework Working Group, 2009). DoDAF is the most popular architecture framework used in the military domain around the world.

To gather the architectural data from DoDAF models into business components, we analyze the relationship between DoDAF models and business components. Based on the five dimensions of a business component, we can find similar concepts in DoDAF 2.0 (DoD Architecture Framework Working

Group, 2009). For example, the concept of business purpose is similar to capability in the capability view of DoDAF, and activity is similar to operational activity in the operational view. Thus, we build a detailed relationship between the five dimensions of business components and models in DoDAF 2.0 (Table 1). As long as we can build this relationship, we can package architectural data into business components, which can be used as integration units.

Table 1 Relationship between the five dimensions of a business component and DoDAF models

Dimension	Concept in DoDAF	Model in DoDAF
Purpose	Capability	CV-2, CV-5, CV-6
Activities	Operational activity	OV-5a, OV-5b
Resources	Resources, organization	OV-2, OV-3, OV-4
Governance	Process, rules	OV-6
Services	Service	SvcV-1, SvcV-5

4 Business component identification

The target of business component identification is to identify business components from business architecture data or models. In this section, the process of business component identification is described (Fig. 1). The inputs of our process are DoDAF models for business architecture and the outputs are the identified business components. The idea of our process is to convert these DoDAF models into a CRUD matrix and model the multi-objective optimization problem based on the matrix and the proposed metrics, and then to solve the problem by using the simulated annealing hybrid genetic algorithm. The details are discussed below.

4.1 Business component formalization

We formalize the business component for ease of calculation. Based on the concept of the business component and the relationship between business components and DoDAF models, the former can be represented as a four-tuple $\mathrm{BC} = (\mathrm{Ca}, A, O, (A, O))$, where Ca is capability in the capability taxonomy (CV-2), A is a set of activities or actions in the operational activity decomposition tree (OV-5a) and the operational activity model (OV-5b), O is the representation of organizations or performers in the operational resource flow description (OV-2) and the organization relationships chart (OV-4), and (A, O) is the relationship between activities and organizations.

According to the formalization of business components, the relationships among activities can be analyzed based on the OV-5a/b model, the relationship among organizations can be analyzed based on the OV-4 model, and the CRUD matrix can be created mainly based on OV-2, OV-3, OV-5b, and OV-6, since these models include relationships between activities and organizations.

4.2 CRUD matrix and technical metrics

We use a four-tuple formalization to represent the business components above, where (A, O) is the relationship between activities and organizations. In DoDAF 2.0, activities (within organizations) can be divided into two types: activity (produce) and activity (consume). The former type produces resource, whereas the latter consumes resource. The basic relationships between activities and organizations are shown in Fig. 2. Thus, we can define the relationship (A, O) as follows:

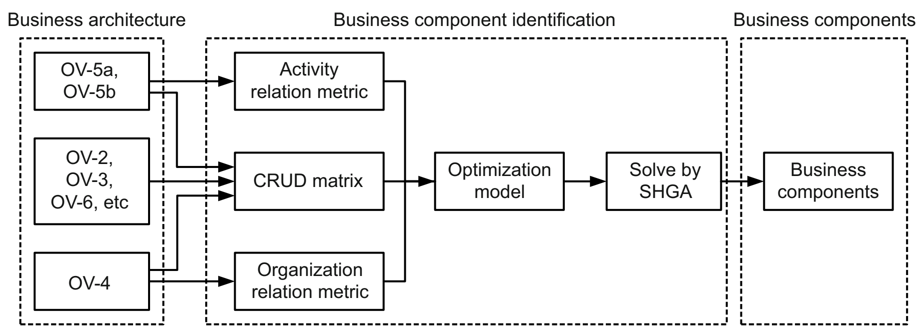

Fig. 1 Process of business component identification

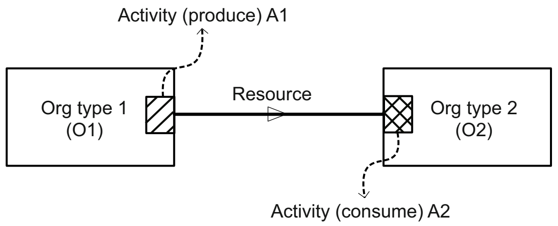

Fig. 2 An example of relationships between activities and organizations

Definition 1 (Relationship (A, O)) We use v to represent the relationship (A, O), $v \in \{C, R, U, D\}$, where C means that an activity is created by an organization, and represents the relationship between org type 1 (O1) and activity (produce) A1, R means that the activity is read by the organization, and represents the relationship between org type 2 (O2) and activity (produce) A2, U means that the activity has been updated by the organization, and D means that the activity has been deleted by the organization.

Through the simplification of computations, we adopt these substitutions: $C = 1$, $U = 0.75$, $D = 0.5$, and $R = 0.25$.

Definition 2 (CRUD matrix) A CRUD matrix can be defined as $\boldsymbol{M} = \{(A_i, O_j) \mid i = 1, 2, \cdots, N_{\mathrm{r}},\ j = 1, 2, \cdots, N_{\mathrm{c}}\}$, where A_i represents the ith activity and O_j the jth organization. N_{r} and N_{c} represent the number of activities and organizations in the model, respectively.

Definition 3 (Cluster of the CRUD matrix) A cluster of the CRUD matrix can be defined as $\boldsymbol{C}_k = \{(A_i, O_j) \mid i = l_1, l_1 + 1, \cdots, h_1,\ j = l_2, l_2 + 1, \cdots, h_2\}$, where $1 \leq l_1 < h_1 \leq N_{\mathrm{r}}$ and $1 \leq l_2 < h_2 \leq N_{\mathrm{c}}$.

We find that a cluster represents a capability in this sense. Indeed, we covert DoDAF models into a CRUD matrix. To automatically identify business components, we use four technical metrics (Jamshidi *et al.*, 2012) of the CRUD matrix: total semantic relationship (TSR), internal semantic dependency (ISD), external semantic dependency (ESD), and cluster semantic affinity (CSA). In our sense, the formulations of these metrics are shown below:

Metric 1 (Total semantic relationship, TSR) TSR is the average value of the total semantic relationships of the clusters. For cluster $\boldsymbol{C}_k$, $\mathrm{TSR}(\boldsymbol{C}_k)$ is its total semantic relationship:

$$\mathrm{TSR}(\boldsymbol{C}_k) = \sum_{j=l_2}^{h_2} \sum_{i=l_1}^{h_1} v_{ij}. \tag{2}$$

For cluster set C_{set}, $\mathrm{TSR}(C_{\mathrm{set}})$ is the average value:

$$\mathrm{TSR}(C_{\mathrm{set}}) = \sum_{k=1}^{|C_{\mathrm{set}}|} \mathrm{TSR}(\boldsymbol{C}_k) \Big/ |C_{\mathrm{set}}|, \tag{3}$$

where $|C_{\mathrm{set}}|$ is the number of clusters in C_{set}. Fig. 3 illustrates the TSR calculation process using a simple example.

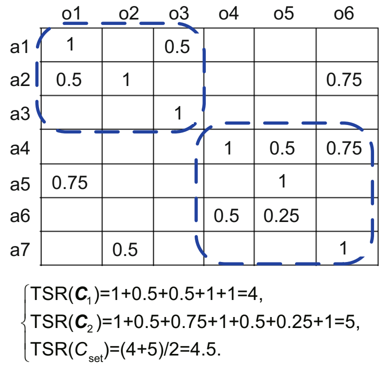

Fig. 3 An example of the total semantic relationship calculation process

Metric 2 (Internal semantic dependency, ISD) ISD is the metric that measures the strength of the semantic dependency between clusters. $\mathrm{ISD}(\boldsymbol{C}_k)$ is the ISD of cluster $\boldsymbol{C}_k$, and $\mathrm{ISD}(C_{\mathrm{set}})$ is the average ISD of cluster set C_{set}:

$$\mathrm{ISD}(\boldsymbol{C}_k) = \frac{\sum\limits_{q=l_2}^{h_2} \sum\limits_{p=l_1}^{h_1} \sum\limits_{j=l_2}^{h_2} \sum\limits_{i=l_1}^{h_1} \min\limits_{i=p \| j=q} (v_{pq}, v_{ij})}{\sum\limits_{q=l_2}^{h_2} \sum\limits_{p=l_1}^{h_1} \sum\limits_{j=l_2}^{h_2} \sum\limits_{i=l_1}^{h_1} \min(v_{pq}, v_{ij})}, \tag{4}$$

$$\mathrm{ISD}(C_{\mathrm{set}}) = \frac{\sum\limits_{k=1}^{|C_{\mathrm{set}}|} \mathrm{ISD}(\boldsymbol{C}_k) \cdot \mathrm{TSR}(\boldsymbol{C}_k)}{\sum\limits_{j=1}^{N_{\mathrm{c}}} \sum\limits_{i=1}^{N_{\mathrm{r}}} v_{ij}}. \tag{5}$$

Fig. 4 illustrates the ISD calculation process using a simple example.

Metric 3 (External semantic dependency, ESD) ESD is a measure of the relative interdependence among semantic relationships within the cluster and those outside the cluster. $\mathrm{ESD}(\boldsymbol{C}_k)$ is the ESD of cluster $\boldsymbol{C}_k$ and $\mathrm{ESD}(C_{\mathrm{set}})$ is the summation ESD of cluster set C_{set}:

$$\mathrm{ESD}(\boldsymbol{C}_k) = \sum_{j=1}^{N_{\mathrm{c}}} \sum_{i=l_1}^{h_1} v_{ij} - \mathrm{TSR}(\boldsymbol{C}_k), \tag{6}$$

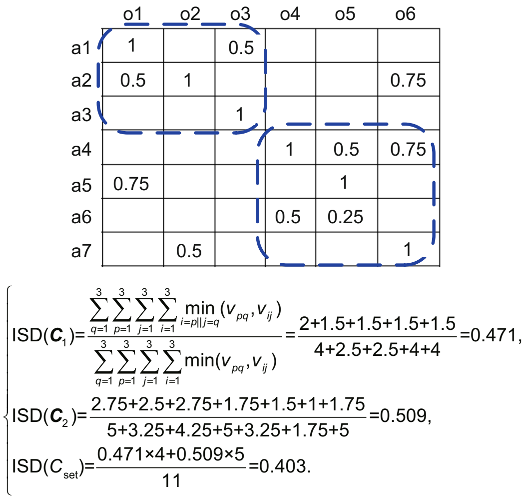

Fig. 4 An example of the internal semantic dependency calculation process

$$\mathrm{ESD}(C_{\rm set}) = \sum_{k=1}^{|C_{\rm set}|} \mathrm{ESD}(\boldsymbol{C}_k). \tag{7}$$

Fig. 5 illustrates the ESD calculation process using a simple example.

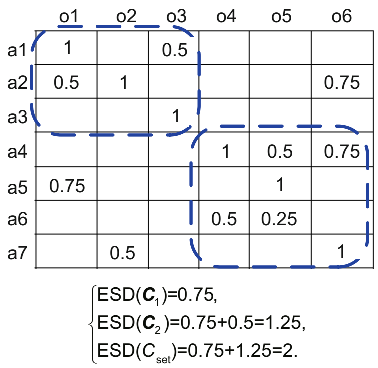

Fig. 5 An example of the external semantic dependency calculation process

Metric 4 (Cluster semantic affinity, CSA) CSA is a measure of the density of the semantic relationships in the cluster. $\mathrm{CSA}(\boldsymbol{C}_k)$ is the CSA of cluster $\boldsymbol{C}_k$ and $\mathrm{CSA}(C_{\rm set})$ is the average CSA of cluster set $C_{\rm set}$:

$$\mathrm{CSA}(\boldsymbol{C}_k) = \sum_{j=l_2}^{h_2}\sum_{i=l_1}^{h_1} \frac{v_{ij}}{(h_2-l_2+1)(h_1-l_1+1)}, \tag{8}$$

$$\mathrm{CSA}(C_{\rm set}) = \frac{\mathrm{TSR}}{\sum\limits_{k=1}^{|C_{\rm set}|}(h_2-l_2+1)(h_1-l_1+1)}. \tag{9}$$

Fig. 6 illustrates the CSA calculation process using a simple example.

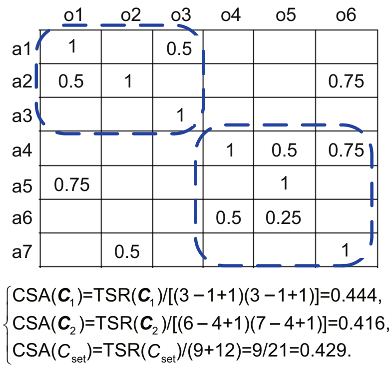

Fig. 6 An example of the cluster semantic affinity calculation process

Since the CRUD matrix does not consider the relationships of activities and those of organizations, we propose two metrics for such identification: organization entity affinity (OEA) and activity entity affinity (AEA).

Metric 5 (Organization entity affinity, OEA) OEA is defined as a measure of the density of semantic dependency among organizational entities in the cluster. The semantic dependency of organizational entities can be described by a relational tree graph or undirected graph, which is usually a description model of OV-4 or OV-2. We use the relative distance between organizational entities in the relational graph as a representative of their semantic dependency. The relational graph can be defined as $T=\{(O_i)\mid i=1,2,\cdots,N_{\rm org}\}$, where O_i is the ith organization entity and $N_{\rm org}$ is the number of organization entities. We set o_{ij} as the semantic dependency between O_i and O_j, and $O_i, O_m, \ldots, O_n, O_j$ as the shortest path between O_i and O_j, where the number of $O_m, \ldots, O_n$ is set to l_{ij}. Then,

$$o_{ij} = \begin{cases} \dfrac{1}{l_{ij}+1}, & i \neq j, \\ 0, & i = j. \end{cases} \tag{10}$$

$\mathrm{OEA}(\boldsymbol{C}_k)$ is the OEA of cluster $\boldsymbol{C}_k$:

$$\mathrm{OEA}(\boldsymbol{C}_k) = \sum_{O_i, O_j \in \boldsymbol{C}_k} o_{ij} \Big/ N_k^2, \tag{11}$$

where N_k is the number of organizational entities in cluster $\boldsymbol{C}_k$.

$\mathrm{OEA}(C_{\rm set})$ is the average OEA of cluster set $C_{\rm set}$:

$$\mathrm{OEA}(C_{\rm set}) = \sum_{k=1}^{|C_{\rm set}|} \mathrm{OEA}(\boldsymbol{C}_k) \Big/ |C_{\rm set}|. \tag{12}$$

Fig. 7 shows the OEA calculation process using a simple example.

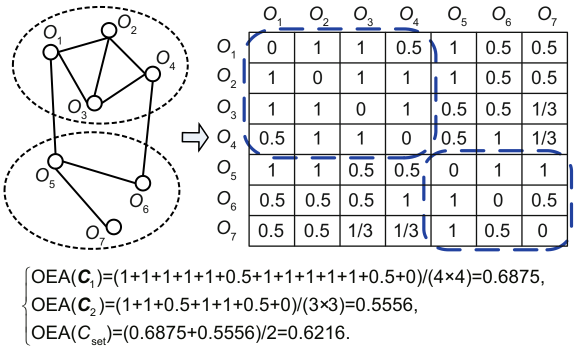

Fig. 7 The organization entity affinity calculation process

Metric 6 (Activity entity affinity, AEA) Similar to OEA, AEA is defined as a measure of the density of semantic dependency among activity entities. The dependence of activity entities can also be described by a relational tree graph or undirected graph, and we set $G = \{(A_i) \mid i = 1, 2, \cdots, N_{\text{act}}\}$ as an activity relational graph, where A_i is the ith activity entity and N_{act} is the number of activity entities. We also use the number of activity entities along the shortest path between two activity entities as semantic dependency a_{ij}:

$$a_{ij} = \begin{cases} \dfrac{1}{l_{ij}+1}, & i \neq j, \\ 0, & i = j. \end{cases} \tag{13}$$

AEA($\boldsymbol{C}_k$) is the AEA of cluster $\boldsymbol{C}_k$ and AEA(C_{set}) is the average AEA of cluster set C_{set}:

$$\text{AEA}(\boldsymbol{C}_k) = \sum_{A_i, A_j \in \boldsymbol{C}_k} a_{ij} \Big/ N_k^2, \tag{14}$$

$$\text{AEA}(C_{\text{set}}) = \sum_{k=1}^{|C_{\text{set}}|} \text{AEA}(\boldsymbol{C}_k) \Big/ |C_{\text{set}}|. \tag{15}$$

4.3 Mathematical formulation

In Section 3, we point out that business component identification is a multi-objective optimization problem that involves maximizing cohesion, maintainability, reusability, and granularity, and minimizing coupling. In this subsection, we analyze the relationships between these five design principles and the six metrics defined in the previous subsection. We then use the six metrics as factors in the multi-objective optimization problem.

During identification, we need high cohesion, low coupling, high maintainability, high reusability, and high granularity. According to the meaning of cohesion, high cohesion requires high ISD and CSA, following which cohesion is positively correlated with ISD and CSA. Since there is a negative correlation between ISD and ESD, and ISD is positively correlated with TSR based on Eq. (1), cohesion is positively correlated with TSR and negatively with ESD. Moreover, high cohesion requires high OEA and AEA, in which case cohesion is positively correlated with OEA and AEA. With a similar analysis of coupling, maintainability, reusability, and granularity, we can obtain the correlations between the design principles and the six metrics (Table 2).

Table 2 Relationships between design principles and metrics*

Design principle	TSR	ISD	ESD	CSA	OEA	AEA
Cohesion	+	+	−	+	+	+
Coupling	−	−	+	−	−	−
Maintainability	+	+	−	+	+	+
Reusability	+	+	−	+	+	+
Granularity	+	+	−	+	+	+

* '+' represents positive correlation and '−' negative correlation

Thus, we can use the six metrics to replace the design principles in multi-objective optimization by maximizing TSR, ISD, CSA, OEA, and AEA, and minimizing ESD. Since these design principles cannot be optimal at the same time (Wang *et al.*, 2008), neither can the metrics. We make a trade-off between these objectives, and convert the multi-objective optimization problem to a single-objective problem.

To unify the units of the six metrics, we adjust metric values that are not between 0 and 1 to real numbers between 0 and 1 by multiplying them with parameter α. Therefore, we formulate the objective function as follows:

$$\begin{aligned} F(C_{\text{set}}) = {} & \alpha\text{TSR}(C_{\text{set}}) \cdot \text{ISD}(C_{\text{set}}) \cdot (1 - \alpha\text{ESD}(C_{\text{set}})) \\ & \cdot \text{CSA}(C_{\text{set}}) \cdot \text{OEA}(C_{\text{set}}) \cdot \text{AEA}(C_{\text{set}}), \end{aligned} \tag{16}$$

where

$$\alpha = 1 \Big/ \sum_{j=0}^{N_{\text{c}}} \sum_{i=0}^{N_{\text{r}}} v_{ij}. \tag{17}$$

4.4 Solving the optimization problem by SHGA

Since the matrix-clustering problem is an NP-complete problem (Bui and Moon, 1996) and its solution time is non-polynomial, we use a meta-heuristic method called the 'simulated annealing hybrid genetic algorithm (SHGA)' to generate optimal results. SHGA is a method based on simulated annealing (SA) and a genetic algorithm (GA). SA (Kirkpatrick *et al.*, 1983) is a generic probabilistic meta-heuristic algorithm with a certain global search ability that can converge to the global optimal value in theory. However, its global search ability is limited, and it is not very efficient. GA (Goldberg, 1989) is an adaptive global probabilistic search algorithm widely used in many fields due to its simple operation and fast convergence. However, its local search ability is insufficient and easily leads to premature convergence, which affects the quality of the final results. However, SHGA combines the advantages of both SA and GA and compensates for the shortcomings of these two algorithms. Therefore, this approach yields the advantage of a high convergence speed and near-optimal results in the global optima.

A flowchart of SHGA is shown in Fig. 8.

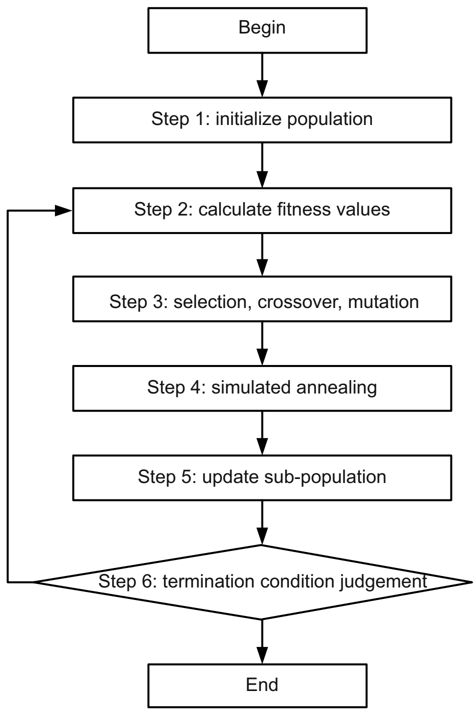

Fig. 8 Flowchart of the simulated annealing hybrid genetic algorithm

Step 1: initialize population

This step contains two sub-steps, chromosome encoding and chromosome initialization.

The common chromosome encoding methods include binary encoding and real number encoding. Since binary encoding affects the algorithm's efficiency and is not conducive to the realization of the hybrid algorithm, we use real number encoding in this study. We set the sum of N_r and N_c of the CRUD matrix as the number of genes in a chromosome, where each gene corresponds to a row or a column in the CRUD matrix (Fig. 9). $|C_\mathrm{set}|$ is initialized by generating a random number between 1 and N_r as the number of clusters in cluster set C_set. Thus, each gene can be encoded by an integer between 1 and $|C_\mathrm{set}|$.

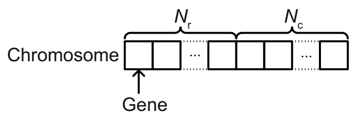

Fig. 9 Chromosome encoding

We use random initialization for chromosome initialization by assigning each gene a number (between 1 and $|C_\mathrm{set}|$) with a uniform probability. If a row and a column have the same number, for example, two, they are in the same cluster, and the cluster number is named two. Hence, we can generate the initial clusters. However, there are inconsistencies whereby a row with a cluster number does not belong to the column clusters, or a column with a cluster number does not belong to the row clusters. We thus need to re-assign such rows and columns new cluster numbers to eliminate inconsistency. The relevant algorithm is shown in Algorithm 1.

Step 2: calculate fitness values

In this step, we need to calculate the fitness values of the initialized population. The fitness function is the objective function (16). Before calculating the fitness values, we need to re-rank the CRUD matrix according to the number of clusters to ensure that they are along the diagonal line of the matrix (Fig. 10).

Step 3: selection, crossover, and mutation

1. Selection operator

We use the roulette wheel-selection method in this algorithm. Roulette wheel selection is based on the fitness rate and helps choose parent chromosomes. A chromosome's selection probability

Algorithm 1 Eliminating inconsistency

Input: CRUD matrix $\boldsymbol{M}$ and unhandled population **Chrom**.

Output: handled population **Chrom**.

```
size(M) = [h, l], size(Chrom) = [popsize, h + l];
Mark N(i, j) = Chrom(i, j);
for (i = 1; i ≤ popsize; i + +) do
    for (p = 1; p ≤ h; p + +) do
        if N(i, p) ∉ B, B = {N(i, q)|q ∈ [1 : l]} then
            N(i, p) = b, b ∈ B;
            Chrom(i, p) = N(i, p);
        end if
    end for
    for (q = 1; q ≤ l; q + +) do
        if N(i, q) ∉ A, A = {N(i, p)|p ∈ [1 : h]} then
            N(i, q) = a, a ∈ A;
            Chrom(i, q) = N(i, q);
        end if
    end for
end for
Return Chrom.
```

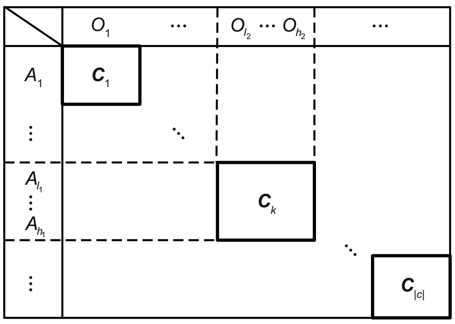

Fig. 10 Clusters on the diagonal line of the matrix

formula is as follows:

$$P_i = \frac{F_i}{\sum\limits_{j=1}^{\text{popsize}} F_j}, \; i = 1, 2, \cdots, \text{popsize}, \tag{18}$$

where P_i is the ith chromosome's selection probability and F_i is the ith chromosome's fitness value.

2. Crossover operator

A crossover operator with a single point is used in this algorithm. The two chromosomes chosen for crossover are correlated with fitness values, and each gene of the chromosome has an equal probability of being chosen as the cross-point. To avoid the dependence of the crossover probability on the initial value, the self-adaptive crossover probability is adopted:

$$P_{\text{cro}} = \begin{cases} P_{\text{cro1}} - \frac{(P_{\text{cro1}} - P_{\text{cro2}})(F' - F_{\text{avg}})}{F_{\max} - F_{\text{avg}}}, & F' \geq F_{\text{avg}}, \\ P_{\text{cro1}}, & F' < F_{\text{avg}}, \end{cases} \tag{19}$$

where F' is the greater fitness of the two crossover chromosomes, $F_{\max}$ and F_{avg} are the maximum fitness and average fitness in the given population, respectively, and P_{cro1} and P_{cro2} are the initial values of the crossover probability.

3. Mutation operator

The discrete mutation operator is applied in this algorithm. Each gene in the chromosomes can be mutated into a new number between 1 and $|C_{\text{set}}|$ with a certain probability. To avoid the dependence of the probability of mutation on the initial value, the self-adaptive mutation probability is adopted:

$$P_{\text{mut}} = \begin{cases} P_{\text{mut1}} - \frac{(P_{\text{mut1}} - P_{\text{mut2}})(F_{\text{avg}} - F)}{F_{\max} - F_{\text{avg}}}, & F \geq F_{\text{avg}}, \\ P_{\text{mut1}}, & F < F_{\text{avg}}, \end{cases} \tag{20}$$

where F is the fitness of the mutated chromosome, $F_{\max}$ and F_{avg} are the maximum fitness and average fitness in the given population, respectively, and P_{mut1} and P_{mut2} are the initial values of the mutation probability.

Step 4: simulated annealing

We set the number of iterations g of GA as the annealing time of SA, T_n the given temperature, and T_0 the initial temperature. When the algorithm executes the simulated annealing operation the first time, $T_n = T_0$. Then, T_n is as follows:

$$T_n = T_{n-1}/g. \tag{21}$$

We assume $\boldsymbol{X}^i = (X_1^i, X_2^i, \cdots, X_{\text{popsize}}^i)$ is the ith individual in the population. To generate a new individual, we create a perturbation function $\boldsymbol{Z}^i = (Z_1^i, Z_2^i, \cdots, Z_{\text{popsize}}^i)$ as follows:

$$Z_j^i = \text{Round}(r_1) = \begin{cases} 1, & 0.5 \leq r_1 \leq 1, \\ 0, & -0.5 < r_1 < 0.5, \\ -1, & -1 \leq r_1 \leq -0.5, \end{cases} \tag{22}$$

where r_1 is a random number of the uniform distribution in $[-1, 1]$.

We can thus obtain the new individual $\boldsymbol{Y}^i = |\boldsymbol{X}^i + \boldsymbol{Z}^i|$. Assuming that its fitness value is $F(\boldsymbol{Y}^i)$ and that P_{sa}^i is the acceptance probability, we have

$$P_{\text{sa}}^i = \min\left\{1, \exp\left(\frac{F(\boldsymbol{Y}^i) - F(\boldsymbol{X}^i)}{T}\right)\right\}. \tag{23}$$

Assume that r_2 is a random number of the uniform distribution in $(0, 1)$. If $r_2 \leq P_{\text{sa}}^i$, we accept $\boldsymbol{Y}^i$ as the new individual instead of $\boldsymbol{X}^i$, that is,

$\boldsymbol{X}^i = \boldsymbol{Y}^i$. For the others in the population, we can apply the previous operation to generate new individuals.

Thus, the simulated annealing operation can be summarized in Algorithm 2.

Algorithm 2 Simulated annealing

Input: current temperature T, annealing time g, and current population **Chrom**.

Output: new population **NewCh** and new temperature T.

```
Set input parameters and annealing time g;
Mark Chrom = [X^1, X^2, ..., X^popsize];
for (i = 1; i ≤ popsize; i++) do
    Calculate Z^i, where Z^i_j = Round(r_1);
    Calculate Y^i = |X^i + Z^i|;
    Calculate fitness value F(Y^i);
    Calculate P^i_sa, Random r_2;
    if r_2 ≤ P^i_sa then
        X^i = Y^i;
    else
        X^i = X^i;
    end if
end for
Calculate temperature T = T/g;
Return NewCh = [X^1, X^2, ..., X^popsize], T.
```

Step 5: update sub-population

Once we have generated the new population, we need to update the sub-population. We reinsert the new population into the parent population and choose individuals based on fitness values. Thus, individuals with greater fitness values will remain in the sub-population.

Step 6: termination condition judgment

The termination condition of the algorithm is the maximum number of iterations g_{num}. If the generation g_{n-1} $(n = 1, 2, \cdots)$ is greater than or equal to g_{num}, the process is stopped, and the chromosome with the best fitness is obtained. Otherwise, $g_n = g_{n-1} + 1$, and go to step 2.

Based on the above analysis, SHGA can be summarized in Algorithm 3.

Algorithm 3 Simulated annealing hybrid genetic algorithm

Input: initial iteration number g, initial temperature T_0, maximum iteration number g_{num}, population size popsize, CRUD matrix $\boldsymbol{M}$, and initial number of clusters $|C_{\text{set}}|$.

Output: chromosome with best fitness **BestCh**.

```
Set input parameters;
Load CRUD matrix M;
Initialize population;
Eliminate inconsistencies by Algorithm 1;
while g < g_num do
    Calculate fitness value F;
    Selection operator;
    Crossover operator;
    Mutation operator;
    Simulated annealing operation by Algorithm 2;
    Eliminate inconsistencies by Algorithm 1;
    Calculate fitness value F;
    Update sub-population;
    g = g + 1;
end while
Return BestCh.
```

5 Evaluation and discussions

We have verified our method by applying it to two case studies. To verify the applicability of the method, we applied it to a classic DoDAF example, search and rescue (SAR) (Maritime and Coastguard Agency, 2008). Then, to evaluate the efficiency of SHGA in our method, we compared it with GA and SA in a sales solution development case (Jamshidi *et al.*, 2012). Following this, we make a discussion about our method.

5.1 Applicability of the method

The applicability of our method was tested by using it to identify the business components of the maritime SAR case, which is a free, publicly available example with a sufficient amount of data to support the relevant DoDAF 2.0 models. Data for the maritime SAR can be found on the website of the chief information officer of the U.S. Department of Defense (DoDAF Development Team, 2010).

SAR is the activity of locating and recovering missing people or people in distress and delivering them to a place of safety. We can regard SAR as a kind of military operation, since it accords with the observation orientation decision action (OODA) process. It is important for integrated SAR services both on land and in water. For this, we first need to identify the business components of both maritime SAR and land SAR. Thus, our case study focused on identifying the business components of maritime SAR.

To obtain the CRUD matrix of the maritime SAR, the relationships (A, O) between activities and organizations needed to be analyzed. In maritime SAR, the organizations in OV-2 and OV-4 included person in distress (o1), monitoring org (o2), tactical C2 org (o3), SAR asset controller (o4), searcher (o5), boat driver (o6), swimmer (o7), and helicopter pilot (o8). There were 10 activities: send distress signal (a1), receive distress signal (a2), send warning order (a3), process warning order (a4), send task (a5), find victim (a6), monitor health (a7), provide medical assistance (a8), recover victim (a9), and transit to SAR operation (a10). By analyzing the relationship type between activities and organizations in OV-2 and OV-5b, the relationships (A, O) can be deduced. We obtain the CRUD matrix of SAR as shown in Table 3.

Then, by analyzing the organizational interactions in OV-4 and OV-2, the semantic dependency matrix for organizational entities can be determined

	o1	o2	o3	o4	o5	o6	o7	o8
o1	0	1/3	1/3	1/3	1/3	1/3	1/3	1/3
o2	1/3	0	1/2	1/2	1/4	1/4	1/4	1/4
o3	1/3	1/2	0	1/2	1/4	1/4	1/4	1/4
o4	1/3	1/2	1/2	0	1/4	1/4	1/4	1/4
o5	1/3	1/4	1/4	1/4	0	1/2	1/4	1/4
o6	1/3	1/4	1/4	1/4	1/2	0	1/4	1/4
o7	1/3	1/4	1/4	1/4	1/4	1/4	0	1/2
o8	1/3	1/4	1/4	1/4	1/4	1/4	1/2	0

Fig. 11 Semantic dependency matrix of organizational entities

(Fig. 11). With a similar analysis to that for OV-5a and OV-5b, the semantic dependency matrix among activity entities is shown in Fig. 12.

The parameters of SHGA in our case study are shown in Table 4. By following Algorithm 3, the generated clustered CRUD matrix is shown in Table 5, and Table 6 lists the identified business components and their related activities and organizations.

By reviewing the identified business components, we can verify the outcome from two aspects. First, from the perspective of semantics, the

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
a1	0	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3
a2	1/3	0	1/2	1/4	1/4	1/2	1/2	1/2	1/2	1/2
a3	1/3	1/2	0	1/4	1/4	1/2	1/2	1/4	1/4	1/4
a4	1/3	1/4	1/4	0	1/2	1/4	1/4	1/4	1/4	1/4
a5	1/3	1/4	1/4	1/2	0	1/4	1/4	1/4	1/4	1/4
a6	1/3	1/2	1/2	1/4	1/4	0	1/2	1/4	1/4	1/4
a7	1/3	1/2	1/2	1/4	1/4	1/2	0	1/4	1/4	1/4
a8	1/3	1/2	1/4	1/4	1/4	1/4	1/4	0	1/2	1/2
a9	1/3	1/2	1/4	1/4	1/4	1/4	1/4	1/2	0	1/2
a10	1/3	1/2	1/4	1/4	1/4	1/4	1/4	1/2	1/2	0

Fig. 12 Semantic dependency matrix of activity entities

Table 4 Parameters of SHGA

Parameter	Value
Population size	popsize = 10
Maximum iteration number	$g_{\text{num}} = 5000$
Initial temperature	$T_0 = 1$
Initial iteration number	$g_0 = 1$
Crossover probability	$P_{\text{cro1}} = 0.9$, $P_{\text{cro2}} = 0.6$
Mutation probability	$P_{\text{mut1}} = 0.1$, $P_{\text{mut2}} = 0.001$

Table 3 CRUD matrix of SAR*

	Person in distress (o1)	Monitoring org (o2)	Tactical C2 org (o3)	SAR asset controller (o4)	Searcher (o5)	Boat driver (o6)	Swimmer (o7)	Helicopter pilot (o8)
Send distress signal (a1)	C							
Receive distress signal (a2)		R			R			
Send warning order (a3)		C	R		U			
Process warning order (a4)			C	R				
Send task (a5)				C	R		R	
Find victim (a6)					C	U		
Monitor health (a7)					U		R	
Provide medical assistance (a8)							C	R
Recover victim (a9)							U	R
Transit to SAR operation (a10)			R					C

* C: create; R: read; U: update

Table 5 Clustered CRUD matrix of SAR*

	Person in distress (o1)	Tactical C2 org (o3)	SAR asset controller (o4)	Monitoring org (o2)	Searcher (o5)	Boat driver (o6)	Swimmer (o7)	Helicopter pilot (o8)
Send distress signal (a1)	C							
Process warning order (a4)		C	R					
Send task (a5)			C		R		R	
Receive distress signal (a2)				R	R			
Send warning order (a3)		R		C	U			
Find victim (a6)					C	U		
Monitor health (a7)					U		R	
Provide medical assistance (a8)							C	R
Recover victim (a9)							U	R
Transit to SAR operation (a10)		R						C

* C: create; R: read; U: update

Table 6 Identified business components of maritime SAR

Business component	Activities	Organizations
C1	a1, a4, a5	o1, o3, o4
C2	a2, a3, a6, a7	o2, o5, o6
C3	a8, a9, a10	o7, o8

identified business components were reasonably good in our case. For instance, send distress signal (a1), process warning order (a4), send task (a5), person in distress (o1), tactical C2 org (o3), and SAR asset controller (o4) were grouped correctly into an enterprise-level business component boundary, which was related to SAR C2. Receive distress signal (a2), send warning order (a3), find victim (a6), and monitor health (a7) were the main activities that formed part of the search operation according to OV-5a. Second, referring to the CV-3 model, the identified business components were appropriate in terms of the granularity of capabilities in CV-3. For example, the identified business components C1, C2, and C3 corresponded with the capabilities SAR C2, search, and recovery in CV-4.

5.2 Efficiency of the method

To verify the efficiency of SHGA in our method, we compared our method with simulated annealing in Jamshidi *et al.* (2012), where SA was used to identify services from the CRUD matrix. We also chose a basic genetic algorithm to compare with our method. We used the sale scenario example in Jamshidi *et al.* (2012). The sale scenario showed that the company focused on the processes of receiving, delivering, processing, and shipping orders daily from customers. The detailed business scenario of the case study has been described by Jamshidi *et al.* (2012). Table 7 shows the CRUD matrix of the sale scenario.

Since Jamshidi *et al.* (2012) did not consider relation metrics from graphs during identification, we ignored metrics 5 and 6, and adjusted the objective function to the one below. The parameters of the three algorithms were set to be the same as those in Table 4. Following the execution of the three algorithms 10 times, the optimal results of the three algorithms were identical (Table 8).

$$F(C_{\text{set}}) = \alpha\text{TSR}(C)\cdot\text{ISD}(C_{\text{set}})\cdot\text{CSA}(C_{\text{set}})\cdot(1-\alpha\text{ESD}(C_{\text{set}})). \quad (24)$$

To compare the efficiency of SHGA, SA, and GA in executing search operations, we ran each algorithm 10 times and calculated the average values of the objective function in each iteration. The results are shown in Fig. 13, where the x- and y-axis show the number of iterations and the average value of the objective function, respectively.

The results show that (1) both GA and SHGA recorded fast search speeds at first (before 1000 iterations), but GA converged prematurely, and might

Table 7 CRUD matrix of the sale scenario (Jamshidi *et al.*, 2012)*

	Customer (1)	Credit (2)	Order (3)	Invoice (4)	Draft (5)	Shipping schedule (6)	Inventory (7)	Account receivable note (8)	Warehouse voucher (9)	Discounts (10)
Add a customer (1)	C	C								
Receive order (2)	R		C							
Check credit (3)	R	R	R							
Calculate discounts (4)			R							R
Check inventory (5)			R				R			
Calculate price (6)			R							R
Issue invoice (7)	R	R	R	C						
Issue draft (8)				R	C	R				
Schedule shipping (9)				R		C				
Add an account receivable note (10)	R	U		R				C		
Add a warehouse voucher (11)	R			R			U		C	
Add an item (12)							C			
Add discounts (13)										C

* C: create; R: read; U: update

Table 8 Identified components of the sale scenario

Cluster	Row number	Column number
C1	1, 3, 10	1, 2, 8
C2	2, 4, 6, 13	3, 10
C3	7, 8, 9	4, 5, 6
C4	5, 11, 12	7, 9

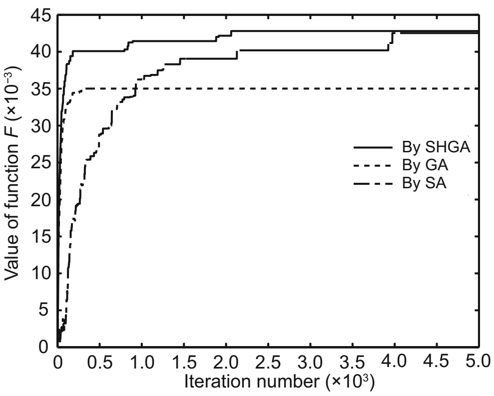

Fig. 13 Comparative results of SA, GA, and SHGA

have failed to find optimal results several times according to its average optimal value (approximately 0.035), and that (2) both SA and SHGA finally found optimal results, but SA was much slower, with convergence after nearly 4000 iterations. On the contrary, it took approximately 2200 iterations to obtain optimal results using SHGA. Thus, SHGA is more suitable and efficient for solving CRUD-based clustering problems.

5.3 Discussions

Our research is aimed to identify enterprise-level business components, which are regarded as the basic units of business architecture integration. We considered cohesion, coupling, granularity, maintainability, and reusability during the identification process and provided an efficient solution. Our research here can provide the technical means for architects or business analysts to identify the core business components or capabilities of both enterprises and militaries.

We conducted a comparison between the representative methods discussed in related work and ours in terms of four criteria: applicability, degree of automation, technical metrics, and identification technique. The applicability criterion shows that the scales of application of the method varied from the enterprise level to the software level and micro-levels. Degree of automation measures the automation of the identification process, from no algorithm to algorithm support or tool support. Technical metrics and the identification technique are important criteria for business component identification. We chose one representative method from each classification (Lee *et al.*, 1999; Yang *et al.*, 2002; Cai *et al.*, 2011) for comparison with our approach. The results are shown in Table 9.

Although both Yang *et al.* (2002)'s approach and our approach can be used for enterprise-level

Table 9 Comparison of business component identification approaches

Criterion	Top-down approach (Yang *et al.*, 2002)	Cohesion coupling-based (Cai *et al.*, 2011)	CRUD-based (Lee *et al.*, 1999)	Our approach
Applicability	Enterprise-level	Software-level	Micro-level	Enterprise-level
Degree of automation	No algorithm support	Algorithm support	Algorithm support	Algorithm support
Technical metrics	No measurable metric	Cohesion and coupling	Cohesion and coupling	Cohesion, coupling, granularity, maintainability, and reusability
Identification technique	CIMOSA function view	Fuzzy formal concept graph clustering	CRUD matrix clustering	Combining graph metrics and matrix clustering

business component identification, Yang *et al.* (2002) have provided no algorithm support or measurable metrics during identification. The approaches proposed by Lee *et al.* (1999) and Cai *et al.* (2011) are used mainly to identify micro- or software-level components from micro-business models, e.g., UML case models, which are considerably different from enterprise business models, e.g., DoDAF models. Furthermore, Lee *et al.* (1999) and Cai *et al.* (2011) used mainly cohesion and coupling as technical metrics during identification. Our approach, however, can satisfy the main capabilities of these important characteristics during enterprise-level business component identification.

Considering security and confidentiality in the military domain, we chose search and rescue (SAR) for case study. However, SAR can be regarded as a typical military operation since its process matches the features of a military operation. Moreover, human-related factors are important aspects of business component identification and design, whereas our research focused on technical factors. Despite these concerns, we believe we have made substantial progress toward our goal. Although our experiments are limited in scope, they did show the applicability and efficiency of identifying enterprise-level business components under the proposed framework.

6 Conclusions

In this study, we have proposed a novel approach to enterprise-level business component identification, one of the primary research problems in component-based business architecture integration. In our approach, we first convert DoDAF models from business architecture to a CRUD matrix, and use six technical metrics as criteria during the identification process. We then convert the business component identification problem into a multi-objective optimization problem and solve it using a meta-heuristic algorithm, called the 'simulated annealing hybrid genetic algorithm'. Finally, the applicability and efficiency of our approach have been evaluated through an enterprise-level case study and several experiments. We have also compared our approach with some other typical identification methods. The results show that our approach is more practical and efficient for enterprise-level business component identification.

There are several directions for future work in the area. First, the difference in technical metrics needs to be considered during identification by discussing the effects of different weights of technical metrics. Moreover, an appropriate method to automatically generate the CRUD matrix and the semantic dependency matrix will be part of our future work. Further, developing a prototype tool for automated enterprise-level business component identification is also an aim.

References

Arch-int, S., Batanov, D.N., 2003. Development of industrial information systems on the web using business components. *Comput. Ind.*, **50**(2):231-250. https://doi.org/10.1016/s0166-3615(02)00122-7

Bui, T.N., Moon, B.R., 1996. Genetic algorithm and graph partitioning. *IEEE Trans. Comput.*, **45**(7):841-855. https://doi.org/10.1109/12.508322

Cai, Z.G., Yang, X.H., Wang, X.Y., *et al.*, 2011. A fuzzy formal concept analysis based approach for business component identification. *J. Zhejiang Univ.-Sci. C (Comput. & Electron.)*, **12**(9):707-720. https://doi.org/10.1631/jzus.c1000337

DoDAF Development Team, 2010. UPDM Search and Rescue (SAR) Diagrams and DoDAF 2 Markups. http://dodcio.defense.gov/Library/DoD-Architecture-Framework/dodaf20journal/

DoD Architecture Framework Working Group, 2009. DoD Architecture Framework Version 2.0, Volume 1: Introduction, Overview, and Concepts, Manager's Guide. Department of Defense, USA. http://www.doc88.com/p-3774254078745.html

Fu, J., Luo, A.M., Luo, X.S., *et al.*, 2016. A component business model-based approach for business architecture integration of C4ISR system. Int. Conf. on Information Science and Control Engineering, p.17-21. https://doi.org/10.1109/icisce.2016.15

Ganesan, R., Sengupta, S., 2001. O2BC: a technique for the design of component-based applications. 39th Int. Conf. and Exhibition on Technology of Object-Oriented Languages and Systems, p.46-55. https://doi.org/10.1109/TOOLS.2001.941658

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston, USA.

IBM Business Consulting Services, 2006. Component Business Models Making Specialization Real. http://ptgmedia.pearsoncmg.com/imprint_downloads/informit/pdfs/cbm.pdf

Jain, H., Chalimeda, N., Ivaturi, N., *et al.*, 2001. Business component identification—a formal approach. IEEE Int. Conf. on Enterprise Distributed Object Computing, p.183-187. https://doi.org/10.1109/edoc.2001.950437

Jamshidi, P., Mansour, S., Sedighiani, K., *et al.*, 2012. An Automated Service Identification Method. Technical Report No. 1, TR-ASER-2012-01, Automated Software Engineering Research Group, Shahid Beheshti University, Iran.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *Science*, **220**(4598):671-680. https://doi.org/10.1126/science.220.4598.671

Lee, J.K., Jung, S.J., Kim, S.D., *et al.*, 2001. Component identification method with coupling and cohesion. 8th Asia-Pacific Software Engineering Conf., p.79-86. https://doi.org/10.1109/apsec.2001.991462

Lee, S.D., Yang, Y.J., Cho, E.S., *et al.*, 1999. COMO: a UML-based component development methodology. Software Engineering Conf., p.54-61. https://doi.org/10.1109/APSEC.1999.809584

Levi, K., Arsanjani, A., 2002. A goal-driven approach to enterprise component identification and specification. *Commun. ACM*, **45**(10):45-52. https://doi.org/10.1145/570907.570930

Maritime and Coastguard Agency, 2008. Search and Rescue Framework for the United Kingdom of Great Britain and Northern Ireland. https://www.gov.uk/government/publications/search-and-rescue-framework-uksar

Meng, F.C., Zhan, D.C., Xu, X.F., 2005. Business component identification of enterprise information system: a hierarchical clustering method. IEEE Int. Conf. on E-Business Engineering, p.473-480. https://doi.org/10.1109/icebe.2005.32

Qin, Z., Liang, Y., Zou, H., 2003. Study on identification method of business process based on CIMOSA. *J. Manag. Sci. China*, **6**(3):13-18 (in Chinese).

Scheer, A.W., 2000. ARIS—Business Process Modeling. Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-642-57108-4

Wang, Z.J., Xu, X.F., Zhan, D.C., 2006. A survey of business component identification methods and related techniques. *Int. J. Inform. Technol.*, **2**(4):229-238.

Yang, Y., Wang, J.F., Song, X.F., *et al.*, 2002. Business system planning based on BPR. *Comput. Eng. Appl.*, **22**:220-223 (in Chinese).

Yuan, X.W., Qin, Z., 2009. A decoupling approach to business component identification. 2nd Int. Symp. on Information Science and Engineering, p.467-471. https://doi.org/10.1109/ISISE.2009.40