Business Componentization: A Guidance to Application Service Design

Chunhua Tian[1], Wei Ding[1], Rongzeng Cao[1], Juhnyoung Lee[2]
1 IBM China Research Lab
Beijing, China
{chtian, dingw, caorongz}@cn.ibm.com
2 IBM T. J. Watson Research Center
New York, USA
jyl@us.ibm.com

Abstract. Aligning IT with business both at the strategic level and at the operation and management level is a challenge in enterprise architecture design. A simplistic approach of linking IT systems with business processes would not sustain because business processes are usually under continuous changes. In this paper, business componentization is proposed to address this challenge. An enterprise can be described as a set of business components with business services as their interaction interfaces. This paper discusses how business components can help the design of enterprise architecture. Also, we propose an interactive quantitative approach for business componentization. A business component is clustered from business activities based on a tightness evaluation of business processes, organizations, and IT systems. This paper presents a heuristic algorithm, and an aggregated clustering algorithm, for developing well-defined business component maps.

1 Introduction

In the face of dynamic business environments and markets, enterprises need to transform their business frequently and rapidly. *Business transformation* is a key executive management initiative that attempts to align technology initiatives of an enterprise closely with its market environment and business strategy, and is achieved through efforts from both of the business and IT sides of the company.

At an IT level, there has been a good deal of work for reducing the cost and effort of such transformations. It includes design methods (e.g., Object-Oriented Programming, Model-Driven Architecture, and Component-Based Development), development technologies (e.g., Integrated Development Environment), architecture

Please use the following format when citing this chapter:

Tian, C., Ding, W., Cao, R., Lee, J., 2006, in International Federation for Information Processing, Volume 205, Research and Practical Issues of Enterprise Information Systems, eds. Tjoa, A.M., Xu, L., Chaudhry, S., (Boston:Springer), pp.97-107.

methods (e.g., Service-Oriented Architecture, Middleware and many application-specific frameworks), software development process management (e.g., Rational United Process), and IT service management methodologies (e.g., IT Infrastructure Library), to name a few. These technologies and methodologies make IT systems and their implementation flexible and efficient for changes. However, it is apparent that an IT-level effort alone is insufficient to address the challenge of providing agile enterprise application systems supporting rapidly changing business requirements. *Enterprise Architecture* [1, 2] is an effort to bridge IT and business to achieve the intended goal of business flexibility. It provides a comprehensive and rigorous method and framework for describing a current or future structure for an enterprise's processes, information systems, personnel and organizational sub-units, so that they align with the organization's core objectives and strategic direction. Besides, it also includes certain methodologies to realize it in an enterprise. Although often associated strictly with information technology, enterprise architecture relates more broadly to the practice of business transformation, and so that it addresses business architecture, performance management and process architecture as well.

Based on the IT trends and EA methodology, the notion of services orientation is extended to an enterprise-level as SOE (Service Oriented Enterprise) 3. While it presents an ideal vision of a world in which resources are cleanly partitioned and consistently represented in terms of services, it also requires an appropriate way for service identification and implementation to make it practical. Until now, there is little work on how to identify applications or IT level services from a business perspective. To fill the gap, we propose a top-down approach using *business componentization.* In information technology, a component as a reusable building block has been an important concept to make IT systems responsive, variable and resilient. In a similar way, a *business component*, i.e., a reusable and loosely-coupled business-level block would make business architecture and operations more responsive, flexible and resilient. A business component is a logical view of part of an enterprise that includes resources, people, and systems necessary to deliver some value. An enterprise can be described as a set of business components with *business services* as the interaction interfaces. This paper discusses how business components can help the design of application services in enterprise architecture. Also, we present an interactive quantitative approach for business componentization.

The rest of this paper is structured as follows: Section 2 describes how business components can help in enterprise architecture design. Section 3 presents the heuristic algorithm for identifying business components from business activities and processes. In Section 4, conclusions are drawn and future work is outlined.

2 Business Component for Application Service Design

2.1 Business Components

In information technology, the notion of componentization is well-rooted. Especially in the hardware domain, the notion of plug-and-play has been widely accepted and

applied. It implies instant connection and operation – ideally, the user should not need to restart your computer, or go through an elaborate installation routine. In the application (or software) domain, the component notion has also been widely adopted, although probably not so complete as that in the hardware domain. In business architecture, the notion of componentization is novel. Most of business design or transformation is based on a business process (or value chain) analysis. However, as market emphasizes the speed of strategic changes, componentization becomes more critical in the business architecture. If a complex business is unbundled into separate components, its strategic changes would be easier to manage. Veryard 0 gives a detailed analysis about the management and technical drivers of business componentization.

In fact, the idea of componentization has been adopted and used in many industry solutions and frameworks, though some in different wording. For example, "main process" in SAP's solution map 5 has business goals/objectives, and is composed of several processes. eTOM 6 is another example. Level 2 processes of the eTOM Framework can be viewed as the components of information and communications services industry. However, the components in most of these process-based frameworks are only high-level business processes (or activities), which cannot be directly used in business reconfiguration. For efficient business transformations, a more service-oriented notion of business components would be needed.

A business component is a unit of business functionalities that serves a unique purpose, and is comprised of a group of cohesive business activities supported by the appropriate information systems, processes, organizational structure, and performance measures. It has the potential to operate independently, in the extreme case as part of another company. The composition of a business component is illustrated in Fig 1.

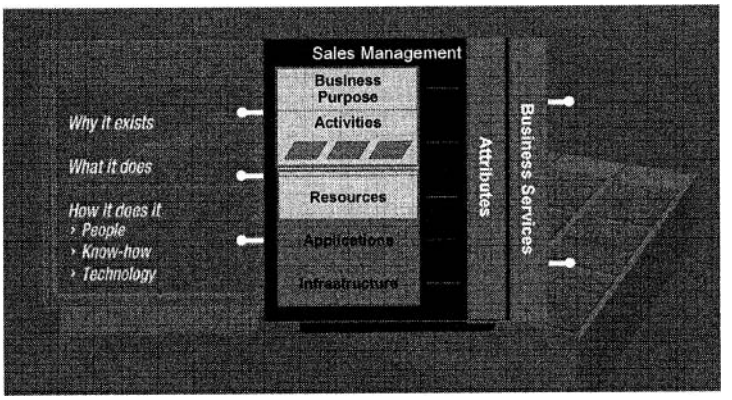

Fig. 1. Business component

From an external perspective, a business component can be described by business services that it offers and refers to, and related KPIs (Key performance Indicators). A business service is usually based on grouping of business functionalities. The grouping may be based on workflow, tasks, activities, and often include implicit or explicit rules. From an internal perspective, a business component can be described by business activities, resources, technology, and business process. The relationship between a business component and other entities is shown in Fig. 2.

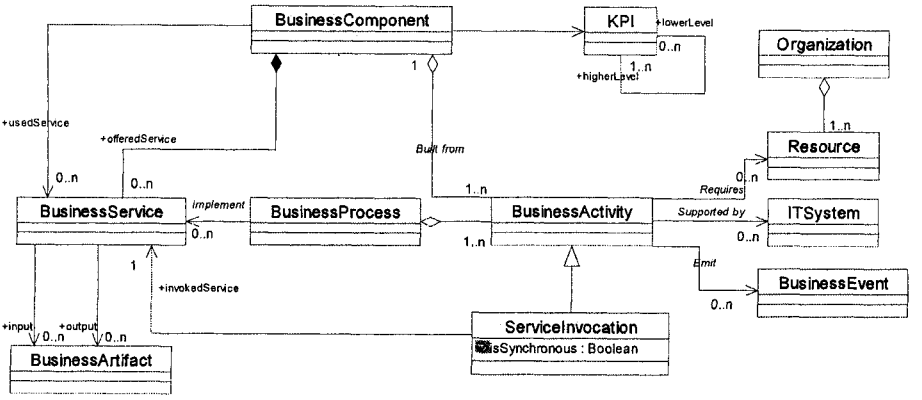

Fig. 2. The relationship between business component and related entities

It is important to note that the notion of business components mentioned in this paper differs significantly from what is presented in other literatures 7. Often, a business component is defined as an IT component that is used in business applications. The business component in presented this paper describes business level content.

Component Business Model (CBM) 8 is a technique for modeling an enterprise as a set of non-overlapping components in order to identify opportunities for innovation and improvement. Business components are organized as tabular view called as *business component map* with business competencies as columns and accountability levels as rows. An *accountability level* characterizes the scope and intent of activity and decision-making. The three levels used in CBM are directing, controlling and executing. Directing is about strategy, overall direction and policy. Controlling is about monitoring, managing exceptions and tactical decision making. Finally, executing is about doing the work. A *business competency* is defined as a large business area with characteristic skills and capabilities, for example, product development or supply chain. A component map is logical view of business operations, so it can be independent from enterprise size. Enterprises in the same industry can share the same or similar component maps, referred to as industry maps.

There is no simple compositional relationship between business processes and business components, because a business process is often a hierarchical structure. A high-level business process may depict the interaction between several business components, while a detail-level business process (or business activity) may lie inside of a business component. A business component can be viewed as a suitable-granularity business entity with a clear interface (i.e., business services). Thus, it is desirable for a business component to be supported by a single application, an IT component, or a minimum number of applications (or IT components) at least 9.

2.2 Business Components for IT Architecture

IT services can be classified into several categories, such as application services, infrastructure services, and enterprise service bus 10. It is hard to identify infrastructure services and enterprise service bus from business perspective, because these services are mostly designed from the IT perspective. Therefore, standard or reference IT service architecture (such as On Demand Operation Environment 10, or Service Oriented Computing 3) is often indispensable in IT architecture design. A business service potentially provides guidance for an application service. Below, an IT service refers to an application service.

Even specified to an application service, there is no simple way to convert a business service to an IT services. For instance, in a bank's business component map, let us assume that a business component, say, "application processing," offers two business services, "customer application submission" and "customer application status query". Also let us assume that it invokes three business services, i.e., the "product information query" service from the component "product profile", the "customer profile" service from the component "customer relationship management", and the "document recording" service from the component "document management". First, it is important to note that not all business services will be implemented as IT services, e.g., the "application submission" service can be a manual process where paper-based applications are transferred between branches. Secondly, a business service may have several implementation patterns. For

example, the "application submission" service can be implemented as a web service, a phone-based service, or even a manual channel.

However, a business service can provide certain guidance for IT service design in the specification of functional requirements, non-functional requirements, and data. The relationship between business services and application services is shown in Fig. 3. Another potential guidance lies in the interaction patterns. The interaction patterns of business components may be used to design the communication patterns of IT components.

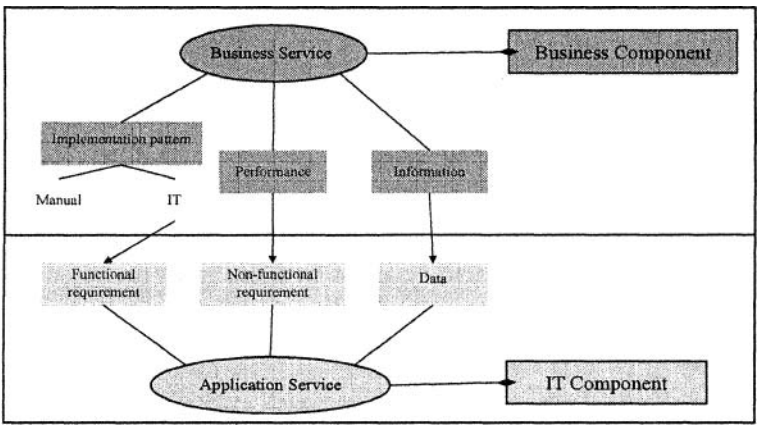

Fig. 3. Business component guidance in application service design

Combined with the existing reference IT service architecture, application services can be identified through business services. Certain common or infrastructure IT services will provided by the IT service architecture. If an application service is well-aligned with a business service, high-level process changes will have little impact on applications, because the interfaces at the business level will not be changed, and the impact of detailed process changes within a business component will be localized.

3 Business Component Identification

In previous sections, we assume that the business component map already exists. In practice, construction of a business component map for an enterprise is a challenging problem. One approach would be to create an enterprise map by customizing reference models such as industry maps developed in previous CBM consulting practices. Another approach would be to build one from existing artifacts such as business processes and activities, applications, organization, etc. This paper focuses on the latter approach and proposes a quantitative method.

A business component is specified by as a set of business activities (or business processes), offered/referred business services, execution organization, and

supporting IT systems. Services are logical artifacts which depend on business component partitions. Without business component partitions, it is difficult to know what business services should be offered or referred. In contrast, business activities (or business process), organization, and IT system are all concrete, and they can be used to identify business components. To construct a general business component, business activities are often used as clustering elements.

A business component can be formed by manually clustering business activities into a component by using human perception of "tightness" among activities. However, the process can be formalized and automated while leveraging human knowledge. We propose an interactive quantitative approach, which consists of the following steps: 1) establish criteria tree to measure the tightness among activities; 2) evaluate the tightness of activities using the criteria tree; and 3) At last, cluster activities into business components by a clustering algorithm which uses the calculated tightness values among activities.

3.1 Criteria Tree and Tightness Evaluation

Similarly to a software component, a business component is expected to be tightly related by internal elements, while loosely-coupled with external components. In an Object-Orient Design 11, there are several metrics for deciding whether a class is appropriate, such as COB (Coupling between Object Classes) which measures the number of related classes, and LCOM (Lack of Cohesion in Method) which measures how methods are invoked by use cases. However, those measures are devised for post-evaluation, that is, measurement is done after the design is completed for checking if the design is appropriate. For the business component design, it would be desirable if it is possible to evaluate them at the design time.

The internal tightness among business activities can be evaluated for three aspects: process (e.g., time constraints and performance influence), organization (e.g., skill level or goal similarity), and IT system. From the process perspective, the interaction among business components should be kept at a low level. From the organization view, the human resource that a component requires should possess similar skill sets. From the IT view, the activities in a component are desired to be supported by the same IT system. Each aspect can be further break down to a criteria tree according to enterprise preference. Through criteria tree and evaluation methodology such as AHP 12 (Analytical Hierarchical Process), the tightness among business activity i and j, f_{ij}, can be evaluated.

3.2 Three-step Partition Procedure

For the clustering process, the following problems need be addressed:

- Input complexity, i.e., the quantity of input needed by the quantitative approach. This requirement is important, because it is usually hard for the user to accept a quantitative approach that requires too much input.

- Computational complexity, the component partition is a combinational problem. Without an appropriate heuristic algorithm, the computational complexity will be prohibitively high. A solution to this problem will be discussed in Section 3.3.
- Flexibility for user, the algorithm should allow user interaction during execution, which will be discussed in Section 3.4.

In business component identification, if the pair-wise comparison is applied to all the n activities in activities matrix, the input complexity will be $O(n^2)$. An activity matrix typically has about 300~500 activities, so overall pair-wise comparison is unsuitable.

To address the input complexity problem, we proposes a multiple-step partition approach, which allows the activity clustering to be carried out in several sections independently, and then components to be clustered from the local components. Below the intersection between an accountability level and a business competency is called as a *cell*. A cell usually contains multiple activities.

The business competency and the accounting level are important description of business activities. It is more likely that the activities with the same business competency and/or accounting level have a tighter relationship. Thus, instead of performing comparison of activity pairs over the entire activity matrix, it is more efficient to first focus on the activities with the same business competency and/or accounting level. Based on this idea, the following 3-step partition procedure is proposed:

1. Partitioning is applied to each cell independently;
2. Then, partitioning is applied to each column, i.e., business competency, independently. The composition is applied to adjacent rows, i.e., the accountability levels.
3. Finally, the partitioning is applied between columns.

Now, let us analyzes the input complexity of the algorithm by using a sample activity matrix with 3 rows and k columns, and m pieces of activity in each cell. If one-shot procedure which requires the pair-wise comparisons among all the $3km$ activities is adopted, the input data complexity will be $O(9k^2m^2)$. For the proposed three-step procedure, the data needed in the first step is $O(3km^2)$. Suppose that there are l components in each cell after the first step. The data complexity in the second step is $O(2k^2l)$. Suppose that there are average j components in each cell after the first step. If a component occupies two rows, it will be regarded as two components in calculating j. The data complexity in the third step is $O(3k^2j^2)$. To sum up, the input complexity of the multiple-step procedure is about $1/3 \sim 1/3k$ of that of the one-shot procedure.

3.3 Aggregated Clustering Algorithm for Activity Clustering

Aggregated clustering algorithm (ACA) tries to cluster most tightly related activities into a business component considering both tightness evaluation and cluster size, because best practices of the CBM methodology shows that the size of individual business components (i.e., the number of contained activities) should not differ significantly. The objects to be clustered can be business activities or components.

For the sake of simplicity, objects (activities or components) to be partitioned are referred to as a general name of elements, and objects after partitioning are referred to as clusters.

The input to ACA includes m elements and their tightness evaluation value. There are three control parameters. L denotes the final number of components. Threshold $\lambda \in [0,1]$ is used to terminate the clustering process when all the element affinities are below the threshold. Parameter α is used to balance the size of different components by increasing the value of α. The computational complexity of ACA is $O(m^2)$.

Step 1 Let the m initial elements be m independent components $\{G_i\}$. Denote the size of cluster G_i (i.e., the number of activities in G_i) as n_i. Total number of activities as N is given by $N = \sum n_i$. The average size of clusters, $\overline{n}$, is given by $\overline{n} = N / m$.

Denote the tightness between cluster G_i and G_j as d_{ij}, which satisfies $d_{ji} = d_{ij}$. To avoid the self-clustering, let $d_{ii} = 0$.

For $i = 1,2,\cdots,m$, $j = i+1, i+2, \cdots, m$, let

$$\rho_{ij} = \left(\frac{n_i + n_j}{2\overline{n}}\right)^{\alpha}, \text{ and } d_{ij} = d_{ji} = f_{ij} / \rho_{ij}.$$

Whence, α is a weight value, which can be adjusted by users.

Step 2 Find $d_{kl} = \max\{d_{ij} : i, j = 1,2,\cdots,m\}$ $(k < l)$. Aggregate G_k and G_l into a new cluster, and let it replace the original G_k. Let $n_k = n_k + n_l$, $\overline{n} = N/(m-1)$,

For $i = 1,\cdots,k-1,k+1,\cdots,l-1,l+1,\cdots,m$, let

$$\rho_{ik} = \left(\frac{n_i + n_k}{2\overline{n}}\right)^{\alpha}, \text{ and } d_{ik} = d_{ki} = \max\{d_{ik}, d_{il}\} / \rho_{ik}.$$

Delete column l and row l from matrix D. Let $m = m - 1$.

Step 3 Repeat Step 2, until the number of clusters decreases to a given number L, or $\max d_{ij} < \lambda$.

3.4 User Interaction

The business component identification algorithms should be able to utilize human knowledge, because the accuracy of the tightness evaluation may heavily depend on domain knowledge. To leverage human knowledge and expertise, the proposed algorithms allow user interactions. Table 1 summarizes several user interaction scenarios, which can be translated as constraints or input of ACA.

Table 1. User interaction and handling method

User scenarios	*Translation to ACA*
Only selected elements will be clustered	Limit the elements input to the algorithm
Existed components cannot be altered	Limit the elements input to the algorithm
Element *i* and *j* must be joined together	$f_{ij} = 1$
Element *i* and *j* cannot be joined together	$f_{ij} = 0$
Delete current components	Free activities from components

3.5 Example

In the activity matrix shown in Fig. 4, there are 20 business activities. The number of comparisons (input complexity) of the one-shot approach would be 190, while that of the three-step approach would be 57. A pure algorithm-based approach is followed. The activities within the same red double block belong to the same business component.

	Business Administration	Product Management	Acquisitions
Directing	Develop business strategy	Develop segmentation strategy	Design and plan campaign
	Identify core capabilities	Define target segments	Define acquisition strategy
	Define organization structure	Track activity against segment plans/budgets	Develop target segment characteristics
	Develop operating model for the organization	Analyze product portfolio	Plan, define and conduct in-market acquisition tests
Controlling	Monitor Performance to Plan	Develop product specifications and features	Develop balanced building campaigns
	Design and develop HR policies		Monitor execution of campaigns
Executing	Train staff	Deploy product	Acquire target prospect list

Fig. 4. Activity clustering

4 Conclusions

Aligning IT with business at the strategic level and also at the operation and management level is a challenge in enterprise architecture design. We argued in this paper that business components provide useful views and guidance for application service design which was difficult with the traditional business process-based models. Also, we proposed an interactive quantitative approach to constructing business components from business activities based on the tightness evaluation of business processes, organizations, and IT systems. This paper presented an aggregated clustering algorithm to help form well-defined business component maps

with input data complexity, computational complexity, and user interaction flexibility.

Business componentization augments Enterprise Architecture with a novel view of businesses and help guide the IT architecture design. In addition, an extension to component business modeling provides analytical capabilities for business transformation and outsourcing. There are a number of interesting technical problems in this new direction of study. For example, it would be useful to identify and compose business services directly from business interaction, once business components are identified. Another interesting problem would be transformation between business services and IT services by using a formal model transformation technique.

References

1. B. Iyer and R. Gottlieb, The Four-Domain Architecture: An Approach to Support Enterprise Architecture Design, *IBM System Journal* **43**(3), 587-597 (2004).
2. http://www.enterprise-architecture.info
3. J. Schekkerman, EA & Services Oriented Enterprise (SOE)/Service Oriented Architecture (SOA) and Service Oriented Computing (SOC); http://www.enterprise-architecture.info/EA_Services-Oriented-Enterprise.htm.
4. R. Veryard, *Towards the Component-Based Business: Plug and Play* (Springer, London, 2000).
5. http://www11.sap.com/solutions/businessmaps/index.aspx
6. http://www.tmforum.org/browse.asp?catID=1647
7. P. Herzum and O. Sims, *Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise* (John Wiley & Sons, 2000).
8. S. Banerjea, Unlocking the Value of Account Opening with Component Business Modeling; http://www-1.ibm.com/services/us/index.wss/xb/imc/a1003029?cntxtId =a1000449.
9. P. Sousa, C. M. Pereira, and J. A. Marques, Enterprise Architecture Alignment Heuristics, *Microsoft Architects Journal*, Issue **4**, 34-39 (2004).
10. http://www-128.ibm.com/developerworks/xml/library/x-odstd/
11. S.R. Chidamber, Toward a Metrics Suite for Object Oriented Design, Special issue of SIGPLAN Notices (OOPSLA 91 Conference Proceedings), 26(11), 197-211.
12. H.F. Wang, *Multicriteria Decision Analysis: From Certainty to Uncertainty* (Ting Lung Book Company, 2004).