# Implementation choice of business rules in e-commerce

Master thesis, 10 credits (15 ECTS), Department of Informatics

*Presented:* June, 2008

*Authors*: Liang Yuhan

Jin Rui

*Supervisor*: Odd T. Steen

*Examiners:* Erik Wallin

Kjell-Ake Holmberg

**Lund University**
**Department of Informatics**

# Implementation choice of business rules in e-commerce

© Liang Yuhan

© Jin Rui

Master thesis presented June, 2008
Size: 70 pages
Supervisor: Odd T. Steen

**Abstract**

The concept of business rule has been developed for several decades. It was initially studied for the purpose of developing rule-based software system. But nowadays, people are aiming at developing knowledge-based software system and applying business rules from business perspective. The technology relating to business rules has been developed to a huge extent, with business rule engine and business rule management system as typical products of this technology.

Within the business rule life cycle, the implementation of business rules is a troublesome step. People are not sure about where they should locate business rules and how to implement business rules. In this study, we are aiming at discovering a methodology to evaluate different implementation choices of business rules. And other people’s opinions in our methodology are collected to identify the shortcomings in the methodology. The shortcomings will implicate what we should do in further study to complete the methodology.

**Key words**
Business rules, BR implementation, methodology

## Acknowledges

In this acknowledge we would like to express our appreciation to several persons who encouraged and supported us during the progress of our thesis. First of all, we should thank our supervisor Odd Steen Ph.D. He helps and guides us carefully with his precious points of view through this research. We also want to thank Bo Andersson who generously provided valuable criticism, which really benefit the improvement of this thesis. Finally, we would like special thanks to the three interviewees, their specific views contributed in our research and we are great appreciating that they are taken their times to participant in our interviews.

Thank you all!

Lund University, June 2008

Jin Rui Liang Yuhan

# Table of Contents

**Table:**

**Figure**

# 1. Introduction

In this chapter, we will firstly introduce some information about the background and problem area in order to offer an initial image in front of our readers. After that we will focus on the research question, purpose, limitation and delimitation.

## 1.1 Background

In the beginning of 1990s when Internet became open to commercial use, electronic commerce (EC) emerged as a “new way of doing business” (Turban, 2000) and brought companies a hopeful future. A large number of companies poured money into developing their own electronic commerce, even though some of them failed at the initial stage. With the improvement of EC knowledge and practical usage, e-commerce has been developed at a rather high speed since 2000 and generated huge profits.

In this paper, we will focus on business-to-customer (B2C), a type of customer-oriented online retailing. Before the emergence of EC, the retailers were facing some problems which blocked their further development. They found it hard to achieve active and quick interactions with customers because of the adoption of intermediaries, found their businesses were limited to local or nearby regions, and worried about the return rate of marketing because their customers were individuals with lots of differences and lacking of loyalties. B2C basically offers a solution to above problems.

The successes of Amazon (online bookstore, found in 1995), Ebay (American web shop) and Dell computer set up good samples to the followings and contributed to the boom of B2C in global scale. According to GVU center’s 10th WWW user survey (Turban, 2000), the top five items purchased online (based on market segments & purchase amount) are software, books, hardware, music and travel.

B2C is characterized by a rapid but personalized response to the customer and web-based, customer-based marketing (Arthur, 2001). In order to keep competitiveness, a B2C company needs to be sensitive to changes coming from both organization itself and external environment. Babis and Anthony (1998) described changes may come from mainly three sources: policy statements and objectives of organization, business process change and external factors (e.g. laws and regulations). Their description is also suitable for B2C. For example, a company may change its discount mechanism according to their business strategy change; it may allow a customer to pay after products delivery (cash on delivery) instead of paying before delivery (delivery against payment); it may respond to a new legislation.

In order to make quick response to changes we mentioned above, a B2C company needs to know how their business is conducted now and how to do their business after making corresponding changes. In other words, they need to capture business rules. Business rules (BR) express how an enterprise conducts its business (Morgan, 2002), so they are captured as a bridge of keeping the entire IS of an organization aligned with its business (Marko Bajec, 2004). Business rules used to be expressed and implemented by coding in general programming languages (GPL, such as Java, C#) or by constraints in data base. But people found the rules implemented by GPL were hard to located and modified, and decided to "outsourcing" the business logic to the business side so that end users or business experts can guide business practices directly (Ronald Ross, 2003). Now people are prone to separate business rules from application codes. Business rules are expressed in standardized natural language or decision table or decision tree (Lublinsky & Tien, 2007). And they may be implemented by BR component (e.g. Prolog, Clips) or business rule engine or business rule management system (BRMS).

Because e-commerce is based on computers network, the business rules can be implemented in several locations (e.g. client end, data base and application server) and in different manner (e.g. scripts, BR components, BR engine and BRMS). Different implementation choices may trigger different cost, return and performance. So B2C companies are looking for some method to help them make wise choices for BR implementation.

## 1.2 Problem Area

The conception of business rule has been developed for several decades, and some standards have been set up to define and capture business rules. As for the implementation of business rules, we only have knowledge about all potential locations (where to implement BR) and all potential manners (how to implement BR). But no guidance relative to how to make a decision for business rule implementation is offered.

Business rules used to be implemented by IT professionals via coding in data base or applications, but now people find more advantages from BR components or BRMS. With BRMS, business rules are formulated in standardized natural language, which can be understood by both IT professionals and non-IT professionals. So BRMS can enable business experts to determine and rewrite the business logic (Morgan, 2002). And we can take advantage of BRMS as a separate service to build up Service Oriented Architecture (SOA) within B2C companies. But everything has two sides. The technologies of BR engine and BRMS are not widely applied, and BRMS products are rather expensive. Some companies with small or medium size may find it unaffordable to invest in a BRMS.

This study will concentrate on the implementation of business rules. Choosing a suitable one from alternative implementation choices could make BR maintenance and updating easier. Because B2C is based on computers network (including Internet), business rules may be distributed in different locations and in different manners. We need to discuss how to implement BR in order to achieve a good modifiability of business rules.

## 1.3 Question

*How should we evaluate different implementation choices of business rules in e-commerce?*

## 1.4 Purpose

The purpose of this study is to find out an evaluation methodology which can be used to evaluate different implementation choices of business rules for a business-to-customer company.

Because business rules can be implemented in different locations and in different manners, several implementation choices may become candidate in one case. For example, an organization can choose to implement business rules by application coding in business logic layer or by stored procedures in data layer or by applying BRMS in business logic layer. Computers network, based on which a B2C business is built up, will make the implementation of business rules more complex. The architects or developers need to make a trade-off between centralized business rules and distributed business rules.

Although there are some candidate choices for BR implementation in one case, it is hard to evaluate them. So we will focus on finding an effective evaluation methodology for BR implementation in this study. The purpose of this methodology is to evaluate each candidate implementation choice, while no subjective judgment (about which choice is better) is offered.

## 1.5 Delimitation

Efraim Turban (2000) classified electronic commerce by the nature of the transactions as following: business-to-consumer (B2C), business-to-business (B2B), consumer-to-business (C2B) and so on. We will delimit our thesis to B2C type in order to keep our study in a smaller and controllable scope. This doesn’t mean other types are not important, but we leave them side for the future studies.

We will introduce an evaluation method to help online retailers make choices for BR implementation. The users of this evaluation method are delimited to the qualified persons who have pre-knowledge relating to business rules and software architecture.

The effectiveness of this kind of evaluation depends on good understandings of a company’s architecture and BR software’s architecture. If the evaluation is made by the buyers of BR software (e.g. BR engine, BRMS), good communications between the buyers and sellers are needed. But it will become more difficult if the evaluation team is from the outside of a buyer company. In our study, we assume the buyer company has qualified staff to do evaluation by itself.

## 1.6 Thesis structure

*Chapter 1*, Introduction
Introduce information about the problem area, research questions, research purpose and delimitation. In this chapter, we can learn the object of the thesis, and identify the research scope.

*Chapter 2*, Literature review
Present and outline the theoretical basis of this thesis. The theories from literatures can let our readers get some knowledge about the topic. And our research is conducted based on these theories.

*Chapter 3,* Method
Describe the method we choose to do the research. In this chapter, our readers can have an insight into how our research is designed, how to collect and analyzed data, and how to keep research quality.

*Chapter 4*, Methodology of evaluating BR implementation
Introduce a methodology, which can be applied to evaluate different implementation choices of business rule. This evaluation methodology is the core of our research because it can realize the research purpose. And an interview is designed to collect other people’s opinions in the methodology.

*Chapter 5*, Apply the methodology in reality
Explain how to apply the methodology in business rules area in reality. A sample is set up to deepen the readers’ understandings.

*Chapter 6*, Research results
Summarize the results from the research, especially the interviews we conducted. The results are mainly related to the interviewees’ opinions in the methodology, from general feelings to low detail opinions.

*Chapter 7,* Discussions
More discussions are involved in chapter 7, based on the research results.

*Chapter 8*, Conclusions
Draw conclusions for this thesis.

## 2. Literature review

In this chapter, we will introduce some theories relating to our research question. Our study will be conducted based on these theories, so it will be helpful for our readers to get some basic knowledge.

### 2.1 Introduction to electronic commerce and B2C

#### 2.1.1 What is electronic commerce?

Electronic commerce started in the early 1970s, and has been developed at a rather high speed since the early 1990s when Internet was opened to commercial usage (Turban, 2000). Recently, EC has been universally applied by medium- and large-size companies.

Electronic commerce is described by Kalakota & Whiston (1997) as a process of buying and selling or exchanging of products, services and information via computer networks including Internet. In simple words, their business are expanded or generated into cyberspace. The huge benefits from electronic commerce contribute to the boom of EC in a global scale.

We would like to introduce the dimensions of electronic commerce, as seen in figure 1 (Choi et al, 1997). This figure shows which kinds of commerce can be considered as EC and its expected pattern. According to figure 1, three dimensions should be included in order to distinguish electronic commerce from traditional commerce: products, agent and process. Each dimension can be physical or digital. In traditional business pattern, a physical product is handled in a physical process by a physical agent. A commerce pattern could be considered as electronic commerce if at least one dimension is digital. Pure EC should have three digital dimensions and it is an ideal model of EC. A good sample of pure EC is online software store.

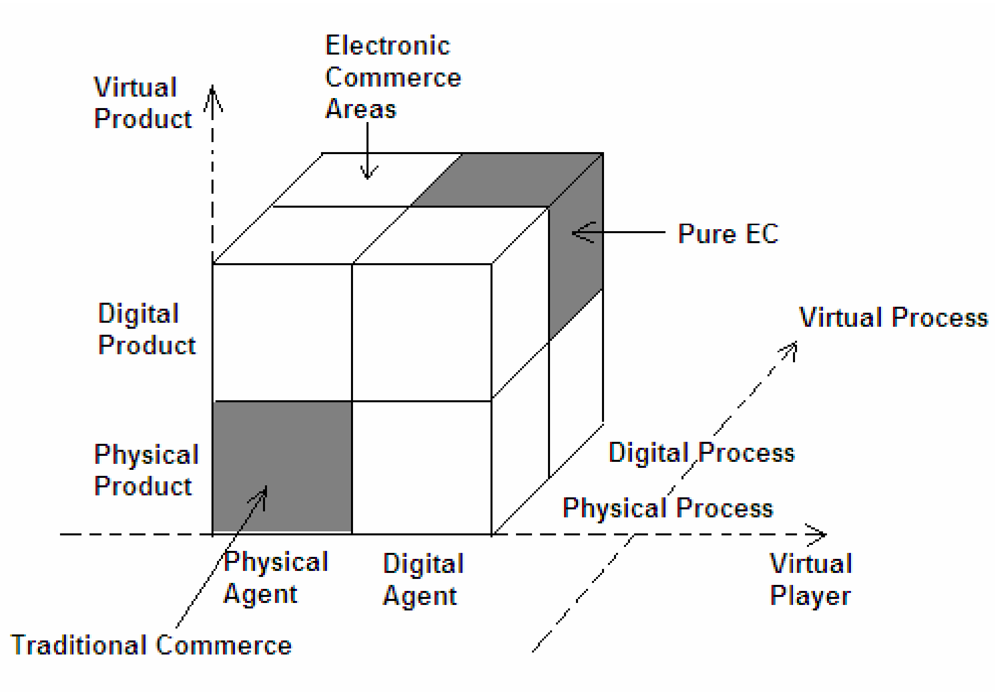


*Figure 1.The dimensions of Electronic Commerce, Source: Choi et al (1997): The Electronic Commerce, Macmillan Technical Publications, p.18*

According to Efraim Turban (2000), electronic commerce has been classified by the nature of transaction as following types: Business-to-business (B2B), Business-to-commerce (B2C), Customers-to-business (C2B) and so on. We will focus on B2C in our study, but it doesn't mean other types are less important. Our research should be expanded to other types in further study.

### 2.1.2 Business-to-commerce (B2C)

Turban (2000) defined Business-to-consumer (B2C) as activities of e-businesses serving end consumers with products and/or services. B2C is a type of customer-oriented online retailing, and enables retailers to put their business into cyber market.

Before the emergence of B2C, the development of retailing professions was obstructed by some problems. Retailers found it hard to react with customers efficiently because of the existence of intermediaries; their businesses were limited to local or nearby regions; the effort of marketing is not obvious because their customers were individuals with lots of differences and lack of loyalties.

The adoption of B2C brings a hopeful future for retailers. Their business can reach international market via Internet, even for small size retailers. Customer-oriented online marketing became an effective means of marketing for online retailers. Other benefits from B2C are the phenomena of disintermediation and re-intermediation. With the development of B2C, the traditional intermediaries (e.g. distributors and wholesalers) are eliminated (so-called disintermediation), and a new kind of electronic intermediaries emerges (so-called re-intermediation).

According to the definition of business-to-customer, B2C includes mainly three types of services (http://projects.bus.lsu.edu/independent_study/vdhing1/b2c/ ):

- Auction stores (e.g. Ebay) -the transactions between business corporations and customers are operated according to the bidding mechanism similar to traditional auctions.
- Online stores (e.g. Amazon) –this type is universally applied. Nowadays there are lots of online stores which are running different sorts of business, such as book, CD, flower and fruit.
- Online services (e.g. www.travelocity.com) –companies offer online service to individual customers, such as travel service.

### 2.2 Business rule

After a simple introduction to electronic commerce and B2C, we start to introduce some conceptions relating to business rules. According to our research question, e-commerce is the context where business rules are implemented.

#### 2.2.1 The conception of business rule

According to Morgan (2002), a business rule is a compact statement about an aspect of a business. The rule can be expressed in terms that can be directly related to the business, using simple, unambiguous language that's accessible to all interested parties: business owner, business analyst, technical architect, and so on. Simply speaking, business rules describe how a company conducts its business. For example, a B2C company requires individual customers to pay for each order before its delivery (delivery against payment). This description displays how individual orders should be handled, so can be identified as a business rule. Learning how business is done right now will facilitate making changes in future.

The Business Rules Group (found in 1989) defined business rule in both business perspective and information system perspectives in their home page (http://www.businessrulesgroup.org). From the business perspective, a business rule is guidance that there is an obligation concerning conduct, action, practice or procedure within a particular activity or sphere. From the information system perspective, a business rule is a statement that defines or constrains some aspect of the business.

According Business Rules Group, IT professionals and non-IT professionals may understand business rules from different perspectives. Business rules may be captured by business experts, business owners or end users for keeping business works. While IT professionals, who can also in charge of BR capture, aim at making their applications usable in reality. So we believe business experts or analysts can understand business rules from more comprehensive perspective.

### 2.2.2 History of business rules

At the initial stage, business rules were captured by IT professionals as “IF-THEN” logics, and implemented by coding in general programming language (e.g. Java). They just believed IF-THEN logics were important for their application development, without clarifying what is BR and how to describe it.

Then people realized they need to define business rule and set up some standards for BR description. At this stage, BR was described in simple natural language, with a serial number (for example, R1: customers must be at least 18 years old). The capture and description of business rules can be done by both business experts and IT professions. And BR was implemented by IT professionals via applications coding. The main purpose of BR capture was to build up rule-based applications.

Nowadays, business rules could be captured by business experts or business owner or end users from different departments, described in natural language, and implemented by BR component (e.g. Prolog) or BR engine or BRMS. Business rules are “outsourced” to the business side (Ronald G. Ross, 2003) and separated from ordinary programming codes. By the adoption of BR engine or BRMS, business experts could realize and rewrite business rules without the help of IT Professionals.

### 2.2.3 Business rule source and rules type

Business rules could be derived from several sources, and the mainly three sources are described as following (Babis and Anthony, 1998): policy statements and objectives of organization, business process and external factors (e.g. laws and regulations). Bajec and Krisper (2004) emphasize the importance of recording BR sources. Because tracing the sources of business rules can help people discover the need of changing BR. If the source changes, the rule relating to this source has to be modified or removed.

Morgan (2002) summarized three main types of business rules as following table 1. There may be some lower classification within each rule type, which we will not introduce here. The Structural rules are corresponding to the relationships between entities in UML diagram. Definitional rules are used to define some terms. Behavioral rules are coming from control flow, information flow, pre-conditions and so on (Marcel Weiden & Leo Hermans, p.3). “IF-THEN” business logics are typical behavioral rules.

**Table 1 Three main types of business rules**

| Rule types | Definition | Sample |
|---|---|---|
| Structural rules | Constraints or relationships among various elements of a system | A customer must own at least one customer account. |
| Behavioral rules | Define a course of action to be followed in particular situation | An individual order must be paid before its delivery. |
| Definitional rules | A definition of a term or a quantitative or qualitative relationship between terms | The total price is defined as product price plus delivery fee. |

### 2.2.4 The description of business rules

Business rules can be described in simple natural language or decision tables or decision trees. While the common used one is simple natural language which could be readable and accessible to all interested parties: business owner, business analyst, technical architect, and so on.

Some typical patterns of business rules described in English-like natural language are displays as following (Morgan, 2002):

- Basic constraint

Example: a customer must be at least 18 years old.

- List constraint

Example: a next-day order must not be accepted if at least one of the following is true:

- Order is received after 15:00, Monday through Friday,
- Order is received anytime on Saturday or Sunday,
- Order is received between 15 December and 5 January,

- Classification

Example: an order is defined as urgent if delivery is required in less than three hours.

- Computation

Example: the total amount is defined as product price plus delivery fee.

- Enumeration

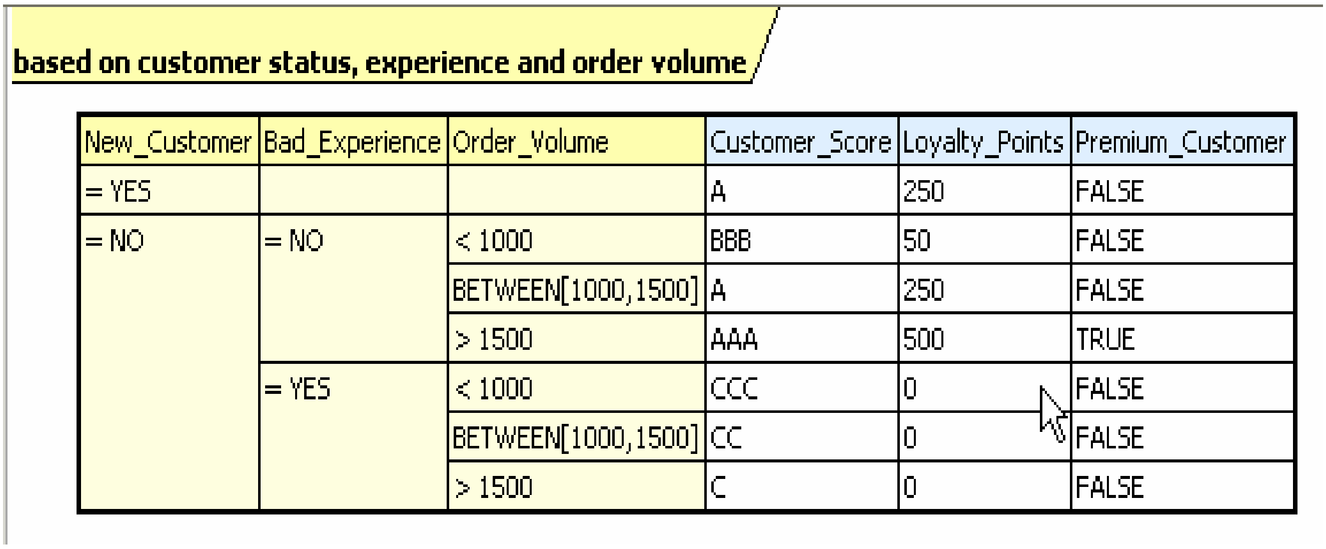
Example: customer standing must be chosen from the following closed enumeration:

- Gold,
- Silver,
- Bronze.

Business rules can also be described by decision table. Figure 2 shows us a good sample of decision table. The columns of Customer Score, Loyalty Point and Premium Customer are decided according to the values of following attributes: New Customer, Bad Experience and Order Volume.

For instance, the fifth line in figure 2 can be explained by the following three natural

language rules.
R1: A customer is defined as premium customer if all of the following are true:
-the customer is not a new customer,
-the customer has no bad experience,
-the customer has spent more than 1500$.
R2: A customer should be given 'AAA' score if all of the following are true:
-the customer is not a new customer,
-the customer has no bad experience,
-the customer has spent more than 1500$.
R3: A customer should be given 500 points for customer loyalty if all of the following are true:
-the customer is not a new customer,
-the customer has no bad experience,
-the customer has spent more than 1500$.
As we can see, if we need to generate rules to deduce some attributes from some other attributes, a decision table will be a wise choice.



based on customer status, experience and order volume

| New_Customer | Bad_Experience | Order_Volume | Customer_Score | Loyalty_Points | Premium_Customer |
|---|---|---|---|---|---|
| = YES | | | A | 250 | FALSE |
| = NO | = NO | < 1000 | BBB | 50 | FALSE |
| | | BETWEEN[1000,1500] | A | 250 | FALSE |
| | | > 1500 | AAA | 500 | TRUE |
| | = YES | < 1000 | CCC | 0 | FALSE |
| | | BETWEEN[1000,1500] | CC | 0 | FALSE |
| | | > 1500 | C | 0 | FALSE |

Figure 2.Decision tables, Source: Demo of Modeling of decision tables with Visual Rules, available at: http://www.visual-rules.com.

### 2.3 Life cycle of business rules

According to Bajec & Krisper (2004), business rules can be realized by following a BR life cycle: business rule discovery, analysis & classification, validation, modeling and implementation. Implementation part will be introduced later in our paper. Now we would like to introduce some theories relating to BR capture.

#### 2.3.1 Capturing business rules

Bajec and Krisper (2004) believe the analysis of Enterprise Model (EM) should be the first step of BR capture, no matter BR is captured from legacy system (or called existing system) or required by a new application development. Business rules are used to describe how business works, so we need to learn EM as the basis for BR capture. Comparing with capturing BR for new system, externalizing rules from the existing system may be more complex (X. Wang & J. Sun, 2004)

Figure 3 is the enterprise model developed by Bajec and Krisper (2004). This model comprises five interactive sub-models:

- Business Process Model: express the processes which will be used to support the achievement of business goals.
- Business Vision Model: describe the overall business strategy, and analyze business goals, goal structure and the problems which must be resolved in order to realize business goals.
- Resources and Actors Model: mainly describe the resources enterprise owns, the structures of resources and the actors who interact with resources
- Conception Model: establish a collection of the conceptions by which the business environment (e.g. products and information resources) is described
- Business Rule Model: capture and maintain business rules, no matter whether the rules are formulated explicitly or existing implicitly in other models.

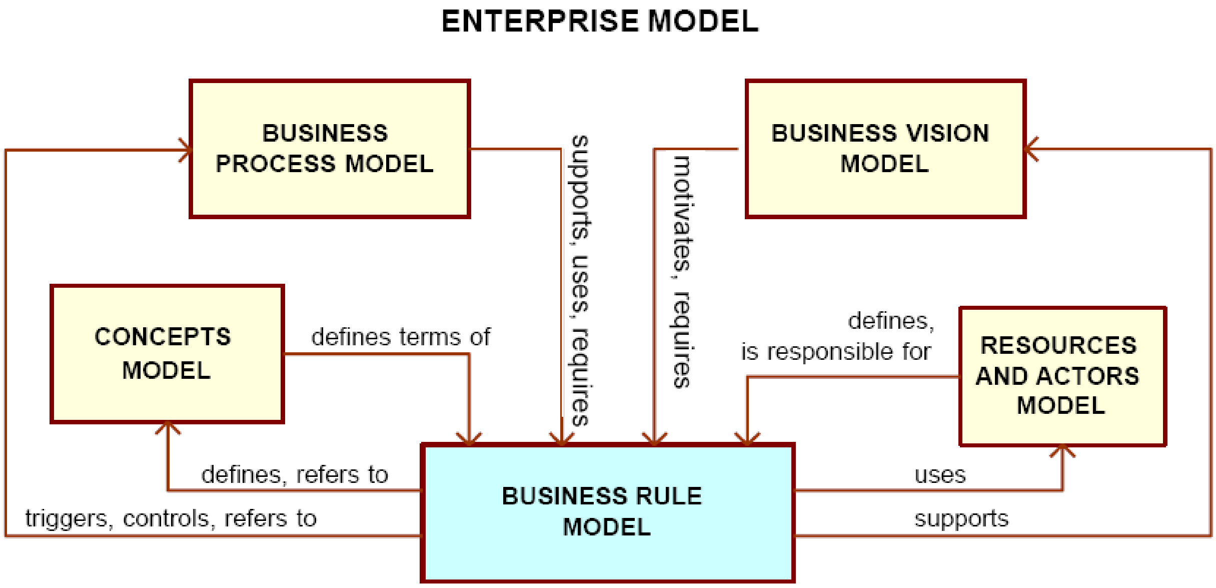


*Figure 3.Enterprise Model, Source: Bajec& Krisper (2004): A methodology and tool support for managing business rules in organizations, p.428.*

As we can see from figure 3, the five sub-models are not isolated, but interactive with each other. Besides business rule model, the analyses of other four sub-models will be helpful for business rules capture. The relationships between business rule model and other sub-models are listed as following (Bajec and Krisper, 2004):

- Business rule - Business vision model
  Business rules will support the realization of business goals, while business goals could be a source of business rules.

- Business rule- Business process model
  Business rules can be extracted from business process model, and will support the operations of business processes.
- Business Rule-Concepts model
  Business rules can define some concepts, and also need to be described according to the vocabulary in concepts model.
- Business Rule –Resources and actors model
  Business Rules are handled by actors in order to take full advantage of enterprise resources. Analysis of resources and actor model will be helpful for learning how business is conducted and discovering business rules.

### 2.3.2 Analyzing & classification, validation and modeling

After the discovery, business rules need to be analyzed and classified. According to P.J. Layzell and P. Loucopoulos (1988), the purpose of this activity is to make sure each rule is atomic, belongs to exactly one category, and is formally or semi-formally described using a predefined rule language.

Consistency and conflict validation is the next step for the classified business rules. A rule can not exist separately, but has relationships with some other rules. Sometimes a rule can be an exception of another rule or conflict with another rule (Ronald G. Ross, 2003). We need to eliminate conflict and make a rule consistent with others at validation step.

After that, business rules need to be represented in a readable way, such as English-like natural language, decision table, decision tree and activity diagram (Bajec & Krisper, 2004).

### 2.3.3 Implementing of business rules

Finally, we need to implement business rules. There are some alternative choices for BR implementation, and we need to select and realize a candidate one which can make BR maintenance easier in future.

## 2.4 business rule tool

### 2.4.1 Business rule component

According to Morgan (2002), a rule component is dedicated to implementing one or more specific rule sets. Simply speaking, business rule component is a component which is applied to handle business rules via the interaction with other components (e.g. Java application). Prolog (Programming in Logic) and Clips (C Language Integrated Production System) are two kinds of languages, which are used to code with business rule component. Both of them are logic programming language.

Business rules implemented in BR component are coded in logic programming language, which is much easier than conventional programming language (e.g. Java, C#) and can be easily learnt by both IT staff and business analysts. BR component can interact with other applications (e.g. java application) via its interface, and thus “outsource” business rules from ordinary application codes.

The following is a simple sample of Prolog language. The first and second lines are two facts. The first line means Eva is a child of Anna. And the second line means Anna is a woman. The third line is a rule. The right part of the rule is equivalent to the condition in “IF-THEN” logic, while the left part is the result. It means X is the mother of Y, if Y is a child of X, and X is a woman. As we can see, Prolog language is made up of two elements: fact and rule. It is a much simpler language comparing with other programming languages.

Sample:

```
child (eva, anna).
woman (anna).
mother (X, Y) :- child (Y, X), woman (X).
```

### 2.4.2 Business Rule Engine

A rules engine, sometimes also known as a logic server, is a specialized component that's designed for the sole purpose of handling rules (Morgan, 2002). Commercial rules engines usually express rules in a proprietary English-like language (Drool rules engine white paper), so it enables business experts to change business rules dynamically without the help of IT professionals.

The core of a rule engine is to manage the collection of working memory and evaluate business rules according to its algorithm. According to current working memory, some rules are able to be executed and are put into a list. Then a rule engine will evaluate rules in the list according to its internal control mechanism and algorithm, and will generate an execution sequence. Because of the execution of rules sequence, new facts will be generated and the working memory needs to be updated. Consequently, some rules will be removed from the list and new rules need to be added into the list.

Figure 4 displays main elements of a typical business rule engine. Source rules are initially compiled by compiler, and stored “in a data structure” (Morgan, 2000). Inference mechanism can identify the data structure, and can locate and identify business rules at runtime. The control mechanism can used to evaluate business rules in the list, which can be executed based on working memory. By using the interface, business rule engine can interact with other processes or applications. Business rule engines based on different algorithms may have difference in rules execution efficiency.

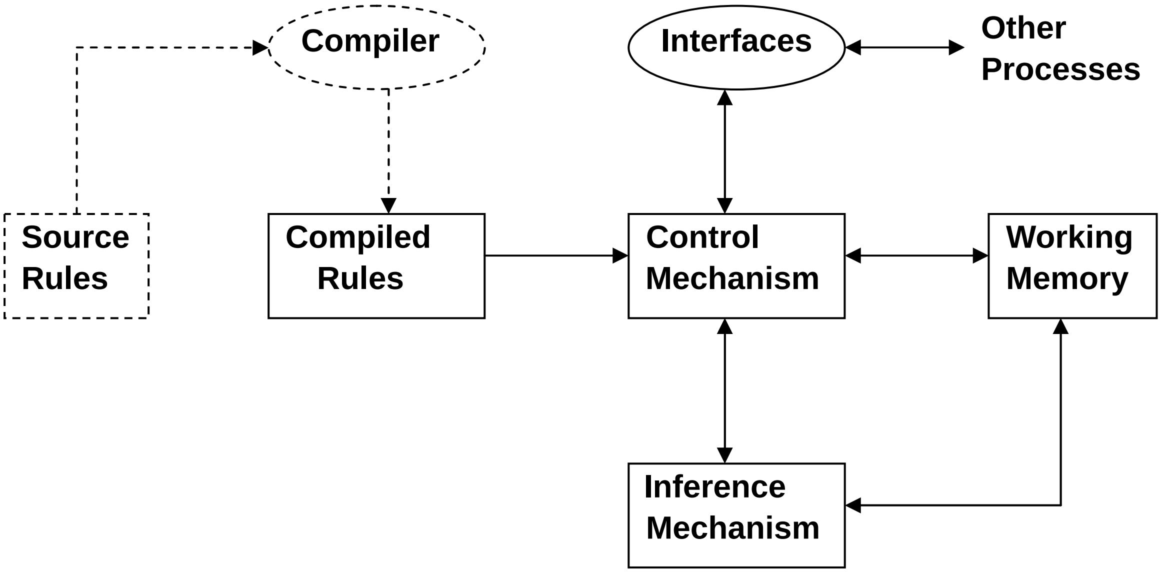


*Figure 4.Main elements of a typical rules engine, Source: Morgan, T (2002): Business rules and information system: aligning IT with business goals.*

### 2.4.3 Business Rule Management System

According to James Taylor, Business Rule Management System (BRMS) is a software system used to manage and support the business rules of an organization or enterprise. This concept of BRMS sounds similar to the definition of business rule engine. Both BR engine and BRMS can separate business rules from conventional programming language, can represent BR in natural language, and can enable business analysts to define and manage BR without the help of IT staff. The main difference between BR engine and BRMS is that BRMS has a centralized rule repository,

The structure of a BRMS can be represented in figure 5. In rule development environment, rules stored in rule repository can be combined into rule-sets to support functions realization. And rule-sets can be combined into rule-flows which arrange rule-sets in an order to achieve business decisions. Decision management service will use rule engine to deal with the inputs from operational application and data base, and generate outputs.

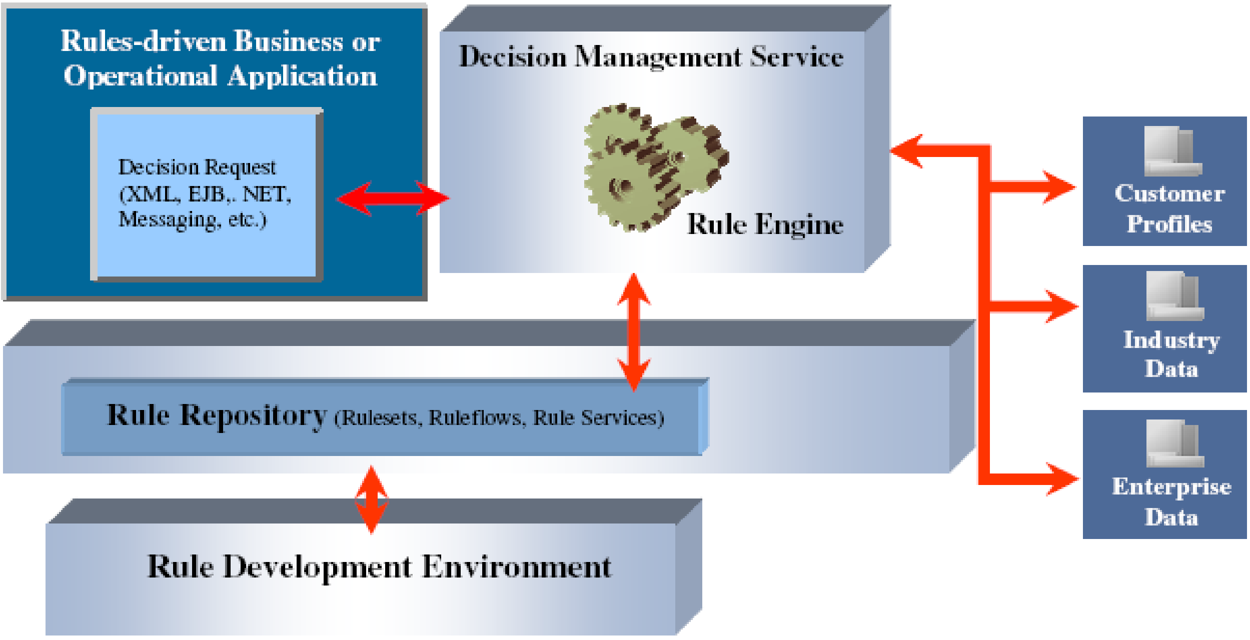

Rules-driven Business or Operational Application
Decision Request (XML, EJB,. NET, Messaging, etc.)
Decision Management Service
Rule Engine
Rule Repository (Rulesets, Ruleflows, Rule Services)
Rule Development Environment
Customer Profiles
Industry Data
Enterprise Data

*Figure 5.BRMS components integrate external applications and data with rules management functions. Source: James Taylor, Achieving Decision Consistency across the SOA-Based Enterprise Using Business Rules Management Systems p.752*

Nowadays, there are some commercial BRMS in the market, such as Jess, ILOG's JRules (the Java version) and Rules (the C/C++ version) and Fair Isaac's Blaze Advisor, JBoss Drools.

### 2.5 implementing of business rules

In this part, we will introduce the theories relating to business rule implementation. How and where to implement BR in computer networks (including Internet) are the topics for our study.

#### 2.5.1 Where to implement business

Morgan (2002) introduced Microsoft's high level version of a Distributed Internet Architecture (DNA). Since it was published by Microsoft Corporation in 1997, DNA has been applied by a growing number of software developers,

DNA is three-tier architecture, listing as following:

- Presentation Layer

In business-to-customer environment, a front end is needed in presentation layer to interact with clients, and handle HTML, Active Server Page (ASP), authorized visits and so on. The front ends in B2C are usually web browsers. Business rules implemented in this layer can support quick interactions with clients.

- Business Logic Layer

Business logic layer is the tier where the bulk of business processes happen (Morgan, 2002). Majority of business processes will be operated in this layer, so this layer may be considered first when someone is trying to implement business rules. In the traditional way, software developers are used to coding business rules into various applications in this layer. But business rule implemented in coding manner is really hard to be updated. It is not very common for software developers to document which parts of their codes are compiled for realizing business rules. Even though Marko Bajec, Marjan Krisper (2004) developed a methodology for managing business rules in organization, it will bring bigger cost and longer developing time if BR is managed by developers manually. And from business perspective, the mobility of IT staff is another factor obstructing the maintenance of BR in coding way. In recent years, people are attracted by the obvious advantages of BR component (e.g. Prolog), BR engine and BRMS, and are prone to applied them in business logic layer.

- Data Base Layer

Some rules relating to constraints to data value, structure and consistency are implemented in data base layer. This can make sure the data will be checked before entering into the data base, and avoid unwanted data transmission.

### 2.5.2 How to implement business rules

Business rules can be implemented in several manners, listing as following:

- Programming code – implement business rules by coding in general programming languages, such as Java and C#. Both online coding (e.g. in ASP) and application coding can be used to implement business rules, but they may be a little different because of different locations of BR implementation.
- Scripts – implement business rules in scripts, which will separate BR from ordinary programming codes. In this way, we can change business rules by updating scripts, without influencing other application codes. Two kinds of typical script languages are VB script and Java script.
- Data base constraints – implement business rules in constraints of data base. Three kinds of constraints are listed in the following:

  Constraints in data value – for instance, R1: a customer mush at least 18 years old. This kind of rule can be realized by domain constraint for an attribute (e.g. age here) or by “CHECK” sentence in table definition.

  Constraints in data structure – constraints in relationships between entities can be expressed in UML diagram and implemented by deriving rational tables from UML diagram.

  Constraints in referential integrity – implement by foreign keys. .
- Stored procedures – implement rules in stored procedures of data base. It can make business rules implemented close to where data is stored, so rules relating to data or information control are recommended to implement by stored procedures.

- Triggers – implement business rules in database triggers, which is similar to stored procedures in functions. The difference is that a trigger can automatically occur once it is triggered.
- BR component – implement business rules in BR component, such as Prolog and Clips.
- BR engine – implement business rules in BR engine
- BRMS – implement business rules in BRMS

### 2.5.3 Implementation choice of business rules

It is not very easy to decide where and how to implement business rules because each manner has its pros and cons. But there are some tendencies we want to mention.

- If a business rule is changed very often, such as fifty times per day, it will be wise to put it as on-line coding in presentation layer to offer quick reaction. The rules controlling authorized visits are in the similar situation.
- The rules relating to data constraints are prone to be put close to data base, so that some unwanted data transmissions could be avoided.
- BR component will be a good choice if we want to develop a rule-based information system. As Morgan (2002) described, BR component is a half way house between in-line code and the more generic rules engine. It can separate rules from ordinary programming codes and make it easy to update rules, but it can not completely put rules management into business side.
- BR engine and BRMS are attractive if we have lots of IF-THEN logics to handle. The pros of BR engine and BRMS are obvious: put business rules' definition and management into business side, centralized management of rules and efficient execution. BR engine and BRMS are usually implemented in business logic layer, but sometimes in data base layer (Morgan, 2002). Applying BR engine and BRMS means a big investment, especially for medium- and small-size companies.

Based on above tendencies, a company needs to choice how to implement rules according to their specific situations. Sometimes a mixture of different manners is used.

## 2.6 Distribution and Centralization of business rules

Today corporations run their businesses in a global scale and build up their infrastructures with computer networks, including internet. Distributed information systems or distributed databases may be applied as a result. So people can choose to distribute business rules in more than one place, or choose to centralize rules over the whole corporation.

Both distribution and centralization have pros and cons. The former one can keep a rule close to its source or its actors or the process supported by it. But it is hard to keep rules consistent because the same rule or rule set may exist in several places. The centralized way will make rules updating much easier, but it requires a large amount of investment in the building of a rule repository.

### 2.6.1 Distributing rules to headquarter and subsidiaries

Theodoulidis and Youdeowei (1998) described a scenario which can be easily imagined: a corporation has it headquarter in location A and two subsidiaries in location B and C, as shown in figure 6. In this situation, they distribute business rules into three places, and apply a business rule engine in each location.

According to their model, the common business rules applied by all locations are only implemented in headquarter (location A). Other locations (location B and C) need to access the rule engine at location A to evaluate common rules. Meanwhile, location B and C are allowed to capture some new rules according to local environments and implement them at local locations.

Generally speaking, common rules are managed by headquarter and specific rules at one location is managed by local subsidiary. In this way, consistent rules can be achieved and executed in high efficiency.

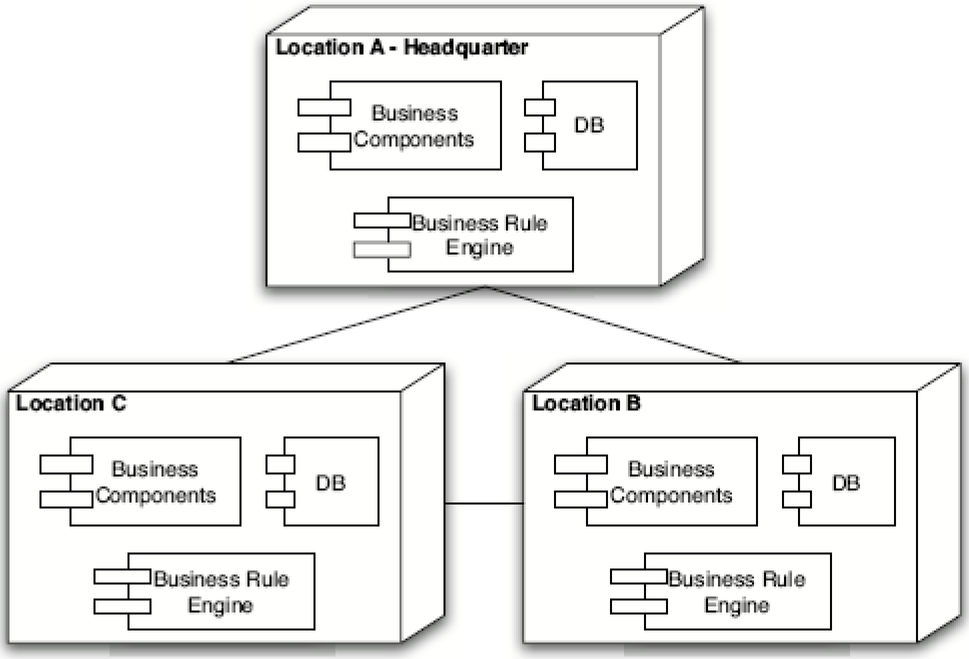


*Figure 6.Distributed business rule scenario, Source: Babis Theodoulidis & Anthony Youdeowei (1998): Business Rules: Towards Effective Information Systems Development*

### 2.6.2 Implementing business rules in SOA

Service Oriented Architecture (SOA) is a computer systems architectural style for creating and using business processes, packaged as services, throughout their lifecycle (http://en.wikipedia.org/wiki/Service-oriented_architecture). SOA separates functions into distinct units (services) (Bell, Michael, 2008), so people could change a service or plug-in a new service without influencing others.

Figure 7 shows us how to make a kind of top-down decomposition in SOA style. Enterprise business process is initially decomposed into several business services. Each service is implemented by a business process which can be recursively decomposed. The leaf nodes are various business components, so enterprise business process is realized by the co-operations of a set of business components (Lublinsky & Didier Le Tien, 2007).

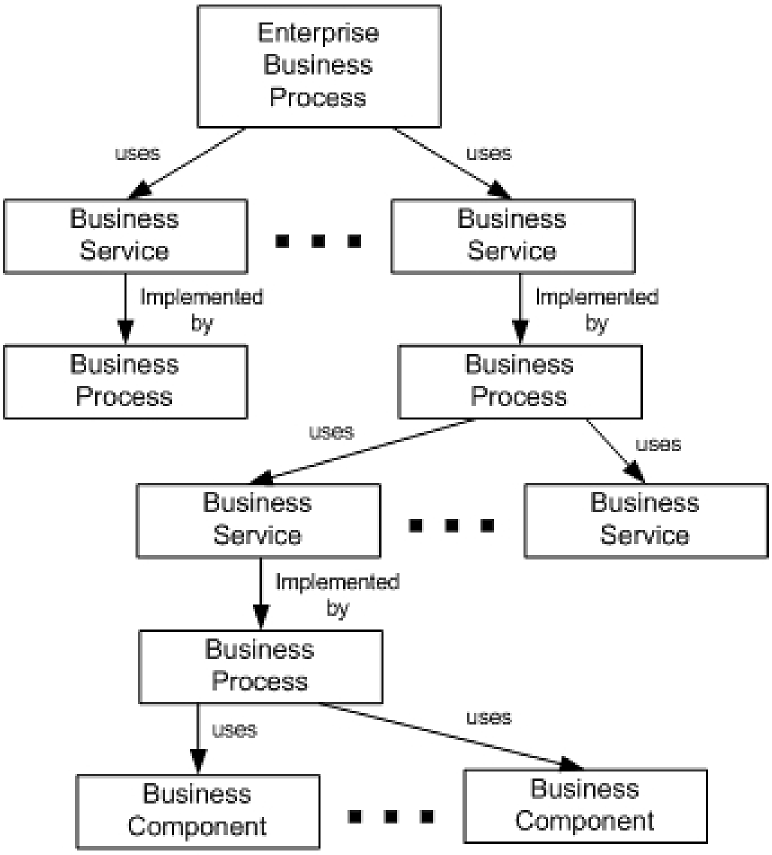


*Figure 7.Hierarchical decomposition of SOA. Source: Boris Lublinsky & Didier Le Tien (2007), Implementation of business rules and business processes in SOA*

There are two options for business rules implementation in SOA. One solution is to adopt BR component (e.g. Prolog) as a kind of business components which are the leaf notes in figure 6. So each top level service-based application is implemented by the cooperation of a BR component with other components. The business rules are distributed into several top-level services. Another choice is to centralized business rules by BR engine or BRMS. The BR engine or BRMS is used as a distinct top-level business service, which needs to be accessed by other top-level services to execute rules.

### 2.7 Scenario-based Software Architecture Evaluation Methods

In this part, we will introduce some theories in scenario-based software architecture evaluation. Our evaluation methodology is based on these theories.

According to Rick Kazman & Gregory Abowd (1996), software architecture (SA) describes a high-level configuration of components that compose the system, and the connections between the components. Software architecture can be described by a diagram which is made up of the components and their connections. In the process of system development, sometimes several candidate architectures are available. So software developers or architects need to evaluate all candidate architectures in order to make a decision about which architecture is better. The evaluation of architecture should help architects find out which part of the architecture needs to be completed.

The architecture of a system should support the achievement of both functional and non-functional quality attributes (Rick Kazman & Mark Klein, 2000). So the scenario-based evaluation of architecture should be conducted with respect to software quality. Some methods have been developed for scenario-based software architecture evaluation, such as SAAM (Software Architecture Analysis Method), ATAM (Architecture Trade-off Analysis Method) and CBAM (Cost Benefit Analysis Method). We will introduce SAAM and CBAM in the following.

#### 2.7. 1 Software Architecture Analysis Method (SAAM)

The SAAM first appeared in 1993 (Rick Kazman, 1994). It is the first developed method for scenario-based architecture analysis. SAAM was initially designed to evaluate whether candidate architectures can support modifiability. But now it is expanded to be able to evaluate any other attribute. So at the beginning of SAAM, people need to select one quality attribute as evaluation standard.

Figure 8 displays the five steps of SAAM and the connections between the steps.

- Step 1: architecture description

The candidate architectures need to be described in a well-understood way. Architecture diagram, which is made up of the components and connections between components, is a good means to describe architecture.

- Step 2: Scenario development

The scenarios need to be developed to capture the significant uses of a system.

- Step 3: Individual scenario evaluation

For each CA (candidate architecture), all scenarios should be evaluated. If a scenario can not be realized by a CA, the scenario is identified as indirect scenario. Otherwise, the satisfied scenarios are called direct scenario. For each indirect scenario, some changes in CA need to be identified and recorded. Making a change here means either modifying an existing component (or connection) or introducing a new component (or connection).

- Step 4: assess scenario interaction

If several indirect scenarios require a modification in the same component, we say the component cause the interaction of these indirect scenarios. According to Rick Kazman & Gregory Abowd (1996), scenario interaction measures the extent to which the architecture supports an appropriate separation of concerns.

- Step 5: Overall evaluation

In this step, people need to make a subjective evaluation for the overall ranking of architectures. They weight all scenarios and scenario interactions according to the relative importance, and make a decision for overall results.

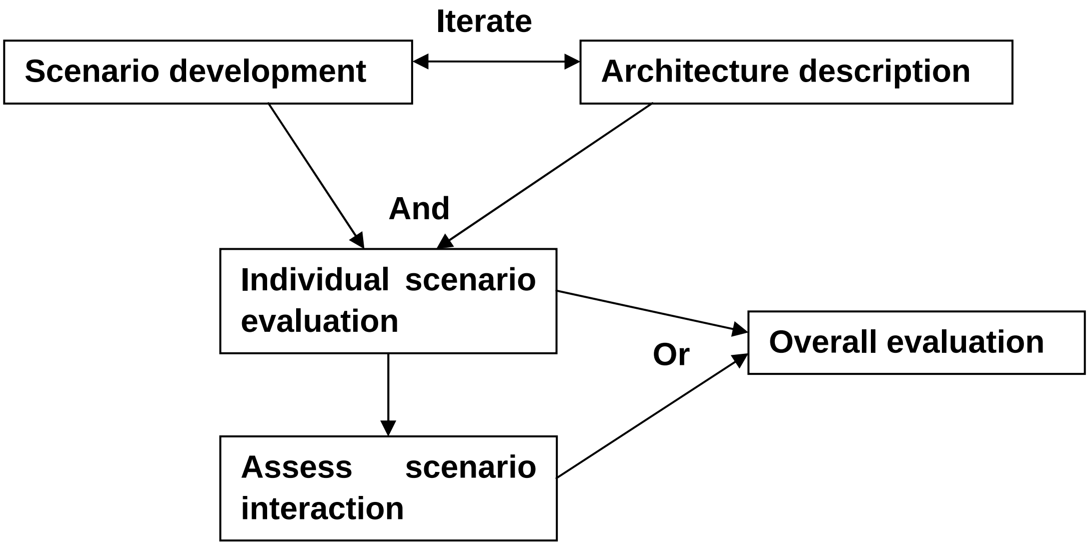


*Figure 8 Steps of SAAM, Source: Rick Kazman & Gregory Abowd (1996), Scenario-Based Analysis of Software Architecture*

### 2.7.2 Cost Benefit Analysis Method (CBAM)

The CBAM was developed as a complement of former scenario-based architecture evaluation methods (e.g. SAAM and ATAM). Different from former methods, CBAM is adding the costs (and implicit budgets or money) as quality attributes (Mugurel Ionita & Dieter K. Hammer), and taking economic factors into account. After the process of SAAM, we find some components (or connections between components) are required to be modified in order to realize indirect scenarios. Making a change on a component can be considered as an architecture strategy (AS). So some architecture strategies are generated as the result of SAAM. CBAM offers us a method to evaluate all architecture strategies. The value of ROI (return on investment) is calculated for each AS. So we can rank the architecture strategies according to the values of ROIs.

Figure 9 shows us the context of CBAM. Business goals of a corporation can affect the architecture strategies (AS) forwarded by architects. The adoption of each AS can promote the achievement of software quality (such as performance, Modifiability and so on), and thus generate benefit. As the same time, a cost will be caused by each AS.

The core of CBAM is to make a trade-off between benefit and cost for each architecture strategy. So ROI is calculated for each AS, and the AS with a higher ROI will be given a relatively higher priority.

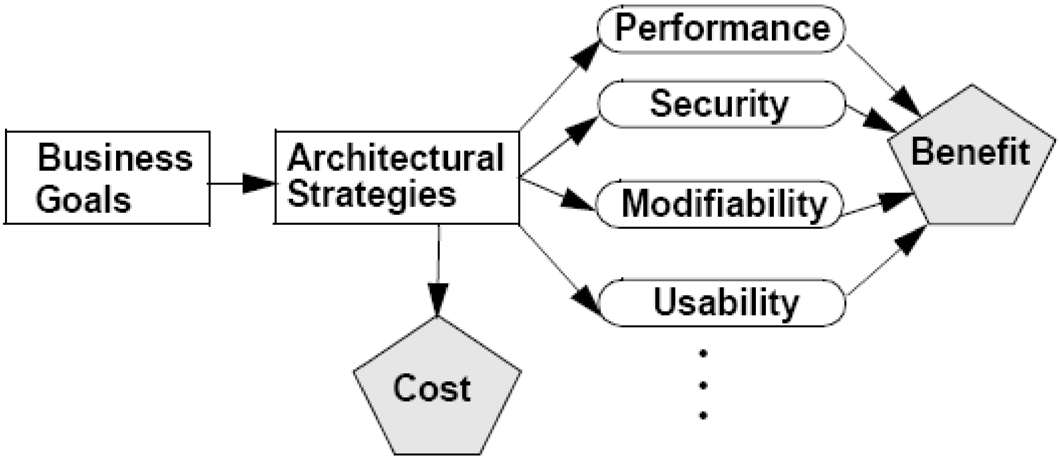


*Figure 9 Context of the CBAM, Source: Rick Kazman & Jai Asundi (2002): Making Architecture Design Decisions: An Economic Approach, Software Engineering Institute.*

## 3. Method

This chapter focuses on the selection of our research approach. The chosen method is followed by how we prepare and conduct our research. And at last we discussed the different variables that influence our research quality.

### 3.1 Research Strategy

There are two different traditional research methods used in science research field: a quantitative and qualitative approach both aim to explaining the complex social phenomena. In our research we chose the qualitative method to understanding our research phenomena via interacting with the experienced respondents. Qualitative research can be conducted when we need a detailed complex understanding of the issues (Creswell 2006).

Plentiful literature review concerning about our area of interests is useful in forming our theory framework. Yin (2003) states it is inevitable to do literature review before a well research question established. This theory framework is also a foundation of our evaluation methodology and is very helpful to creating our interview questions in the after.

Our thesis pays more attention to the *evaluation methodology* we formulated according to our purpose "evaluate implementation choices of business rules". So after the evaluation methodology be formulated, we selected interview as our research approach in order to get some useful feedback, we interacted with some experienced persons, listen to them subjective opinions in our evaluation methodology, and we can also check whether our methodology is clear enough to understand.

### 3.2 Data Collection

As Briony (2006) proposed, when a researcher wants to ask open-ended or complex questions, and wants to achieve detailed information that cannot through pre-defined questionnaire responses or easily be observed, the interview can be deemed to the suitable date collection approach in those situations. The purpose of our research is acquiring empirical information related to our evaluation methodology from certain experienced persons, so that we chose the interview as our main data gathering technique in order to get high-quality feedbacks and also obtain potential intangible information.

Different data collection motivation decides different degree of interview structures. In our research we selected semi-structure interviews, which we have a list of major questions we generated according to our research purpose, but we can change the sequence of the questions during the interview process according to the conversation environment, and we can also ask some additional questions if the interviewees come up with some new issues or ideas that we are not prepared (Briony 2006). As Kvale (1996) described that it will be useful if we taking two guides into account when preparing the interview, one is preparing the primary research questions of the project, another is forming the questions during the interview, both thematic and dynamic dimensions should be thinking over. The advantages of applying this semi-structure interview in our research is that the respondents can not only respond to our questions, but also can express what they think about our evaluation methodology. We allow the interviewee to speak freely during our interview process and make sure we can obtain the information we are searching for.

.

### 3.2.1 Interview guide generation

An interview guide should be created before operating the interview, “interview guide indicates the topics and their sequence in the interview” (Kvale, 1996). Our semi-structure interview guide consist of the main questions we chosen for our research purpose, the respondents talked freely for each question so some related questions might occur during the interview process. It directs us to investigate our research and can be considered as the foundation of the following analysis.

We generate our main questions from three aspects, the individual experiences, evaluation methodology process and future. The questions begin with the individual experience part, which pays attention to the interviewees’ former working experience. Because the people who have different working experience and pre-knowledge may generate different answers to the questions, we need to select interviewees carefully. The second part is the major section of our interview, and focuses on how the interviewees respond to the questions relating to the evaluation methodology process. Most of the new questions will produced during the second part according to their response. The final part is asking about the further information the interviewees would like to talk about. We create the interview guide in appendix A and interview protocol in appendix B.

### 3.2.2 Interviewee selection

After constructing the interview guide, selecting the interviewee become an important step. In order to achieve our thesis purpose in the first chapter “evaluating the BR implementation choice”, so the first precondition of interviewees is that they should be familiar with business rules or at least knows what business rules are. The purpose of the interview is to collect people’s feedbacks on our methodology. The methodology is based on the scenario-based software evaluation method, so it is better if the interviewees have certain knowledge and experience about applying these evaluation methods, or at least they have several years experience of system developing and knows how to evaluating the software architecture in the process of software development.

We searched hard and trying to find the appropriate person who can help us to do this interview. Finally, we found two classmates who are familiar with business rules and have several years working experience in system development. We also found another suitable person coming from the top telecommunication network solution company in China. They all have been working for more than 4 years in software development and have experience to evaluate whether a computing project like software development is good or not, also they have certain knowledge of Business rules. Table 2 shows the detailed information of these interviewees.

**Table 2 Interviews**

| Interviewee | Interviewees' Name | Career Experience | Years of Working |
|---|---|---|---|
| Interview A | Intayoad Wacharawan | Software Engineer in Locus Telecommunication Inc., Bangkok, Thailand<br>Intayoad mainly in charge of software project or module requirement analysis, coding, test and also take part in the evaluation process of software quality. | 5 years |
| Interview B | Cui Li | Advanced Software Engineer in Shenzhen Huawei Technology Co., Ltd.<br>Cui with responsibility for coordinating software requirement, design, developing and test at different operation system and system platform in mobile phone industry. | 6 years |
| Interview C | Chen Xiao | IT System Management Engineer in Shanghai International port group Co., Ltd.<br>Chen take charge of maintain and manage system platform, database or website. | 4 years |

### 3.2.3 Interviewer Conducting

Before we do our interview, we gathered the background information of our interviewees, and also provided them some information about our evaluation methodology in ahead in order to make them have enough time to think about our topics from their perspectives (Briony 2006). Because our mother language is not English, so it is better for us to choose the face-to-face interview rather than telephone interview. As Kvale (1996) states the face-to-face interview is important to achieve contextual information which may be not easy to record such as tone of voice, body language and so on. The interviews with the interviewee A and C are conduct by face-to-face approach, while the interview with interviewee B is accomplished through telephone because of the distance restriction between Cui Li and us.

Our interview began with an introduction of our evaluation methodology and some related background knowledge, we explained each step of the evaluation process and our evaluation purpose. We controlled our time between 50 to 70 minutes to ensure our interviewee not feel too much time consuming, and tried our best to make the respondents feel comfortable and free.

### 3.2.4 Instrumentation

We recorded the whole interview process via audio equipment after we got permission of our interviewee. These were helpful for our following analysis and discuss because it is hard for us to remember all what the respondents answer. Sometimes we can not remember the interviewee’s answer to some question, as Briony (2006) said “our memories are unreliable and prone to bias and error”. So a recorder was used to assist us in remembering what the interviewees said during the interview process. After the interviews, we transcribed it into writing documents, which is easy for us to analyze and extract data.

## 3.3 Data analysis

After we collecting the data from the interview, the following step is to analyze it. It is obvious that the writing document we got after transcribing is a little intricate. We found one analysis method is suitable for our study, which categorizes the data into different categories so that the plentiful data can be reduced and condensed. We categorized the data from mainly three perspectives, individual experiences: evaluation methodology process and future. The evaluation methodology process can be divided into several sub-categories according to the responses of the interviewees for every step. After the process of categorizing, the information of each category can be summarized, so that the information used for the final analysis and discuss will be much clearer.

## 3.4 Research quality

As researcher it is important for us to justify if the research is meaningful, reliability and validity are the two common criterions to evaluating the research quality.

### 3.4.1 Reliability

Reliability responds the consistency findings of the research (Kvale 1996). Kvale also states “though increasing the reliability of the interview findings is desirable in order to counteract haphazard subjectivity." For example, we doing the same project, often we are trying to ask the same questions but we use different wordings, which may lead the interviewee producing the different comprehension and response the research answers. Therefore we must agree on certain reliability during our interview. The reliability is also valuable when we transfer our recording to the writing documentary. Kvale (1996) recommend it is better to have two persons to check the same recording independently and compare the different understanding. On our research we listen the recording independently and discussed our comprehension of each answer finally manage to consistency.

### 3.4.2 Validity

Kvale (1996) describes validity is "a method investigates what it is intended to investigate." In our study, the interview validity are mainly involves if we can correctly understand what the interviewee mean. As Kvale (1996) states when choose the transcript's linguistic style it is also important to achieve valid translation from the oral to written documentation. Therefore, we checked the text with our interviewee after transcribing. We send them our transcription of their opinion, if some parts of the transcription they do not agree, we will communicate with them and modify the transcription.

## 3.5 Ethics

Ethical issue is the important factor need to be considering about for acquire faithful relationship with interviewees. As Israel (2006) mentioned in his book, ethical behavior helps protect individuals, communities and environments, and offers the potential to increase the sum of good in the world. As responsible researchers, we need to make sure our behaviors are correct in moral perspective and what we do will not harm participants in the process of research or after the final report is published.

Informed consent is one of the ethical guidelines that Kvale (1996) advocates. For our research, informed consents include the whole purpose of the research. Our interviewees will clearly understand our research topics and the involved information, for instance, who is the researcher, how long is the interview will take, how we will use their data and how to disseminating the research findings (Briony 2006). Meanwhile the interviewees have right to not participant our interview and can withdraw from our interview at any time they want. In addition, we were considering our interview carefully in order to avoid the unnecessarily intruding into our interviewee's activities.

## 3.6 Bias

In order to ensure our study is high quality we also need to be thinking over the *bias*. Harmmersley and Gomm (1997) describes that bias is deem to a positive feature, it reveals important aspects of phenomena that are hidden from other perspectives. Bias might arises from different person’s subjective judgment. Business owns and system developers may have different point of view when they design the scenarios for evaluating the BR-based systems. Certain bias can provide profitable information, for instance, when bias happened during the evaluation process, it force us to thinking over the evaluation process more carefully and avoid miss important standpoint.

## 4. A methodology of evaluating BR implementation

In this chapter, we will introduce a methodology of evaluating business rules implementation. This methodology is derived from the theories in scenario-based software architecture analysis methods, and is formulated to adapt to the business rule field.

### 4.1 Motivation of developing this methodology

As we introduced in chapter 2, the implementation of business rules is the last step before business rules maintenance, and is hard to control because the following two reasons:

Firstly, business rules can be implemented in several places in three-tier architecture which is commonly applied by B2C companies, and in several different manners. So there are too many alternative choices when people want to decide where and how to implement BR.

Secondly, architects make decisions for business rule implementation according to their experience, and there is no formal guidance in BR implementation decisions. Some vague tendencies for business rules implementation exist. For example, the rules need to be changed frequently are prone to be implemented by in-line codes in client side; rules relating to data control are recommended to implement in data layer to avoid unwanted data transmission; majority of rules are implemented in business logic layer. But these tendencies are not enough for making BR implementation choices in such a complex computer-networked environment.

So we need to develop a methodology to help people evaluate candidate choices for BR implementation.

Another motivation is that the implementation choices of business rules will affect the achievement of software quality. Software quality here means how well software is designed to realize the functionality relating to business rules. We pick out performance and modifiability as two of the most important software quality attributes, listing as following:

- Performance – from client perspective, individual customers in B2C commerce require a quick response from systems and corporations, so business rules need to be checked and executed efficiently. The same request is forwarded from corporation perspective.
- Modifiability – from business perspective, business rules should be easily modified. Modifiability, like other software quality attributes, can be divided into more detailed items, such as adding new rules, removing meaningless rules, changing existing rules and restructuring business rules.

Above two quality attributes are not separate from each other, but there are some interdependent relationships between them. Sometimes a modification in a component may trigger reverse effects on them. For instance, removing some rules to client side may contribute to better performance, but lower the ability of BR modification. So trade-off between performance and modifiability is required when evaluating BR implementation.

### 4.2 The purpose of evaluation

At BR implementation stage, people need to choose where to implement business rules and how to, and thus generate different candidate implementation choices. The evaluation is not aimed at deciding which choice is better than others or should be adopted, because there is no absolutely better.

The purpose of developing this methodology is to provide stakeholders of a better formulated method to evaluate all candidate choices. The evaluation results will be shown in front of people to help them learn more about each choice, but no subjective suggestions will be offered. The right to select and drop is assigned to software designer after the process of evaluation.

### 4.3 Strategy of methodology design

Our methodology is a kind of scenario-based software architecture analysis. It is formulated by referring to the theories in scenario-based software architecture analysis methods and specifying into business rules field. Software Architecture Analysis Method (SAAM) and Cost Benefit Analysis Method (CBAM) are two theories we mainly referred to.

#### 4.3.1 Why need software architecture analysis in our methodology?

Rick Kazman and Len Bass (1999) described three perspectives of architecture design of a software system – functionality, structure and allocation of domain function to structure. They said architecture design is to understand the way in which the intended functionality is achieved by the developed system (Rick Kazman, 1999). Mapping from function to structure in a way could generate a candidate Software Architecture (SA).

In our case, the intended functionality of executing business rules is achieved by mapping the same functionality to different structural alternatives. By this mapping process, several candidate architectures are generated and act as one sort of input in our methodology. For instance, the same set of modeled business rules can be implemented via applications coding in business logic layer or via adopting a BRMS in business logic layer. These two different mappings will produce two candidate architectures, which need to be evaluated with respect to software quality.

Generally speaking, the process of identifying candidate choices of BR implementation will generate several candidate architectures. So the evaluation of candidate implementation choices is equivalent to the evaluation of candidate architectures.

### 4.3.2 Why need the selection of software quality attributes?

Architecture is related to software quality based on the following assumption:

> *Software architecture analysis is based upon the assumption that the product's architecture is responsible for a substantial amount of its non-functional qualities, such as portability, modifiability and extendibility. (Mauricio De Simone & Rick Kazman)*

According to above assumption, a good architecture will support the achievement of selected quality attributes (such as performance and modifiability in our study). So if the selected attributes can not be satisfied, we can conclude there is something wrong with the software architecture (SA) and SA needs to be modified.

So before SA analysis, we need to select some quality attributes from non-functional attributes (e.g. modifiability, portability and extendibility) and functional attributes (e.g. performance, availability and reliability). The set of attributes you choose will become the standard of your software architecture analysis.

### 4.3.3 Why base the evaluation on scenarios?

We use scenario-based analysis method in our methodology, because scenarios can offer us good contexts for architecture evaluation. And scenarios can be easily discovered and understood by all stakeholders, even for non IT-professional staff.

Scenarios are defined as a short statement describing an interaction of one of the stakeholders with the system (Rick Kazman & Mark Klein, 2000). Scenarios are like a kind of task description and forwarded by stakeholders based on their daily work or expected working patterns in future. .Because scenarios which are described in natural language can be easily understood by people from different fields, the analysis based on scenarios will become more accessible to all stakeholders, such as business analysts, individual customers, architects, software developers, maintainers and so on.

We assume that “good” candidate architecture (CA) should support the realization of all selected scenarios. If some scenarios can not be supported by a CA, it means the candidate architecture needs to be modified.

## 4.4 Methodology of evaluating BR implementation

### 4.4.1 Description of the methodology

Figure 10 shows the main activities in our methodology. We will introduce this methodology at first to offer a rough outline.

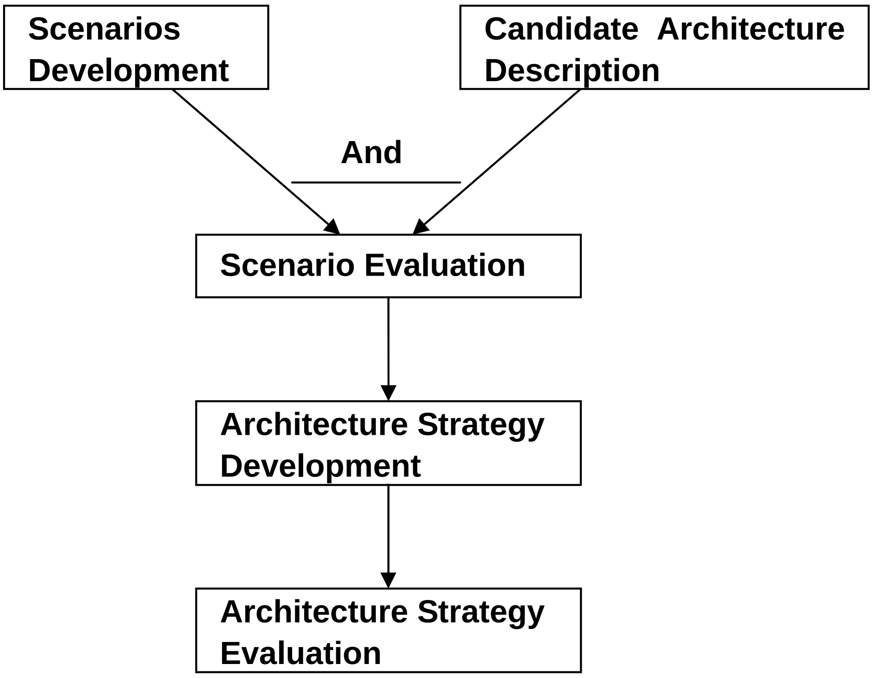

Scenarios Development
Candidate Architecture Description
And
Scenario Evaluation
Architecture Strategy Development
Architecture Strategy Evaluation

*Figure 10 methodology of evaluating BR implementation*

As shown in figure 10, two inputs are needed in our methodology: scenarios and candidate architectures. So the scenarios need to be developed, and the candidate architectures need to be described. Then scenario-based evaluations are conducted for each candidate architecture on the basis of the two inputs. As a result, we may find some scenarios can not be directly realized by candidate architecture and some components need to be modified. Making a change on a component can be identified as an Architecture Strategy (AS). For instance, removing some rules to client side in order to provide quick response to customers can be recognized as a AS. So some architecture strategies are developed by architects in the step of architecture strategy development. Modifying a component will bring some cost to the corporations, and it is nearly impossible for corporations to realize all architecture strategies because of their limited resources. So we need to take architecture strategy evaluation as our last step in order to calculate ROI (Return on Investment) for each architecture strategy and select a subset from all strategies to be realized.

Form above description, you can see how we design our methodology. We take scenario-based architecture analysis as our evaluation method in order to discover which components need to be modified to support all scenarios we select. Because of the limited resources for most corporations, we need to take economic factors into our account. A trade-off between cost and benefit is needed to decide which architecture strategy has higher ROI and should be given a higher priority. The basis concepts in this methodology come from the methods of scenario-based architecture analysis, such as SAAM and CBAM.

### 4.4.2 Steps of the methodology

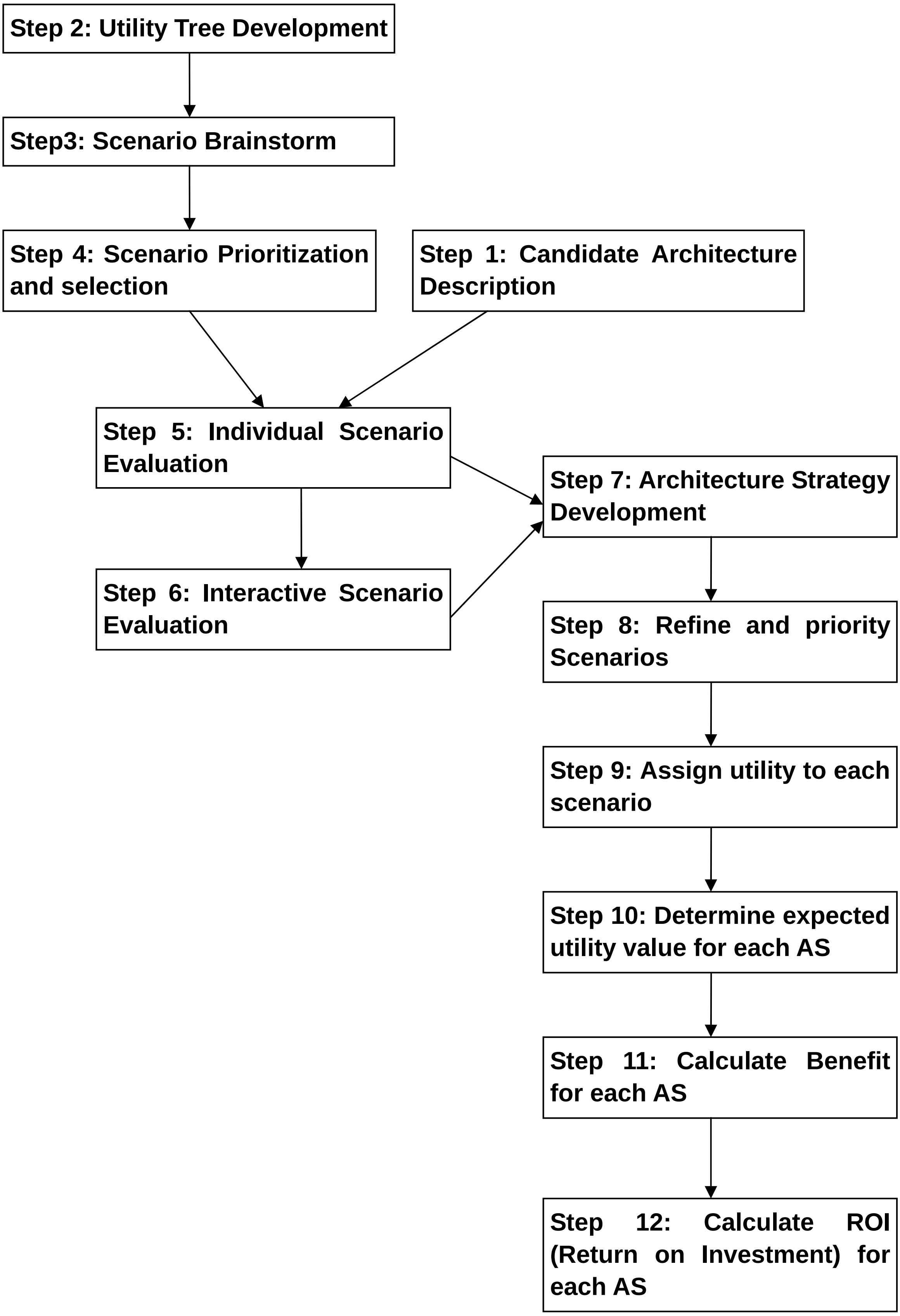

Step 2: Utility Tree Development
Step3: Scenario Brainstorm
Step 4: Scenario Prioritization and selection
Step 1: Candidate Architecture Description
Step 5: Individual Scenario Evaluation
Step 6: Interactive Scenario Evaluation
Step 7: Architecture Strategy Development
Step 8: Refine and priority Scenarios
Step 9: Assign utility to each scenario
Step 10: Determine expected utility value for each AS
Step 11: Calculate Benefit for each AS
Step 12: Calculate ROI (Return on Investment) for each AS

*Figure 11.Steps of our methodology*

Figure 11 shows the steps in our methodology, and we will offer low detail introduction for every step in the following parts. This methodology is derived from the theories in SAAM and CBAM, two kinds of scenario-based architecture analysis. This methodology is applicable to business rules field, and supports our initial ideas for the evaluation of BR implementation.

- Step 1: candidate architecture description

Candidate architectures need to be described in step one as a kind of input in our scenario-based architecture analysis. The architecture is usually described in a common syntactic (e.g. structural) notation (Mauricio De Simone & Rick Kazman, 1995). Architecture diagram is recommended as a kind of graphic description and is made up of components (e.g. computation components and data components) and the connections between components.

In our study, the components within which business rules are implemented should be included in the architecture, as well as some other components which will not directly implement BR, but interact with the components implementing BR. After the process of business rules modeling, BR developers will put forward some candidate choices of BR implementation (where and how) according to their experience and the characteristics of modeled rules. And some candidate architectures will be generated.

This architecture description is conducted by members in the evaluation team (usually 4-5 people) which is formed at the beginning of the evaluation. Architects and system developers are usually involved in the evaluation team and in charge of describing candidate architectures.

- Step2: utility tree development

In our study, we pick out performance and modifiability as our main focuses because the following two reasons: firstly, in B2C rapidly changing environment, business rules are required to be updated easily in order to make the corporations achieve superiorities against their competitors and attract individual customers . Secondly, a large number of individual customers may require completing the same interaction (e.g. search product information and place an order) with the corporation at the same time. So the rules need to be executed efficiently.

But only the descriptions of performance and modifiability are not enough for well understanding the request on software quality. We need to produce ramifications for the two attributes. A unity tree is an effective means to understand what functional or non-functional quality attributes should be included and what are the sub-factors for each attribute. A sample of utility tree is shown as following.

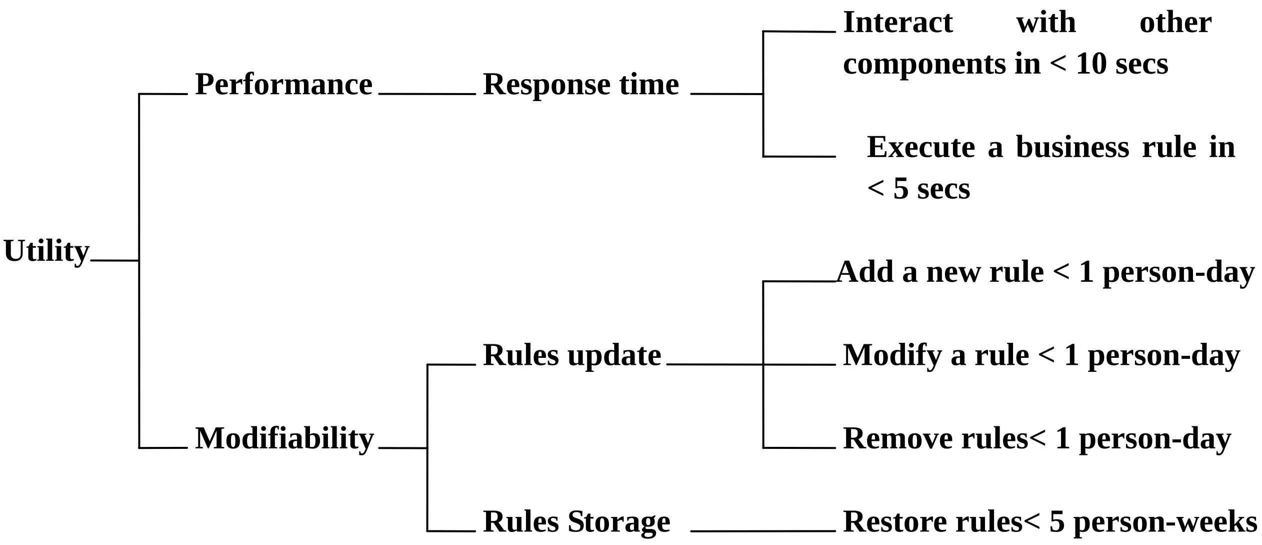

Utility
Performance
Response time
Interact with other components in < 10 secs
Execute a business rule in < 5 secs
Modifiability
Rules update
Add a new rule < 1 person-day
Modify a rule < 1 person-day
Remove rules< 1 person-day
Rules Storage
Restore rules< 5 person-weeks

*Figure 12.Smaple of utility tree*

As seen from figure 12, utility is the root note of utility tree. Performance and modifiability are two quality attributes we select. Then we divide modifiability into two sub-factors, “Rules update” and “Rules storage”. While the “Rules update” is continue to be refined into three sub-factors. After the development of utility tree, the leaf notes will be specific enough to be understood.

The utility tree should be developed at the beginning of the evaluation because it can help people deepen their understandings in quality attributes. The utility tree needs to be offered to all stakeholders before scenarios development because it, as Mauricio Simone & Rick Kazmen (1995) described, is an expression of the overall “good” of the system.

- Step3: scenario brainstorm

In step 3, a brainstorm will be applied to collect scenarios. The attendants are not only coming from the evaluation team, but also from external stakeholders who are not directly involved in the evaluation process, but can also provide some important scenarios. Samples of external stakeholders in our study are business analysts and business owners.

- Step 4：scenario prioritization and selection

In this step, scenarios collected from step 3 are prioritized by voting, and some of the most important scenarios will be selected from all scenarios. In our methodology, we will choose the top 30% of all scenarios according to their importance.

For instance, if we totally collect 30 scenarios from last step, 9 scenarios (30%* 30=9) should be selected as a result. Then each stakeholder will have 9 votes in its hand, and is requested to vote for the nine most important scenarios. By counting the votes for each scenario, the top nine with higher votes will be selected. As a result, some scenarios are picked out by all stakeholders

After scenario brainstorm, lots of scenarios are generated. But some of them are more important, and some others are less important. So we need to make a selection according to vote.

- Step 5: individual scenario evaluation

In this step, we will take the selected scenarios (from step 4) and candidate architectures as two kinds of inputs to evaluate whether candidate architecture (CA) can support the achievement of scenarios. For each CA, the scenarios are divided into two kinds: direct scenario and indirect scenario. The former kind can be directly satisfied by candidate architecture, but the later one can only be satisfied by making some modifications in candidate architecture. The required changes (in the components or the connections between components) need to be described and the total number of changes is also recorded. The results from step 5 should be recorded in a table like table 3.

The thing is not as easy as “the fewer changes the better” and we need to find out a method to evaluate different changes. The method of evaluating changes will be included in later step.

**Table 3 Sample of scenario evaluation (based on Rick Kazman, 1996)**

| Scenario | Description | Direct/Indirect | Change Required | # of changes |
|---|---|---|---|---|
| 1 | Description to scenario1 | Indirect | Require modifications to component A and B | 2 |
| 2 | Description to scenario2 | Direct | | |
| 3 | Description to scenario3 | Indirect | Require modification to component A and C | 2 |

- Step 6: interactive scenario evaluation

In this step, we need to evaluate the interactions of different scenarios. Two or more scenarios are defined as interactive scenarios when they necessitate modifications to the same component or connection between components (Rick Kazman & Gregory Abowd, 1996).

From step 5, we already know which component or components are required to be modified, as shown in the manner of table 3. If two or more scenarios are found to require a change on the same component, they are said to have scenario interaction. According to the samples in table 3, component A needs to be modified in order to support the achievements of scenario 1 and 3. So the two scenarios have interaction via component A.

Simone and Kazman (1995) describe component A in above sample as a potential trouble spot, and consider this kind of spots should be avoided. So the component A needs to be modified or separated into two components to avoid scenario interaction.

The results from scenario interaction evaluation need to be recorded as a table, like table 4. Three columns are included: the component need to be modified, the number of change requests per component and the scenario or scenarios which generate change. If the value of the second column is more than one, it means the change is required by two or more scenarios and scenario interaction happens. The scenarios which necessitate changes are also recorded.

From table 3, we can discover two scenarios (S1 & S3) require change in component A. For B or C, only one scenario requires its change. Every detail can be found in table 4.

**Table 4 Sample of scenario interaction evaluation (based on Rick Kazman, 1996)**

| Component | # of change requests per component | Scenarios |
|---|---|---|
| A | 2 | S1, S3 |
| B | 1 | S1 |
| C | 1 | S3 |

- Step 7: Architecture strategy development

According to the former steps, some changes are required in order to support the satisfaction of all selected scenarios. Each change on a component or connection between components can generate an architecture strategy (AS). So some architecture strategies will be produced after step seven.

Because of the limited resources in each corporation, it is nearly impossible to realize all architecture strategies. So we need to use some method to pick out a sub-set of architecture strategies to realize in reality. The following steps will introduce a ROI-based method to evaluate each AS, and this ROI (return on investment) calculation is the core part of CBAM. CBAM takes cost as an extra attribute of software quality and calculate ROI for each AS.’

- Step 8: Refine and prioritize scenarios

In this step, we need to decide QA (quality attribute) response goals for each scenario, including worst case, best case, current case and desired case. QA will respond to each scenario in the four cases, as we can see from table 5. For instance, scenario 3 is that corporation needs to reduce the number of failing orders. In worst case, 10% of the total number will fail; in best case, no order fails; the fail rates in current and desired cases are 5% and 1% separately (the third line in table 5).

The scenarios are prioritized by giving a weight to each scenario according to their relative importance. Different scenarios are supposed to have different degrees of importance. So all stakeholders should discuss together and give a weight to each scenario after reaching an agreement. In our method, the total weight for all scenarios is 100 (15+25+20+15+25=100), and then a figure should be assigned to each scenario, as shown in the second column in table 5. The bigger the figure is, the more important the scenario is.

**Table 5 Sample of refining and prioritizing scenarios (based on Kazman, 2002)**

| Scenario | Weight | Response Goals | | | |
|---|---|---|---|---|---|
| | | Worst | Current | Desired | Best |
| 1 | 15 | 10% hung | 5% hung | 1% hung | 0% hung |
| 2 | 25 | >5% lost | <1% lost | 0% lost | 0% lost |
| 3 | 20 | 10% fail | 5% fail | 1% fail | 0% fail |
| 4 | 15 | 10% hung | 5% hung | 1% hung | 0% hung |
| 5 | 25 | 10% lost | <1% lost | 0% lost | 0% lost |

The weight of each scenario is applied in the equation of calculating benefit for each architecture strategy, as seen from following equations (Len Bass & Paul Clements, 2005). $\mathrm{B_i}$ is the total benefit for architecture strategy i, while $\mathrm{W_j}$ is the weight for scenario j and $\mathrm{b_{ij}}$ is the benefit of strategy i coming from its affect on scenario j. Here we use utility to calculate $\mathrm{b_{ij}}$, which is the utility of expected value of architecture strategy i minus the utility of current value (Len Bass & Paul Clements, 2005).

$$\mathrm{B_i} = \sum b_{ij} * Wj$$

$$b_{ij} = U_{\exp ected} - U_{current}$$

- Step 9: Assign utility to each scenario

In this step, utility scores for each case in each scenario is estimated by stakeholders and recorded in a table, in a manner like table 6. A consensus on each score needs to be achieved by all stakeholders. The utility with score 0 means there is no utility at all, while the utility with score 100 stands for the most utility. The worst and best cases will be scored at first to set up references. Then current and desired cases which are what we really concentrate on will be scored. We use utilities to estimate the benefit for each strategy, but utilities must be scored according to some scenario.

**Table 6 Sample of assigning utility scores (based on Kazman, 2002)**

| Scenario | Weight | Utility Scores | | | |
|---|---|---|---|---|---|
| | | Worst | Current | Desired | Best |
| 1 | 15 | 10 | 80 | 95 | 100 |
| 2 | 25 | 0 | 70 | 100 | 100 |
| 3 | 20 | 25 | 70 | 100 | 100 |
| 4 | 15 | 10 | 80 | 95 | 100 |
| 5 | 25 | 0 | 70 | 100 | 100 |

- Step 10: Determine expected utility value for each AS

In this step, we need to determine the expected utility value for each architecture strategy.

First of all, we need to list architecture strategies (developed by architects in step 7) and the scenarios affected by ASs in a table, which takes table 7 as a sample. As seen from table 7, an architecture strategy may affect more than one scenario and its expected response to a scenario can be estimated according to the current and desired responses. For instance, AS1 will affect scenario 3 and 5. From table 5, we get the current and desired responses to scenario 3 are ‘5% fail’ and ‘1% fail’ separately. Then AS1’s expected response to scenario 3 is established to be ‘2% fail’.

**Table 7 Sample of AS and scenarios addressed (based on Kazman, 2002)**

| AS | AS Name | Scenarios Affected | Current Response | Expected Response |
|---|---|---|---|---|
| 1 | Order persistence | 3 | 5% fail | 2% fail |
| | | 5 | <1% lost | 2% lost |
| 2 | Order segmentation | 4 | 5% hung | 2% hung |
| 3 | Order reassignment | 1 | 5% hung | 2% hung |
| 4 | Order retry | 4 | 5% hung | 3% hung |
| 5 | Forced order completion | 1 | 5% hung | 3% hung |

Based on the results from table 7, expected utility of an architecture strategy to some scenario will be evaluated by stakeholders, and then reach consensus. The expected value is generated by referring to current and desired utilities. Table 8 is a sample of how to record expected values for each AS. .

**Table 8 Sample of expected utility value (based on Kazman, 2002)**

| AS | AS Name | Scenarios Affected | Current Utility | Expected Utility |
|---|---|---|---|---|
| 1 | Order persistence | 3 | 70 | 90 |
| | | 5 | 70 | 65 |
| 2 | Order segmentation | 4 | 80 | 90 |
| 3 | Order reassignment | 1 | 80 | 92 |
| 4 | Order retry | 4 | 80 | 85 |
| 5 | Forced order completion | 1 | 80 | 87 |

- Step 11: Calculate Benefit for each AS

In this step, we need to calculate total benefit for each architecture strategy. Table 9 is a sample of the results generated. Raw strategy benefit is the result of expected utility minus current utility. Normalized benefit is the result of multiplying raw AS benefit by scenario weight, and it takes the importance of each scenario into account. The total benefit of an architecture strategy is calculated by adding all normalized benefits of a strategy with respect to affected scenarios. If we express in equation, the following one should be adopted in the calculation:

$$\mathrm{B_i} = \sum (\mathrm{U_{expected}} - \mathrm{U_{current}}) * Wj$$

**Table 9 Sample of final benefits of ASs (based on Kazman, 2002)**

| Strategy | Scenario Affected | Scenario Weight | Raw AS benefit | Normalized AS benefit | Total AS benefit |
|---|---|---|---|---|---|
| 1 | 3 | 20 | 20 | 400 | 275 |
| 1 | 5 | 25 | -5 | -125 | |
| | 4 | 15 | 10 | 150 | 150 |
| 2 | 1 | 15 | 12 | 180 | 180 |
| 3 | 4 | 15 | 5 | 75 | 75 |
| 4 | 1 | 15 | 7 | 105 | 105 |

From table 9, we can note that the raw AS1 benefit to scenario 3 is a positive value, while the raw AS1 benefit to scenario 5 is a negative value. This means the change made by strategy 1 will increase the degree to which architecture supports scenario 3 and reduce the degree with respect to scenario 5. Because of the interaction of scenario 3 and 5, the two raw benefits need to be added up when calculating the total benefit for architecture strategy 1.

- Step 12: Calculate ROI (Return on Investment) for each AS

In this step, the cost of each architecture strategy is estimated by the evaluation team. Return on investment (ROI) for each architecture strategy is calculated according to the following equation (Len Bass & Paul Clements, 2005):

$$\mathrm{ROI}_i = \frac{B_i}{C_i}$$

Within this equation, $B_i$ means the total benefit for architecture strategy i, while $C_i$ means the estimated cost for architecture strategy i. By the calculation of ROIs, we could evaluate AS from economic perspective and give each AS a rank. Table 10 is a sample of calculating ROIs for architecture strategies. As seen from table 10, strategy 1 has the highest rank and we have tendency to consider its realization first because of its priority.

**Table 10 Sample of calculating ROI for each AS (based on Kazman, 2002)**

| AS | Cost | Total AS Benefit | AS ROI | AS Rank |
|---|---|---|---|---|
| 1 | 350 | 275 | 0.79 | 1 |
| 2 | 300 | 150 | 0.5 | 2 |
| 3 | 600 | 180 | 0.3 | 4 |
| 4 | 300 | 75 | 0.25 | 5 |
| 5 | 300 | 105 | 0.35 | 3 |

From step 8 to step 12, we are calculating ROIs step by step. The ROI calculation method is coming from the CBAM, and the sample tables are based on a case study of Rick Kazman and Jai Asundi (2002).

## 5. How to apply the methodology in reality?

In order to explain how to use our methodology to evaluate BR implementation in reality, we would like to set up a sample by ourselves. The purpose of this sample is just to display how to realize our methodology step by step. Because it is not a real case study, there is no practical meaning for this sample. In order to develop the practical experience of applying this methodology, some case studies are needed in the further study.

Here we image a B2C company which is selling their products on-line to individual customers. Their system is built on a service-oriented architecture, Thousands of individual customers who are sitting in front of web browsers are connected to web servers via Internet. Web servers will be in charge of receiving and handling the requests of access to the applications. All applications are built based on services, mainly including the applications of processing orders, managing customers and managing products. And all the applications are assumed to exist in the sample computer. The data base is centralized in one place, but is locating in a different computer from where applications exist. This architecture is only a top-level description and much simpler than architectures in real situations.

Under this architecture, we have more than one choice to implement business rules. For instance, prolog, a kind of BR component, can be used to implement rules in business logic layer. We can assign a prolog file to each application to implement the rules relating to that application. If we choose to implement business rules in this way, a kind of candidate architecture will be generated and we call it CA1. Of course, there are other choices, such as applying a BRMS to interact with all applications, which will generate other candidate architectures. We concentrate on how to use our methodology to evaluate CA1 in this discussion.

Before the process of BR implementation, a set of rules are modeled in natural language in our sample. Some rule samples are listed as following:
R1: Customers must log in before they access to products information.
R2: Customer discount must be 10% if the customer class is platinum.
R3: Customer discount must be 7% if the customer class is gold.
R4: Customer discount must be 5% if the customer class is silver.
R5: Customer discount must be 3% if the customer class is bronze.
R6: Customer discount must be 0% if the customer class is standard.

Under the situations we set up, out methodology is performed step by step as following:

- Step one: candidate architecture 1 (we mentioned above) and other candidate architectures need to be described in architecture diagram.

- Step two: in our methodology, modifiability and performance are identified as high-level factors which influence utility. Sub-factors need to be specified for the two high-level factors.

- Step three: except evaluation team, external stakeholders, mainly business analysts and business owner (who is in charge of complete business rules), need to be included to perform scenario brainstorm.

- Step four: Assign priorities by voting and select the top 30% scenarios as inputs for scenario evaluation.

In our sample, the following two scenarios are selected and described, but the selected scenarios should not be limited to the two.
Scenario 1: the time spent in checking log-in status before loading products information should be less than 3 seconds.
Scenario 2: the cost of changing discount for different customer classes should be less than 1 day-person.

For scenario 1, each time when a customer requests to read products information, the customer’s status needs to be checked before loading products information. If the customer has logged in, products information will be displayed on the web page. But if the customer just walks through without log-in, some notice message (e.g. “Please log in before you browse products information”) will be shown on web page instead of products information. In other words, R1 needs to be checked each time when products information is required. Because thousands of customers may need to check R1 at the same time, the response time in checking R1 should be less than 3 seconds.

For scenario 2, customer discounts are always changed according to corporations’ business strategies, so rules relating to discount (R2/ R3/ R4/ R5/ R6) should be able to be modified easily. So the cost should be less than one day-person.

- Step 5: based on the selected scenarios (1 & 2 in this sample) and candidate architectures, we evaluate each individual scenario.

- Step 6: we evaluate interactive scenarios. After step 5 and step 6, we found CA1 can support the achievement of scenario 2, but can not support scenario 1. The results are shown in the following table 11 and no interaction is discovered between scenario 1 and 2. Maybe there are some other indirect scenarios (like 3 & 4) in our sample, but we put them aside at the moment.

**Table 11 Results of scenario evaluation**

| Scenario | Description | Direct/ Indirect | Change Required | # of changes |
|---|---|---|---|---|
| 1 | Time spent in checking R1 should be less than 3 SECs | Indirect | Changing the prolog file containing R1 | 1 |
| 2 | Cost of changing discount should be less than 1 day-person | Direct | | |
| 3 | ………….. | indirect | ……………. | ……. |
| 4 | …………… | Indirect | ………..…… | ……. |

- Step 7: identify architecture strategy. AS1: Remove R1 to presentation layer and take the manner of ASP Script.

- Step 8: refine and priority scenarios as following table 12. The weight of scenario 1 is given according to its importance and other indirect scenarios' importance. The total number of scenarios' weights is 100. QA respond to each scenario in four kinds of cases. We take scenario 1 as a sample, the execution time of R1 is more than 10 seconds in worst case, and is less than 1 second in best case. The desired execution time is less than 3 seconds. While the execution time in current case is 5 seconds.

**Table 12 Results of refining and prioritizing scenarios**

| Scenario | Weight | Response Goals | | | |
|---|---|---|---|---|---|
| | | Worst | Current | Desired | Best |
| 1 | 30 | >10 secs | 5 secs | <3 secs | <1 sec |
| 3 | 35 | ………….. | …………. | ………..... | ……….. |
| 4 | 35 | …………… | …………. | …………. | ………… |

- Step 9: Assign utility to each scenario and the results is shown in table 13. Utilities are assigned according to stakeholders' estimation and their consensus.

**Table 13 Results of assigning utility**

| Scenario | Weight | Response Goals | | | |
|---|---|---|---|---|---|
| | | Worst | Current | Desired | Best |
| 1 | 30 | 0 | 70 | 90 | 100 |
| 3 | 35 | ………….. | …………. | ………..... | ……….. |
| 4 | 35 | …………… | …………. | …………. | ………… |

- Step 10: Determine expected utility value for each AS. Table 14 shows the expected response for AS1 with respect to scenario 1, and this expected value is estimated according to current and desired response. Table 15 shows us the expected utilities for AS1 and other strategies.

**Table 14 Results of architecture strategies and scenarios addressed**

| AS | AS Name | Scenarios Affected | Current Response | Expected Response |
|---|---|---|---|---|
| 1 | Remove R1 | 1 | 5 secs | 2 secs |
| 2 | …………… | 3 | …………… | …………… |
| 3 | …………… | 4 | …………… | …………… |

**Table 15 Results of determine expected utility value for each strategy**

| AS | AS Name | Scenarios Affected | Current Utility | Expected Utility |
|---|---|---|---|---|
| 1 | Remove R1 | 1 | 70 | 95 |
| 2 | …………… | 3 | …………… | …………… |
| 3 | …………… | 4 | …………… | …………… |

- Step 11: calculate benefit for each AS. From table 15, we get raw AS1 benefit is 95 minus 70. Normalized benefit for AS1 is 30 (scenario weight) multiplied by 25. Because AS1 can only affect one scenario (scenario 1), the total benefit for AS 1 is equal to normalized benefit for AS1.

**Table 16 Results of benefits for architecture strategies**

| Strategy | Scenario Affected | Scenario Weight | Raw AS benefit | Normalized AS benefit | Total AS benefit |
|---|---|---|---|---|---|
| 1 | 1 | 30 | 25 | 750 | 750 |
| 2 | 3 | 35 | …… | …… | …… |
| 3 | 4 | 35 | …… | …… | …… |

- Step 12: calculate ROI for each strategy. The evaluation team estimates a cost for AS 1. Then ROI for architecture strategy 1 is deriving from total benefit (750) divided by cost (980). According to ROI for strategy 1 and ROIs for other strategies, we can rank all architecture strategies.

**Table 17 Results of calculating ROI for each architecture strategy**

| AS | Cost | Total AS Benefit | AS ROI | AS Rank |
|---|---|---|---|---|
| 1 | 980 | 750 | 0.77 | |
| 2 | …….. | ……. | ……. | ……. |
| 3 | ……. | ……. | ……. | ……. |

## 6. Research result

In this chapter, we will present the information we get from the interviews. As mentioned in our method part, we take advantage of interview to collect other people's opinions about our methodology. So the results from interviews will be described from interviewees' perspective.

### 6.1 Opinions about BR implementation

The interviewees agree that the technologies relating business rules are meaningful for both software system development and business perspective. Interviewee A who has been a java programmer for five years emphasizes the importance of capturing business rules in building rule-based software systems. She is more interested in BR engine and BRMS than other BR implementation methods. When we stress the relatively high cost of applying a BRMS, such as ILOG, she answered "The cost is a problem for corporations, but the BRMS will enable ordinary people to maintain business rules, not only the IT professionals". And the interviewees think the cost is not a problem if a corporation has spent so much time in analyzing business rules.

The interviewees can not offer us clear answer when we ask them whether they have some personal tendencies or guides when they are facing choice for BR implementation. They only put forward some vague guides. They said they would assign a candidate implementation choice of BR a higher priority if it can be easily implemented, and make BR executed well and modified easily. Above vague guide is not enough to help people make a decision for BR implementation.

Generally speaking, the interviewees tend to make a choice for BR implementation according to their feelings and former working experience. As interviewee B said it is hard for him to describe how to choose.

The interviewees consider business rule technology is relative new and there are some problems in its real application at recent stage. But it has a bright future because it will enable employees from all departments (not only IT department) to update business rules relating to their working field (stated by interviewee C).

The interviewees all agree that it will be nice if there is a way for them to evaluate their implementation choice for business rules (Where and how to implement BR) before they really realize their choice. They said the application of business rule technology is facing a similar problem to the application of other IT technologies. Sometimes problems can only be discovered if it brings some trouble to the system. If there is some way to filter candidate implementation choices of BR at an early stage, it will avoid some unwanted problems.

## 6.2 Overall evaluation to our methodology

The interviewees consider our methodology as a nice try. They think it will be quite helpful for their BR implementation choice if it can be applied in reality.

The adoption of scenarios is given praise by all interviewees. They think scenarios can offer a context for BR implementation judging (whether implement BR in the right location and right manner). Scenarios can be easily forwarded by all stakeholders according to their daily work, and can be understood easily. Even though only two interviewees among the totally three have heard of the conception of scenario-based architecture analysis, they think our methodology is not so hard to understand. The interviewees can understand how this methodology is designed, and are able to follow our introduction to the methodology step by step.

But at the same time, the interviewees have something uncertain in this methodology. Fist of all, they are not sure whether this methodology can be realized in reality because it lacks of evidences from case study and empirical findings. They also doubt the reliability of ROI calculation in the methodology. Lastly, they believe this methodology should be evolved in future because the real situations are more complex than our assumption.

Even though there are some challenges in our methodology, the interviewees totally confirm our methodology as a nice try. They believe there is long way to go before it is completed, but we are in the right starting point.

## 6.3 Detail evaluation to our methodology

In this part, we will summarize some results we got from the interview questions. By the interactions with our interviewees in this question-answer way, we found some opinions in their views.

### 6.3.1 Interviewees' opinions in quality attribute selection

The interviewees said we should include other software quality attributes (like integrability and security) into our utility tree development when they were asking whether other quality attributes should be taken into account. They all agree that performance and modifiability are two of the most important attributes we should pay attention to. But besides the two, there are some other attributes need to be trade-off as well.

Interviewee A believes that integrability is an important attribute for evaluating business rules implementation and should be assessed by our methodology. She said, “I used to program a java application which was integrated with a prolog file via ODBC. Sometimes I found connection errors, while sometimes I didn't get expected outputs from prolog”. She believes, when we use business rule product (e.g. Prolog, BR engine and BRMS) to externalize business rules from programming codes and store them in another file or mechanism, how to make our applications interact well with a BR file or product is an important issue.

Interviewee B offered another attribute from business perspective. He thinks security should be considered because business rules are relating to how the business is conducted and belong to a kind of corporation secrecy. Business rules used to be distributed into programming codes, and were hard to be tracked. But nowadays, business rules locations are obvious to corporation staff if using BRMS, as well as to intended attackers. Especially in B2C environment, internet brings some risks for security problems, so corporations need to consider security issue when implementing rules. The BR component or BR engine needs to integrate with other applications. If a hycker can attack the applications, he can also change business rules illegally.

### 6.3.2 Interviewees’ opinions in scenarios selection

In our methodology, we select only 30% of the all scenarios collecting from brainstorm to enter evaluation step. For this figure 30%, our interviewees doubt why we take this figure, not others.

Interviewee C also mentioned that, if the total number of scenarios from brainstorm is less than 15, 30% of the total number should be less than 5. In this case, the number of selected scenarios is not enough to offer reliable evaluation.

Interviewee C recommends double standards for how many scenarios should be selected. If the total number of scenarios from scenario brainstorm is more than 30, we take 30% of them to enter next step. Otherwise, we should select the top ten scenarios according to their relative importance. Of course, if the total number is less than 10, we select all of them.

### 6.3.3 Interviewees’ opinions in AS evaluation

We take advantage of ROI (return on investment) to evaluate candidate architecture strategies. The interviewees consider calculating ROI is an effective means for evaluating different architecture strategies, because a rational decision maker will always make a trade-off between benefit and cost. Like interviewee A said she also took advantage of ROI in her former working experience.

But the interviewees think that, besides ROI, some other factors should be judged as well. The three interviewees’ opinions are listed as following:

Interviewee A said we should consider the time spent in changing architecture as well. The time spent in realizing an architecture strategy is an important factor for corporations, if the corporations have time limit in system development or their schedules have already been delayed. If two strategies have similar values of ROIs to the same scenario, we are prone to choose the one with less time cost.

Interviewee B thinks we should consider technology support as well. If a corporation has no technology support for realizing an architecture strategy, it has to outsource the change task to a third party. Sometimes the corporation may think it is not convenient to outsource.

Interviewee C believed we should give an architecture strategy a higher priority if the component it changed will affect more than one indirect scenario. In other words, the components which cause the interactions of scenarios have higher priority. This interviewee's opinion is similar to our tendency in the step of interactive scenarios evaluation.

### 6.3.4 Interviewees' opinions in the calculation of ROI

The interviewees reach an agreement that the calculation of ROI is not very hard to understand, but they challenge the reliability of the values of ROIs. In the calculation method, too many values (such as cost, utility and expected response) are coming from the estimation and consensus of evaluation team members. There are too many subjective judgments involved, so our interviewees question the reliability of the calculation results.

## 6.4 Further development of the methodology

All interviewees believe we have two urgent tasks in further development of this methodology. Firstly, we need to develop a more reliable ROI calculation method. Secondly, more attentions should be paid to make a trade-off among several quality attributes, such as performance, modifiability, security and integrability. They believe our methodology is interesting, but have a long way to go before it is further completed.

## 7. Discussions

The results from the interviews are similar to what we expect from self-judgment. Our methodology is based on the scenario-based architecture analysis methods, especially the SAAM and CBAM. We combine the two methods in understandable way and specify this methodology to business rules field.

Two main shortcomings mentioned by our interviewees are agreed by us and we need to try to find solutions to them in the further study. The two shortcomings are listed as following:

First one is that we need to include security and integrability as two quality attributes besides performance and modifiability. A good trade-off among different software quality attributes is required to be realized in further study.

Secondly, the calculation method of ROI needs to be completed in further version of our methodology because too many subjective factors are involved in the process of estimating utility, cost and expect response. This calculation method for ROI is coming from CBAM, which is used to make a trade-off between cost and benefit. We think we need to search for a better ROI calculation method in further study.

## 8. Conclusions

In this paper, we can get conclusions that the scenario-based software architecture analysis method can be used to evaluate different implementation choices of business rules. And our methodology can work in business rule field and the shortcomings in this methodology are also obvious.

The methodology is considered as an initial vision, which needs to be modified in further study. The performance and modifiability are not enough to stand for the software quality attributes. Other attributes, such as security and integrability, need to be taken into account. The ROI calculation method is not reliable, and we should search for more reliable method.

The methodology in our study is a try to evaluate BR implementation choices. This study proves the evaluation of BR implementation can be realized in the manner of scenario-based analysis.

**Reference**

Arthur Gingrande (2001): *Relationship marketing: a competitive advantage for B2C*, http://www.kmworld.com/Articles/Editorial/Feature/Relationship-marketing-a-competitive-advantage-for-B2C-9240.aspx

Babis Theodoulidis & Anthony Youdeowei (1998): *Business Rules: Towards Effective Information Systems Development*, Department of Computation, UMIST

Bell, Michael (2008): *Service-Oriented Modeling: Service Analysis, Design, and Architecture*, Wiley & Sons, 3. ISBN 978-0-470-14111-3.

Boris Lublinsky and Didier Le Tien (2007): *Implementation of business rules and business processes in SOA*, http://www.infoq.com/articles/business-rules-processes.

Briony, J Oates (2006): *Researching Information Systems and Computing*. Sage Publications, Thousand Oaks, Calif., ISBN 1-4129-0223-1

Creswell, J.W. (2006): *Qualitative inquiry and research design: choosing among five traditions, 2nd edition*, Sage Publications, Thousand Oaks, Calif., ISBN 9781412916073.

Efraim Turban & Jae Lee (2000): *Electronic Commerce: A Managerial Perspective*, Prentice Hall international, INC, ISBN 0-13-018866-2

Hammersley, M. & Gomm, R. (1997): *Bias in Social Research*, Sociological Research Online, http://www.socresonline.org.uk/2/1/2.html,

Israel, M. & Hay, I. (2006): *Research ethics for social scientists: between ethical conduct and regulatory compliance*. Sage, London; Thousand Oaks, Calif., ISBN 1412903904 (pbk), 193 s

James Taylor: *Achieving Decision Consistency across the SOA-Based Enterprise Using Business Rules Management Systems*, Fair Isaac Corporation, p. 751-761.

J. Morabito, I. Sack, A. Bhate (2001): *Organization Modeling, Innovative Architectures for the 21st Century*, Prentice-Hall, NJ.

Kenneth L. Kraemer & Jason Dedrick (2001): *Dell Computer: Using E-commerce To Support the Virtual Company*, Center for research on information technology and organizations, University of California.

Kvale, S. (1996): *Interviews: an introduction to qualitative research interviewing*, SAGE, Thousand Oaks, CA, 326pp.

Len Bass & Paul Clements (2005): *Software architecture in practice, second edition, Person Education*, Inc, ISBN 0-321-15495-9, p.307-325.

Marcel Weiden & Leo Hermans: *Classification and Representation of Business Rules*, Social Science Informatics, university of Amsterdam. .

Marko Bajec & Marjan Krisper (2004): *A methodology and tool support for managing business rules in organizations*, Information Systems 30 (2005) 423–443

Mauricio De Simone & Rick Kazman (1995): *Software Architecture Analysis: An Experience Report*, University of Waterloo, Canada.

Morgan, T (2002): *Business rules and information system: aligning IT with business goals*. Addison-Wesley, Boston, ISBN 0-201-74391-4

Muhammed Ali Babar & Ian Gorton: *Comparison of Scenario-based Software Architecture Evaluation Methods*, Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC' 04), 1530-1362/04.

P.J. Layzell, P. Loucopoulos (1988): *A rule-based approach to the construction and evolution of business information systems*, Proceedings of the Fourth IEEE International Conference on Software Maintenance, Phoenix, Arizona, USA, , pp. 258–264

R Kalakota & A.B. Whinston (1997): *Electronic Commerce: A Manager's Guide*

Rick Kazman (1994): *SAAM: A Method for Analyzing the Properties of Software Architectures*, Proc. of the 16th ICSE.

Rick Kazman & Gregory Abowd (1996): *Scenario-Based Analysis of Software Architecture*, appear in IEEE Software, November 1996

Rick Kazman & Len Bass (1999): *SAAM: A Method for Analyzing the Properties of Software Architectures*, available at: http://elin.lub.lu.se.ludwig.lub.lu.se/elin

Rick Kazman & Mark Klein (2000): *ATAM: Method for Architecture Evaluation*, Carnegie Mellon University.

Rick Kazman & Jai Asundi (2002): *Making Architecture Design Decisions: An Economic Approach*, Software Engineering Institute.

Ronald G. Ross (2003): *The business rule approach*, Business rule solutions, p.85-87.

X. Wang, J. Sun, X. Yang, Z. He, and S. Maddineni (2004): *Business rules extraction from large legacy systems*. In Proceedings of the Eighth European Conference on Software Maintenance and Reengineering, pages 249–258,

Yin, R. K. (2003): *Case study research: design and methods, 3rd ed.*, Sage Public actions, Thousand Oaks, CA, 181pp.
Internet materials

http://www.ecommerce-land.com/history_ecommerce.html

http://www.bricklin.com/b2csuccess.htm

http://projects.bus.lsu.edu/independent_study/vdhing1/b2c/

# Appendix A: Interview guide

| ***Interview guide*** |
|---|
| ***Individual experiences*** |
| How do you perceive BR implementation? Are there any benchmarks that guide you to choose locations if you need to implement BR? |
| Have you ever experienced scenario-based software architecture evaluation method, Such as SAAM or ATTM? |
| Do you think scenario based evaluation method is advisable applied in our study, or could you advise some other additional evaluation method? |
| ***Evaluation methodology process*** |
| Do you think our methodology is easy to understand? |
| In the Brainstorm selection process, we select 30% scenarios via voting, do you think this voting method is reliable or do you have other ideas? |
| The attendants in scenario selection process should include external stakeholders, not only the evaluation groups. What kinds of external stakeholders would you advise in this selection process? |
| We focus on two quality attributes performance and modifiability in our study. Is there any other quality attributes you advised in the utility tree? |
| How do you consider the selecting of Architecture Strategy? Do you think the ROI are suitable tool to analysis AS and is this calculation method reliable? |
| ***Future*** |
| Do you think our evaluation methodology is appropriate for evaluating the BR implementation choice, is it easy or convenience to executing? |
| Is there any further information in evaluation BR implementation choice you would like to advise? |

# Appendix B: Interview protocols

**Transcription of the interview with Intayoad Wacharawan (interviewee A)**

R stands for the researcher; A stands for the interviewee A.

R: Hi, I am Jin. And this is my partner, Liang.
A: Hi.
R: I think you have already had some brief understandings about what we are going to do though our last email, right?
A: Yes.
R: Ok, we want to ask you some questions relating to business rules and our methodology. Are you ready? Well, we would like to give more explain about our methodology [……..] Do you have more questions?
A: No.
R: Ok, let us start. As we know, you have been a software developer for several years, right?
A: Yes, I am a Java programmer.
R: What is your understanding about business rules? Do you know business rule engine or BRMS?
A: Well, I think business rules are very important for system developing. You know, if I want to make my application work smoothly, I need to know how business processes. I have used Prolog to interact with my application. And I also heard about BR engine and BRMS. But my company hasn't had a BRMS till now. Actually, for me, I am more interested in BR engine and BRMS than other implementing manners.
R: Really? Why? Don't you think investing on BRMS is expensive?
A: Yeah, a little expensive. But the cost is a problem for the owner of the company. I think, BRMS can make people who are not really IT people, such as manager, change rules by themselves. So could make organization rapid change.
R: Ok. When you implement business rules, do you have some personal standard for deciding where and how to implement BR? I mean, your own guidance.
A: Well, as a developer, I will pick out a way in which business rules are easily implemented and maintained, and will make application stable. Make the application run smoothly.
R: Do you learn about scenario-based architecture evaluation methods, such as SAAM?
A: I have heard a little about it, but not using it in reality.
R: Ok, let's talk about our methodology. Do you think it is easy to understand?
A: Yes, I think it is easy to follow.
R: In our methodology, after a brainstorm for potential scenarios, we pick out 30% of the most important scenarios by voting. Do you think 30% is enough? Do you think this voting method is reliable?
A: I don't real consider about the number as soon as we got the scenarios we want to

pick. I think 30% is enough. You know, if we pick out too many, such as 50%, we will have too many potential changes to be evaluated.
R: We should include some external stakeholders to participate the scenarios brainstorm, not only the evaluation team. If for you, which kinds of external stakeholders should you take?
A: I am not sure, maybe business owner.
R: Except modifiability and performance, do you think some other software attribute should be selected, for example availability, security or portability?
A: Well, I think intergrability should be involved. You know, a BRMS is going to be connected with any application in organization. It should has ability to integrate with other applications. I used to apply prolog to interact with my java program, when I wanted to do something, sometimes it is impossible to do. I don't think it has good performance because of the bad interaction between java and prolog.
R: Have you heard about ROI?
A: Yeah, I also use it in former working.
R: How do you think about our ROI calculation? Do you think the calculation method is reliable?
A: I think you are doing in a similar way with me. I compare benefit and cost as well. I am not sure whether the calculation is reliable because too many figures are coming from the estimate.
R: Except return on investment, do you think there is some other factor affecting the choice of BR implementation?
A: I think an implementation choice should be given a higher priority if it can be easily implemented, and make BR executed well and modified easily.
R: Do you think our evaluation methodology is appropriate for evaluating the BR implementation choice. Is it easy or convenience to executing?
A: I think it is nice. Even though it is not perfect, I can understand. It is new.
R: Thank you. Could you give us some suggestions for further development of the methodology?
A: Well, I think you should improve the calculation method of ROI. Make it more reliable. And some case studies should be done to apply it in practice.
R: Ok, I think we finish our interview now. Thank you very much for your participate and for all the information.
A: You are welcome.

**Transcription of the interview with Cui Li (interviewee B)**

R stands for the researcher; B stands for the interviewee B.

R: Hello, is it Ms. Li? [through telephone]
B: Yes.
R: Hi, I am Jin Rui. We have made an appointment with you for a telephone interview.
B: yes, I remember. We can start now.

R: Ok, than you very much for your time. Well, I think you have got some information about our purpose and methodology though our last email. Do you have more questions to ask me?

B: [……..] Ok, I don't have more questions.

R: Great. Let us start. What is your understanding about business rules? Do you know business rule engine or BRMS?

B: Well, I think business rules have been developed a lot, from coding in applications to BR engine. The developing of BR engine or BRMS has shown us how important business rules are. I think BRMS gives us a bright future. It can enable employees from all departments (not only IT department) to update rules relating to their working field.

R: Ok. When you implement business rules, do you have some personal standard for deciding where and how to implement BR? I mean, your own guidance.

B: Well, it is really hard to decide where and how to implement BR. I just make decisions according to what I feel at that time. Yes, I can not describe a certain standard for you.

R: Do you learn about scenario-based architecture evaluation methods, such as SAAM?

B: Well, I know a little.

R: Ok, let's talk about our methodology. Do you think it is easy to understand?

B: It is fine, not very difficult to understand after having your detail information.

R: In our methodology, after a brainstorm for potential scenarios, we pick out 30% of the most important scenarios by voting. Do you think 30% is enough?

B: I don't know why you pick out 30%, not other figures. Maybe you need to explain more.

R: We should include some external stakeholders to participate the scenarios brainstorm, not only the evaluation team. If for you, which kinds of external stakeholders should you take?

B: I think, except the developer, the people who are in charge of modifying business rules should be involved. Also the end users, because they may have some requires for business rules according to their daily work.

R: Ok, next one. Except modifiability and performance, do you think some other software attribute should be selected?

B: Well, I think security is another important factor. If business rules are distributed into different applications, it is really hard for attackers to steal rules. But if we are using a BRMS or BR engine, a hycker may change rules illegally by entering the application which can interact with BRMS. Because the rules are exactly how the business is done, and they are secrecy for company, we need to consider security.

R: Have you heard about ROI?

B: A little.

R: How do you think about our ROI calculation? Do you think the calculation method is reliable?

B: Well, it seems not very reliable. But I don't know another method for calculating ROI. It is hard.

R: Ok. Except return on investment, do you think there is some other factor affecting the choice of BR implementation?
B: Maybe the technology support. Sometimes a company needs to outsource if they don't have their own technology support to realize an architecture strategy. Well, some corporations don't like outsource.
R: Do you think our evaluation methodology is appropriate for evaluating the BR implementation choice. Is it easy or convenience to executing?
B: I think it is interesting. But you need to modify it to become more reliable.
R: Thank you. Could you give us some suggestions for further development of the methodology?
B: Maybe you should also include security as a selected software attribute.
R: Ok, I think we finish our interview now. Thank you very much for your participate and for all the information.
B: You are welcome.

**Transcription of the interview with Chen Xiao (interviewee C)**

R stands for the researcher; C stands for the interviewee C.

R: Hi, Chen. Thank you for attend this interview.
C: My pleasure.
R: I think you have read our email, and got some basic information about what we will do during this interview. We would like to take this interview to learn what you think about business rules and our methodology.
C: Yes, I know.
R: Ok, Before ask you some questions, we would like to give more explain about our methodology [……...] Do you have something to ask me?
C: No.
R: Ok, let us start. What is your understanding about business rules? Do you know business rule engine or BRMS?
C: Well, the capture of business rules is always essential, no matter you consider it from IT perspective or business perspective. I have leant a little about business rules technology. I think business rule technology will be powerful in supporting business process in future, although it is rather new now.
R: How do you think about BR engine or BRMS? You know, they are a little different from other implement manners.
C: I think the maintenance of business rules will be easier if applying a BRMS. I used to implement rules in data base or application codes. But maintaining rules was a difficult task. I need to recall the location of a rule before really changing.
R: Ok. When you implement business rules, do you have some personal standard for deciding where and how to implement BR? I mean, your own guidance.
C: Well, I am not sure how you define the meaning of 'standard'. But I will choose a BR implementation choice if it is easy to be implemented. And it contributes good

effect and makes system development easier.

R: Do you learn about scenario-based architecture evaluation methods, such as SAAM?

C: Well, I know nothing about them before this interview. But after you explain, I don’t think scenario-based evaluation is hard to understand. Scenario can be easily understood by all stakeholders, and it is a useful means.

R: Yes, it is. Ok, let’s talk about our methodology. Do you think it is easy to understand?

C: Yes, I think it is not so hard.

R: In our methodology, we pick out 30% of the most important scenarios by voting. Do you think 30% is enough? Do you think this voting method is reliable?

C: Well, I doubt that the figure of 30% is reliable. Well, there is another problem. For example, if only ten scenarios are forwarded after the brainstorm. The 30% of the total number is only three. I don’t think three is enough to make a reliable evaluation. Don’t you think so?

R: Well, yes, three is not enough. I think we should make a modification here.

C: Maybe you can have more than one option here. If the total number is more than 30, for example, you can pick out 30% of the total. But if the total number is less then 10, you should include all of them. Maybe between 10 and 30, you just pick out 10. This is only one suggestion. You can do more modification by yourself.

R: That is a good suggestion. Ok, we continue to next question. Which kinds of external stakeholders should be included in the process of brainstorm?

C: Well, maybe business owner or end users.

R: Except modifiability and performance, do you think some other software attribute should be selected, for example availability, security or portability?

C: Well, I think the two are enough.

R: Have you heard about ROI?

C: Yes, I know the concept of return on investment. But I didn’t use it in reality.

R: How do you think about our ROI calculation? Do you think the calculation method is reliable?

C: I don’t think your calculation is reliable because you only let evaluation team estimate, and the standard for the estimate is also vague. Maybe you need to do more explanations about how to estimate.

R: Ok. Next question, except return on investment, do you think there is some other factor affecting the choice of BR implementation?

C: Well, if a modification on a component can make more than one indirect scenario become direct, I will do this modification first. And I also consider the cost of maintaining business rules. If an implementation choice can make the process of BR maintenance easier, with less cost, I will choose that choice first.

R: Do you think our evaluation methodology is appropriate for evaluating the BR implementation choice. Is it easy or convenience to executing?

C: I think it is a good try. But it still needs to be modified if you want to make it more reliable.

R: Could you give us some suggestions for further development of the methodology?

C: Well, maybe you need to find out a more reliable method of ROI calculation.

R: We will. Ok, I think we finish our interview now. Thank you very much for your participate and for all the information.

C: It’s my pleasure.