RC24127 (W0611-209) November 29, 2006
Computer Science

IBM Research Report

Business Componentization: A Principle for Enterprise Architecture Design

Chunhua Tian, Wei Ding, Rongzeng Cao
IBM Research Division
China Research Laboratory
HaoHai Building, No. 7, 5th Street
ShangDi, Beijing 100085
China

Juhnyoung Lee
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Business Componentization: a Principle for Enterprise Architecture 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 work because business processes are usually under continuous changes. Business componentization is proposed to address the challenge. 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 enterprise architecture. Also, we propose an interactive quantitative approach for business componentization. A business component is clustered from business activities based on tightness evaluation of business processes, organizations, and IT systems. This paper presents two heuristic algorithms, a fuzzy clustering algorithm and an aggregated clustering algorithm, which help form 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 the technology initiatives of an enterprise closely with its business strategy and vision, and is achieved through efforts from both the business and IT sides of the company. However, the technology side of the company often emphasizes functions and capabilities, while the business side focuses on business impact and value. Because of this "business-IT gap," business transformation processes for IT and services are long and costly.

For closing the business-IT gap, it is critical for enterprises to build flexible and robust application systems to support business transformation in an efficient and flexible manner. There has been a good deal of work done in IT to meet this requirement. It includes design methods (e.g., Object-Oriented Programming, Model-Driven Architecture, and Component-Based Development), development technologies (e.g., Integrated Development Environment), architecture 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 clear that an IT-level effort alone is insufficient to address the challenge of providing agile enterprise application systems supporting rapidly changing business needs.

Enterprise Architecture is an effort to bridge IT and business to achieve the intended goal of business flexibility. It is the practice of applying a comprehensive and rigorous method 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 goals and strategic direction. Although often associated strictly with information technology, it relates more broadly to the practice of business transformation in that it addresses business architecture, performance management and process architecture as well. Enterprise Architecture is a significant practice within the U.S. Federal Government as a means of addressing persistent weaknesses in information technology investments. As it gains popularity, many companies in the private sector also have applied Enterprise Architecture to improve their business architectures as well in order to improve business performance and productivity.

To address the business-IT alignment problem, we propose to augment Enterprise Architecture with *business componentization*. In information technology, component as a reusable building block has been an important concept to make IT systems responsive, variable and resilient. In a similar fashion, *business components*, i.e., reusable and loosely-coupled business-level blocks, would make business architecture and operations more responsive, variable and resilient. A business component is a part of an enterprise that has the potential to operate independently, in the extreme case as a separate company, or as part of another company. A business component is a logical view of part of an enterprise that includes the resources, people, technology and know-how 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 enterprise architecture. Also, we present an interactive quantitative approach for business componentization. A business component is clustered from business activities based on tightness evaluation of business processes, organizations, and IT systems. We present two heuristic algorithms, a fuzzy clustering algorithm and an aggregated clustering algorithm, which help form well-defined business component maps.

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 algorithms for identifying business components from business activities and processes. In Section 4, conclusions are drawn and future work is outlined.

2 Business Component for Enterprise Architecture

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 is widely accepted. 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 is also widely adopted, although probably not so complete as that in the hardware domain.

In business architecture, the notion of componentization is a novel concept. Most of business design or transformation is based on a business process (or value chain) analysis. However, as market emphasizes the speed of strategy change, componentization becomes more critical in business architecture. If a complex business is unbundled into separate components, or if a new enterprise is configured from separate components, individual components would be easier to manage. Veryard[4] 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.

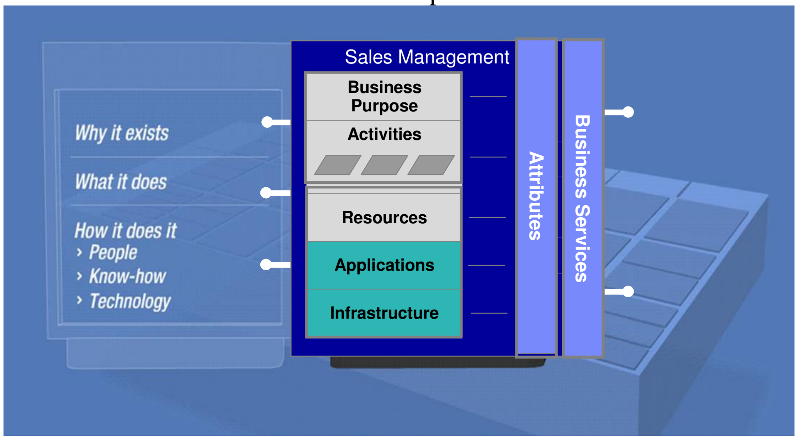

Fig. 1. Business Component

A business component is a unit of business functionalities that serves a unique purpose, which 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 data structure of a business component is illustrated in Fig 1.

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.

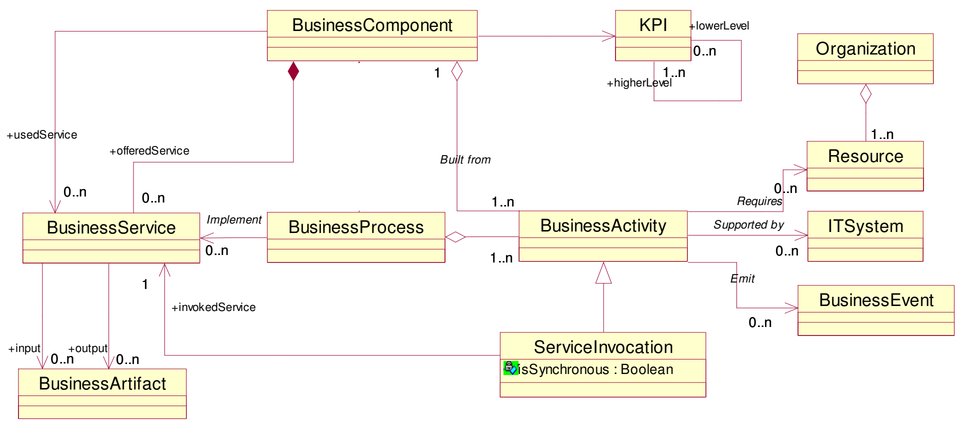

Fig. 2. The relationship between business component and related entities

It is important to note that the concept of business components proposed in this paper differs significantly from what is presented in other literatures [7][8]. Often, a business component is defined as an IT component that is used in business applications. The business component in this paper is about business level content.

A typical enterprise comprises of 60~100 business components and around 500~1000 business activities. To address the complexity, an organized arrangement of the components is required. There are several logical organization methods existing for different purposes in different reference frameworks. For example, in e-TOM, the components are organized according to key functional areas and vertical end-to-end processes. However, most of the existing approaches can be used only for specified areas.

Component Business Model (CBM) [9] provides a more general organization of business components in form of business component map, where business components are organized in a matrix with business competencies as columns and accounting levels as rows. A *business component map* is a tabular view of the business components in scope. 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. 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.

Component business modeling provides a technique for modeling an enterprise as a set of non-overlapping components in order to identify opportunities for innovation and improvement. The modeling is of the business itself, not of applications or technology. Also, CBM provides an analytical tool, and so it is sometimes referred to as a lens through which a business can be examined and analyzed. CBM is complementary to process modeling techniques. A business process can be interpreted in CBM as collaboration among a network of business components. Conversely, from a process perspective, a business component is a closely related group of sub-processes.

2.2 Business Components for IT Architecture

There is no simple, explicit compositional relationship between business processes and business components, because a business process is often a hierarchical structure. A high-level business process may interact with several business components, while detail-level business process (or business activity) may lie inside of a business component. A business component can be viewed as a suitably-grained 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.

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

Even specified as an application service, there is no simple way to convert a business service to an IT services (e.g., in SOA framework). For instance, in a bank's component business 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., "product information query" service from the component "product profile", "customer profile" service from the component "customer relationship management", and "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., "application submission" service can be a manual process where paper-based applications are transferred from branches. Secondly, a business service may have several implementation patterns. For example, "application submission" service can be implemented through a computer network, a phone network, or 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 IT 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.

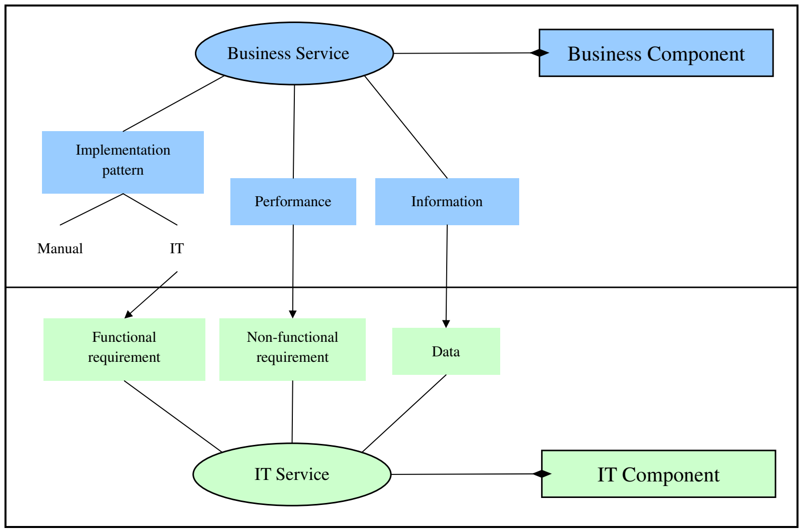

Fig. 3. Business component guidance in IT architecture design

Combined with the existing reference IT service architecture framework, application related IT 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 are not changed, and the impact of detailed process changes within a business component will be localized.

3 Business Component Identification

In previous sections, we assumed the business component map already exists. In practice, construction of a business component map for an enterprise is a key problem. One approach is to create an enterprise map by customizing reference models such as a universal component map and industry maps previously developed in CBM consulting practices. Another approach is to build it from existing artifacts such as business processes and activities, applications, organization, etc. The latter approach is useful for creating a more precise map. This paper focuses on this approach and provides a quantitative method.

A business component is specified by as set of business activities (or business processes), a set of 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 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:

- First, establish a criteria tree to measure the tightness among activities;
- Then, evaluate the tightness of activities using the criteria tree; and
- 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 (business activities), while loosely-coupled with external components. In an Object-Orient design[11], there are several metrics to measure 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 of classes is completed to decide whether the design is appropriate. For the business component design, we want to evaluate them at a design time.

The internal tightness among business activities can be evaluated from 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 system, the activities in a component should be supported by the same IT system.

By using a criteria tree, the degree of tightness among business activities can be evaluated. Due to the uncertainty in evaluation, fuzzy value is adopted in our approach.

1) The evaluation criteria tree is constructed with leaf criteria set:

$$C = \{c_1, c_2, \cdots, c_n\}$$

where *n* is the number of leaf criteria.

2) For each activity pair (*i, j*) ($i \neq j$), their tightness according to each criteria c_k, $m_{ij}(c_k) \in [0,1]$ ($k = 1,2,\cdots,n$), is assessed by domain experts.

3) For an activity pair (*i, j*), the membership function of fuzzy similarity relationship,

$$f_{ij} = f\big(m_{ij}(c_1), m_{ij}(c_2), \cdots, m_{ij}(c_n)\big) \in [0,1],$$

is defined. Thus, the tightness degree f_{ij} for each activity pair (*i, j*) can be calculated.

For example, f_{ij} can be defined as

$$f_{ij} = \sum m_{ij}(c_k) w(c_k) / \sum w(c_k)$$

where $w(c_k)$ is the weight of criteria c_k, which can be obtained through AHP[12] (Analytical Hierarchical Process), a formalized way to obtain the weights of evaluation criteria through pair-wise comparisons.

3.2 Issues in Activity Clustering

For this algorithm-based method, 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. 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.
- Computational complexity: The component partition is a combinational problem. Without an appropriate heuristic algorithm, the computational complexity will be inhibitively high.
- Flexibility for user: The algorithm should allow user interaction during execution.

To address the input complexity problem, Section 3.3 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. To reduce the computational complexity, two heuristic algorithms – a fuzzy clustering algorithm and an aggregated clustering algorithm – are proposed in Section 3.4. They perform with low computational complexity, and the latter allows the user to define expect number of components.

To increase the user interaction flexibility, the activity clustering procedure also offers the following two features:

- Interaction patterns: It allows the current business components unchanged; it reserves the current components, but allows adding activities to them; and it allows deleting the current components.
- Rule: It allows the user to indicate pairs of business activities which should belong to a component, or should not be in the same component.

These features will be translated into the algorithms as constraints or pre-handled as input to algorithm. They will be described in detail in Section 3.5.

3.3 Three-step Partition Procedure

Below the intersection between an accountability level and a business competency is called as a *cell*. A cell usually contains multiple activities.

A business competency and an accounting level are important description of business activities. It is 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., accountability level.
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.

3.4 Heuristic Algorithm for Activity Clustering

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.

Suppose the number of initial elements is m, which is also the dimension of the matrix R. In heuristic algorithms, a threshold $\lambda \in [0,1]$ is often adopted to determine whether two objects can be clustered.

Two algorithms are presented in this section. The fuzzy clustering algorithm is directly related with fuzzy variables, and is low in computation complexity, while the aggregated clustering algorithm allows more user interactions.

3.4.1 Fuzzy Clustering Algorithm (FCA)

Step 1 (Fuzzy Equivalent Matrix Calculation) The fuzzy equivalent matrix $\overline{R}$ can be obtained by the following fuzzy operations:

$$\overline{R} = R^{m-1} = R \circ R \circ \cdots \circ R$$

$$R \circ R = [r'_{ij}], \quad r'_{ij} = \max_k \{\min(f_{ik}, f_{kj})\}$$

Step 2 (Elements Partition) Given a threshold $\lambda \in [0,1]$, the partition is handled according to the following rule: if $r'_{ij} \geq \lambda$, element *i* and element *j* are clustered into the same component. Otherwise, they belong to different components. By regulating the value of *λ,* we can obtain different-granularity components.

3.4.2 Aggregated Clustering Algorithm (ACA)

In FCA, only the tightness degree among elements is considered. Best practices of the CBM methodology requires that the size of individual business components (i.e., the number of contained activities) should not differ significantly. ACA takes the size of clusters into consideration.

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. Denote the total number of activities as *N*, which 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

$$\rho_{ij} = \left(\frac{n_i + n_j}{2\overline{n}}\right)^{\alpha},$$

$$d_{ij} = d_{ji} = f_{ij} / \rho_{ij}.$$

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

In the tightness calculation, the cluster size is taken into account through denominator ρ_{ij}.

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

$$\rho_{ik} = \left(\frac{n_i + n_k}{2\overline{n}}\right)^{\alpha},$$

$$d_{ik} = d_{ki} = \max\{d_{ik}, d_{il}\} / \rho_{ik}.$$

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 $\max d_{ij} < \lambda$.

3.4.3 Algorithm Comparison

The two algorithms can be compared from three aspects shown in Table 1:

1) Computational complexity;
2) Configuration parameter, more configuration parameter means more freedom for user.

3) Control approach, that is, how the algorithm determines how many components will be attained. A direct control approach allows the user to directly determine the number of components generated. An indirect control approach allows the user to impact the final number of components by changing certain parameter values, but does not provide the exact number before the algorithm is executed.

Table 1. Comparison of two algorithms

	Computational Complexity[*]	***Configuration Parameter***	***Control Approach***
FCA	$O(n^4)$	λ	Indirect
ACA	$O(n^2)$	α, λ, L	Direct/Indirect

[*] The complexity of FCA is calculated according the number of comparison, while the complexity of ACA is computed according to the number of multiply/division. So FCA may have lower computational complexity.

3.5 User Interaction

The business component identification algorithms should utilize human knowledge, because the tightness evaluation heavily depends on domain knowledge. To leverage human knowledge, the proposed algorithms allow user interactions. Table 2 summarizes several user interaction scenarios and corresponding handling methods.

Table 2. User interaction and handling method

User scenarios	***Handling method***
Only select elements will be clustered	Limit the elements input to the algorithm
Existed components can not 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

Also, Fig. 4 illustrates the user interactions in the clustering algorithms.

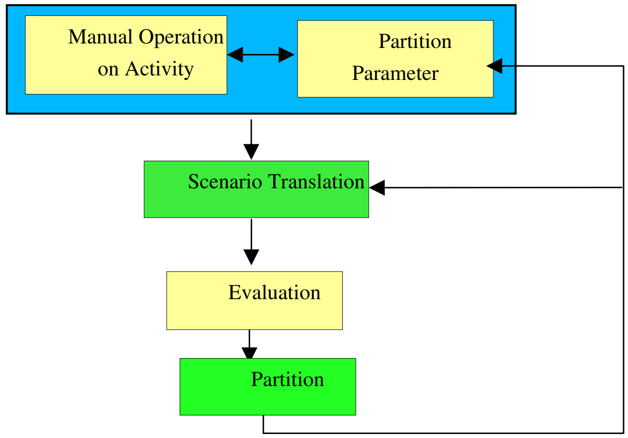

Fig. 4. Interactive algorithm-based clustering

3.6 Example

In the activity matrix shown in Fig. 5, 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
Planning & Analysis	Develop business strategy	Develop segmentation strategy	Design and plan campaign
	Identify core capabilities	Define customer target segments	Define acquisition strategy
	Define organization structure	Develop and track activity against segment plans and budgets	Develop target segment characteristics
	Develop operating model for the organization	Analyze product portfolio	Plan, define and conduct in-market acquisition tests (champion/challenger)
Checks & Controls	Monitor Performance to Plan	Develop product specifications and features	Develop balanced building campaigns
	Design and develop HR policies		Monitor execution of campaigns
Execution	Train staff	Deploy product	Acquire target prospect list

Fig. 5. Activity clustering

4 Conclusions

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 work because business processes are usually under continuous changes. We argued in this paper that business components provide useful views and guidance for enterprise architecture design which the traditional business process-based models do not. We discussed that, with well-defined business components, business services can be adopted to design application services. Also, we proposed an interactive quantitative approach to constructing business components from raw information of business activities. A business component is clustered from business activities based on tightness evaluation of business processes, organizations, and IT systems. This paper presented two heuristic algorithms, a fuzzy clustering algorithm and an aggregated clustering algorithm, which help form well-defined business component maps. We analyzed the proposed algorithms by their 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 an analytical tool for business transformation and outsourcing. This paper presents intermediate result from an ongoing research project on business design and transformation at IBM Research Center. There are a number of interesting technical problems in this new direction of Enterprise Architecture. For example, it would be useful to identify and compose business services directly from businesses and interaction[13], once business components are identified. Another interesting problem would be transformation between business services and IT services by using a formal model transformation technique.

5 References

[1] B. Iyer and R. Gottlieb. The Four-Domain Architecture: An approach to support enterprise architecture design. IBM System Journal, Volume 43, Number 3, 2004, pp. 587-597.

[2] http://www.enterprise-architecture.info

[3] Pedro Sousa, Carla Marques Pereira and Jose Alves Marques. Enterprise Architecture Alignment Heuristics. Microsoft Architects Journal, Issue 4, 2004, pp. 34-39.

[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. The Business Component Approach, http://jeffsutherland.org/oopsla98/sims.html.

[8] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/bpsdk /BPIG_Ch01_TypesOfIntegrations1048325.asp

[9] Sunny 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.

[10] http://www-128.ibm.com/developerworks/xml/library/x-odstd/

[11] Shyam R. Chidamber. Toward a metrics suite for Object Oriented Design, Special issue of SIGPLAN Notices (OOPSLA 91 Conference Proceedings), Vol. 26, No 11, pp. 197-211.
[12] Hsiao-Fan Wang. Multicriteria Decision Analysis-From Certainty to Uncertainty, Ting Lung Book Co., 2004.
[13] Gerke Geurts and Adrie Geelhoed. Business Process Decomposition and Service Identification Using Communication Patterns. http://msdn.microsoft.com /architecture/journ/default.aspx?pull=/library/en-us/dnmaj/html/aj1bizproc.asp.