แบบจำลองนามธรรมสำหรับตัวแทนเคลื่อนที่ที่สามารถปรับตัวอัตโนมัติ

นางสาวดุษฎี ประเสริฐธิติพงษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2549

AN ABSTRACT MODEL FOR AUTOMATED ADAPTABLE MOBILE AGENT

Miss Dussadee Praserttitipong

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program Program in Computer Science
Department of Mathematics
Faculty of Science
Chulalongkorn University
Academic year 2006

Thesis Title	AN ABSTRACT MODEL FOR AUTOMATED MOBILE AGENT
By	Miss Dussadee Praserttitipong
Field of Study	Computer Science
Thesis Advisor	Associate Professor Peraphon Sophatsathit, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

.... Dean of the Faculty of Science

(Professor Piamsak Menasveta, Ph.D.)

THESIS COMMITTEE

.... Chairman

(Professor Chidchanok Lursinsap, Ph.D.)

.... Thesis Advisor

(Associate Professor Peraphon Sophatsathit, Ph.D.)

.... Member

(Assistant Professor Pattarasinee Bhattarakosol, Ph.D.)

.... Member

(Assistant Professor Somjai Boonsiri, Ph.D.)

.... Member

(Associate Professor Somchai Prasitjutrakul, Ph.D.)

.... Member

(Surapun Mekanawin, Ph.D.)

คุษฎี ประเสริฐธิติพงษ์ : แบบจำลองนามธรรมสำหรับตัวแทนเคลื่อนที่ที่สามารถปรับตัวอัตโนมัติ.(AN ABSTRACT MODEL FOR AUTOMATED ADAPTABLE MOBILE AGENT) อ. ที่ปรึกษา: รศ. ดร. พีระพนธ์ โสพัศสถิตย์ 135 หน้า.

โดยทั่วไปตัวแทนเคลื่อนที่ทำงานในสภาพแวดล้อมแบบเปิด พลศาสตร์ และไม่สามารถทำนายได้ ดังนั้นจึงเป็นการยากที่จะคาดเดาสถานการณ์ที่เป็นไปได้ทั้งหมดที่ตัวแทนเคลื่อนที่แต่ละตัวจะต้องพบในขั้นตอนการออกแบบระบบ เพื่อให้ตัวแทนเคลื่อนที่ทำงานได้บรรลุเป้าหมายด้วยความสามารถอันฉลาดรู้จักคิด ตัวแทนเคลื่อนที่จะต้องสามารถปรับตัวได้ด้วยตัวเองไปกับสภาพแวดล้อมที่จะต้องพบ เป้าหมายของการศึกษานี้คือทำแบบจำลองเคลื่อนที่ซึ่งสามารถปรับตัวและสร้างกลไกการปรับตัวพลศาสตร์ที่จำเป็นสำหรับตัวแทนเคลื่อนที่ซึ่งจะทำงานในสภาพแวดล้อมใด ๆ ภายใต้การสืบค้นบริการเพื่อให้ตัวแทนเคลื่อนที่เกิดความสามารถของการปรับตัวอันฉลาดรู้จักคิดให้ทำงานได้บรรลุเป้าหมายของผู้ใช้ รายละเอียดพฤติกรรมที่สามารถกำหนดได้เป็นสิ่งสำคัญที่จะต้องกำหนดขึ้น มีการประยุกต์ความสามารถด้านการขยายตัวเข้ากับรายละเอียดดังกล่าว เพื่อให้สอดคล้องกับแบนด์วิดธ์ของสภาพแวดล้อมที่ทำงานอยู่ วิทยานิพนธ์นี้ได้นำเสนอวิธีของปรับตัวสองระดับ ในขั้นวิธีแรกเป็นรายละเอียดเจตนาแบบกล่องเทาที่สามารถกำหนดได้ โดยนำเสนอเป็นแบบจำลองซึ่งหุ้มพฤติกรรมของตัวแทนเคลื่อนที่ไว้ในรูปแบบที่ถูกนอร์มัลไลซ์ให้เหมาะกับการทำงานในสภาพที่มีแบนด์วิธสูง และพัฒนาการประเมินการถูกต้องแบบละเอียดเพื่อตรวจสอบความถูกต้องของผลลัพธ์การจับคู่ ขั้นที่สองประมวลย่อของรายละเอียดเจตนาที่สามารถกำหนดได้ โดยนำเสนอตัวแทนเคลื่อนที่มีขนาดกระชับ ช่วยให้ทำงานในสภาพที่มีแบนด์วิดธ์ต่ำ ใช้การประเมินความถูกต้องแบบหยาบสอด คล้องกับการวิเคราะห์ประมวลย่อ ความถูกต้องของผลลัพธ์การศึกษานี้วัดโดยวิธีพรีซิชันและรีคอลเพื่อแสดงว่าแบบจำลองที่นำเสนอทำงานได้ดีเพียงใด ผลลัพธ์ทั้งหมดชี้ว่าแบบจำลองที่เสนอจะช่วยให้ตัวแทนเคลื่อนที่มีศักยภาพเพียงพอที่จะทำงานได้สำเร็จตามความพึงพอใจของผู้ใช้

ภาควิชา..........**คณิตศาสตร์**...................... ลายมือชื่อนิสิต.......คุษฎี ประเสริฐธิติพงษ์.........
สาขาวิชา..........**วิทยาการคอมพิวเตอร์**............ ลายมือชื่ออาจารย์ที่ปรึกษา.......Peraphon Sophatsathit
ปีการศึกษา 2549

4573816423 : MAJOR COMPUTER SCIENCE

KEY WORD: SERVICE DISCOVERY / ADAPTABLE AGENT.

DUSSADEE PRASERTTITIPONG: AN ABSTRACT MODEL FOR AUTOMATED MOBILE AGENT. THESIS ADVISOR: ASSOC.PROF. PERAPHON SOPHATSATHIT, Ph.D., 135 pp.

A mobile agent (MA) typically operates in open, dynamic, and unpredictable environments. Therefore, when designing MA systems, it is difficult to anticipate all possible situations an MA may encounter. In order to reach its missions with intelligence capability, the MA must be able to adapt itself to the underlying environment. The goals of this study are to model adaptable MA and devise dynamic adaptation mechanisms necessary for the MA to operate in such environments by means of service discovery. In order to enable intelligence adaptation capabilities of the MA to accommodate the user's purpose, deterministic behavioral specifications are vital issues that must be addressed. Some scalable capabilities are applied toward the specifications so as to accommodate the bandwidth of its working environments. Two levels of adaptation are proposed. First, a deterministic gray box intention specification is proposed as a model which encapsulates the MA's behavior in normalized form, operating with high available bandwidth. In which case, a detailed validation assessment is formulated to support a precision of matchmaking results. Second, a synopsis of deterministic intention specification is nominated in a compact form of MA's intention that must operate with low available bandwidth. A rough validation assessment corresponding with this synopsis analysis is presented. Accuracy of the results is measured by precision and recall as to how well the proposed model performs. The overall results indicate that this proposed model will permit the MA to be potent enough to accomplish its tasks within users' satisfaction.

Department:.........**Mathematics**..................Student's Signature:..............................

Field of Study:......**Computer Science**...........Advisor's Signature:..............................

Academic Year:....**2006**............................

Acknowledgments

During my years as a Ph.D. student, I have received a lot of tuition, care, and friendship from several people, of whom I wish to express my appreciation here.

I would like to express my deepest gratitude to my supervisor, Assoc.Prof. Peraphon Sophatsathit, for his practical guidance and kind support which make this dissertation possible. His care, encouragement, and understanding help me overcome the necessary difficult points along my quest. I wish to extend my sincere gratitude to Assoc.Prof.Dr. Arkady Zaslavsky of Monash University, Australia. His helpful advice has provided a good application for my research work. I am really grateful to my dissertation committee, Prof.Dr. Chidchanok Lursinsap, Assist.Prof.Dr. Pattarasinee Bhattarakosol, Assist.Prof.Dr. Somjai Boonsiri, Assoc.Prof.Dr. Somchai Prasitjutrakul, and Dr.Surapun Mekanawin, for their meticulous review, constructive criticisms, and excellent advices about my research work. My grateful thanks also go to my respected educators at the Faculty of Science, Chiang Mai University, Assist.Prof. Panutson Chuehongthong, Assoc.Prof. Petcharat Chotigarpa, Assist.Prof. Darunee Smavatkul, and Dr.Watcharee Jumpamule, for their generous care and support. I also wish to thank my colleagues at Advanced Virtual and Intelligent Computing (AVIC) Center for theirs extensive discussions and interesting exploration on my work that proved to be very helpful. I wish to convey my thank to Mr.Wijak Srisujjalertvaja for his untiring support during my difficult moments. Furthermore, I would like to thank the Royal Thai Government who furnishes the financial support for my research scholarship.

Finally, I owe a debt of gratitude to my parents and my brother. Their tremendous love, great care, and unconditional support are my passport to sustain in this world.

Table of Contents

List of Tables

List of Figures

Figure Page

List of Symbols

β	an agent intention specification describing particular tasks or duties to be undertaken by the MA
π	a service advertisement characterizing tasks which can be accomplished by that service
$\mathcal{F}$	a 4-tuple functional specification
$\mathcal{I}$	a behavioral specification
$\mathcal{B}$	a 5-tuple behavioral specification
Γ	the function that performs a relaxation matchmaking over a behavioral specification
ϑ	the reliability degree of a matchmaking function
$\mathcal{X}$	a discrete random variable representing the possible purposeful degree of an agent executing by means of service π for a matchmaking function
$\mathcal{A}$	a set of concerning attributes for each object domain
ρ	a number of concerning attribute in each domain
$\mathcal{A}(k)$	a set of concerning attribute values for any concerning attribute
p_k	a number of concerning attribute values belonging to attribute $\mathcal{A}(k)$
$\mathcal{A}'$	an attribute column vector
$\mathcal{N}$	the number of overall attribute values or decision criteria
λ	the number of declared attribute values or decision criteria
$\mathcal{M}$	the number of possible condition statements
α	the number of declared conditional statements
$\widetilde{\mathcal{C}}$	a set of conditional statements encompassing ordered pairs
δ	an $\alpha \times n$ decision making situation matrix

Ω'	a decision action matrix
$\mathcal{I}^{WB}$	a white box deterministic intention specification of either β or π
$\mathcal{I}^{GB}$	a deterministic gray box specification of either β or π
$\mathcal{I}^{SY}$	a synopsis of deterministic intention specification of either β or π
$\mathcal{W}$	an adjusted learning weight matrix
b	an adjusted learning bias matrix
$\bar{\mathcal{I}}$	the generic (unspecified to β or π) deterministic white box specification
$\hat{\mathcal{I}}$	the normalized deterministic behavior specification
$\mathcal{I}'$	the scaled intention specification
$t\Theta$	the accuracy of normalization process
$t\nabla$	acceptable value of a discrepancy degree
$maxIter$	the maximum iteration for performing a normalization process accordingly
$\Theta^{\beta,\pi}$	the compatibility degree of reusing a service π in place of β
$\Delta(\beta.\Omega(i), \pi.\Omega(i))$	the similarity of $\beta.\Theta(i)$ and $\pi.\Theta(i)$
$\Gamma^{\beta,\pi}$	the resemblance degree, the ability for preserving the principal characteristics of β
$\nabla_{\mathcal{E}}^{\beta,\pi}$	the degree of discrepancy between the capabilities specified in β and π
$\mathcal{E}^{\beta}(j)$ and $\mathcal{E}^{\pi}(j)$	the entropy of β and π over a concerning attribute value j^{th}
$\mathcal{P}^{\beta}(j)$ and $\mathcal{P}^{\pi}(j)$	the probability of β and π over a concerning attribute value j^{th} toward the FV class
$\hat{\beta}$	a normalized intention specification
$\hat{\pi}$	a normalized service advertisement
$\mathcal{V}^{SY}$	a visualized model of a synopsis deterministic intention specification of either β or π

$\mathcal{V}(j)$	an x-y coordinate, $(\mathcal{E}(i), \mathcal{P}(i))$, of a concerning attribute value j^{th}
$\mathcal{G}(j)$	an edge from a vertex $\mathcal{V}(j)$ to a vertex $\mathcal{V}(j+1)$
σ	a user satisfaction level over reusing a service π in place of β
$\tilde{\tau}$	a set of candidate services
τ	a set of the most proper service
ζ	the expectation percentage over compatible degree
$\mathcal{B}roker^{ND}$	a broker that performs service validation algorithm based a nondeterministic behavioral specifications approach
$\mathcal{B}roker^{WB}$	a broker that performs matchmaking algorithm based on $\mathcal{I}^{WB}$ which is a deterministic white box behavioral specification approach
$\mathcal{B}roker^{BL}$	a broker that performs matchmaking algorithm based on $\mathcal{I}^{WB}$ and is used in controlled environment
$\mathcal{B}roker^{BP}$	a broker that performs service validation algorithm based on $\mathcal{I}^{GB}$ which is a deterministic gray box behavioral specification approach
$\mathcal{B}roker^{SY}$	a broker that performs service validation algorithm based on $\mathcal{I}^{SY}$ which is a deterministic behavioral specification approach
$\mathcal{B}roker^{GB}$	a broker that performs service validation algorithm based on $\mathcal{I}^{GB}$ which is a deterministic gray box behavioral specification approach The behavioral elicitation algorithm concerned both the 1^{st} and the 2^{nd} *constraints*. Furthermore, an extended-$\mathcal{I}^{GB}$ elicitation algorithm was taken into account in case of an inequivalent-λ.
$\mathcal{G}$	a group of complete basis intention blueprints
$\mathcal{G}^1$	a group of synthetic intention blueprints
$\mathcal{G}^2$	a group of incomplete basis intention blueprints
$\mathcal{G}^3$, $\mathcal{G}^4$	a group of synthetic intention blueprints with newly added attribute

CHAPTER I

INTRODUCTION

1.1 Introduction and Problem Review

A number of computer science disciplines and technologies have been used in developing intelligent systems, starting from traditional information systems and databases, to modern distributed systems and the Internet. Recent incorporation of Artificial Intelligence (AI) and computational intelligence research have led to the development of intelligent software agents.

A software agent, also called agent for short, is a piece of software that is authorized to act for or in the place of an application having major characteristics of autonomy, reactiveness, pro-activeness, cooperative, adaptation, and mobility. Intelligent software agent is a type of software agent which has built-in intelligent behavior to response to its tasks in accordance with a priori knowledge. The pervasive needs for interoperability in the Internet environment has brought about the notion of distributed application. As a consequence, a mobile agent (MA) has emerged as a paradigm for structuring computational intelligence of distributed applications. Based on mobility characteristics, Lange [1] defined an MA as

> *An MA is not bound to the system where it begins execution. It has the unique ability to transport itself from one system in a network to another. The ability to travel, allows a mobile agent to move to a system that contains*

an object with which the agent wants to interact, and then to take advantage of being in the same host or network as the object.

The advantages of MA paradigm [1] are: (1) reduce the network load, (2) overcome network latency, (3) encapsulate protocols, (4) execute asynchronously and autonomously, (5) adapt dynamically, (6) are naturally heterogeneous, and (7) are robust and fault-tolerant.

On the other hand, some inherent problems concerning MA operating in open, dynamic, and unpredictable environments still persist. These problems will be addressed, along with a novel approach to establish the solutions to the problems. Therefore, the aim of this proposed dissertation is to introduce an abstract model for an automated adaptable MA.

Consider the advent of Internet connectivity which creates an information-centric society where millions of people access large amounts of information stored in distributed networks on a daily basis. Frequently, ***pieces of information require evaluation with the help of expert knowledge to make them more meaningful***. In conventional client-server system, such an evaluation processing normally takes place at the client site. The requests usually are posted from the client to the server and the result data will be transferred across the network back to the client site. This will increase the overall network congestion. Bearing this scheme in mind, the MA technology can be employed to carry out the evaluation process at the server site. Thus, the resulting information will be sent back to the client via the network, whereby reducing the overhead of transmitting huge amount or raw data.

However, due to the nature of operating environment of the MA, the needed resources are usually not all known or adequately characterized in advance, and may change with time. This is, in part, owing to the fact that different hosts contain different resources which dynamically bound to the host system. Hence, the required resources cannot be anticipated ahead of time. Moreover, the MA must perform its tasks rapidly and independent of different working conditions imposed by the foreign environment. Under such mandate, the MA must act as a cooperative assistant to carry out those tasks on behalf of the user, rather than just serves as a conventional participating application program that is remotely manipulated by the user. In order to model such an intelligent MA that can operate alone smoothly in any foreign environment, all types of possible scenarios must be taken into account which unfortunately lead to an imploded MA. Furthermore, it is difficult to anticipate all possible situations that the MA must encounter during design time. As the prior knowledge will not be suitable for the new situation, some adaptive mechanisms are called for to transform this MA into target-compatible unit capable of coping with new unknowns.

To utilize an adaptation mechanism, one of the fundamental principles is to rely on service discovery technology. The MA contacts a middle agent, called **service broker**, to determine the most suitable service according to the agent's **intention specification**. Upon receiving such **an agent intention specification**, the service broker performs service discovery procedures by means of a **matchmaking process** to make a recommendation for the most suitable services based upon **a service advertisement**. In so doing, the agent will be able to accomplish its intention in the dynamic and uncertain environments.

1.2 Research Objectives

The objectives of this dissertation are:

1. model an adaptable MA
2. devise dynamic adaptation mechanisms necessary for the MA to operate in the open, dynamic, and distributed networks.

1.3 Scopes of the Study

This dissertation shall focus on the following scenarios for modeling support of an intelligent MA by integrating research on agent technology and service discovery.

1. **An automated adaptable MA system, encompassing**
   (a) A reference architecture of an automated adaptable MA.
   (b) A deterministic intention specification.
   (c) Model of an automated adaptable MA.
2. **A knowledge-based service discovery system, consisting of**
   (a) A deterministic intention specification.
   (b) A matchmaking process.
3. **Conduct an experiment to validate the proposed method.**

1.4 Research Advantages

This framework will in turn accomplish the following additional aspects:

- An architectural framework for light-weighted MA.
- An MA potent enough to accomplish its missions within the acceptable users' satisfaction.

CHAPTER II

THEORIES AND LITERATURE REVIEWS

According to the scope of this dissertation, there are two main related principles, namely, software agent technology and service discovery. Software agent technology considers the environment of this study. Service discovery is a thoughtful approach that is applied to solve the agent's adaptation problem. Some theories and literature reviews of both principles are introduced in the sections that follow.

2.1 Software Agent Technology

2.1.1 Overview

A description of of software agent is stated in [2] as

> *A software agent is in general a software entity that performs actions for its owner. An agent is described as being autonomous, goal-oriented, and having social ability to communicate with other agents.*

Agent technology can be articulated in two contexts: (1) the single-agent system, local agents, and networked agents, and (2) multi-agent system (MAS), Distributed Artificial Intelligence (DAI)-based agents, and Mobile Agents (MA). The main difference of these two contexts is the agents of MAS may extensively cooperate with each other to achieve individual goals. In the DAI-based system, the coordination of intelligent agents is concerned. In this type of system, an agent may communicate with the user, system resources, and other agent [3]. Some general characteristics of a software agent are proposed by Sanneck et al. [4] as autonomous (to act on its own), reactiveness (to process external events), proactiveness (to reach goals), cooperative (to efficiently and effectively solve tasks), adaptation (to learn by experience), and mobility (migration to new places).

2.1.2 Architecture

To specify the behavior of MAS, Wooldridge [5] presented a new logic. In this logic, agents are viewed as BDI systems, in that their state is characterized in terms of beliefs, desires, and intentions. The semantics of the BDI component of the logic are based on the well-known system of Rao and Georgeff [6, 7]. A definition of an intelligent agent is proposed as a system that is situated in a dynamic environment, of which it has an incomplete view, and over which it can exert partial control through performing the actions.

The architecture of a BDI agent [5], as shown in Figure 2.1, consists of four key data structures. They are

- *An agent's beliefs* correspond to information the agent has about the world, which may be incomplete or incorrect.

- *An agent's desires* intuitively correspond to the task allocated to it. (Implemented BDI agents require that desires be consistent, although human desires often fail in this respect.)

- *An agent's intentions* represent desires that the agent has committed to achieving. The intuition is that an agent will not be able to achieve all its desires, even if these desires are consistent. Agents must therefore fix upon some subset of available desires and commit resources to achieving them. These chosen desires are intentions. An agent will typically continue trying to achieve an intention until it believes the intention is either satisfied or no longer achievable.

- *A plan library* is a set of plans that specify courses of action that may be followed by an agent in order to achieve its intentions. An agent's plan library represents its procedural knowledge or know-how. A plan contains two parts: a body, or program, which defines a course of action; and a descriptor, which states both the circumstances under which the plan can be used (i.e., its pre-condition), and what intentions the plan may be used in order to achieve (i.e., its post-condition).

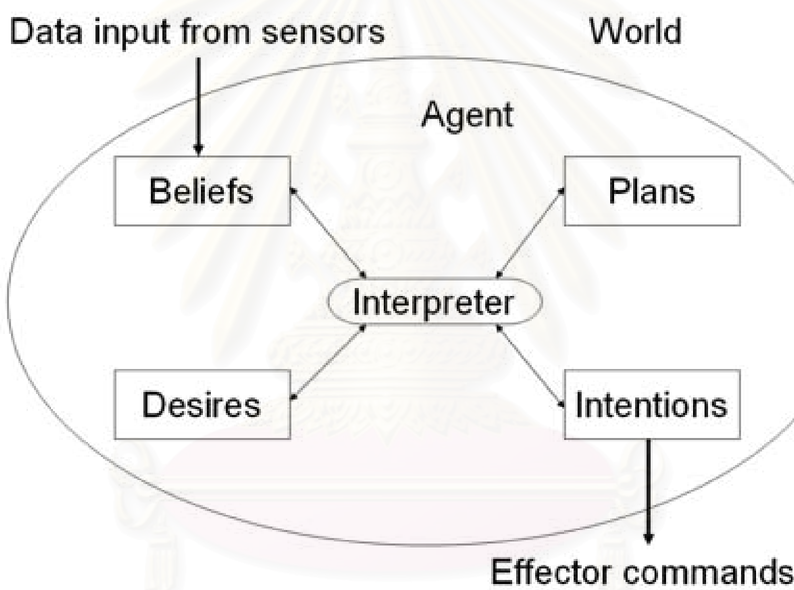

Figure 2.1: Architecture of a BDI agent [5]

The agent's plan, which is a test-bed prototype of this study, is defined by [5] as follows:

> A *plan* is a pair (β, δ), where $\beta \in D_B$ is a plan body, and $\delta \in \Delta$ is a plan descriptor, intended to represent the behavior of β. Let $D_\Pi = D_B \times \Delta$ be the set of all plans, and use π (with decorations: $\pi', \pi_1, \ldots$) to stand for members of D_Π. If $\pi \in D_\Pi$, then let $\hat{\beta}(\pi) \in D_B$ denote the body of π, and $\hat{\delta}(\pi) \in \Delta$ denote the descriptor in π. Thus, dom$\hat{\pi}$ represents the pre-condition and ran$\hat{\delta}(\pi)$ represents the post-condition of π. If $s \in \text{dom}\hat{\delta}(\pi)$, then let $\hat{\delta}(\pi)(s)$ denote the image of s through $\hat{\delta}(\pi)$, i.e., $\hat{\delta}(\pi)(s) = \left\{ s' | (s, s' \in \hat{\delta}(\pi) \right\}$.

As mentioned earlier that agent technique is a goal-orient system, Braubach et al. [8] classified the agent's types of goals depending on its behavior as achieve, maintain, cease, avoid, optimize, test, query, perform, and preserve. They investigated four main types of behavioral goals, namely,

- A *perform goal* specifies some activities to be done, therefore the outcome of the goal depends only on the fact if activities are performed.

- An *achieve goal* represents a goal in the classical sense by specifying what kind of world state an agent wants to bring about the future. This target state is represented by a target condition.

- A *query goal* is used to inquire information about a specified issue. Therefore, the goal is used to retrieve a result for a query and does not necessarily cause the agent to engage in actions.

- A *maintain goal* has the purpose to observe some desired world state and the agent actively tries to re-establish this state when it is violated. The perform, achieve, and query goal types represent goals that continuously cause the execution of plans while they are active. In contrast, an activated maintain goal may not instantly cause any plan to be executed.

2.2 Service Discovery Technology

To utilize an adaptation mechanism, one of the fundamental principles is to rely on service discovery technology. Currently, service discovery technology integrates researches from several communities [9, 10], such as reuse of Component-Based Software Engineering (CBSE), information retrieval of natural language service description, etc. There are two processes involved in the service discovery mechanisms, i.e., **service specification advertising** and **matchmaking process**. The former is a service specification advertisement focuses on how to advertise the service, while the latter is a matchmaking process focuses on how to select the most suitable service. The details of service discovery and its related technologies are introduced below.

2.2.1 Component-Based Software Engineering

A closely related research with the proposed model is a framework for automating specification-based component retrieval and adaptation (SPARTACAS) [13]. The work proposes the use of specifications to abstractly represent implementations. This allows automated theorem-provers to formally verify logical reusability relationships between specifications. These logical relationships are used to evaluate the feasibility of reusing the implementation of components to implement a problem. Components that partially satisfy the constraints of a design problem are adapted using adaptation architectures. Adaptation architectures modify the behavior of a software component based on the functionality specified in the problem and the partially-matched components. A subproblem that specifies the missing functionality is synthesized.

2.2.2 Knowledge Engineering

Ontologies and problem-solving methods (PSM) [14] are promising candidates for reuse in Knowledge Engineering. An *ontology* is a shared and common understanding of some domains that can be communicated across people and computers. Ontologies can therefore be shared and reused among different applications. *PSM* decomposes the reasoning task in a number of subtasks and inference actions. Both type of components can be viewed as complementary entities that can be used to configure new knowledge systems from the existing ones.

One instance of the research applying ontology and PSM concepts is IBROW3 project, an Intelligent Brokering Service for Knowledge-Component Reuse on the World-Wide Web [14]. The aim of this project is to develop an intelligent brokering service that enables third party knowledge-component reuse through the WWW. Suppliers provide libraries of knowledge components adhering to some standards. In order to configure a knowledge system suited to their needs, customers consult these libraries through intelligent brokers, of which ontologies form an essential part for selecting, adapting, and configuring PSMs.

2.2.3 Web Services Technology

The goal of Web services, presented by van der Aalst [15], is to exploit XML technology and the Internet to integrate applications ready for publication, search, and invocation over the Web. Gottschalk et al. [16] expressed that a Web service is an interface that describes a collection of operations that are network-accessible through standardized XML messaging. A Web service performs a specific task or a set of tasks. A service-oriented architecture of Web services, shown in Figure 2.2, consists of the service actors, the objects, and the operations. The service actors play three important roles, namely, the service provider, the service requester, and the service registry. The objects acted upon are the service and the service description. The operations performed by the actors in those objects are publish, find, and bind.

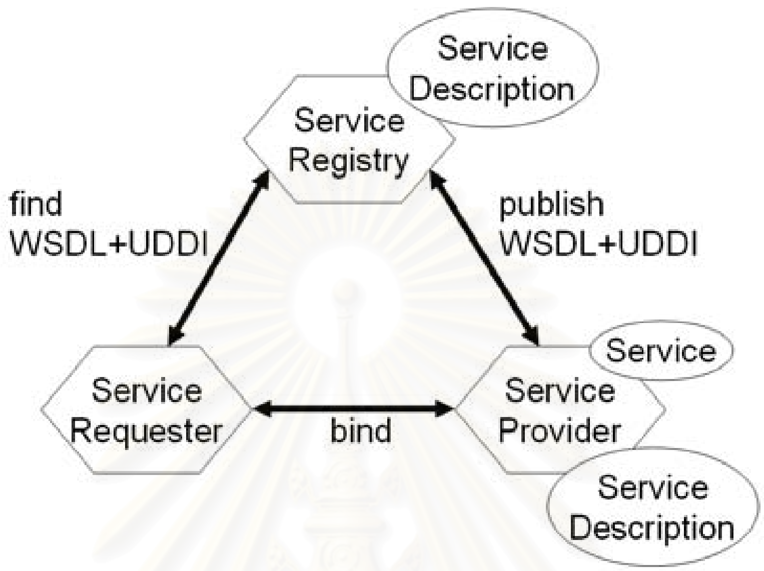

Figure 2.2: Web service actors, objects, and operations [16]

In Figure 2.2, a Web service and its service definition are created by a service provider and published with a service registry based on a standard called the Universal Description, Discovery, and Integration (UDDI) specification [21]. Once a Web service is published, a service requester may find the service via the UDDI interface. The UDDI registry provides a WSDL (Web Services Description Language [22]) service description and a URL (Uniform Resource Locator) pointing to the service itself for the service requester. The service requester may then use this information to directly bind to the service and invoke it.

2.2.4 Service Discovery Techniques

A number of research endeavors have been attempted to arrive at some approaches of service discovery techniques outlined as follows:

1. **Table-based service discovery** is the conventional approach for most of the service search technologies, for example, Jini^{TM}, eSpeck [19], CORBA trading service [20], UDDI. The *service advertisement* are the attribute values describing properties of the service, whereas the *matchmaking* between the provided service and the request is done by considering about the fixed number of attribute value pairs describing properties of the services.

   This model provides good qualities of retrieving results because of additional properties of the required services, e.g., parameter types, return types, calling protocols, etc. For example, the Web Services Description Language (WSDL) is an XML format published for describing public interface to the Web services. This is an XML-based service description on how to communicate using the web service, that is, the protocol bindings and message format required to interact with the web services listed in its directory. The supported operations and messages are described abstractly, and then bound to a concrete network protocol and message format. WSDL specifications of service-providing components are published in UDDI registries. UDDI is an on-line marketplace providing a standardized format for general business discovery. Developers can browse and query a UDDI registry using the UDDI API to identify businesses that offer services in a particular business category and/or services that provided by a certain service provider.

   All of these approaches define the term *relevant service* based on keyword mapping. The problem will occur when the requirement and the service specification are represented in different terminology but in fact they are synonymous, such as "buy" and "purchase." The proposed mechanism cannot detect such similarity. Another problem is that this mechanism cannot detect contextual wordings, such as "order", which has different meanings depending on the context.

2. **Semantic-based service discovery** is based upon concept-based information retrieval model [23], which improves search quality over the keywords based retrieval through the support of semantic web. The semantic web vision is to make web resources accessible by contents as well as by keywords. The *service advertisement* consists of attribute values describing properties of the service combining with semantic meanings. For instance, DAML-S [24] is a service specification language that supports the specification of semantic information. A service can be described in natural language by a profile (what the service does), a process model (how the service works), and a grounding (how to access the service). By the same token, *matchmaking* between the provided service and the agent intention specification is done with the semantic description.

3. **Context, ontology-based service discovery** is an approach for service discovery technique [17] that combines the concepts of context-aware service discovery with semantic-based service retrieval. This approach [25, 26, 27, 28, 29] regards the user's and the service providers' context by determining *where* the execution will take place. Since it is designed based on ubiquitous computing and continuous execution notion, the agent can continuously execute the suitable service anywhere.

   Ontology, on the other hand, furnishes semantic meaning and location information of shared and common understanding domains to be used by the service advertisement. Many researches suggest that there have been growing interests in focusing on functional specification rather than behavioral specification. The reason is that functional specification operates on properties of the service in the following terms:

- Action templates describes the functions that can be formed to bring about the intention.
- Preconditions are states obtained from the world before the actions take place.
- Postconditions are states exposed to the world once the actions took place.
- Constraints are mandates which are associated with the intention.

However, representation of "how the service does" is usually specified in natural language which contains insufficient and ambiguous information describing service behavior. As the matchmaking process performing on the behavioral specification relies on the principles of information retrieval [23], it remains unconvinced by the current evidence whether an agent encompassing with the recommended service is potent enough to accomplish its tasks within an agent's purposes. Given such an imprecise representation, these kind of investigations are denoted in this paper as *a conventional nondeterministic specification*. Acquiring a service with loosely or lack of concern on a service behavior leads to a nondeterministic manner toward an agent execution.

It would thus be imperative to focus on research investigating on deterministic service discovery establishing over a given system whose all possible situations can be anticipated at designed time. For instance, a **deductive service discovery** [30, 31, 32], which aims at expressing the service behavior bases on the formal specification, expresses the service semantics using logic modeling. Despite this approach offers formal description to represent service behavior in a deterministic fashion, the matching process of this method has high complexity and is hard to used in practical way.

In order to employ a more practical matchmaking process, a software component matrix represented the formal specification is proposed [33]. The logical form of a behavioral model refines the service capabilities in the form of state transition diagram, whereas the physical form is represented by a matrix. Thus, the matching process can be conducted with the help of matrix manipulation operations.

Measurements of the quality of service matchmaking [9, 17] assessment are **precision** and **recall**.

1. *Recall* is defined as the number of relevant services retrieved divided by the total number of relevant services at the broker site. The highest value of recall is achieved when all relevant services are retrieved.

2. *Precision* is the number of relevant services retrieved divided by the total number of services retrieved. The highest value of precision is achieved when only relevant services are retrieved.

2.2.5 Matching Degree

The matchmaking (or matching) process focuses on how to select the most suitable service by comparing user's request and service description. To rate the matches, Broens et al. [34] defined a quality measure called matching degree as follows.

Consider a user's request R and a service description S. To rate how relevant a particular match between R and S is, the number of service properties (i.e., type, inputs, outputs, and contextual attributes) is computed from the request R that are not presented in S. Based on those missing properties, they classified the match in five different categories, defined by Li and Horrocks [35].

The first category indicates an *exact match* which means the request has the same properties as the service description, i.e., there is no missing properties. This is the best possible match. The second category is called *plug-in match* that represents the second best match. It indicates that the service is more capable than the requester wants. The third and fourth categories, called *subsume match* and *intersection match*, respectively, indicate that the service can only partially provide what the user wants, i.e., the number of missing properties is larger than zero. The fifth category indicates a *disjoint match*, i.e., the request and the service do not share any properties.

Broen et al.'s approach used this initial classification to further classified matches in three types that became useful for the user.

- *Precise match*: Exact and Plug-in matches. The service is capable of providing the requested functionality or more.

- *Approximate match*: Subsume and Intersection matches. The service is capable of providing part of the requested functionality.

- *Mismatch*: Disjoint match. The service is not capable of providing the requested functionality and will not return any results to the user.

2.3 Literature Reviews on Software Agent and Adaptation Behavior

Agent technology has been employed in numerous application areas, namely, artificial intelligence, software technology, network management, and robotics. Early agent applications were largely user-manipulated that carried all predetermined requirements and functionalities. The advent of distributed processing notions have brought about new paradigms of mobile, dynamically-bound agent applications that are capable of remote execution. The MAs are thus precipitated from such notions as consigned applications running on target environments. The idiosyncrasies of different hosts inevitably force these MAs to be flexible and adaptable to suit the underlying environment. A number of research endeavors have been attempted to arrive at some forms of suitable MA configuration as outlined in the sections that follow.

In [36], dynamic adaptation technique is presented to equip mobile agents with a flexible capacity to adapt to a range of different environment on demand. Traditional adaptive techniques are static and dynamic adaptation. Static adaptation is a technique that generally is applied at compilation time by code reuse. Unfortunately, such an effort is not adequate in that continuous adaptation modifies a running application by tuning parameters. The proposed technique, dynamic adaptation, offers a technique for creating mobile agents which are able to adapt themselves to the environment where they are currently running. Adaptation of mobile agents occurs without termination of the agent. If the core mobile agent moves to a new host, the dynamic adaptation procedure is initiated. The input of dynamic adaptation is a set of environment dependent implementations, an environment independent core agent, and a description of the current environment. The result of dynamic adaptation is the selection of right implementation for the environment and linking of the selected implementation to the core. In the meantime, dynamic adaptation selects an implementation from an existing set of environment specific implementations, exchanges code, and instantiates code dynamically.

Brazier et al. [12, 37] proposed an approach to migrate agents between non-identical platforms that need not be written in the same language. In this approach, agents can be migrated across heterogeneous code bases and platforms by reconfiguration of the agents upon arrival at a new location. The principal basis of migration is accomplished by the agent's blueprint of functionality. A blueprint is a high level specification of the functionality and operational semantics of an agent. The specification describes the behavior of an agent in terms of its basic building blocks: components, control flow, and data flow. The resulting blueprint is expressed in a high level specification language: the blueprint language. At a new location, the agent is regenerated according to this blueprint using components specific to local agent platforms. The functional components can be another code base other than from the originating agent, as well as different platforms. Hence, interoperability between agent platforms can be realized. This approach of generative mobility not only has implications on interoperability but also on security.

As can be seen in Table 2.1, two relevant approaches are compared with this proposed approach from various perspectives, namely, the dynamic adaptation of mobile agents in heterogeneous environments [36] proposed by Brandt R. and Reiser H. and Reiser approach and Agent Factory [12] proposed by Brazier F. M.T., Overeinder B. J., van Steen M., and Wijngaards N. J.E. These aspects will be further discussed in the chapters that follow.

Table 2.1: Comparison of agent adaptation approaches

Topics	Brandt and Reiser approach	Agent Factory Approach	Proposed approach
Heterogeneous code bases	No	Yes	Yes
Heterogeneous platform	Yes	Yes	Yes
Mobility	Yes	Yes	Yes
Light-weight concern	Yes	Yes	Yes
Dynamic Adaptation	Yes	Yes	Yes
What to migrate	Core-code	Blueprint	Wish-list
Repository	local	remote, in-house	remote, third-party

continue on the next page

Topics	Brandt and Reiser approach	Agent Factory Approach	Proposed approach
Problem area	Network management configuration	Information retrieval (not expertise information evaluation)	Problem solving with expertise evaluation (Required expert knowledge and adaptation for computational)

CHAPTER III

PROPOSED METHOD

In this dissertation, some essentials must be established for use in subsequent in-depth analyzes and experiments.

1. Model an automated adaptable MA.
2. Summarizing conventional behavioral specifications.
3. Establish a conceptual model for deterministic behavioral specifications.
4. Establish a matchmaking process
5. Model a reference architecture of an automated adaptable MA.

The following sections explain the finding of each task.

3.1 A Model of an Automated Adaptable MA

The structure of the proposed MA is defined based on the research achievements of intelligent agent and BDI architecture (see Section 2.1.2). The main characteristics of intelligent agent from a general point of view are as follows:

- ***Autonomy*** When an agent runs, it needs not be controlled by human or any control mechanisms because the agent will control its own behavior and inner states in some degrees.

- ***Social capabilities*** An agent can exchange information with other agents by some agent communication languages.

- ***Reactivity*** An agent must have an ability to provide intelligent responses to the external events.

- ***Pro-activeness*** The action of an agent should be active or spontaneous.

Adaptation of the MA can be performed autonomously or externally. If the target environment is similar to the originating environment, adaptation can be carried out immediately. Otherwise, the MA must resort to its prior knowledge (in autonomous case) or consult a service agent (in external case) so that the intention can be fulfilled. The adaptation is said to complete. At which point, execution can commence according to the stated mission. The mechanisms which the intention is filled rest on structural, functional, and behavioral resemblance of the intention objects and what is available on the target environment. The selection process shifts toward the paradigm known as *service discovery*. Services can be viewed as standard commercial software without any application in mind. Generally, service and knowledge services are designed by a third party that allow the MA to fulfill its mission with pre-existing knowledge services. However, the literatures suggest that there have been growing interests in focusing on functional specification rather than on behavioral specification. It remains unclear whether the details about how the actual service works in each execution step covering user's intention. Such loose or lack of concern on service behavior often leads to nondeterministic agent execution. In order to arrive at some forms of deterministic capability, the behavior information is incorporated into the proposed model.

The framework of an abstract model for this proposed automated adaptable MA is designed based on the main characteristics of an intelligent agent defined by recent research achievements. Figure 3.1 shows the essential components of an adaptable MA which consists of:

- **intention component**: contains configurations about client's requirements and client's profile in machine-independent format.

- **mobility component**: provides the ability for autonomous mobility.

- **environment interaction**: enables reactivity and pro-activeness abilities of the MA.

- **social interaction**: provides the ability to exchange information with other agents; the new knowledge will be sent to update other components.

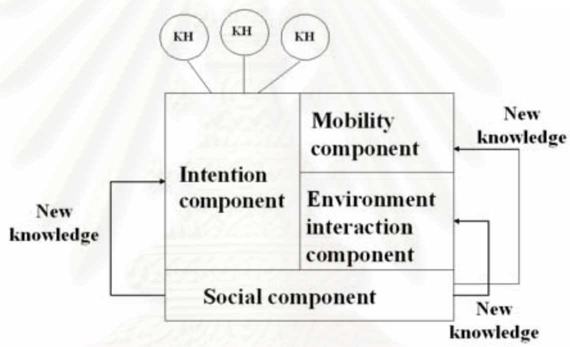

Figure 3.1: A framework model of an adaptable MA

In this work, the agent contacts a middle agent, called **service broker**, to determine the most suitable service according to the agent's **intention specification**, encompassing in an *intention component*. Upon receiving the intention specification, the service broker performs a sequence of service discovery using **matchmaking** procedures. This process provides recommendations for suitable services based upon **service advertisement**. As shown in Figure 3.1, the services, denoted by KH, are loaded from external service agents at runtime through the social interaction component. In so doing, the agent will be able to accomplish its intention in the dynamic and uncertain environments.

3.2 The Reference Architecture of an Automated Adaptable MA

As mentioned earlier, the nature of MA development is changing toward a distributive plug and play process. This requires a new way of managing MA by the so called software brokers. The objective of the broker is to configure a knowledge system for a customer. In other words, the broker provides the user with a reasoning service.

IBROW3 [38] proposes the three different possibilities as where to execute the configured knowledge system, that is, how to provide the reasoning service. They are:

- The broker retrieves the problem-solving methods (PSMs) from the libraries so that the customer can download the problem-solver from the broker. The customer's domain knowledge stays at customers site.

- The problem solver runs at the brokers site. The broker retrieves the PSMs from the libraries and takes care of interoperability problems (if the PSMs are implemented in different languages).

- The problem solver runs distributively. The broker does not physically retrieve the PSMs from the libraries, but they are executed in the libraries where they are found. Interoperability needs to be taken care of, i.e., by means of a middleware.

However, these reference architectures of knowledge system are suitable for the MA approach which encompasses mainly the concepts to move the computational part to any target hosts holding the data. To further enhance such an idea, a reference architecture is proposed in the following sections.

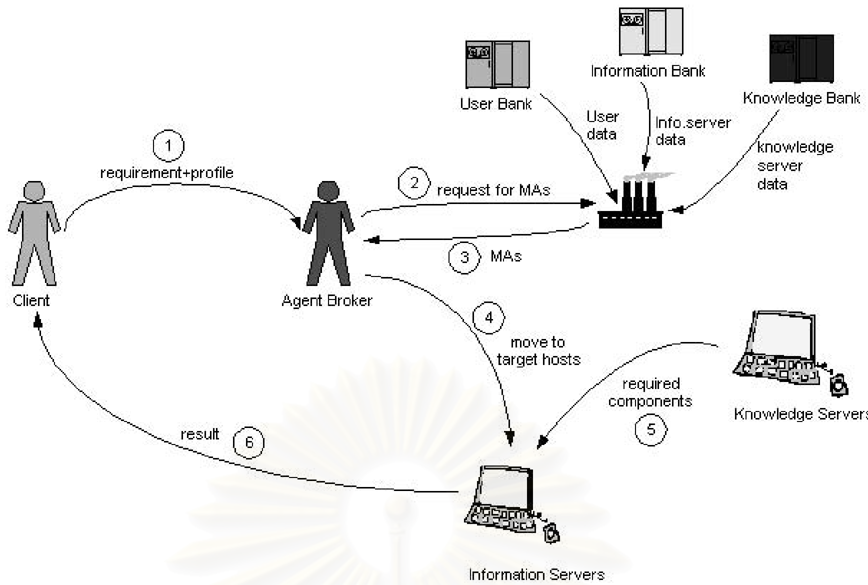

Figure 3.2: A reference architecture for an automated adaptable MA

3.3 The Reference Architecture

In this section, a reference architecture is proposed for suitable working with an MA as shown in Figure 3.2. The MAs themselves may be transmitted across computer networks and executed remotely. Based on this architecture, an MA is transmitted to the target host where data are processed. The results are then sent back to the originating host. A critical problem of this architecture concerns with the network bandwidth. To provide the quality of service (QoS) to the users, a two-phase intention specification adaptation is proposed as the MA design framework to cope with the network bandwidth problem.

- A deterministic gray box intention specification model is proposed as a model which encapsulates the MA's behavior in normalized form. This model is suitable for a network that is not congested. It provides a detailed validation assessment to enable higher result precision of the matchmaking process. This is described in Section 3.5.4.

- If the network is congested, which often leads low available bandwidth, a synopsis of deterministic intention specification is nominated in a compact form of MA's intention to alleviate the congestion problem. A rough validation assessment is supported in this model which is given in Section 3.5.7.

The system workflow of the reference architecture for the proposed MA is described as follows.

1. Client *sends* a request to the agent broker.
   - This request consists of the client's requirements and profile.
   - This request is handled by a Client Agent (CA) on the client machine.
2. The agent broker *assigns* a configuration for the MA. In this step, the desire, belief, and intention of each agent are defined.
   (a) The agent broker *decomposes* the client's requirements into tasks.
   (b) The agent broker *decides* the target hosts that provides the necessary resources for accomplishing the client's requirements.
   (c) The relevant context of each target host is *gathered*, including available network bandwidths.
   (d) The agent broker *establishes* the intention specification model for the MA based on the network bandwidth between the agent broker and the target host.
      - If the available bandwidth exceeds a predefined threshold, the broker applies the synopsis of deterministic intention specification to the MA.
      - Otherwise, the available bandwidth is lower than the predefined threshold and the deterministic gray box intention specification is used in the MA configuration.

3. The MAs are *generated* corresponding to the number of the designated target hosts.

   - Each MA specifies its missions based on defined tasks, the target host, and the configuration.

4. Each MA *migrates* to its target host.

5. The MA *requests* the necessary services the system's service broker based on its configuration. This is a service matchmaking process.

   (a) The MA contacts the service broker for its required services

   (b) The service broker matches the required services with the pre-registered services.

   (c) The service broker sends the appropriate services' detail to the requesting MA.

   (d) The MA requests the appropriate service from the service provider.

   (e) The requested services are *sent* back to the requesting MA.

   (f) The MA is *regenerated* from various sub-services.

   (g) The selected knowledge services are *translated*.

   (h) The regenerated MA *executes* its tasks.

6. After the processing finished, each MA returns to the server broker machine. The primary results of each MA are *acknowledged* to the service broker which controls the entire MA mission.

7. The final result is *replied* to the requesting Client.

3.3.1 A Sample Application of the Reference Architecture

The notion of MA can be applied to many application areas such as travel agent problem [39], e-learning, network management (see [40] for more detail). In this dissertation, a shopping assistant system is introduced. The system acts as a shopping assistant with the objective to providing the candidate stores for the customers to choose from, depending on their requirements and preferences. There are an agent broker that offers knowledge, such as shopping stores and customers' preferences, and a service broker that offer knowledge about service components. The customers, who have some shopping plan, are the target groups of this system. Some assumptions of the system are as follows:

- There is an agent broker that keeps the historical users' preferences, stores' profile.
- There is a service broker that controls service components' advertisement.
- Customers have individual preferences such as their behavior and preferred stores.
- The stores have their own shopping systems which are registered to the agent broker.
- Shopping assistant services are third party services and advertise their services to the service broker.

According to the reference architecture and the sample application, the state diagram of the system is presented in Figure 3.3 and the system work flow thus proceeds as follows:

1. When the customers need to go shopping, they must specify their shopping objectives, for examples, date, time, destination area, and shopping list.
2. Then, an appropriate customer desire is initiated at the agent broker. The customer leaves the finding duty to the broker.

3. The agent broker generates MAs, including belief, desire, and intention, based on the customer desire.

4. The MAs migrate to the store's systems.

5. Each MA requests its required services from the service broker.

6. The service broker matches the required services with the advertised services and informs the MA with the candidate services' detail.

7. The MAs request the candidate services from the service providers and regenerated themselves.

8. The MAs execute the necessary tasks.

9. The results are returned to the agent broker which in turn forwarded to the customer when the user is online.

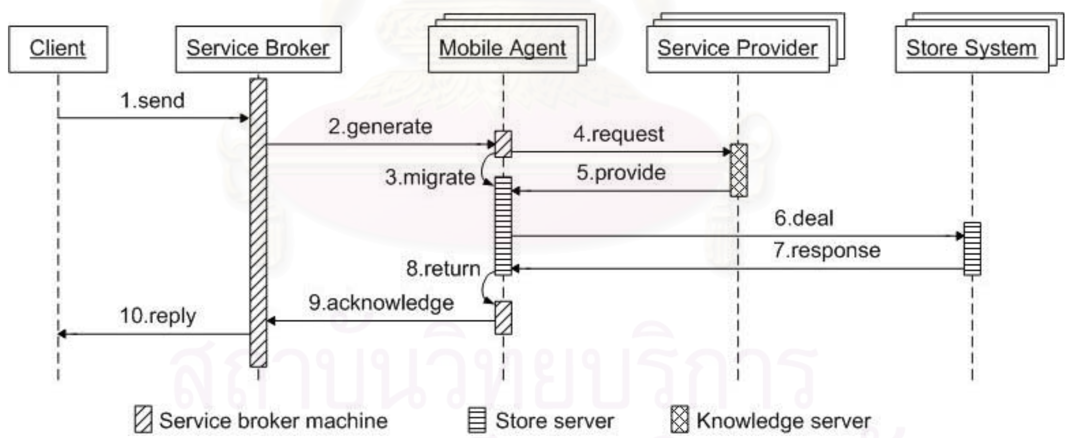

Figure 3.3: State diagram of the shopping assistant system

3.4 Summarizing Conventional Behavioral Specifications

In this section, some relevant definitions and nomenclatures of either nondeterministic behavioral specification, modeling in table-based service discovery approach, or deterministic behavioral specification, modeling in deductive service discovery approach are introduced. A few important aspects are also addressed.

> Let β be an agent intention specification describing particular tasks or duties to be undertaken by the MA. Let π be a service advertisement characterizing tasks which can be accomplished by that service.

1. A functional specification, $\mathcal{F}$, is a 4-tuple $(\mathcal{A}ct, \mathcal{P}re, \mathcal{P}ost, \mathcal{C}ons)$ defined as follows:
   (a) $\mathcal{A}ct$ describes functions of the required service.
   (b) $\mathcal{P}re$ denotes states that must be obtained in the world before the actions are taken.
   (c) $\mathcal{P}ost$ denotes states that will yield in the world once the actions have been implemented.
   (d) $\mathcal{C}ons$ encompasses a list of constraints which are associated with the intention.
2. A behavioral specification, denoted by $\mathcal{I}$, furnishes essential characteristics of the conventional approaches in two major classes, namely,
   (a) *Nondeterministic behavioral specification*, summarized in Section 3.4.1, and
   (b) *Deterministic behavioral specification*, summarized in Section 3.4.2.

3.4.1 Nondeterministic Behavioral Specification

A nondeterministic behavioral specification, denoted by $\mathcal{B}^{NDM}$, is a 5-tuple (Ω', Σ, $\mathcal{S}state$, $\mathcal{E}state$, δ), where

1. Ω' is a finite set of states;

2. Σ is a set of agent's contextual attributes or problem domain contextual attributes;

3. $\mathcal{S}state \in \Omega'$ denotes the *starting* state;

4. $\mathcal{E}state \subseteq \Omega'$ denotes a set of *ending* states; and

5. $\delta^*(\mathcal{S}state, \mathcal{P}) = \mathcal{E}State$ denotes the *transition relation* from the stating state to the designated ending state, with respect to the agent's contextual attributes.

Due to unpredictability of the MA's behavior, there may be situations that allow possible different moves by the MA in some unspecified way. In other words, the MA can guess a move.

MA is an adaptable agent carrying β to find the best matching service π in *nondeterministic way*, denoted by $\mathcal{M}A \overset{NDM}{\Longleftarrow} \pi$, if the following conditions hold true:

1. $\mathcal{P}^\pi \equiv \mathcal{P}^\beta$

2. $\mathcal{F}^\pi \equiv \mathcal{F}^\beta$

3. For any $\delta{*}^\beta(\mathcal{S}state, \mathcal{P}^\beta) = \mathcal{E}state^\beta$ and $\delta{*}^\pi(\mathcal{S}state, \mathcal{P}^\pi) = \mathcal{E}state^\pi$, $\mathcal{E}state^\beta \equiv \mathcal{E}state^\pi$

4. $\mathcal{P}re^\pi \equiv \mathcal{P}re^\beta$

5. $\mathcal{P}ost^\pi \equiv \mathcal{P}ost^\beta$

6. $\mathcal{C}ons^\pi \equiv \mathcal{C}ons^\beta$

The *matchmaking function* for a nondeterministic behavioral specification, denoted by $\Gamma^{\mathcal{N}DM}(\beta, \pi)$, is the function that performs a *relaxation matchmaking* over a *nondeterministic behavior specification.* A relaxation matchmaking is conducted according to the semantic-based service retrieval principles [25, 26, 27, 28, 29]. The result of $\Gamma^{\mathcal{N}DM}(\beta, \pi)$ will yield the most suitable advertised service π based on the given β.

The reliability degree of this matchmaking function can be defined in terms of the probability of user's purpose, denoted by $\vartheta^{\mathcal{N}DM}$.

Let $\delta^*(q_i, \mathcal{P}) = q_j$, where $q_i \neq \mathcal{S}state$ and $q_j \notin \mathcal{E}state$ be an *intermediate transition.* $\mathcal{X}^{\mathcal{N}DM}$ be a discrete random variable representing the possible purposeful degree of an agent executing by means of service π. The value that $\mathcal{X}^{\mathcal{N}DM}$ can take on has two aspects:

1. The value of $\mathcal{X}^{\mathcal{N}DM}$ over the *intermediate transitions* is given by

$$\mathcal{X}^{\mathcal{N}DM} = \begin{cases} 1 & \text{for exactly successful execution} \\ 0 & \text{for failure execution} \end{cases}$$

   According to the state transitions of $\mathcal{B}^{NDM}$, assessment of any explicitly specified intermediate execution results of π cannot be decided to be either successful or failure. Thus, the probability of $\mathcal{X}^{\mathcal{N}DM}$ falling in one of these two cases is 0.5.

2. The value of $\mathcal{X}^{\mathcal{N}DM}$ over the *final transition* is

$$\mathcal{X}^{\mathcal{N}DM} = \begin{cases} 1 & \text{for exact successful execution} \\ \nu & \text{for relaxation successful execution} \\ 0 & \text{for failure execution} \end{cases}$$

   where ν is the expected accuracy attained from $\Gamma^{\mathcal{N}DM}(\beta, \pi)$, $threshold \leq \nu \leq 1$, and ν is almost 1. For nondeterministic matchmaking process, the execution results of the *final transition* of service π will fall into just only the exact and relaxation successful execution.

Consequently, the reliability degree, $\vartheta^{\mathcal{N}DM}$, is determined by

$$\begin{aligned}\vartheta^{\mathcal{N}DM} &= \frac{\left[\sum_{i=1}^{n-1}\left(\sum_{X=0}^{1}\mathcal{X}P(X)\right)\right]+1}{n}\\ &= \frac{\left[\sum_{i=1}^{n-1}\left(0\times\frac{1}{2}+1\times\frac{1}{2}\right)\right]+1}{n}\\ &\approx 0.5\end{aligned}$$

However, it remains unconvinced by this nondeterministic model whether an agent encompassing with the recommended service conducted as $\Gamma^{\mathcal{N}DM}(\beta,\pi)$ is potent enough to accomplish its tasks within the agent's purposes.

3.4.2 Deterministic Behavioral Specification

The deductive service discovery approach [30, 32], introduced in component-based software engineering paradigm, offers a formal description for modeling service functionality and behavior. The behavioral specification can be represented as rule-based like structures. This kind of deterministic behavioral specification is similar to the way human experts express their decision makings, which decision statements are represented in both conjunctive normal form and disjunctive normal form. The explicit specific next state or next action of an agent basing upon an agent's contextual values is specified in a clear-sighted fashion. Essential characteristics of a specification and matchmaking function of this approach are summarized below.

A deterministic behavioral specification, denoted by $\mathcal{B}^{DM}$, is a 5-tuples (Ω', Σ, $\mathcal{S}state$, $\mathcal{E}state$, δ), where

1. Ω' is a finite set of states;
2. Σ is a set of agent's contextual attributes or problem domain contextual attributes;
3. $\mathcal{S}state \in \Omega'$ denotes the *starting* state;

4. $\mathcal{E}state \subseteq \Omega'$ denotes a set of *ending* states; and

5. $\delta : \Omega' \times (\Sigma) \rightarrow 2^{\Omega'}$ denotes the *transition* relation.

Determinism results form actions of the MA for the given decision situations that can be specified in a predictable way.

MA is an adaptable agent carrying β to find the best matching service π in *deterministic way*, denoted by $\mathcal{M}A \overset{\mathcal{D}M}{\Longleftarrow} \pi$, if these following conditions hold true:

1. $\mathcal{P}^{\pi} \equiv \mathcal{P}^{\beta}$

2. $\mathcal{F}^{\pi} \equiv \mathcal{F}^{\beta}$

3. For any $\delta^{\pi}(q_i, \mathcal{P}^{\beta}) = q_j$ and $\delta(q_i, \mathcal{P}^{\pi}) = q_k$, for each i, $q_k = q_j$

4. $\mathcal{P}re^{\pi} \equiv \mathcal{P}re^{\beta}$

5. $\mathcal{P}ost^{\pi} \equiv \mathcal{P}ost^{\beta}$

6. $\mathcal{C}ons^{\pi} \equiv \mathcal{C}ons^{\beta}$

The *matchmaking function* for this deterministic behavioral specification, $\Gamma^{\mathcal{D}M}(\beta, \pi)$, is the function that performs a *relaxation matchmaking* on a *deterministic behavior specification.* A relaxation matchmaking is conducted according to the semantic-based service retrieval principles. The result of $\Gamma^{\mathcal{D}M}(\beta, \pi)$ will yield the most suitable advertised service π based on the given β.

Moreover, the reliability degree of this matchmaking function can be defined as the probability of user's purpose satisfaction, denoted by $\vartheta^{\mathcal{D}M}$. The value of $\mathcal{X}^{\mathcal{D}M}$ over the *intermediate transition* is

$$\mathcal{X}^{\mathcal{D}M} = \begin{cases} 1 & \text{for exact successful execution} \\ \nu & \text{for relaxation successful execution} \\ 0 & \text{for failure execution} \end{cases}$$

where ν is the expected accuracy attained from $\Gamma^{\mathcal{D}M}(\beta, \pi)$, $threshold \leq \nu \leq 1$, and ν is almost 1. For nondeterministic matchmaking process, the execution results of the *transition* of service π will fall into just only the exact and relaxation successful execution. Then, $P(X = 1)$, $P(X = \nu)$ and $P(X = 0)$ are given by

$$\begin{aligned} P(X = 1) &= p \\ P(X = \nu) &= 1 - p - \epsilon \\ P(X = 0) &= \epsilon \end{aligned}$$

For general determinism matchmaking, the value of ϵ must be close to 0. The reliability degree, $\vartheta^{\mathcal{D}M}$, can be determined by

$$\begin{aligned} \vartheta^{\mathcal{D}M} &= 1(p) + \nu(1 - p - \epsilon) + 0(\epsilon) \\ &\approx p + \nu - \nu p \\ &\approx p + 1 - p \\ &\approx 1 \end{aligned}$$

However, some inherent problems concerning the MA operating in uncertainty environment still persist. This is, because, the uncertainty environment notably differs from the certainty environment in which under uncertainty environment the resources are not all known or adequately characterized in advance. Thus, the characteristics of the required services and the available services operating under uncertainty environment may change with time based on contextual values. The MA is not able to anticipate all possible actions that it would be in order to react under the uncertainty environment. Thus, these problems will be addressed along with a novel approach to establish the solutions. Moreover, this framework will in turn accomplish the following additional aspects:

- An architectural framework for light-weighted MA.
- An MA potent enough to accomplish its missions with a wide array of target hosts within the acceptable users' satisfaction. In other words, the value of ϑ acquired from the proposed model must be close to 1, or 100%.

3.5 Conceptual Model for Deterministic Behavioral Specifications

The emphasized of this research relies on making some progress toward the conventional table-based specification encompassing deterministic behavioral specification. A matchmaking procedure is conducted according to the semantic-based service retrieval principles [25, 26, 27, 28, 29] which the standard models of knowledge representation, taxonomies, vocabularies, and domain terminologies are explicit studied in the field of ontological engineering. This study offers some insight to the behavioral specification over dynamic and unpredictable environments. The processes of deriving such a determinism behavior specification are carried out in the sections that follow.

3.5.1 Terminology

In order to model the behavioral specification of an agent intention, some related terminologies are defined as follows:

Definition 1. ***Set of concerning attributes***

A set of concerning attributes for each **object domain**, $\mathcal{A}$, is defined as follows:

$$\mathcal{A} = \{\mathcal{A}(1), \mathcal{A}(2), \ldots, \mathcal{A}(\rho)\},$$

where ρ is a number of concerning attribute in each domain.

Definition 2. ***Set of concerning attribute values***

A set of concerning attribute values for any concerning attribute, $\mathcal{A}(k)$, is defined as follows:

$$\mathcal{A}(k) = \left\{a(k)^1, a(k)^2, \ldots, a(k)^{p_k}\right\},$$

where p_k is a number of concerning attribute values belonging to attribute $\mathcal{A}(k)$.

This definition implies an important characteristic that the proposed model does not handle continuous attributes, thus these concerning attribute values need to be discretized beforehand. In order to succinctly refer to a *concerning attribute*, $\mathcal{A}(k)$ having the value $a(k)^r$, or $\mathcal{A}(k) = a(k)^r$. We will refer to this expression as a *column number* of a *column vector* which will be defined below.

Definition 3. ***Attribute column vector***

An attribute column vector, denoted as $\mathcal{A}'$. The position of an attribute $\mathcal{A}(k)$ having the value $a(k)^r$, denoted as column jth in vector $\mathcal{A}'$, can be calculated by

$$j = \sum_{x=1}^{k-1} \|\mathcal{A}(x)\| + r \tag{3.1}$$

where $\|\mathcal{A}(x)\|$ is the number of concerning attribute values belonging to attribute $\mathcal{A}(x)$.

Definition 4. ***Number of attribute values***

The number of attribute values or decision criteria, hereafter referred to as $\mathcal{N}$, is given by

$$\mathcal{N} = \sum_{k=1}^{\rho} p_k$$

where p_k is the number of concerning attribute values belonging to $\mathcal{A}(k)$, ρ is the number of concerning attributes in each domain.

Definition 5. ***Number of possible conditional statements***

The number of possible condition statements, hereafter referred to as $\mathcal{M}$, is as follows:

$$\mathcal{M} = \prod_{k=1}^{\rho} p_k$$

where p_k is the number of concerning attribute values belonging to $\mathcal{A}(k)$.

Definition 6. ***Number of declared conditional statements***

Let α represent the number of declared conditional statements. In case of a **complete behavioral specification**, the value of α is equal to $\mathcal{M}$. By contrast, for an **incomplete behavioral specification** whose partial situations are specified, the variable α is the number of declared situations.

Definition 7. ***Conditional statement matrix***

Let $\widetilde{\mathcal{C}}$ be a set of conditional statements encompassing ordered pairs produced by a *cross-product* of sets $\mathcal{A}(1) \times \mathcal{A}(2) \times \ldots \times \mathcal{A}(\rho)$ taken the form

$$\widetilde{\mathcal{C}} = \{(a(1), a(2), \ldots, a(p)) \mid a(1) \in \mathcal{A}(1), a(2) \in \mathcal{A}(2), \ldots \text{ , and } a(p) \in \mathcal{A}(\rho)\}$$

A conditional statement matrix of $\alpha \times \rho$, denoted by $\mathcal{C}$. For a **complete behavioral specification**, $\mathcal{C}^c$ is used in place of $\mathcal{C}$, where $\mathcal{C}^c$ is an $\mathcal{M} \times \rho$

matrix. Assign $\mathcal{C}(i,k) = r$, where $\mathcal{A}(k)$ having the value $a(k)^r$ and r is an index of a concerning value. Thus, $\mathcal{A}(k) = a(k)^r$ is a concerning attribute value of the condition statement $\mathcal{C}(i)$.

Definition 8. ***Decision making situation matrix***

Let δ be an $\alpha \times \mathcal{N}$ decision making situation matrix. Using Definition 3, $\mathcal{A}(k) = a(k)^r$ can be represented as the j^{th} element in column vector $\mathcal{A}'$. The index j can be calculated from Equation 3.1. $\delta(i,j)$ is equal to 1 if $\mathcal{A}(k) = a(k)^r$ is a concerning attribute value of a condition statement $\mathcal{C}(i)$. Otherwise, $\delta(i,j)$ is 0.

Definition 9. ***Decision action matrix***

A decision action matrix, Ω', is an $\alpha \times 1$ matrix representing decision actions reacted by an agent based on the given decision situations. Ω' takes the form

$$\Omega' = \begin{bmatrix} \Omega'(1) \\ \Omega'(2) \\ \cdots\cdots \\ \Omega'(\alpha) \end{bmatrix}$$

In order to clarify the meaning of these definitions, an example is given in Table 3.5.1. Some instances of relevant elements are demonstrated below.

- Let $\mathcal{A}(1), \mathcal{A}(2), \mathcal{A}(3)$, and $\mathcal{A}(4)$ denote outlook, temperature, humidity, and windy attributes, respectively. The set of concerning attributes is written as

$$\mathcal{A} = \{\mathcal{A}(1), \mathcal{A}(2), \mathcal{A}(3), \mathcal{A}(4)\}.$$

  In addition, a set of concerning attribute values, say $\mathcal{A}(1)$, can take the form

$$\mathcal{A}(1) = \{overcast,\ rain,\ sunny\}.$$

Table 3.1: The golf data set

The golf data set					
Cases	Outlook	Temperature	Humidity	Windy	Actions
$\mathcal{C}(1)$	sunny	hot	high	false	Don't Play
$\mathcal{C}(2)$	sunny	hot	high	true	Don't Play
$\mathcal{C}(3)$	overcast	hot	high	false	Play
$\mathcal{C}(4)$	rain	mild	high	false	Play
$\mathcal{C}(5)$	rain	cool	normal	false	Play
$\mathcal{C}(6)$	rain	cool	normal	true	Don't Play
$\mathcal{C}(7)$	overcast	cool	normal	true	Play
$\mathcal{C}(8)$	sunny	mild	high	false	Don't Play
$\mathcal{C}(9)$	sunny	cool	normal	false	Play
$\mathcal{C}(10)$	rain	mild	normal	false	Play
$\mathcal{C}(11)$	sunny	mild	normal	true	Play
$\mathcal{C}(12)$	overcast	mild	high	true	Play
$\mathcal{C}(13)$	overcast	hot	normal	false	Play
$\mathcal{C}(14)$	rain	mild	high	true	Don't Play

- An attribute column vector $\mathcal{A}'$ thus becomes

$$\mathcal{A}' = \left[\overbrace{sunny\ overcast\ rain}^{outlook}\ \overbrace{hot\ mild\ cool}^{temperature}\ \overbrace{high\ normal}^{humidity}\ \overbrace{false\ true}^{windy} \right]$$

- The number of overall attribute values can be calculated by $\mathcal{N} = 3+3+2+2 = 10$. Similarly, the number of all possible condition statements becomes $\mathcal{M} = 3 \times 3 \times 2 \times 2 = 36$. The value of α is valued of 14.

- An arbitrary proposition as "*if* outlook=overcast *and* temperature=hot *and* humidity=high *and* windy=false *then* an action will be Play", consists of two main parts as follows:

  1. "outlook=overcast *and* temperature=hot *and* humidity=high *and* windy=false" are ordered pairs specified in $\widetilde{\mathcal{C}}$, such that

     $$\widetilde{\mathcal{C}}(i) = [overcast\ hot\ high\ false]$$

     Thus the i^{th} row in matrix $\mathcal{C}$ can be written as

     $$\mathcal{C}(i) = [2\ 3\ 1\ 1]$$

     This information infers that the i^{th} row of matrix δ becomes

     $$\delta(i) = [0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1]$$

  2. An action "Play" is an instance of one element in set Ω'.

3.5.2 Fundamentals

The emphasis of this research rests on the notion of a *conventional nondeterministic specification.* There are four crucial points that need to be addressed.

1. The uncertainty environment notably differs over the certainty environment in which under uncertainty environment the resources are not all known or adequately characterized in advance. Thus, the characteristics of the required services and the available services operating under uncertainty environment may change based on contextual values with time. The agent is not able to anticipate all possible actions that they would turn out in order to react under the uncertainty environment. Moreover, the set of states or actions tend to be **infinite**.

2. When designing the agent, it is difficult to anticipate all complete situations the agent will encounter. In most cases, the agent intention specifications have a tendency to be incomplete. In contrast, service advertisements are mostly specified in the form of the complete specifications. As such, the values of α^{β} and α^{π} are not equal. This case is noted as an *inequivalent-α matchmaking*. Otherwise, when the α^{β} and α^{π} are equal, the matchmaking is referred to as an *equivalent-α matchmaking*. In addition, conventional specification matching is performed over fixed number of attributes, or white box specification in this literature. Any unspecified or new attributes are considered indescribable which, from theoretical standpoint, must be extended (scaling) to accommodate the extraneous matching requirements. Unfortunately, scalable capability is unlikely. Such a limitation imposes difficulties in dealing with those indescribable situations. Thus, the **scalability over incomplete declared intention** is taken into account for modeling the behavioral specification.

3. The resources under agent environments are referred to as the **condition criteria**. Despite the characteristics of an agent executing under uncertainty environment, a variety of unknown factors may not be adequately characterized in advance and may change with time. The similarity between β and π needs a step further to

look over the equivalent level between the numbers of concerning attributes. The characteristics of matchmaking process can be stated in three categories based on gap analysis technique. Gap analysis [41, 31] technique is an approach for determinating the recorded discrepancies of capability specified in the agent intention specifications that are not fulfilled by the selected service. Figure 3.4 shows the results of gap analysis evaluation.

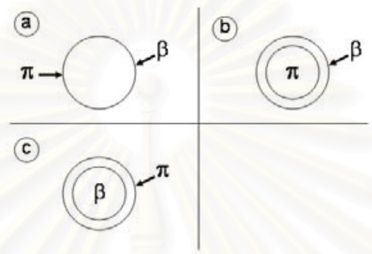

Figure 3.4: Results of gap analysis evaluation

Hence, the definition of service matchmaking result stated in the gap analysis is modified by extending the concepts of attributes matchmaking relation between β and π. Thus, there are three possibilities that could be obtained from the above measurement with regard to the attributes of matchmaking relation between β and π, namely,

(a) **Exact-match:** denotes the capabilities of an element in π that *match exactly* with an element of the corresponding requirements in β and the relationship of concerning attributes between β and π is characterized as one-to-one relationship.

(b) **Plug-in-match:** denotes the capabilities of an element in π that *partially fulfill* with an element of the corresponding requirements in β and does not provide any inherent capabilities that exceed the above requirements.

(c) **Subsume-match:** denotes the capabilities of an element in π that *fulfill all* of the requirements of an element in β, but also incorporates additional capabilities that exceed the original system.

In order to cope with these problems, a **scalability over inequivalent decision criteria** is served as the fundamental principles that called for.

4. Modeling such an intelligent MA that can adapt itself to any foreign environments, all type of possible scenarios must be taken into account which unfortunately leads to an imploded MA. To move an MA over the network, the MA must be compacted to reduce transmission bandwidth. Thus, the size of intention specification is taken into account as one important factor for modeling the proposed intention specification.

3.5.3 A Deterministic White Box Intention Specification

In order to overcome an **infinite set** of possible actions, the crucial preamble assumption of this proposed approach stipulates that an infinite set of possible actions might be reduced to the limited values or categories. Thus, the elements in the set of possible actions are reduced to the user's attitude toward the particular decision criteria. The definition of attitude is defined in [42] as follows:

> *Attitude can be defined as a predisposition that is learned in order to respond in a consistent way, either in favorable or unfavorable manner toward a specific object.*

Hence, the qualitative judgment of each condition statement is formalized by a bipolar attitude scale as **favorable** and **unfavorable** classes toward the agent's decision (that is formulated from the users attitude) acting on behalf of the user. In so doing, possible actions stating in this proposed behavioral specification model can be narrated in discrete or finite terms. Obviously, using discrete/finite terms is more practical than descriptive/infinite terms. The necessary preamble assumption of this proposed approach stipulates that an infinite set of possible actions could be reduced to a corresponding finite set of values or categories. Thus, a decision action matrix, Ω', can be arranged in a new decision actions matrix, Ω, which takes the form

$$\Omega(i) = \begin{cases} 1 & \text{if } \Omega'(i) \in FV\ \mathcal{C}lass \\ \\ 0 & \text{if } \Omega'(i) \in UFV\ \mathcal{C}lass \end{cases}$$

In order to facilitate a practical matchmaking process, a behavioral model in logical view, defined as state transition definition, are refined to physical form, as matrix representation. A *conventional deterministic intention specification* is organized as a *white box deterministic intention specification* defined in a definition given below.

Definition 10. *A white box deterministic intention specification*

Let $\mathcal{I}^{WB}$ denote a white box deterministic intention specification of either β or π. This specification is defined as (δ, Ω), where

1. δ is an $\alpha \times \mathcal{N}$ matrix. $\mathcal{N}$ is the number of concerning attribute values. If the j^{th} attribute value represents the concerning attribute of i^{th} condition, $\delta(i, j)$ set to 1. Otherwise, $\delta(i, j)$ is assigned with 0.

2. Ω is an $\alpha \times 1$ matrix. For any i^{th} row in an $\alpha \times 1$ matrix Ω, $\Omega(i)$ represents a class of actions (the i^{th} row of Ω) which are relevant to a condition statement $\delta(i)$. $\Omega(i)$ is set to 1 if the condition $\delta(i)$ is $FV\ \mathcal{C}lass$, otherwise $\Omega(i)$ is 0.

Denote the instance $\mathcal{I}$ of $\mathcal{I}^{WB}$ by $\mathcal{I}.\delta$ and $\mathcal{I}.\Omega$ for matrix δ and matrix Ω which represent the elements of $\mathcal{I}^{WB}$ specification. In case of a **complete deterministic white box**

specification, all possible situations must be specified or assessed with their relevant actions. Thus, the variable α is equal to $\mathcal{M}$, where $\mathcal{M}$ is a number of all possible situations. By contrast, for an **incomplete deterministic white box specification** whose partial situations are specified with their relevant actions, the variable α is equal to the number of declared situations.

3.5.4 A Deterministic Gray Box Intention Specification

One step toward modeling such a deterministic intention is proposed relies on the basic idea of **inductive learning** approach. One of the most successful machine learning algorithm that can be conducted over **inductive learning** approach is back-propagation (BP) technique, reported in [43]. The primitive form of this inductive back-propagation learning is of the form

$$\Omega = f(\mathcal{W}\mathcal{C} + b),$$

where a is a matrix representation of specific goals or actions, $\mathcal{W}$ and b are both *adjustable* parameters. The parameter $\mathcal{W}$ is an $S \times \mathcal{N}$ representing a weight matrix, where S is the number of network neurons and $\mathcal{N}$ is the number of inputs. According to *parameters adjusting step*, the constraint which must hold true before terminating learning process is specified as parameters $\mathcal{W}$ and b are adjusted until the input/output relationship meets the specific goals.

By applying inductive learning approach, the output of this process is coming in some form of the deterministic behavioral specification. A new or unknown decision making situations over an incomplete declared specification can be inducted from these specification parameters. The definition of a deterministic specification by induction learning is defined by the following definition.

Definition 11. ***A deterministic gray box specification***

Let $\mathcal{I}^{GB}$ denote a deterministic gray box specification of either β or π. This specification comprises of 2-tuples defined as $(\mathcal{W}, b)$, where

1. $\mathcal{W}$ represents an adjusted learning **weight** matrix

2. b represents an adjusted learning **bias** matrix

In order to add up scalable capabilities over an *incomplete declared* intention, an **ordinary deterministic white box specification**, which can be either a complete or an incomplete specification, is normalized to a proposed deterministic behavioral specification. The proposed specification, or a **deterministic gray box specification**, is originated from a *machine learning* process, hereafter referred to as a **normalized deterministic behavior specification**. The learning parameters characterizing the behavioral of a service are obtained from machine learning process in terms of *learning parameters*. These parameters are embedded as the attribute values of the proposed specification. To elaborate the values of these parameters for this kind of specification, not only the constraint for parameters adjusting step is concerned, but also the characteristics preserving issue must be taken into account. Thus, the constraints for a deterministic gray box behavioral elaborating algorithm become

- 1^{st} **constraint:** The parameter adjusting step must be performed until the input/output relationship meets the specific goals.

- 2^{nd} **constraint:** The principal characteristics of the ordinary specification must be preserved, especially for the predominant ones.

Before any matchmaking commences, *complete deterministic white box specifications* with new or unknown decision making situations are **inducted** from the learning parameters. These forthcoming specifications are called **scaled intention specifications**.

The agent intention specification is called β and the service advertisement is referred to as π. Both scaled-β and scaled-π are arranged in matrix format of unequal size. These matrices are transformed by the proposed algorithm to equal-sized matrices. Consequently, the indescribable problems become scaled intention specification.

Back-propagation learning algorithm [44] is selected as a behavioral elicitation algorithm, using multilayers feed-forward network with two-layers $log - sigmoid$ transfer function. The input vector, δ, represents the declared situations in matrix form operating on three neurons. The output layer utilizes one neuron. The hidden layer, encompassing a log-sigmoid transfer function, employs one neuron. A training function based on *conjugate gradient back-propagation with Fletcher-Reeves updates* is also embedded in the hidden layer.

> Let $\bar{\mathcal{I}}$, $\hat{\mathcal{I}}$, and $\mathcal{I}'$ denote the generic (unspecified to β or π) deterministic white box specification, the normalized deterministic behavior specification, and the scaled intention specification, respectively. $\bar{\mathcal{I}}$ and $\mathcal{I}'$ are structured in the form of $\mathcal{I}^{WB}$, whereas $\hat{\mathcal{I}}$ is arranged in $\mathcal{I}^{GB}$. Denote $\Theta^{\bar{\mathcal{I}},\hat{\mathcal{I}}}$, $t\Theta$, and $maxIter$ as the accuracy of normalization process, acceptable value of the discrepancy degree, and the maximum iteration for performing a normalization process, respectively. The default values of $t\Theta$, $t\nabla$ and $maxIter$ are set to 90%, 0.1, and 20, respectively.

A pivotal process of this algorithm is $\bar{\mathcal{I}}$-$\hat{\mathcal{I}}$ normalization to determine $\hat{\mathcal{I}}$ which, in general, approximately is equal or close to the value specified in $\bar{\mathcal{I}}.\Omega$. Thus, the results of this step yield components of *temporary* $\hat{\mathcal{I}}$, namely,

1. $\mathcal{W}^1$ is a weight matrix of the input (1^{st}) layer represented by an $S \times \mathcal{N}$ matrix,

2. $\mathcal{W}^2$ is a weight matrix of the hidden (2^{nd}) layer represented by a $1 \times S$ matrix,

3. b^1 is a bias matrix of the input (1^{st}) layer represented by an $S \times 1$ matrix; and,

4. b^2 is a bias matrix of the hidden (2^{nd}) layer represented by a 1×1 matrix

where S is the number of learning neurons.

A deterministic gray box behavioral elaborating algorithm

1. Create $\bar{\mathcal{I}}$ consisting of matrices $\bar{\mathcal{I}}.\delta$ and $\bar{\mathcal{I}}.\Omega$ from an ordinary rule-based specification, which is a deterministic white box intention specification.

   Let $\bar{\mathcal{I}}.\delta$ be a condition part, denoted by an $\alpha \times \mathcal{N}$ matrix. The matrix $\bar{\mathcal{I}}.\delta$ is given by

$$\bar{\mathcal{I}}.\delta(i,j) = \begin{cases} 1 & \text{if } \mathcal{A}'(j) \in \bar{\mathcal{I}}.\mathcal{C}(i) \\ 0 & \text{if } \mathcal{A}'(j) \notin \bar{\mathcal{I}}.\mathcal{C}(i) \end{cases}$$

   Similarly, let $\bar{\mathcal{I}}.\Omega$ be an action part, denoted by an $\alpha \times 1$ matrix. The matrix $\bar{\mathcal{I}}.\Omega$ can be expressed as

$$\bar{\mathcal{I}}.\Omega(i) = \begin{cases} 1 & \text{if } \bar{\mathcal{I}}.\Omega(i) \in FV\ \mathcal{C}lass \\ 0 & \text{if } \bar{\mathcal{I}}.\Omega(i) \in UFV\ \mathcal{C}lass \end{cases}$$

2. Create a *complete decision making situation* matrix

   Let δ^c be an $\mathcal{M} \times \mathcal{N}$ matrix representing the complete decision making situations. The matrix δ^c takes the form

$$\delta^c(i,j) = \begin{cases} 1 & \text{if } \mathcal{A}'(j) \in \mathcal{C}^c(i) \\ 0 & \text{if } \mathcal{A}'(j) \notin \mathcal{C}^c(i) \end{cases}$$

3. **REPEAT**

(a) Normalize $\bar{\mathcal{I}}$ to $\hat{\mathcal{I}}$ by creating a *2-layers feed forward* network with *log-sigmoid* transfer function

$$(1^{st}\text{ layer:})\ \mathcal{Y} = \frac{1}{1+e^{-\mathcal{W}^1\bar{\mathcal{I}}.\delta+b^1}}$$

$$(2^{nd}\text{ layer:})\ \hat{\mathcal{I}}.\Omega = \frac{1}{1+e^{-\mathcal{W}^2\mathcal{Y}+b^2}}$$

(b) Compute the *compatible degree* $\Theta^{\bar{\mathcal{I}},\hat{\mathcal{I}}}$ of reusing $\hat{\mathcal{I}}$ in place of $\bar{\mathcal{I}}$ by means of Equation 3.2.

(c) Scale up $\hat{\mathcal{I}}$ to $\mathcal{I}'$ according to δ^c. The scaling is carried out as follows:

$$\mathcal{Y} = \frac{1}{1+e^{-\hat{\mathcal{I}}.\mathcal{W}^1\delta^c+\hat{\mathcal{I}}.b^1}}$$

$$\mathcal{I}'.\Omega = \frac{1}{1+e^{-\hat{\mathcal{I}}.\mathcal{W}^2\mathcal{Y}+\hat{\mathcal{I}}.b^2}}$$

(d) Calculate the *discrepancy degree* $\nabla_{\mathcal{E}}^{\bar{\mathcal{I}},\mathcal{I}'}$ of $\bar{\mathcal{I}}$ and $\mathcal{I}'$ based on Equation 3.5.

(e) **IF** $[iteration = 1]$ **OR** $\left[\Theta^{\bar{\mathcal{I}},\hat{\mathcal{I}}} < min(\Theta^{\bar{\mathcal{I}},\hat{\mathcal{I}}})\right]$

OR $\left[(\Theta^{\bar{\mathcal{I}},\hat{\mathcal{I}}} = min(\Theta^{\bar{\mathcal{I}},\hat{\mathcal{I}}}))\ \textbf{AND}\ (\nabla_{\mathcal{E}}^{\bar{\mathcal{I}},\mathcal{I}'} < min(\nabla_{\mathcal{E}}^{\bar{\mathcal{I}},\mathcal{I}'}))\right]$

i. $min(\Theta^{\bar{\mathcal{I}},\hat{\mathcal{I}}}) = \Theta^{\bar{\mathcal{I}},\hat{\mathcal{I}}}$

ii. $min(\nabla_{\mathcal{E}}^{\bar{\mathcal{I}},\mathcal{I}'}) = \nabla_{\mathcal{E}}^{\bar{\mathcal{I}},\mathcal{I}'}$

iii. Set output $\hat{\mathcal{I}} = \hat{\mathcal{I}}$

ENDIF

(f) Increase the value of *iteration* by 1

UNTIL $\left[(\Theta^{\bar{\mathcal{I}},\hat{\mathcal{I}}<t\Theta)\ \textbf{AND}\ (\nabla_{\mathcal{E}}^{\bar{\mathcal{I}},\mathcal{I}'}<t\nabla))}\right]$ **OR** $[iteration > maxIter]$

According to the processes specified in this algorithm, the following conditions must hold true:

1. $\Theta^{\bar{\mathcal{I}},\hat{\mathcal{I}}}$ must be equal or close to 100%, i.e., $\bar{\mathcal{I}}.\Omega \equiv \hat{\mathcal{I}}.\Omega$, that the decision making of $\hat{\mathcal{I}}$ according to the information specified in $\bar{\mathcal{I}}$ must be equivalence. This condition corresponds to the 1^{st} *constraint*.

2. $\nabla_{\mathcal{E}}^{\hat{\mathcal{I}},\mathcal{I}'}$ must be equal or close to 0. The entropy of the original specification $\bar{\mathcal{I}}$ and the assessment of $\mathcal{I}'$ must be almost equal. This condition respects to the 2^{nd} *constraint*.

For illustrating purpose based on the sample data in Table 3.5.1, the inputs of $\mathcal{I}^{WB}$ are

$$\bar{\mathcal{I}}.\delta = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \quad \bar{\mathcal{I}}.\Omega = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

and the output $\hat{\mathcal{I}}$, or $\mathcal{I}^{GB}$, becomes

$$\hat{\mathcal{I}}.\mathcal{W}^1 = \begin{pmatrix} -0.71 & 5.1 & -2.6 & 2.2 & 3.2 & -0.5 & -0.9 & -2.5 & -0.5 & -1.9 \\ -0.3 & 1.2 & 0.9 & 2.2 & -2.8 & -0.8 & -4.9 & 1.4 & 2.8 & -0.2 \\ 2.7 & 2.6 & -1.7 & -2.3 & 1.8 & 3.5 & -1.9 & 2.0 & 1.1 & 2.4 \\ 3.3 & -3.4 & -2.3 & 1.2 & -1.0 & 2.3 & 0.4 & -2.7 & 2.6 & -1.5 \\ 0.2 & 1.7 & -2.6 & 3.5 & 0.3 & 2.1 & 2.1 & -3.7 & -3.6 & 3.1 \\ 0.7 & 3.0 & -3.2 & -3.6 & 2.1 & -2.3 & -1.6 & 1.8 & 3.98 & -1.8 \\ 3.5 & -4.6 & -1.8 & 2.2 & 2.2 & 2.3 & -1.3 & 1.2 & -1.9 & 1.5 \\ 1.4 & 0.2 & 2.9 & -1.8 & -1.6 & 3.3 & 3.1 & 0.7 & 3.3 & 1.0 \\ -4.1 & -2.5 & 3.7 & 1.1 & 0.7 & -1.0 & -1.9 & 1.8 & 0.1 & 2.5 \\ 0.5 & 0.2 & 0.8 & -0.4 & 2.5 & 4.0 & -3.2 & 3.6 & -1.0 & 1.2 \end{pmatrix}$$

$$\hat{\mathcal{I}}.\mathcal{W}^2 = \begin{pmatrix} 3.2 & 1.4 & 3.2 & -4.8 & -5.7 & 6.8 & -4.7 & -1.0 & -3.2 & 3.1 \end{pmatrix}$$

$$\hat{\mathcal{I}}.b^1 = \begin{pmatrix} -4.2 \\ 3.3 \\ -7.4 \\ -2.4 \\ 3.0 \\ 2.2 \\ 0.4 \\ -4.6 \\ -4.5 \\ -2.1 \end{pmatrix} \hat{\mathcal{I}}.b^2 = \begin{pmatrix} 2.5 \end{pmatrix}$$

3.5.5 A Detailed Validation Assessment

The detailed quantitative measurement for the **compatibility degree of reusing a service** π **in place of** β can be derived from [33] as follows:

$$\Theta^{\beta,\pi} = \frac{\left[\sum_{i=1}^{\alpha} \Delta\Big(\beta.\Omega(i), \pi.\Omega(i)\Big)\right]}{\alpha} \times 100 \tag{3.2}$$

where α denotes the number of the *describable* cases which can be assessed by a minimum value between $\alpha^{\mathcal{I}_a}$ and $\alpha^{\mathcal{I}_b}$. $\Theta^{\mathcal{I}_a,\mathcal{I}_b}$ is the percentage of compatible degree between $\mathcal{I}_a$ and $\mathcal{I}_b$ (how many elements of $\mathcal{I}_a$ match/similar to $\mathcal{I}_b$). The compatibility value for the i^{th} conditional statement, $\Theta_i^{\mathcal{I}_a,\mathcal{I}_b}$, can be expressed as follows:

$$\Delta(\beta.\Omega(i), \pi.\Omega(i)) = \begin{cases} 1 & \text{if } \beta.\Omega(i) = \pi.\Omega(i) \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where $\beta.\Omega(i) = \pi.\Omega(i)$ if either ($\beta.\Omega(i) \in FV$ and $\pi.\Omega(i) \in FV$) or ($\beta.\Omega(i) \in UFV$ and $\pi.\Omega(i) \in UFV$). $\Delta(\beta.\Omega(i), \pi.\Omega(i))$ determine the value in the interval $[0 : 1]$.

Furthermore, the ability for preserving the principal characteristics of β is called a **resemblance degree**, which can be quantified as

$$\Gamma^{\beta,\pi} = 1/[1 + \nabla_{\mathcal{E}}^{\beta,\pi}] \tag{3.4}$$

where $\nabla_{\mathcal{E}}^{\beta,\pi}$ is the degree of discrepancy between the capabilities specified in β and π given by

$$\nabla_{\mathcal{E}}^{\beta,\pi} = \sum_{j=1}^{n} [\nabla_{\mathcal{E}}^{\beta,\pi}(j)] \tag{3.5}$$

where

1. $\nabla_{\mathcal{E}}^{\beta,\pi}(j)$ can be assessed as

$$\nabla_{\mathcal{E}}^{\beta,\pi}(j) = \begin{cases} 4.757 * \left|\mathcal{P}^{\beta}(j) - \mathcal{P}^{\pi}(j)\right| & \text{if } \left|\mathcal{P}^{\beta}(j) - \mathcal{P}^{\pi}(j)\right| > 0.4 \\ e^{[\mathcal{E}^{\beta}(j) - \mathcal{E}^{\pi}(j)]} \times \left|\mathcal{P}^{\beta}(j) - \mathcal{P}^{\pi}(j)\right| & \text{otherwise} \end{cases}$$

2. $\mathcal{E}^{\beta}(j)$ and $\mathcal{E}^{\pi}(j)$ are the entropy of $\mathcal{I}_a$ and $\mathcal{I}_b$ over a concerning attribute value j^{th}. $\mathcal{E}(j)$ of either β, π is given by

$$\begin{aligned} \mathcal{E}(j) &= -([\mathcal{P}(j) \times log_2\mathcal{P}(j)] \\ &\quad +[(1 - \mathcal{P}(j)) \times log_2(1 - \mathcal{P}(j))]) \end{aligned} \tag{3.6}$$

3. $\mathcal{P}^{\beta}(j)$ and $\mathcal{P}^{\pi}(j)$ are the probability of $\mathcal{I}_a$ and $\mathcal{I}_b$ over a concerning attribute value j^{th} toward the FV class. $\mathcal{P}(j)$ of either β, π becomes

$$\mathcal{P}(j) = \frac{\text{number of } \mathcal{A}(j) \text{ belonging to } FV \text{ class}}{\text{number of occurrences } \mathcal{A}'(j)} \tag{3.7}$$

3.5.6 An Extended-$\mathcal{I}^{GB}$ Elicitation Algorithm

In order to enable *scalability over inequivalent decision criteria* to the proposed $\mathcal{I}^{GB}$, the extended elicitation algorithm is presented. An input of this algorithm can be either a normalized intention specification, previously noted by $\hat{\beta}$, or a normalized service advertisement, previously noted as $\hat{\pi}$. An output of this algorithm is a normalized specification.

Instead of conducting an iterative learning process, this algorithm takes action in order to add some information to the primer $\mathcal{I}^{GB}$ derived from the learning process. The $\mathcal{I}^{GB}$ will embed some neutral values, such as 0 or 1.

An extended-$\mathcal{I}^{GB}$ elicitation algorithm

1. In case of **Exact match**, where $\mathcal{N}^{\beta} = \mathcal{N}^{\pi}$, there is no further process in this phase.

2. For **Plug-in match**, where $\mathcal{N}^{\beta} \succ \mathcal{N}^{\pi}$, some additional process are called for in order to extend the size of $\hat{\pi}$ from $\mathcal{N}^{\pi}$ to $\mathcal{N} = \mathcal{N}^{\beta}$. The data preparation process for a plug-in match is, therefore, Extending the matrix $\hat{\pi}.\mathcal{W}^1$ is extended from $S \times \mathcal{N}^{\pi}$ to $S \times \mathcal{N}^{\beta}$, denoted by $\hat{\pi}.\mathcal{W}^1$, which can be expressed as

$$\hat{\pi}.\mathcal{W}^1(j,k) = \begin{cases} \hat{\pi}.\mathcal{W}^1(j,k) & \text{if } j \leq \mathcal{N}^{\pi} \text{ and } k \leq \mathcal{N}^{\pi} \\ 0 & \text{if } j \succ \mathcal{N}^{\pi} \text{ or } k \succ \mathcal{N}^{\pi} \end{cases}$$

Thus, the result matrix $\hat{\pi}.\mathcal{W}^1$ becomes

$$\hat{\pi}.\mathcal{W}^1 = \begin{bmatrix} \hat{\pi}.\mathcal{W}^1(1,1) & \hat{\pi}.\mathcal{W}^1(1,2) & \dots & \hat{\pi}.\mathcal{W}^1(1,\mathcal{N}^{\pi}) & \vdots & 0 & 0 & \dots & 0 \\ \hat{\pi}.\mathcal{W}^1(2,1) & \hat{\pi}.\mathcal{W}^1(2,2) & \dots & \hat{\pi}.\mathcal{W}^1(2,\mathcal{N}^{\pi}) & \vdots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \hat{\pi}.\mathcal{W}^1(\mathcal{N}^{\pi},1) & \hat{\pi}.\mathcal{W}^1(\mathcal{N}^{\pi},2) & \dots & \hat{\pi}.\mathcal{W}^1(\mathcal{N}^{\pi},\mathcal{N}^{\pi}) & \vdots & 0 & 0 & \dots & 0 \end{bmatrix}$$

3. For **Subsume match**, where $\mathcal{N}^{\beta} \prec \mathcal{N}^{\pi}$, some additional processes are called for in order to extend the size of $\hat{\beta}$ from $\mathcal{N}^{\beta}$ to $\mathcal{N} = \mathcal{N}^{\pi}$. Data preparation process for a subsume match is as follows: The matrix $\hat{\beta}.\mathcal{W}^1$ is extended from $S \times \mathcal{N}^{\beta}$ to

$S \times \mathcal{N}^{\pi}$, denoted by $\tilde{\beta}.\mathcal{W}^1$, which can be expressed as

$$\hat{\beta}.\mathcal{W}^1(j,k) = \begin{cases} \hat{\beta}.\mathcal{W}^1(j,k) & \text{if } j \leq \mathcal{N}^{\beta} \text{ and } k \leq \mathcal{N}^{\beta} \\ 0 & \text{if } j \succ \mathcal{N}^{\beta} \text{ or } k \succ \mathcal{N}^{\beta} \end{cases}$$

Thus, the matrix $\hat{\beta}.\mathcal{W}^1$ becomes

$$\hat{\beta}.\mathcal{W}^1 = \begin{bmatrix} \hat{\beta}.\mathcal{W}^1(1,1) & \hat{\beta}.\mathcal{W}^1(1,2) & \ldots & \hat{\beta}.\mathcal{W}^1(1,\mathcal{N}^{\beta}) & \vdots & 0 & 0 & \ldots & 0 \\ \hat{\beta}.\mathcal{W}^1(2,1) & \hat{\beta}.\mathcal{W}^1(2,2) & \ldots & \hat{\beta}.\mathcal{W}^1(2,\mathcal{N}^{\beta}) & \vdots & 0 & 0 & \ldots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hat{\beta}.\mathcal{W}^1(\mathcal{N}^{\beta},1) & \hat{\beta}.\mathcal{W}^1(\mathcal{N}^{\beta},2) & \ldots & \hat{\beta}.\mathcal{W}^1(\mathcal{N}^{\beta},\mathcal{N}^{\beta}) & \vdots & 0 & 0 & \ldots & 0 \end{bmatrix}$$

3.5.7 A Synopsis of Deterministic Intention Specification

In this work, a synopsis of deterministic intention specification is nominated as a level of adaptation. This addresses the aforementioned significant issue as the desired behaviors of an MA are collected in a compact form. Instead of using narrative description, the synopsis is formulated and described by graphical representation to enable direct visualization.

A **synopsis of deterministic intention specification** represents the characteristics of $\mathcal{I}$ ascribing the following definitions:

Definition 12. ***A synopsis of deterministic intention specification***
Let $\mathcal{I}^{SY}$ be a synopsis of deterministic intention specification of either β or π. This specification can be modeled as a 2-tuple $(\mathcal{E}, \mathcal{P})$.

1. $\mathcal{E}$ is $1 \times \mathcal{N}$ matrix representing an entropy

2. $\mathcal{P}$ is $1 \times \mathcal{N}$ matrix representing a probability

Given this definition, the ordinary intention specifications are arranged in standard form with equal size to overcome the limitations of *lacking of scalability over incomplete declared intention*, whereby assessment of compatibility degree over the inequivalent match can be determined. An algorithm for elicitation this proposed model is as follows:

A synopsis of deterministic behavior elicitation algorithm

1. Given $\mathcal{E}(j)$ be an entropy of an intention specification $\mathcal{I}$, the j^{th} concerning attribute value, $1 \leq j \leq \mathcal{N}$, The value of $\mathcal{E}(j)$ can be calculated as Equation 3.6.

2. Given $\mathcal{P}(j)$ be a probability of an intention specification $\mathcal{I}$ over the FV class toward the j^{th} concerning attribute value, $1 \leq j \leq \mathcal{N}$. The value of $\mathcal{P}(j)$ takes a form as Equation 3.7

The output $\mathcal{I}^{SY}$ corresponding to the sample data in Table 3.5.1, takes a form

$$\mathcal{I}.\mathcal{E} = \begin{pmatrix} 0.97 & 0.00 & 0.97 & 1.00 & 0.92 & 0.81 & 0.99 & 0.59 & 0.81 & 1.00 \end{pmatrix}$$

$$\mathcal{I}.\mathcal{P} = \begin{pmatrix} 0.40 & 1.00 & 0.60 & 0.50 & 0.67 & 0.75 & 0.43 & 0.86 & 0.75 & 0.50 \end{pmatrix}$$

3.5.8 A Rough Validation Assessment

The rough validation of the *compatibility degree of reusing a service* π *in place of* β is proposed based on an assumption defined as

> *If an agent executing its tasks by means of reusing service* π *can preserve its principal characteristics declared in* β*,* π *will be revealed as a compatible service of* β*.*

In order to enable direct visualization, a visualized model is presented. Instead of using narrative description, the synopsis is formulated and described by graphical representation to enable direct visualization. A visualized model of the specification synopsis is drawn in a 2-D shape, where the x-axis accounts for the degree of entropy and the y-axis represents the probability of intention degree.

Definition 13. *A visualized model of a synopsis deterministic intention specification*

Let $\mathcal{V}^{SY}$ be a visualized model of a synopsis deterministic intention specification of either β or π. This specification is modeled with vertices ($\mathcal{V}$) and edges ($\mathcal{G}$).

1. $\mathcal{V}(j)$ denotes an x-y coordinate, $(\mathcal{E}(i), \mathcal{P}(i))$, of a concerning attribute value j^{th}.
2. $\mathcal{G}(j)$ denotes an edge from a vertex $\mathcal{V}(j)$ to a vertex $\mathcal{V}(j+1)$. The final $\mathcal{V}(n)$ wraps around to $\mathcal{V}(1)$, completing the representation cycle.

Figure 3.5 demonstrates three graphic examples of β and π on the left and right, respectively. As can be seen, the graphs depicting $\pi(1)$ in comparison with $\beta(1)$ are virtually no resemblance. The second one exhibits minor difference between $\beta(2)$ and $\pi(2)$. In contrast, the third graphs are strikingly similar.

The resulting compatible degrees of πs and βs are given as 31.71%, 70.60%, and 89.81%, respectively. The lowest compatibility degree is reported in the first instance as it is noticeable with the highest shape difference. On the other hand, the third example has the highest compatibility degree, where the graphs are very much alike in many ways. This can be inferred that the compatible degree of π comparing with β is related to the similarity in shape of π liken to β. Thus, the compatibility degree can somehow be calculated from the degree of characteristics preservation.

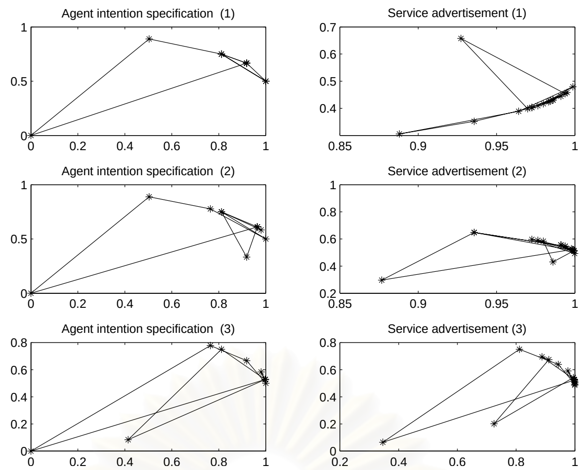

Figure 3.5: Examples of visualized model

In order to confirm the correctness of the proposed concepts, statistical methods are chosen to conduct validation experiments. The variable $\Theta^{\beta,\pi}$ is mapped as *dependent variable*, whereas $\nabla_{\mathcal{E}}^{\beta,\pi}$ is reduced to an *independent variable*. Regression and correlation analyses, namely, linear, quadratic, and cubic regression are employed. Mathematically, the linear, quadratic, and cubic equations can be expressed as (3.8), (3.9), and (3.10), respectively.

$$\Theta^{\beta,\pi} = \rho_0 + \rho_1 \nabla_{\mathcal{E}}^{\beta,\pi} + \epsilon \tag{3.8}$$

$$\Theta^{\beta,\pi} = \rho_0 + \rho_1 \nabla_{\mathcal{E}}^{\beta,\pi} + \rho_2 [\nabla_{\mathcal{E}}^{\beta,\pi}]^2 + \epsilon \tag{3.9}$$

$$\Theta^{\beta,\pi} = \rho_0 + \rho_1 \nabla_{\mathcal{E}}^{\beta,\pi} + \rho_2 [\nabla_{\mathcal{E}}^{\beta,\pi}]^2 + \rho_3 [\nabla_{\mathcal{E}}^{\beta,\pi}]^3 + \epsilon \tag{3.10}$$

where ρ_0, ρ_1, ρ_2 and ρ_3 are regression coefficients.

There are two aspects that must be addressed in these analyses. The first aspect concerns with the *nature of the relationship*, where mathematical relation is used to predict the value of $\Theta^{\beta,\pi}$ from $\nabla_{\mathcal{E}}^{\beta,\pi}$. The second aspect concerns with the *degree of the relationship*, which refers to the strength of the relationship between $\Theta^{\beta,\pi}$ and $\nabla_{\mathcal{E}}^{\beta,\pi}$.

Additionally, hypothesis testing is applied to determine whether there exists an actual relationship between $\Theta^{\beta,\pi}$ and $\nabla_{\mathcal{E}}^{\beta,\pi}$. The basic hypotheses are defined as follows:

$$H_0 : \rho_k = 0$$

$$H_1 : \rho_k \neq 0$$

where ρ_k is the regression coefficient of order k^{th}. For the linear, quadratic, and cubic relation testings, k is equal to 1, 2 and 3, respectively. If the null hypothesis, H_0, holds true, it can be inferred that there exists a relationship between $\Theta^{\beta,\pi}$ and $\nabla_{\mathcal{E}}^{\beta,\pi}$. The contribution precipitating from these hypothesis testings are mathematical models that can be used for rough estimation of the compatibility degree. Furthermore, the synopsis model is also verified whether it is an alternative way for specifying the agent's behavior.

3.5.9 An Extended-$\mathcal{I}^{SY}$ Elicitation Algorithm

In order to enable scalability over inequivalent decision criteria to an $\mathcal{I}^{SY}$, some additional steps are called for, namely, an extended-$\mathcal{I}^{SY}$ elicitation algorithm. The essential steps of this algorithm are presented below.

An extended-$\mathcal{I}^{SY}$ elicitation algorithm

1. In case of **Exact match**, where $\mathcal{N}^{\beta} = \mathcal{N}^{\pi}$, there is no further process in this phase.

2. For **Plug-in match**, where $\mathcal{N}^{\beta} \succ \mathcal{N}^{\pi}$, some additional processes are called for in order to extend the size of π from $\mathcal{N}^{\pi}$ to $\mathcal{N}^{\beta}$; and $\mathcal{N}$ is set to $\mathcal{N}^{\beta}$. The data preparation process for a plug-in match are as follows:

(a) The matrix $\pi.\mathcal{E}$ is extended as

$$\pi.\mathcal{E}(j) = \begin{cases} \pi.\mathcal{E}(j) & \text{if } j \leq \mathcal{N}^{\pi} \text{ and } k \leq \mathcal{N}^{\pi} \\ 1 & \text{if } j \succ \mathcal{N}^{\pi} \text{ or } k \succ \mathcal{N}^{\pi} \end{cases}$$

Thus, the matrix $\pi.\mathcal{E}$ is given by

$$\pi.\mathcal{E} = \begin{bmatrix} \pi.\mathcal{E}(1,1) & \pi.\mathcal{E}(1,2) & \ldots & \pi.\mathcal{E}(1,\mathcal{N}^{\pi}) & \vdots & 1 & 1 & \ldots & 1 \end{bmatrix}$$

(b) The matrix $\pi.\mathcal{P}$ is extended as

$$\pi.\mathcal{P}(j) = \begin{cases} \pi.\mathcal{P}(j) & \text{if } j \leq \mathcal{N}^{\pi} \text{ and } k \leq \mathcal{N}^{\pi} \\ 1 & \text{if } j \succ \mathcal{N}^{\pi} \text{ or } k \succ \mathcal{N}^{\pi} \end{cases}$$

Thus, the matrix $\pi.\mathcal{P}$ becomes

$$\pi.\mathcal{P} = \begin{bmatrix} \pi.\mathcal{P}(1,1) & \pi.\mathcal{E}(1,2) & \ldots & \pi.\mathcal{E}(1,\mathcal{N}^{\pi}) & \vdots & 0.5 & 0.5 & \ldots & 0.5 \end{bmatrix}$$

3. For **Subsume match**, where $\mathcal{N}^{\beta} \prec \mathcal{N}^{\pi}$, some additional processes are called for in order to extend the size of β from $\mathcal{N}^{\beta}$ to $\mathcal{N}^{\pi}$; and $\mathcal{N}$ is set to $\mathcal{N}^{\pi}$. The data preparation process for a subsume match are as follows:

(a) The matrix $\beta.\mathcal{E}$ takes the form

$$\beta.\mathcal{E}(j) = \begin{cases} \beta.\mathcal{E}(j) & \text{if } j \leq \mathcal{N}^{\beta} \text{ and } k \leq \mathcal{N}^{\beta} \\ 1 & \text{if } j \succ \mathcal{N}^{\beta} \text{ or } k \succ \mathcal{N}^{\beta} \end{cases}$$

Thus, the matrix $\beta.\mathcal{E}$ is

$$\beta.\mathcal{E} = \begin{bmatrix} \beta.\mathcal{E}(1,1) & \beta.\mathcal{E}(1,2) & \ldots & \beta.\mathcal{E}(1,\mathcal{N}^{\beta}) & \vdots & 1 & 1 & \ldots & 1 \end{bmatrix}$$

(b) The matrix $\beta.\mathcal{P}$ is extended as

$$\beta.\mathcal{P}(j) = \begin{cases} \beta.\mathcal{P}(j) & \text{if } j \leq \mathcal{N}^{\beta} \text{ and } k \leq \mathcal{N}^{\beta} \\ 1 & \text{if } j \succ \mathcal{N}^{\beta} \text{ or } k \succ \mathcal{N}^{\beta} \end{cases}$$

Thus, the matrix $\beta.\mathcal{P}$ becomes

$$\beta.\mathcal{P} = \left[\beta.\mathcal{P}(1,1) \quad \beta.\mathcal{E}(1,2) \quad \ldots \quad \beta.\mathcal{E}(1,\mathcal{N}^{\beta}) \;\vdots\; 0.5 \quad 0.5 \quad \ldots \quad 0.5\right]$$

3.6 Matchmaking Process

In this work, the MA contacts a middle agent, called **service broker**, to determine the most suitable service according to the MA is **intention specification**. Upon receiving the intention specification, the service broker performs a sequence of service discovery using **matchmaking** procedures. This process provides recommendations for suitable services based upon **service advertisement**.

The matchmaking process is the process of finding the most relevant service comparing with β. To equip the matchmaking process which accommodate the user's purpose, some related terms are defined as

1. σ is a user satisfaction level over compatible degree of reusing a service π in place of β;
2. $\tilde{\tau}$ is a **set of candidate services**;
3. τ is a **set of the most proper service**; and
4. ζ is a expectation percentage over compatible degree.

The process is described in this following algorithm.

A matchmaking algorithm

A matchmaking algorithm

1. In case of both β and π are normalized to the synopsis model, $\Theta^{\beta^{SY},\pi^{SY}}$ is assessed as follows:

$$\Theta^{\beta^{SY},\pi^{SY}} = 91.764 - 1.985\nabla_{\mathcal{E}}^{\beta^{SY},\pi^{SY}} + 0.033[\nabla_{\mathcal{E}}^{\beta^{SY},\pi^{SY}}]^2 - 0.00029[\nabla_{\mathcal{E}}^{\beta^{SY},\pi^{SY}}]^3 \quad (3.11)$$

2. In case of both β and π are normalized to the deterministic gray box intention specification model, the following steps must be performed.

   (a) Create a *complete decision making situation* matrix

   Let δ^c be an $\mathcal{M} \times \mathcal{N}$ matrix representing the complete decision making situation. The matrix δ^c takes the form

$$\delta^c(i,j) = \begin{cases} 1 & \text{if } \mathcal{A}'(j) \in \mathcal{C}^c(i) \\ 0 & \text{if } \mathcal{A}'(j) \notin \mathcal{C}^c(i) \end{cases}$$

   (b) **Assess** the decision making of $\hat{\beta}$ for all possible situations collecting from δ^c. $\hat{\beta}.\Omega$ can be calculated by

$$\mathcal{Y} = \frac{1}{1 + e^{-\hat{\beta}.\mathcal{W}^1\delta^c + \hat{\beta}.b^1}}$$

$$\hat{\beta}.\Omega = \frac{1}{1 + e^{-\hat{\beta}.\mathcal{W}^2\mathcal{Y} + \hat{\beta}.b^2}}$$

   (c) **Assess** the decision making of $\hat{\pi}$ for all possible situations collecting from δ^c. $\hat{\pi}.\Omega$ can be calculated by

$$\mathcal{Y} = \frac{1}{1 + e^{-\hat{\pi}.\mathcal{W}^1\delta^c + \hat{\pi}.b^1}}$$

$$\hat{\pi}.\Omega = \frac{1}{1 + e^{-\hat{\pi}.\mathcal{W}^2\mathcal{Y} + \hat{\pi}.b^2}}$$

(d) Calculate the *compatible degree* of $\hat{\pi}$ with $\hat{\beta}$, where $\Theta^{\hat{\beta},\hat{\pi}}$ is assessed by a **detailed validation assessment**.

3. Find a set of candidate services ($\tilde{\tau}$) having the compatible degree with in the value defined in σ, where σ is an acceptable compatible degree of reusing π in place of β.

$$\tilde{\tau} : \tau \subseteq \Pi, \tilde{\tau} = \{\tilde{\tau}|\ \forall(\tilde{\tau}),\text{where } \Theta^{\beta,\tilde{\tau}} \geq \sigma\}$$

and Π is a set of all available πs.

4. Find the most proper service that has the highest compatible degree. The result of this step is the most proper service(τ). This service is the service with the maximum value of $\Theta^{\hat{\beta},\hat{\pi}}$

$$\tau : \tau \subseteq \tilde{\tau}, \tau = \{\tau|\ \forall(\tau), \text{ where } \Gamma(\beta,\tau) = \max\left(\Gamma^{\beta,\tilde{\tau}}\right)\}$$

Furthermore, the expectation percentage over compatible degree can be assessed as

$$\zeta = \max\left(\Gamma^{\beta,\tilde{\tau}}\right)$$

CHAPTER IV

EXPERIMENTAL RESULTS

Some preliminary investigations of the experiment are

1. Defining standard terms to carry out the experiment.
2. Assigning parameters to the proposed model.
3. Conducting fundamental experiment for equivalent matchmaking.

Furthermore, from the vital issues stated in Chapter 3, there are two main tasks that have been addressed:

1. Scalability over an incomplete declared intention
2. Scalability over an inequivalent decision criterion

These issues are assessed by two phases of experiment described below.

4.1 Experiment Approaches and Data

There were four fundamental approaches conducted in the experiment.

1. **Conventional nondeterministic approach:** $\mathcal{B}roker^{ND}$ performed service validation algorithm based a nondeterministic behavioral specifications approach.
2. **Conventional deterministic approaches:** In this work, two brokers were conducted as the instances of this approach.

(a) $\mathcal{B}roker^{WB}$ performed matchmaking algorithm based on $\mathcal{I}^{WB}$ which was a deterministic white box behavioral specification approach.

(b) $\mathcal{B}roker^{BL}$ also performed matchmaking algorithm based on $\mathcal{I}^{WB}$ which was a deterministic white box behavioral specification approach. This approach was used as the controlled environment. Thus, all situations and their corresponding actions were all known.

3. **Applied conventional inductive learning to model intention specification:** $\mathcal{B}roker^{BP}$ performed service validation algorithm based on $\mathcal{I}^{GB}$ which was a deterministic gray box behavioral specification approach. The behavioral elicitation algorithm considered just only the 1^{st} *constraint* and did not performed an extended-$\mathcal{I}^{GB}$ elicitation algorithm.

4. **Proposed approaches:** Two deterministic intention specification model were proposed which were conducted over the following brokers.

(a) $\mathcal{B}roker^{SY}$ performed service validation algorithm based on $\mathcal{I}^{SY}$ which was a deterministic behavioral specification approach.

(b) $\mathcal{B}roker^{GB}$ performed service validation algorithm based on $\mathcal{I}^{GB}$ which was a deterministic gray box behavioral specification approach. The behavioral elicitation algorithm concerned both the 1^{st} and the 2^{nd} *constraints*. Furthermore, an extended-$\mathcal{I}^{GB}$ elicitation algorithm was taken into account in case of an inequivalent-λ.

In this research, we states our focus on the problem of identifying the execution results of an agent, by means of reusing external service, under uncertainty environments. The emphasized of this research relies on the attempt to extending the concept of a deterministic behavioral specification, which potent enough to work on the uncertainty environment and this novel specification must practical for using as an added part to the conventional table-based specification approach.

Thus, the concept of reducible decision problem in theory of computation [45] is defined as follows:

> *If we can establish that one decision problem, $\mathcal{P}1$, can be reduced to another, $\mathcal{P}2$, or that having a general solution to $\mathcal{P}2$ would guarantee a general solution to $\mathcal{P}1$, then it is reasonable to say informally that $\mathcal{P}1$ is no harder than $\mathcal{P}2$. It should follow that if $\mathcal{P}2$ is solvable, $\mathcal{P}1$ is solvable.*

From the above problem reducible concept, we can issue our work as:

> *If we can reduce this particular stated problem to another well-known problem, which is intensively studied in other domains, it should follow that if there are some successful algorithms solve that well-known problem, our particular stated problem is solvable.*

In general, there are many problems in science, statistics, and technology, which can be effectively modeled or reduced as learning of input-output mapping given, based upon some data set. In order to solve the proposed issued in this dissertation, the problem of an agent's decision makings are reduced to a *pattern classification* problem. Some reduction descriptions are declared as follows:

1. the agent's decision attributes can be reduced to *features*,
2. the agent's decision conditions are represented as *pattern*; and
3. the agent's actions are reduced to *pattern category*.

The experiment was simulated based on MATLAB 7.0 environment. Furthermore, five groups of the intention blueprints were created as test cases. The methods for generating all test cases were relied on discrete uniform random numbers. Descriptions of these intention blueprints are defined below.

1. The description of a group of *basis intention blueprints* is defined as follows:

   Let $\mathcal{G}^0$ be a group of *complete basis intention blueprints*. For any sample t^{th} of a group of *basis intention blueprints*, its rule-based conditional statements over the class of favorable or unfavorable were *generated randomly*.

2. The description of a group of *synthetic intention blueprints* is defined as follows:

   Let $\mathcal{G}^1$ be a group of *synthetic intention blueprints*. For any sample t^{th} of a group of *synthetic intention blueprints*, its rule-based conditional statements over the class of favorable or unfavorable were *generated with some fixed conditions*. The percent of the similarity between the intention blueprints in group $\mathcal{G}^0$ and $\mathcal{G}^1$ varied from 30-90%.

3. The description of a group of *incomplete basis intention blueprints* is defined as follows:

   Let $\mathcal{G}^2$ be a group of *incomplete basis intention blueprints*. For any sample t^{th} of a group of *incomplete basis intention blueprints*, its explicit rule-based conditional statements over the class of favorable or unfavorable were *subsets* of the rule-based conditional statements specified in a sample t^{th} of a group of *complete basis intention blueprints*.

4. The descriptions for two groups of *synthetic intention blueprints with new added up attributes* are defined as follows:

Let $\mathcal{G}^i$ be a group of *synthetic intention blueprints with new added up attributes*, where $i = 3, 4$. For any sample t^{th} of a group of *synthetic intention blueprints with new added up attribute*, its explicit rule-based conditional statements over the class of favorable or unfavorable were *supersets* of the rule-based conditional statements specified in the t^{th} sample of a group of *complete basis intention blueprints*. The new concerning attribute was added to intention blueprints in group *complete basis intention blueprints*. This new concerning attribute provided two attribute values. The average percentage of similarity between $\mathcal{G}^0(t)$ and $\mathcal{G}^i(t)$ are 93.41% and 50.04% with respect to the order of i.

4.2 Parameters Tuning

4.2.1 Evaluation for Parameters of Synopsis Specifications Model

The equivalent-α matchmaking of complete agent intention specification and complete service advertisement was experimented under a controlled environment. An assumption described the compatibility degree ($\Theta^{\beta,\pi}$) of β and π in terms of the discrepancy degree $\nabla_{\mathcal{E}}^{\beta,\pi}$. The crucial issue was whether there was any relationship between $\Theta^{\beta,\pi}$ and $\nabla_{\mathcal{E}}^{\beta,\pi}$.

The results obtained from $\mathcal{B}roker^{WB}$ were noted as **controlled results** of this equivalent-α matchmaking. These results were subsequently employed in hypothesis testing. Mathematical evaluation formulated from the hypothesis testing step was carried out as the validation assessment procedure for $\mathcal{B}roker^{SY}$.

4.2.1.1 Hypothesis Testing

The experiment performed over 1500 training cases where every training case was formulated as a **complete intention specification**. The f-statistic was applied on three regression analysts by $\mathcal{B}roker^{SY}$ having significance rejecting level of 0.05%. Figure 4.1 shows a comparison between the degree of compatibility and the degree of discrepancy. The x-axis represents the number of test cases. In case of the solid-line, the y-axis is the percentage of compatibility degree. For the dash line, the y-axis presents discrepancy degree. These results suggest that the discrepancy degree tends to be reversely related to the compatibility degree. The compatibility degree of the results increases as the discrepancy degree decreases. Consequently, the curve estimation procedure was carried out to determine the relationships between the degree of compatibility and the degree of discrepancy. The data were plotted as scatter plots presented in Figure 4.2. From these scatter plots, linear, quadratic, and cubic curve estimations were employed in order to determine which relationship ough to be used.

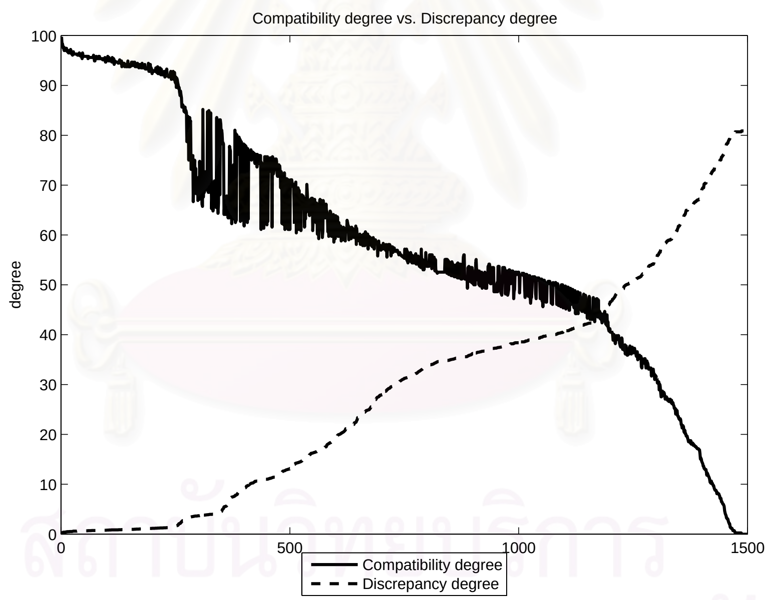

Figure 4.1: Comparison between compatibility degrees and discrepancy degrees

Table 4.1 summarizes significant values of 0.00 on all analyses. This implies that the null hypothesis is obviously rejected as the significance values are less than 0.05. This can be inferred that there exists a relationship between $\Theta^{\beta,\pi}$ and $\nabla_{\mathcal{E}}^{\beta,\pi}$. The R-

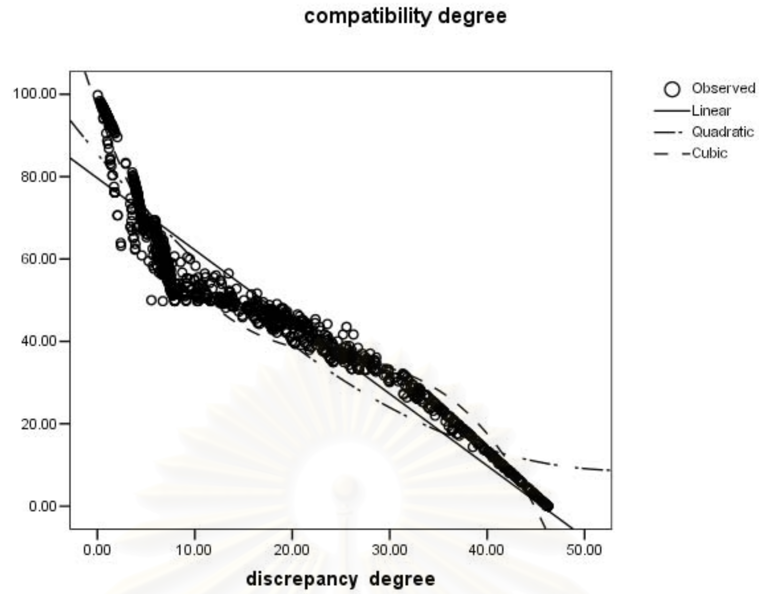

Figure 4.2: Results from curve fittings

square values are proportionated to the variations of $\Theta^{\beta,\pi}$ in the range of 94.6%, 94.6%, and 96.1% for linear, quadratic, and cubic regression, respectively. The corresponding standard of errors are 5.681%, 5.682%, and 4.875%, respectively. This means that $\nabla_{\mathcal{E}}^{\beta,\pi}$ is a decisive factor influencing on the compatibility degree. Equations (3.8), (3.9), and (3.10) thus become

$$\Theta^{\beta,\pi} = 88.264 - 1.055\nabla_{\mathcal{E}}^{\beta,\pi} \tag{4.1}$$

$$\Theta^{\beta,\pi} = 88.403 - 1.071\nabla_{\mathcal{E}}^{\beta,\pi} + 0.00022[\nabla_{\mathcal{E}}^{\beta,\pi}]^2 \tag{4.2}$$

$$\Theta^{\beta,\pi} = 91.764 - 1.985\nabla_{\mathcal{E}}^{\beta,\pi} + 0.033[\nabla_{\mathcal{E}}^{\beta,\pi}]^2 - 0.00029[\nabla_{\mathcal{E}}^{\beta,\pi}]^3 \tag{4.3}$$

4.2.1.2 Validation Assessment of an Accuracy Regression Models

The above three equations were employed by $\mathcal{B}roker^{SY}$. The results of the estimation compatibility degree obtained from $\mathcal{B}roker^{SY}$ over those equations were drawn analogies against results obtained from both $\mathcal{B}roker^{WB}$. As shown in Figure 4.3, the compatibility

Table 4.1: Model Summary

Equations	R-Square	standard error	F	df1	df2	Sig.
Linear	0.946	5.681	26438.095	1	1498	0.000
Quadratic	0.946	5.682	13216.727	2	1497	0.000
Cubic	0.961	4.875	12147.971	3	1496	0.000

degrees acquired from $\mathcal{B}roker^{SY}$ were almost the same as the compatibility degree of the results achieved from $\mathcal{B}roker^{WB}$ on all analyses. Of the three analyses, the controlled results from cubic fitting obtained from $\mathcal{B}roker^{WB}$ shown in Figure 4.3(c), were most similar to the compatibility degree.

Table 4.2 sums up the information about percentage of training accuracy from applying those equations. The corresponding box-plots are shown in Figure 4.4. These validation results indicate that cubic fitting is the most suitable regression model for representing the relationship between the compatibility degree and the discrepancy degree. The average accuracy of $\mathcal{B}roker^{SY}$ with cubic fitting is 96.51%, having the standard deviation of 3.43%.

Table 4.2: Summary of training accuracy from synopsis model

Equations	mean	median	std	min	max
Linear	95.97	97.09	4.00	80.49	99.99
Quadratic	95.87	97.00	3.93	80.41	100.00
Cubic	96.51	97.29	3.43	80.31	100.00

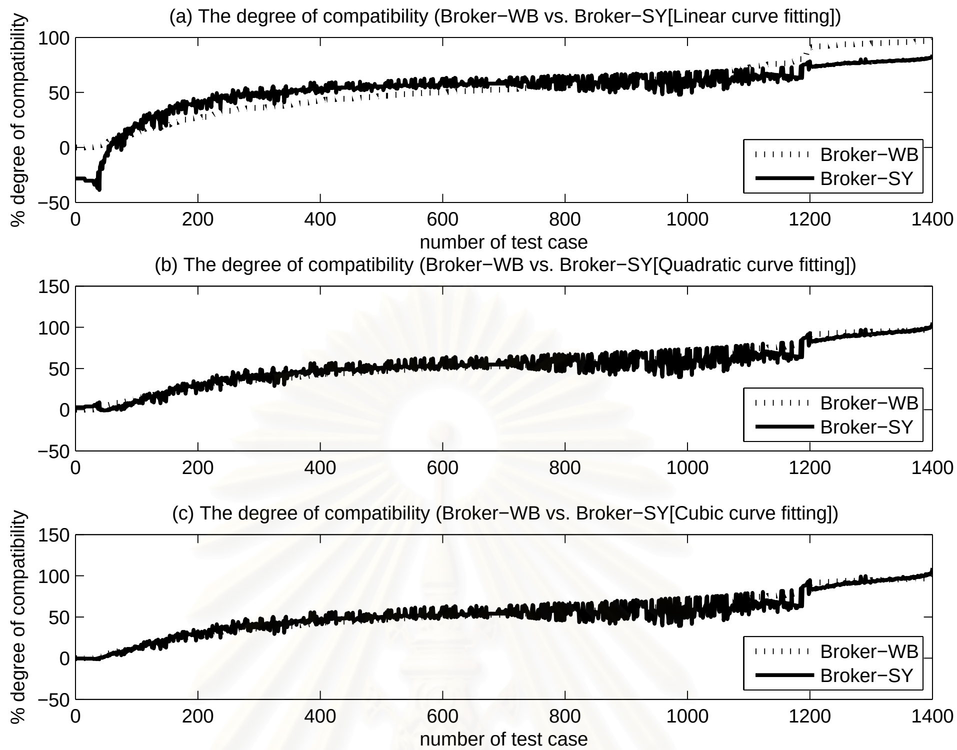

Figure 4.3: Comparisons of estimated results from three fitting model over training data

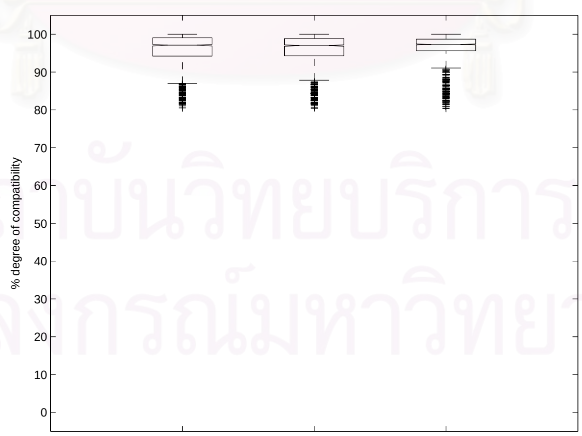

Figure 4.4: Box plots of training accuracy from synopsis model

4.2.2 Evaluation for Parameters of Deterministic Gray Box Specification Model

For deterministic gray box specification, the size depended on that of parameters $\mathcal{W}^1$, which previously was specified as $S \times \mathcal{N}$. As such, S neurons were employed in the backpropagation learning algorithm.

Table 4.3 presents the percentage of accuracy based on some influential factors, namely,

- the number of learning neurons(S),
- the number of overall attribute values($\mathcal{N}$), and
- the number possible condition statements($\mathcal{M}$).

The visualized model of these comparison are shown in Figure 4.5.

Table 4.3: Influence factors over percentage of accuracy

case no.	$\mathcal{N}$	S	S/$\mathcal{N}$ in %	$\mathcal{M}$	% of accuracy results
1	13	13	100.00	108	100.00
2	13	9	69.23	108	99.07
3	13	7	53.85	108	100.00
4	13	6	46.15	108	100.00
5	13	5	38.46	108	98.15
6	13	13	100.00	108	100.00
7	13	9	69.23	108	100.00
8	13	7	53.85	108	100.00
9	13	6	46.15	108	98.15

continue on the next page

Table 4.3: (*continued*)

case no.	$\mathcal{N}$	S	S/$\mathcal{N}$ in %	$\mathcal{M}$	% of accuracy results
10	13	5	38.46	108	91.67
11	13	13	100.00	108	100.00
12	13	9	69.23	108	99.07
13	13	7	53.85	108	99.07
14	13	6	46.15	108	99.07
15	13	5	38.46	108	95.37
16	13	13	100.00	108	100.00
17	13	9	69.23	108	99.07
18	13	7	53.85	108	99.07
19	13	6	46.15	108	98.15
20	13	5	38.46	108	98.15
21	13	13	100.00	108	100.00
22	13	9	69.23	108	99.07
23	13	7	53.85	108	98.15
24	13	6	46.15	108	98.15
25	13	5	38.46	108	98.15
26	14	14	100.00	162	100.00
27	14	10	71.43	162	99.38
28	14	7	50.00	162	95.68
29	14	6	42.86	162	96.30
30	14	5	35.71	162	93.21

continue on the next page

Table 4.3: (*continued*)

case no.	$\mathcal{N}$	S	S/$\mathcal{N}$ in %	$\mathcal{M}$	% of accuracy results
31	14	14	100.00	162	100.00
32	14	10	71.43	162	98.77
33	14	7	50.00	162	98.77
34	14	6	42.86	162	94.44
35	14	5	35.71	162	92.59
36	14	14	100.00	162	97.53
37	14	10	71.43	162	99.38
38	14	7	50.00	162	94.44
39	14	6	42.86	162	94.44
40	14	5	35.71	162	96.30
41	14	14	100.00	162	98.77
42	14	10	71.43	162	98.77
43	14	7	50.00	162	98.77
44	14	6	42.86	162	93.83
45	14	5	35.71	162	95.06
46	14	14	100.00	162	98.77
47	14	10	71.43	162	98.77
48	14	7	50.00	162	98.77
49	14	6	42.86	162	94.44
50	14	5	35.71	162	93.83
51	15	15	100.00	216	98.61

continue on the next page

Table 4.3: (*continued*)

case no.	$\mathcal{N}$	S	S/$\mathcal{N}$ in %	$\mathcal{M}$	% of accuracy results
52	15	10	66.67	216	98.15
53	15	8	53.33	216	96.76
54	15	6	40.00	216	91.67
55	15	5	33.33	216	90.74
56	15	15	100.00	216	99.54
57	15	10	66.67	216	97.69
58	15	8	53.33	216	97.22
59	15	6	40.00	216	90.74
60	15	5	33.33	216	87.04
61	15	15	100.00	216	99.07
62	15	10	66.67	216	97.22
63	15	8	53.33	216	98.15
64	15	6	40.00	216	93.06
65	15	5	33.33	216	93.52
66	15	15	100.00	216	98.61
67	15	10	66.67	216	98.15
68	15	8	53.33	216	94.91
69	15	6	40.00	216	92.13
70	15	5	33.33	216	92.13
71	15	15	100.00	216	99.54
72	15	10	66.67	216	95.37

continue on the next page

Table 4.3: (*continued*)

case no.	$\mathcal{N}$	S	S/$\mathcal{N}$ in %	$\mathcal{M}$	% of accuracy results
73	15	8	53.33	216	97.22
74	15	6	40.00	216	91.67
75	15	5	33.33	216	89.81
76	18	18	100.00	648	96.60
77	18	12	66.67	648	93.21
78	18	9	50.00	648	87.50
79	18	8	44.44	648	83.95
80	18	6	33.33	648	79.63
81	18	18	100.00	648	96.14
82	18	12	66.67	648	90.74
83	18	9	50.00	648	89.04
84	18	8	44.44	648	87.96
85	18	6	33.33	648	79.94
86	18	18	100.00	648	96.45
87	18	12	66.67	648	92.75
88	18	9	50.00	648	87.65
89	18	8	44.44	648	82.56
90	18	6	33.33	648	82.41
91	18	18	100.00	648	96.60
92	18	12	66.67	648	88.27
93	18	9	50.00	648	86.73

continue on the next page

Table 4.3: (*continued*)

case no.	$\mathcal{N}$	S	S/$\mathcal{N}$ in %	$\mathcal{M}$	% of accuracy results
94	18	8	44.44	648	87.04
95	18	6	33.33	648	80.25
96	18	18	100.00	648	96.45
97	18	12	66.67	648	95.52
98	18	9	50.00	648	87.50
99	18	8	44.44	648	87.65
100	18	6	33.33	648	81.48

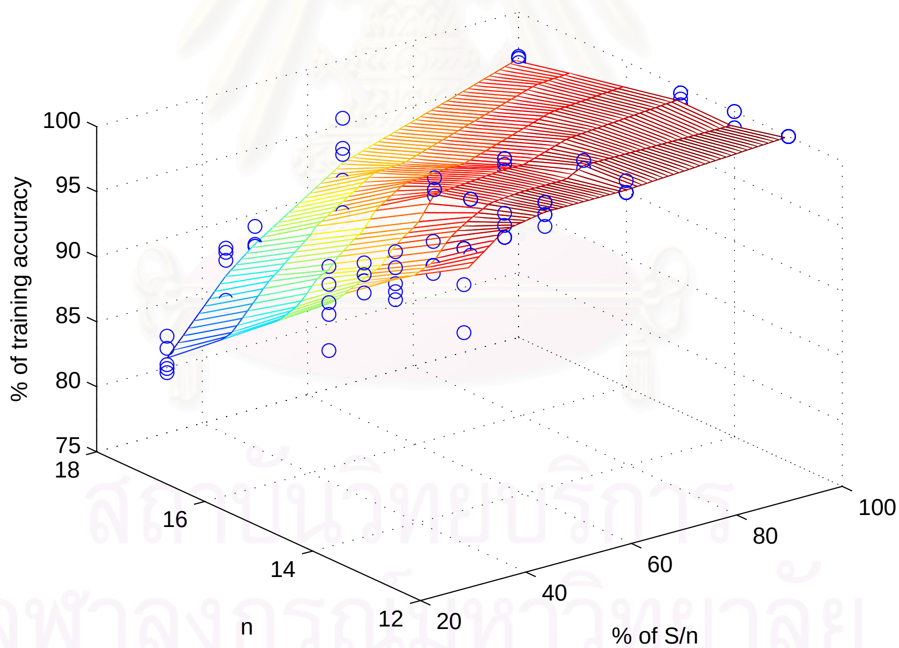

Figure 4.5: Influence factors over percentage of accuracy results

For small value of $\mathcal{N}$, say $\mathcal{N} = 12$, the percentage of accuracy indicated the impact on different values of S. As $\mathcal{N}$ increases, say $\mathcal{N} = 18$, the percentage of accuracy flattened. In general, higher percentage of accuracy is achieved when the value of S increases.

It seems that a proper value of S might be equal to $\mathcal{N}$, yielding almost 100% accuracy. However, for compact MA, this value may be less given a pre-specified percentage of accuracy by the user as ϑ.

4.3 Application of The Proposed Approach

In order to summarized the applicability of the proposed model, a *simple sealed first-price, lady products E-auction system* was demonstrated. Let us assume that there was one market place with an auction broker agent presenting the products. In this auction, all bidders had to simultaneously submit bids in such a way that no bidder knew the bid of the rest. There were five bidder agents, which were working on behalf of the same customer, participating in this auction. They were implemented with different service discovery techniques for selecting the execution service as follows:

1. *Agent 1* employed a nondeterministic behavioral specification approach by contacting with $\mathcal{B}roker^{ND}$
2. *Agent 2* employed a deterministic behavioral specification approach by contacting with $\mathcal{B}roker^{WB}$
3. *Agent 3* employed a deterministic behavioral specification approach by contacting with $\mathcal{B}roker^{GB}$
4. *Agent 4* employed a deterministic behavioral specification approach by contacting with $\mathcal{B}roker^{SY}$
5. *Agent 5* employed a deterministic behavioral specification approach by contacting with $\mathcal{B}roker^{BP}$

The following protocols were asserted to establish a viable model:

- The seller might specify a reserved price (which was hidden from the buyers) for an item posted for auction.
- A reserved price was the minimum price a seller was willing to accept for the item.
- Each bidder agent could place one request per product.
- At the end of the auction, all bidders were notified of the result including the details of all bidding transaction and the reserved price. If no bid was higher than the reserved price, the seller was not obligated to sell the item.

The system work flow for this sample scenario is described in Figure 4.3

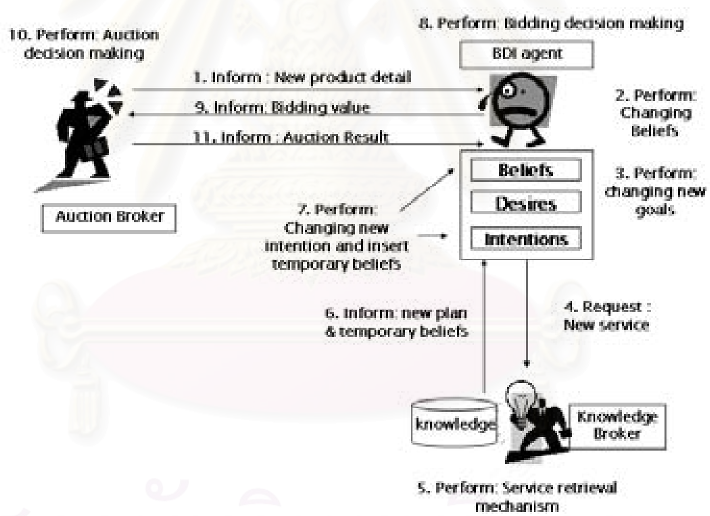

Figure 4.6: Work flow of a sample scenario

1. The auction broker agent sends the informed message to tell the bidder agent about the new auction product.
2. The beliefs of the bidder are changed according to the informed message.

3. The changed beliefs, in turns, lead to changing of the new intentions.

4. To achieve its new intentions, the bidder agent sends the message to a service broker agent requesting a new *service*.

5. The agent performs the service retrieval mechanism.

6. The agent sends the most suitable *service* encompassing some new beliefs to the bidder agent.

7. The bidder agent adapts itself according to the new intentions and temporary beliefs.

8. The bidder agent executes bidding *decision*.

9. The bidder agent sends the message to inform the auction broker agent about its bidding propose.

10. The auction broker executes all bidding proposals from the joining agents.

11. The auction broker notifies all bidder agents with the auction results.

The assumptions on the experiments presented below are based on matchmaking of *behavioral intention specifications*. Besides, the matchmaking of *functional intention specifications* assumes that all services on *functional* characteristics are fully completed. Furthermore, from the important issues presented in Chapter 3, there are two main tasks addressed:

1. Evaluation of equivalent matchmaking. The example of this evaluation is demonstrated in Section 4.3.1, and the detailed of this experiment will be described in Section 4.4.

2. Scalability over an incomplete declared intention. The instance of this issue is demonstrated in Section 4.3.2, and the detailed of this experiment will be described in Section 4.5.

3. Scalability over an inequivalent decision criterion. The example of this topic is illustrated in Section 4.3.3. For further detailed, the experiment will be presented in Section 4.6.

These issues are exemplified with the scenario described as the agent modeled for a 20-year old student and the current product for bidding was a robot.

4.3.1 Evaluation of Equivalent Matchmaking

The samples β and π of group $\mathcal{G}^0$ and $\mathcal{G}^1$ are shown in Figure 4.10 and Figure 4.12, respectively. The value of σ was 90%. The symbol Os represent the actions in set $FVClass$, #s represent the actions in set $UFVClass$, and blanks represent undeclared intention. Comparative results from $\mathcal{B}roker^{WB}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ are assessed below, where the results obtained from $\mathcal{B}roker^{WB}$ are **controlled results**.

Figure 4.9 presents the evaluation result obtained from the three concerning brokers. The results indicate that *Agent 2* was not recommend to performed bidding for this product. Because there was no proper service that could be performed on this task. For example, let β and π were intention blueprint in group $\mathcal{G}^0$ and $\mathcal{G}^1$, respectively. The matchmaking results obtained from $\mathcal{B}roker^{WB}$ was 66.20% which lower than the value of σ. Moreover, both *Agent 3* and *Agent 4* were recommended by $\mathcal{B}roker^{GB}$ and $\mathcal{B}roker^{SY}$ with the same information.

Since an MA works under dynamic and uncertain environments, the equivalent matchmaking cannot be satisfactorily determined. It can be noted that the further evaluation procedures are required to progress the matchmaking process to others group

sword								flag								balloon										
red		yellow		green		blue		red		yellow		green		blue		red		yellow		green		blue				
y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	smile	body	head
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	rou
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	squ
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	oct

Figure 4.7: A complete decision making encompassing behavioral intention of group $\mathcal{G}^0$

sword								flag								balloon										
red		yellow		green		blue		red		yellow		green		blue		red		yellow		green		blue				
y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	smile	body	head
O	O	#	O	#	O	O	O	O	#	O	#	#	O	O	O	O	#	#	O	O	O	O	O	y		
O	#	#	O	#	#	O	O	#	O	O	#	#	O	O	O	O	O	O	O	#	#	O	O	n	rou	
#	O	O	O	#	#	O	O	#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	y		
O	O	#	#	#	O	O	O	#	O	#	#	O	#	O	O	#	#	#	#	O	O	O	O	n	squ	
O	#	O	O	O	#	O	O	#	O	O	O	#	O	O	O	O	O	O	O	O	O	O	O	y		
#	#	#	O	O	#	O	O	#	#	#	O	#	#	O	O	O	O	O	#	O	O	O	O	n	oct	rou
O	O	O	#	O	O	O	O	#	O	#	O	#	O	O	O	#	O	#	#	#	#	O	O	y		
#	#	O	O	#	O	O	O	#	#	O	#	#	O	O	O	O	O	O	O	O	O	O	O	n	rou	
#	#	O	#	O	#	O	O	O	O	O	#	O	#	O	O	O	#	O	O	O	#	O	O	y		
O	#	O	O	O	#	O	O	O	#	#	#	O	#	O	O	O	O	O	#	O	O	O	O	n	squ	
#	O	O	#	O	#	O	O	#	#	#	#	#	O	O	O	#	O	#	#	#	#	O	O	y		
#	#	#	#	O	#	O	O	#	#	O	O	#	#	O	O	O	O	#	#	#	#	O	O	n	oct	squ
O	O	#	O	O	#	O	O	#	#	#	#	#	O	O	O	#	O	O	#	O	O	O	O	y		
#	#	#	O	#	#	O	O	#	O	O	#	#	O	O	O	O	O	O	O	O	#	O	O	n	rou	
#	#	O	#	#	#	O	O	#	O	O	O	O	#	O	O	O	#	#	O	#	#	O	O	y		
#	O	O	#	#	O	O	O	O	O	O	O	O	#	O	O	#	#	#	#	O	#	O	O	n	squ	
#	O	#	#	#	O	O	O	#	#	#	O	O	O	O	O	#	O	#	O	O	O	O	O	y		
#	O	O	O	#	#	O	O	#	O	O	O	#	#	O	O	#	#	O	O	O	O	O	O	n	oct	oct

Figure 4.8: A complete decision making encompassing behavioral intention of group $\mathcal{G}^1$

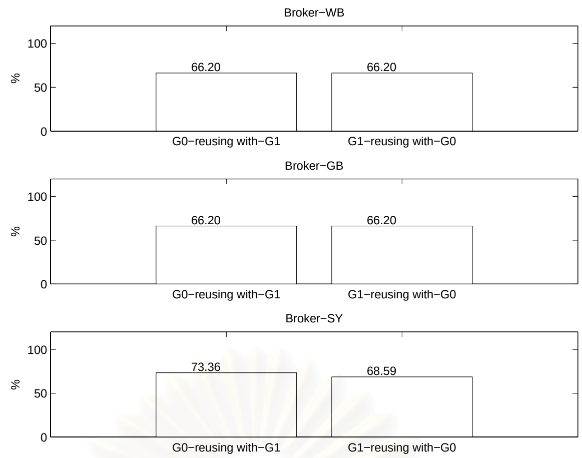

Figure 4.9: Compatibility degree from different brokers

of intention blueprint.

4.3.2 Scalability over an Incomplete Declared Intention

The samples β and π of group $\mathcal{G}^2$ and $\mathcal{G}^0$ are shown in Figure 4.10 and Figure 4.12, respectively. Comparative results from $\mathcal{B}roker^{WB}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ are assessed below.

1. According to the results of $\mathcal{B}roker^{ND}$, *Agent 1* was recommended with service π of group $\mathcal{G}^0$. However, this recommended information did not inform how well *Agent 1* reused the service π of group $\mathcal{G}^0$.

2. $\mathcal{B}roker^{WB}$ informed *Agent 2* with $\tau = \varnothing$, i.e., there was no proper service that could execute on this matter with compatibility degree greater than σ. $\mathcal{B}roker^{WB}$ conducted the case-by-case comparison which yielded the compatibility degree at 26.39%. From the above assumption, this result indicated that some relevant services were **unnoticed** by $\mathcal{B}roker^{WB}$ due to scalability limitations of $\mathcal{B}roker^{WB}$.

3. In case of $\mathcal{B}roker^{GB}$, β of group $\mathcal{G}^2$ denoting *incomplete agent intention specifi-*

sword								flag								balloon										
red		yellow		green		blue		red		yellow		green		blue		red		yellow		green		blue				
y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	smile	body	head
	O	O	O	#		#		O			O				#									y		
			O				#				O			#	#	O	O			O	O			n	rou	
				O						O	O	#				O	O	O	O		O	#		y		
				O			#	O		O	O					O		O	O					n	squ	
		#				#							#	#		#				#				y		
#	#	#					#						#		#							#		n	oct	rou
O	O					#	#	O	O										O		O	#		y		
			O											#			O							n	rou	
					O		#		O	#						O			O	#	#		#	y		
			O						O			O	O									#		n	squ	
				O				#		#	#		#							#				y		
	#		#			#						#			#	#		#					#	n	oct	squ
O		O		O											#		O						#	y		
		O											O		#	O								n	rou	
			O			#						O					O							y		
			O		O				O							O					O	#		n	squ	
					O	O									#	#		#	#			#		y		
#	#										#		#			#					#			n	oct	oct

Figure 4.10: An incomplete decision making encompassing behavioral intention β of group $\mathcal{G}^2$

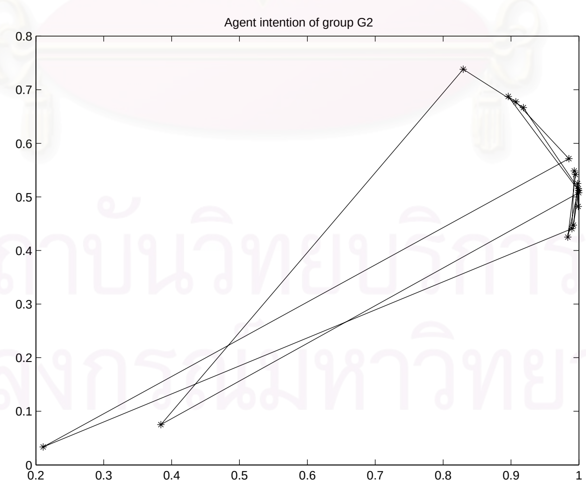

Figure 4.11: Visualized models of β from group $\mathcal{G}^2$

sword								flag								balloon										
red		yellow		green		blue		red		yellow		green		blue		red		yellow		green		blue				
y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	smile	body	head
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	rou
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	squ
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	oct

Figure 4.12: Decision making encompassing behavioral intention π^0 of group $\mathcal{G}^0$

sword								flag								balloon										
red		yellow		green		blue		red		yellow		green		blue		red		yellow		green		blue				
y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	has tie		
																								smile	body	head
O	O	O	O	X	O	#	#	O	O	O	O	X	X	#	#	O	O	O	O	X	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	X	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	X	X	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	X	X	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	rou
O	O	O	O	O	O	#	#	O	O	O	O	X	X	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	X	O	X	O	#	#	O	O	O	O	X	X	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	squ
O	O	O	O	O	O	#	#	O	O	O	O	X	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	X	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	X	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	oct

Figure 4.13: A scaled-behavioral intention blueprint of β

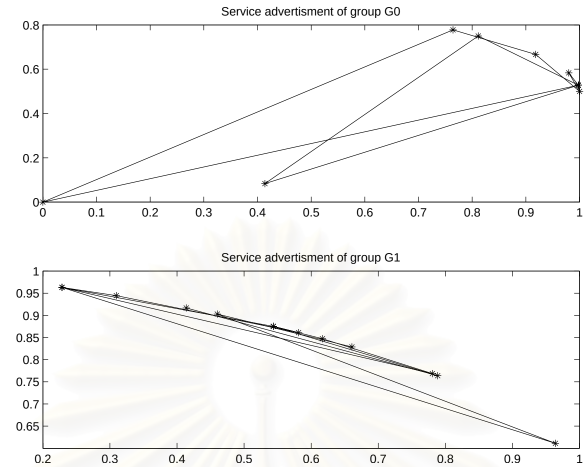

Figure 4.14: Visualized model of π^0 and π^1

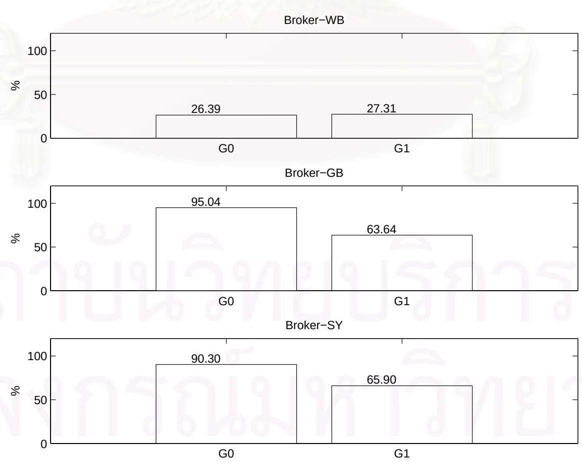

Figure 4.15: Compatibility degree from different brokers

cation, was input to the *deterministic gray box behavioral elaborating algorithm.* Consequently, the compatibility degree was assessed with the help of *matchmaking algorithm.* The *incomplete agent intention specification* would turn to be the *complete agent intention specification.* The result of this scaling is shown in Figure 4.13, where the miss declared actions are depicted. Symbol Xs represent the scaled actions which are incorrect compared with π of group $\mathcal{G}^1$. The compatibility degree of reusing service π of group $\mathcal{G}^0$ in place of agent intention in group $\mathcal{G}^2$ was 95.04%. Thus, *Agent 3* which contacted $\mathcal{B}roker^{GB}$ was recommended with service π of group $\mathcal{G}^0$.

4. *Agent 4* contacting $\mathcal{B}roker^{SY}$ was also suggested with service π of group $\mathcal{G}^0$. Figure 4.11 and Figure 4.14 demonstrate graphic examples of either β and π. As can be seen, the graphs depicting π^0 in comparison with β are virtually similar. On the other hand, the visual model of π^1 is different from the visual model of β. The compatibility degree of reusing the service π of group $\mathcal{G}^0$ was 90.30%.

4.3.3 Scalability over an inequivalent decision criterion

The sample β of group $\mathcal{G}^0$ are shown in Figure 4.16. The number of concerning attribute values of this intention is 17, i.e., $\lambda = 17$. Besides, the instance of π represented as an intention blueprint of group $\mathcal{G}^4$. The decision making of this service was performed with $\lambda = 19$ and was illustrated in Figure 4.18. Comparative results from $\mathcal{B}roker^{BL}$, $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ are assessed below.

1. The *Agent 1* was suggested with service π of group $\mathcal{G}^4$ and $\mathcal{G}^5$, but there was no information provided about the compatibility degree.

2. $\mathcal{B}roker^{BP}$ informed the *Agent 5* with $\tau = \varnothing$, i.e., there was no proper service that could execute on this matter with compatibility degree greater than σ. This

sword								flag								balloon										
red		yellow		green		blue		red		yellow		green		blue		red		yellow		green		blue				
y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	smile	body	head
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y	rou	rou
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y	squ	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y	oct	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y	rou	squ
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y	squ	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y	oct	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y	rou	oct
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y	squ	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y	oct	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n		

Figure 4.16: Decision making encompassing behavioral intention $\beta(t)$ of group $\mathcal{G}^0$

Figure 4.17: Visualized model of β from $\mathcal{G}^0$

waring hat = yes

																								smile	body	head
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	rou
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	squ
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	oct

y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	has tile
red		yellow		green		blue		red		yellow		green		blue		red		yellow		green		blue		
sword								flag								balloon								

																								smile	body	head
O	O	O	O	O	O	#	#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	O	O	O	O	O	O	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	rou
O	O	O	O	O	O	#	#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	O	O	O	O	O	O	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	squ
O	O	O	O	O	O	#	#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	O	O	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	O	O	O	O	O	O	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	oct

waring hat = no

Figure 4.18: Decision making encompassing behavioral intention π^4 of group $\mathcal{G}^4$

waring hat = yes																								smile	body	head
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	rou
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	squ
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	oct
y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	has tile		
red		yellow		green		blue		red		yellow		green		blue		red		yellow		green		blue				
sword								flag								balloon										
																								smile	body	head
O	O	O	O	O	O	#	#	O	O	O	O	O	O	X	X	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	X	X	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	X	X	X	X	X	X	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	rou
O	O	O	O	O	O	#	#	O	O	O	O	O	O	X	X	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	X	X	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	X	X	X	X	X	X	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	squ
O	O	O	O	O	O	#	#	O	O	O	O	O	O	X	X	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	rou	
O	O	O	O	O	O	#	#	O	O	O	O	O	O	X	X	O	O	O	O	O	O	#	#	y		
O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	O	O	O	O	O	O	#	#	n	squ	
#	#	#	#	O	O	#	#	X	X	X	X	X	X	#	#	#	#	#	#	#	#	#	#	y		
#	#	#	#	O	O	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	n	oct	oct
waring hat = no																										

Figure 4.19: A scaled up decision making over an inequivalent decision criterion $\beta(t)$ of group $\mathcal{G}^0$

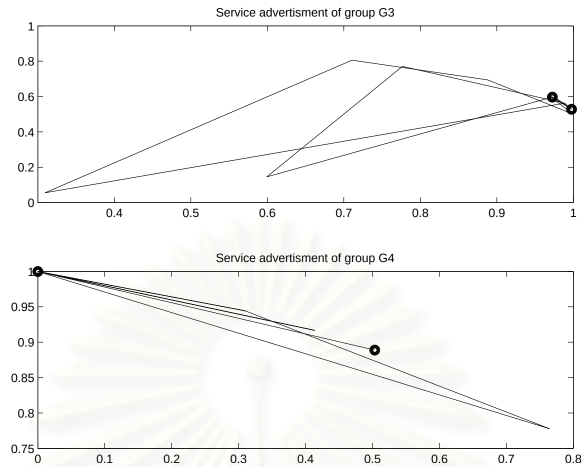

Figure 4.20: Visualized model of π^3 and π^4

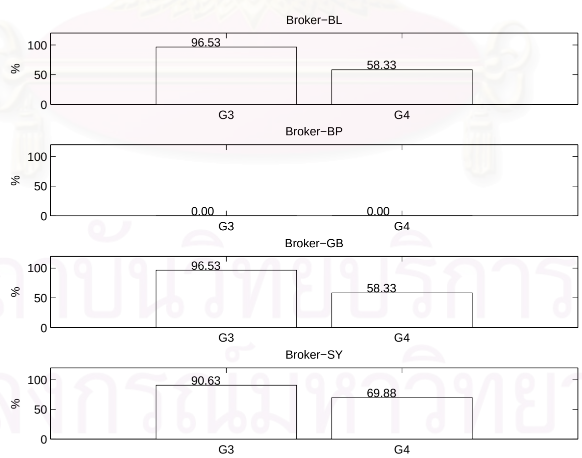

Figure 4.21: Compatibility degree from different brokers

was because $\mathcal{B}roker^{BP}$ had not further its capability to scale over an inequivalent decision criterion. The comparison between β and π could not be conducted.

3. In case of $\mathcal{B}roker^{GB}$, β of group $\mathcal{G}^1$ was normalized by means of the *deterministic gray box behavioral elaborating algorithm*. Then the extended-$\mathcal{I}^{GB}$ was applied in order to enable the scale capability of β. The result of this scaling is shown in Figure 4.19, where the ordinary β with $\lambda = 17$ is scaled up to the new β toward $\lambda = 19$. The compatibility degree of reusing service π of group $\mathcal{G}^3$ and $\mathcal{G}^4$ in place of agent intention in group $\mathcal{G}^1$ were 96.53% and 58.33%, respectively.

4. In case of $\mathcal{B}roker^{SY}$, the extended-$\mathcal{I}^{SY}$ was applied to β enhancing the scalable capability in the form of $\mathcal{I}^{SY}$. Figure 4.17 and Figure 4.20 present the visualized models of β and π. The circle symbols illustrated in 4.20 represent the positions of newly added on attribute values. As can be seen, these values of π from group $\mathcal{G}^3$ exhibit less significant impact on decision criteria, i.e., the entropy of these attribute values are almost 1. This means that these newly added attribute values do not excerpt any significant effects that will impact the rest of the data. On the other hand, the newly added attribute values of π from group $\mathcal{G}^4$ exhibit high significant impact on decision criteria. Thus, the service π of group $\mathcal{G}^4$ tends to be improper for reusing. The resulting compatibility degree of reusing service π of group $\mathcal{G}^3$ and $\mathcal{G}^4$ in place of agent intention in group $\mathcal{G}^0$ were 90.63% and 69.88%, respectively.

4.4 Experiment: Phase I (Evaluation of Equivalent Matchmaking)

To evaluate the accuracy of the proposed model, the *complete* service advertisement over the *complete* agent intention specification were compared. The experiment was performed over the **complete intention blueprints** in group $\mathcal{G}^0$ and $\mathcal{G}^1$. The sizes of conditional statements were all equal, whereby the test cases could be categorized as **equivalent-α match**, i.e., $\alpha^\beta = \alpha^\pi$.

This empirical experiment employed $\mathcal{B}roker^{WB}$, $\mathcal{B}roker^{SY}$, and $\mathcal{B}roker^{GB}$. The results obtained from $\mathcal{B}roker^{WB}$ were noted as **controlled results** of this equivalent-α matchmaking, while the results obtained from $\mathcal{B}roker^{SY}$ and $\mathcal{B}roker^{GB}$ were noted as **test results**. Experiment validation experiment was carried out on 900 additional cases. The t^{th} comparative results of an agent intention specification in group $\mathcal{G}^0$ executing its designated task by means of the corresponding t^{th} service in group $\mathcal{G}^1$ were reported to be the compatibility degree of validation. The accuracy of a synopsis specification and a deterministic gray box specification model could be performed by comparing testing results against controlled results. The higher accuracy degrees were acquired when stronger similarity between controlled results and test results were attained.

4.4.1 Experiment of the Synopsis Specification Model

The validation results obtained from both controlled results and testing results are depicted in Figure 4.22. Solid and dashed lines represent the compatibility degree of validation obtained from controlled results and test results, respectively. As presented in Figure 4.22, the compatibility degree of the results acquired from $\mathcal{B}roker^{SY}$ are nearly equal to the compatibility degree of the results achieved from $\mathcal{B}roker^{WB}$ in all aspects.

As shown in Table 4.4, the percentage of test accuracy are 95.96%, 95.56%, and

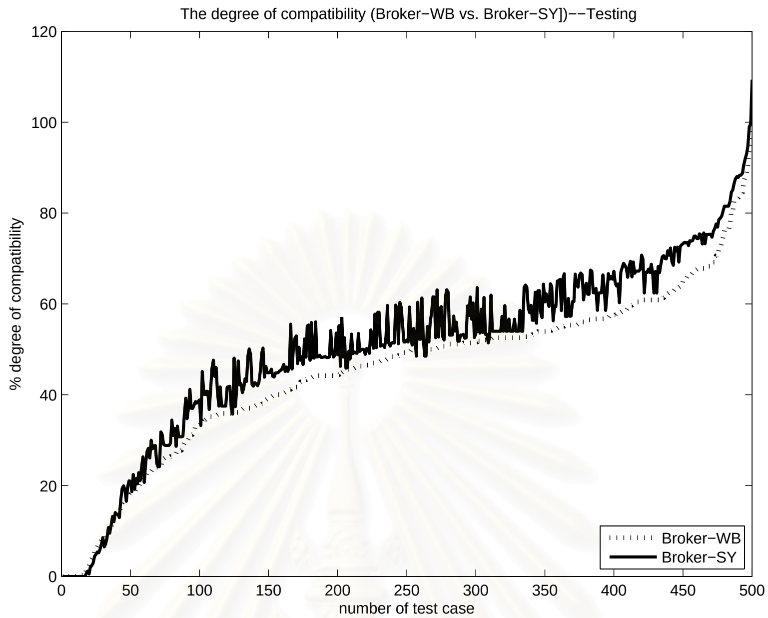

Figure 4.22: Results from equivalent-α matchmaking of $\mathcal{B}roker^{WB}$ and $\mathcal{B}roker^{SY}$

Table 4.4: Summary of equivalent-α matchmaking accuracy degree from $\mathcal{B}roker^{SY}$

Equations	mean	median	std	min	max
Linear	95.96	97.18	4.54	80.49	100.00
Quadratic	95.56	96.88	4.47	80.41	100.00
Cubic	96.94	97.00	4.05	80.31	100.00

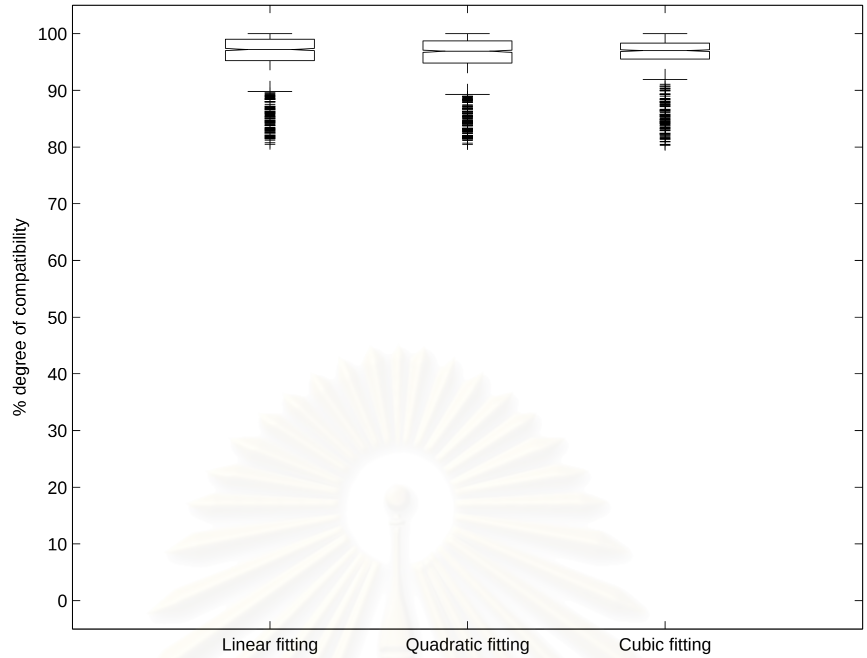

Figure 4.23: Box plots of equivalent-α matchmaking accuracy from $\mathcal{B}roker^{SY}$

96.94% achieved from applying linear, quadratic, and cubic curve fitting, respectively. The corresponding standard deviation are 4.54%, 4.47%, and 4.05%, respectively. The box-plots of these training accuracy percentage are shown in Figure 4.23. These validation results indicate that cubic fitting is the most suitable regression model for representing the relationship between the compatibility degree and the discrepancy degree. The average accuracy of $\mathcal{B}roker^{SY}$ with cubic fitting is 96.94% which is the highest value.

4.4.2 Experiment of the Deterministic Gray Box Specification Model

The validation results obtained from both $\mathcal{B}roker^{WB}$ and $\mathcal{B}roker^{GB}$ are illustrated in Figure 4.24. The compatibility results acquired from $\mathcal{B}roker^{GB}$, represented by the dotted-line, are almost the same as the compatibility results achieved from $\mathcal{B}roker^{WB}$, represented by solid-line. The compatibility results acquired from $\mathcal{B}roker^{GB}$ and $\mathcal{B}roker^{WB}$

are almost the same. Table 4.5 summarizes test accuracy statistics. The average accuracy achieved from this comparison is 95.52%, with a standard deviation of 3.46%. Generally, the lower difference acquired reflects the higher accuracy degree achieved. Degrees of test accuracy are summarized in Table 4.5.

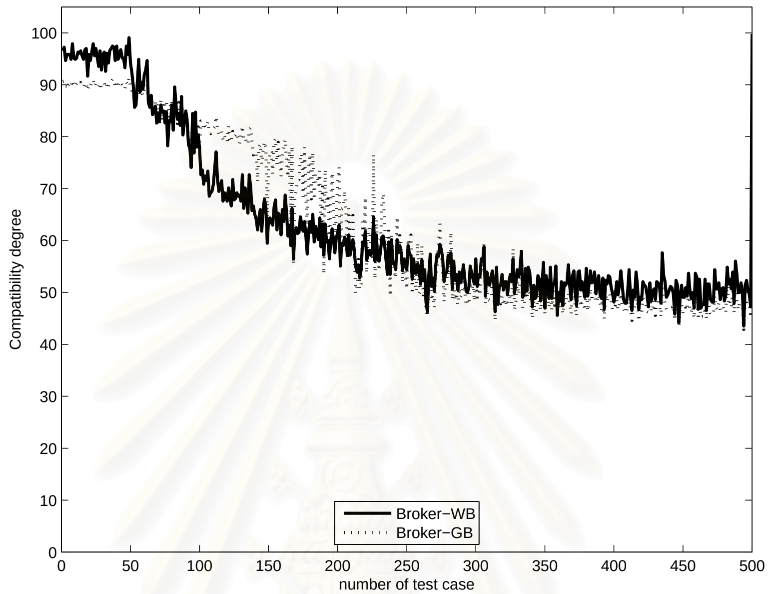

Figure 4.24: Results from equivalent-α matchmaking of both $\mathcal{B}roker^{WB}$ and $\mathcal{B}roker^{GB}$

Table 4.5: Summary of equivalent-α matchmaking accuracy degree from $\mathcal{B}roker^{GB}$

$Broker$	mean	median	std	min	max
$\mathcal{B}roker^{GB}$	95.52	96.64	3.46	85.57	99.98

4.5 Experiment: Phase II (Scalability over an Incomplete Declared Intention)

The objective of this experiment to evaluate the **scalability an incomplete declared intention** of the proposed approach against others. This experiment considers the matchmaking between *incomplete agent intention specifications* and *complete service advertisements*, having α^{β} and α^{π} being not equal. The value of α^{π} is equal to $\mathcal{M}$, whereas α^{β} is less than $\mathcal{M}$. This experiment was conducted basing upon the intention blueprints of $\mathcal{G}^0$ and $\mathcal{G}^2$. The intention blueprints of $\mathcal{G}^0$ were all *complete* agent intention specifications, but the intention blueprints of $\mathcal{G}^2$ was an *incomplete* agent intention specification. Set $\mathcal{G}^0$ to π and $\mathcal{G}^2$ to β. The experiment was performed over 500 test cases. The controlling factors stipulated that the i^{th} intention specification was a *subset* of the i^{th} service advertisement, i.e., $\alpha^{\beta} \prec \alpha^{\pi}$. Thus, the assumption of this experiment presumed the ideal results of compatibility degree between agent intention specifications and service advertisements which theoretically should be close to 100%.

4.5.1 Experiment of the Synopsis Specification Model

Compatibility results obtained from $\mathcal{B}roker^{WB}$ and $\mathcal{B}roker^{SY}$ over 500 cases of inequivalent matchmaking are shown in Figure 4.25, where the x-axis denotes number of test cases and the y-axis is compatibility degree. Table 4.6 shows the percentage of scaling test accuracy from this proposed model. The compatible degrees acquired from $\mathcal{B}roker^{WB}$ and $\mathcal{B}roker^{SY}$ are 24.77 % and 88.92%, respectively. The standard deviation degrees are reported to be 17.37% from $\mathcal{B}roker^{WB}$ and 3.96% from $\mathcal{B}roker^{SY}$.

As revealed by the box-plots in Figure 4.26, the results from $\mathcal{B}roker^{SY}$ are four times higher than those from $\mathcal{B}roker^{WB}$. These results satisfy the above assumption, thus it can be concluded that the synopsis model employing $\mathcal{B}roker^{SY}$ attributing to

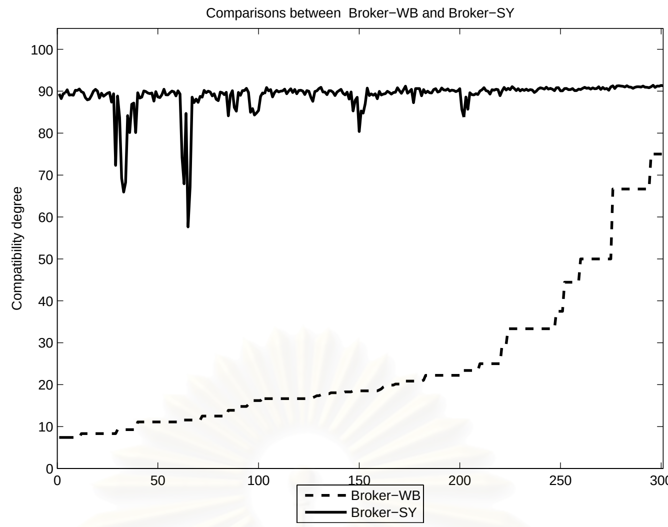

Figure 4.25: Results from inequivalent-α matchmaking of $\mathcal{B}roker^{WB}$ and $\mathcal{B}roker^{SY}$

Table 4.6: Summary of inequivalent-α matchmaking accuracy degree from $\mathcal{B}roker^{SY}$

$Broker$	mean	median	std	min	max
$\mathcal{B}roker^{WB}$	24.77	18.52	17.37	7.41	75.00
$\mathcal{B}roker^{SY}$	88.92	89.97	3.96	57.62	91.41

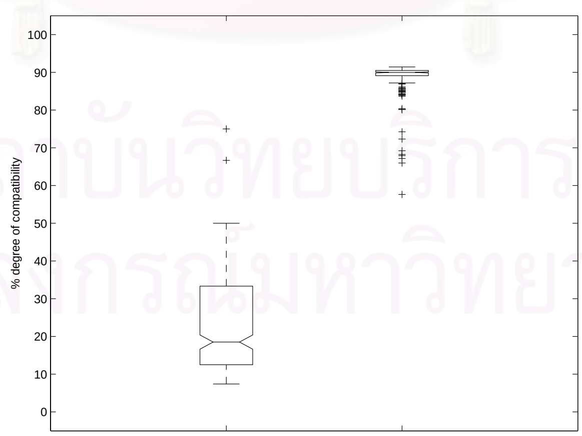

Figure 4.26: Box plots of inequivalent-α matchmaking accuracy from $\mathcal{B}roker^{SY}$

the limitation of $\mathcal{B}roker^{WB}$ on inequivalent-α matchmaking.

4.5.2 Experiment of the Deterministic Gray Box Specification Model

Compatibility results obtained from both $\mathcal{B}roker^{WB}$ and $\mathcal{B}roker^{GB}$ are compared. The results in Table 4.7 show that the compatible degree of $\mathcal{B}roker^{WB}$ and $\mathcal{B}roker^{GB}$ are 24.77%, 72.96%, and the corresponding standard deviation are 17.37% and 15.88%, respectively.

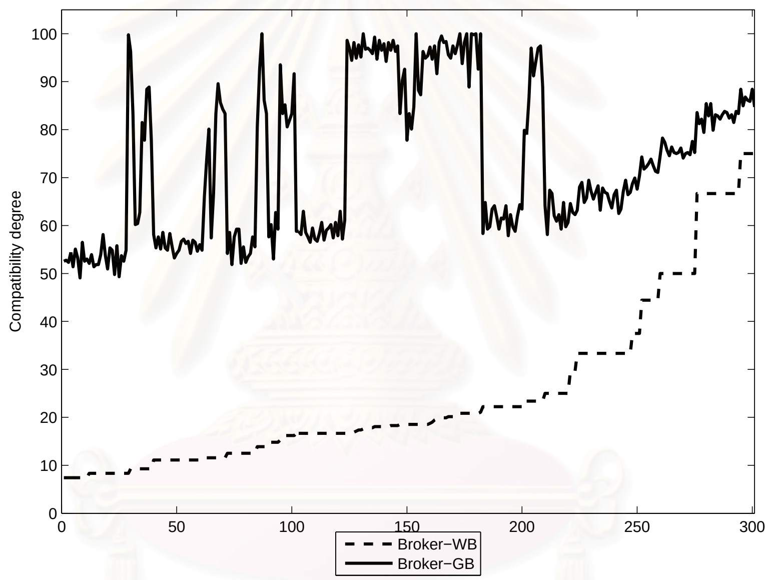

Figure 4.27: Results from inequivalent-α matchmaking of $\mathcal{B}roker^{WB}$ and $\mathcal{B}roker^{GB}$

Table 4.7: Summary of inequivalent-α matchmaking accuracy degree from $\mathcal{B}roker^{GB}$

$Broker$	mean	median	std	min	max
$\mathcal{B}roker^{WB}$	24.77	18.52	17.37	7.41	75.00
$\mathcal{B}roker^{GB}$	72.96	68.75	15.88	49.07	100.00

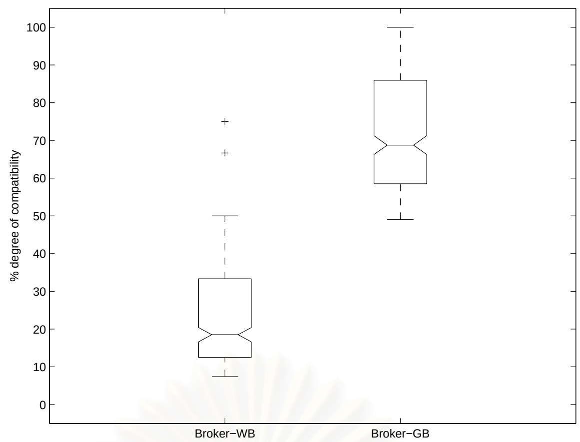

Figure 4.28: Box plots of equivalent-α matchmaking accuracy from $\mathcal{B}roker^{GB}$

As mention earlier, the compatibility degree obtained from either $\mathcal{B}roker^{WB}$ or $\mathcal{B}roker^{GB}$ might be close to 100%. As depicted by the box-plots in Figure 4.28, the results from $\mathcal{B}roker^{GB}$ are three times higher than those from $\mathcal{B}roker^{WB}$. However, the results from $\mathcal{B}roker^{GB}$, conduting one of the proposed approach, are slightly different from those ideal results. In addition, these results are somewhat less than the average results achieved from another proposed approach conducted by $\mathcal{B}roker^{SY}$. Thus, some affecting factors over this phenomena must be addressed.

Figure 4.29 presents some factors that probably influence the assessed compatibility degree of the proposed approach employed by $\mathcal{B}roker^{GB}$. The x-axis denotes number of test cases and the y-axis is the percentage of completeness, i.e., $\lambda \prec n$. For example, the value 100%, shown over dotted-line, indicates that all attributes values have been included as the concerning attributes. In addition, the dashed-line on y-axis describes percentage of *completeness specified conditional statements*, i.e., $\alpha \prec \mathcal{M}$. On the contrary, the value 50% experssed by dashed-line implies that all attributes values have been included as the concerning attributes.

Based on these results, high percentages of both *completeness specified attributes* and

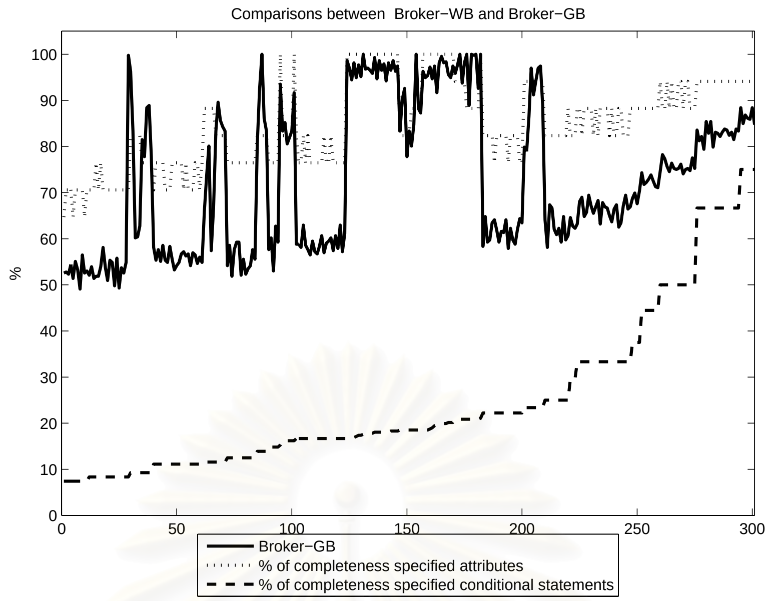

Figure 4.29: Influential factors over scalable capability of $\mathcal{B}roker^{GB}$

completeness specified conditional statements lead to high compatibility degree attained from $\mathcal{B}roker^{GB}$. This implies that compatibility degrees are directly proportional to percentages of completeness specified attributes. As illustrated in Figure 4.29, increasing of *completeness specified attributes* percentages causes the compatibility degree to increase. On the other hand, the percentages of *completeness specified conditional statements* have moderately less implication on compatibility degrees comparing with the influence from the percentages of completeness specified attributes.

Additionally, the evaluations are conducive to considering the results obtained from the case of 100% of *completeness specified attributes.* As demonstrated in Figure 4.30, resulting compatibility degree acquired from $\mathcal{B}roker^{GB}$ is the highest score of the results obtained from all brokers. Given these, it can be issued that the *completeness specified attributes* are the key factors that influece the accuracy of $\mathcal{B}roker^{GB}$. To achieve the highest accuracy, all attributes must be stated as concerning attributes with random probability. In other words, all attributes must relate to one or more conditions specified

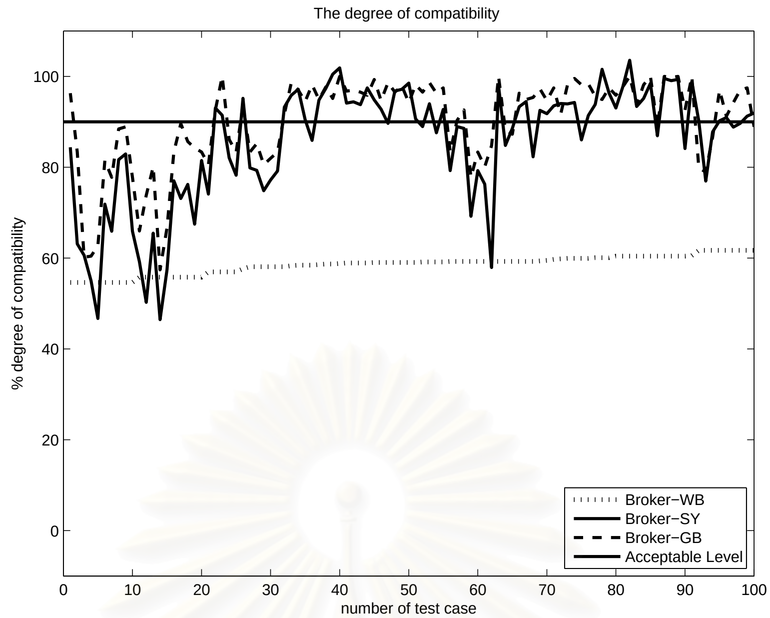

Figure 4.30: Results from inequivalent-α matchmaking of $\mathcal{B}roker^{WB}$, $\mathcal{B}roker^{SY}$, and $\mathcal{B}roker^{GB}$

in the intention specifications.

4.6 Experiment: Phase III (Scalability over an inequivalent decision criterion)

The objectives of these experiments are to compare the results obtained from $\mathcal{B}roker^{BL}$, $\mathcal{B}roker^{ND}$, $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ in two main aspects toward the **scalability over an inequivalent decision criteria**.

1. Comparison between the results attained from $\mathcal{B}roker^{BL}$ and $\mathcal{B}roker^{ND}$ are illustrated. To put it more simply, the advantages of a deterministic behavioral specification approach over a nondeterministic behavioral specification approach are evaluated.

2. Comparison between the results attained from $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$

are also illustrated. These three brokers were conducted with service validation algorithm basing on deterministic behavioral specification approach. The objective of this aspect is to underpin the advantages of the proposed validation algorithm employed by $\mathcal{B}roker^{GB}$ and $\mathcal{B}roker^{SY}$ over $\mathcal{B}roker^{BP}$ algorithm.

4.6.1 Evaluation of Validation Process

The objective of this section is to compare results obtained from $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ with *baseline results* conducted by $\mathcal{B}roker^{BL}$. Besides, the accuracy of matchmaking algorithms over behavioral validation are also evaluated.

Table 4.8 presents the results attained from $\mathcal{B}roker^{BL}$ comparing with the results obtained from $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$. The majority of test subjects yields the results acquired from these three brokers that are more or less the same as the results attained from $\mathcal{B}roker^{BL}$. With an exception of test subjects relating to intention blueprints in group $\mathcal{G}^0$ conducted by $\mathcal{B}roker^{BP}$, the results are roughly 0.0%. These discrepancies can be attributed to the overlooking of $\mathcal{B}roker^{BP}$ either in case of plug-in match or subsume match. In order to carry out evaluations of the proposed algorithms, the results obtained from $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ are proportioned to the results attained from $\mathcal{B}roker^{BL}$. These proportionate ratios are reported as percentage of accuracy results. The variables ϑ^{BP}, ϑ^{GB}, and ϑ^{SY} of **scalability over an inequivalent decision criteria** hold percentages of accuracy results obtained from those corresponding three brokers, respectively.

Table 4.9 summarized the percentages of accuracy results based upon the possibility characteristics of service matching results. Figure 4.31 illustrates the box-plots of these summarized results. The scope of every test subject and its results is elucidated below.

1. **Self validation testings**

Table 4.8: Results from service validation algorithms

(a) $\mathcal{B}roker^{BL}$				
β		$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$
π	$\mathcal{G}^0$	100.00	94.43	52.17
	$\mathcal{G}^3$	94.43	100.00	50.98
	$\mathcal{G}^4$	52.17	50.98	100.00

(b) $\mathcal{B}roker^{BP}$				
β		$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$
π	$\mathcal{G}^0$	97.41	0.00	0.00
	$\mathcal{G}^3$	0.00	97.82	51.13
	$\mathcal{G}^4$	0.00	51.13	98.70

(c) $\mathcal{B}roker^{GB}$				
β		$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$
π	$\mathcal{G}^0$	98.23	92.88	51.86
	$\mathcal{G}^3$	91.83	98.68	50.73
	$\mathcal{G}^4$	51.84	50.96	99.98

(d) $\mathcal{B}roker^{SY}$				
β		$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$
π	$\mathcal{G}^0$	100.00	93.15	49.59
	$\mathcal{G}^3$	93.15	100.00	48.62
	$\mathcal{G}^4$	49.59	48.62	100.00

Table 4.9: Percentage of accuracy results from service validation algorithms

(a) Self validation				(b) Exact match			
Approach	$\mathcal{B}roker^{BP}$	$\mathcal{B}roker^{GB}$	$\mathcal{B}roker^{SY}$	Approach	$\mathcal{B}roker^{BP}$	$\mathcal{B}roker^{GB}$	$\mathcal{B}roker^{SY}$
Mean	98.73	99.07	100.00	Mean	100.00	100.00	95.25
Median	98.79	98.96	99.21	Median	99.29	99.77	94.74
Std	1.05	1.08	1.07	Std	3.98	0.38	2.55
Min	95.60	95.60	95.60	Min	60.19	97.69	85.10
Max	100.00	100.00	100.00	Max	100.00	100.00	100.00
(c) Plug-in match				(d) Subsume match			
Approach	$\mathcal{B}roker^{BP}$	$\mathcal{B}roker^{SY}$	$\mathcal{B}roker^{GB}$	Approach	$\mathcal{B}roker^{BP}$	$\mathcal{B}roker^{SY}$	$\mathcal{B}roker^{GB}$
Mean	0.00	99.07	96.90	Mean	0.00	98.84	96.90
Median	0.00	98.94	95.56	Median	0.00	98.41	95.56
Std	0.00	0.92	4.40	Std	0.00	1.52	4.40
Min	0.00	96.06	39.76	Min	0.00	93.52	39.76
Max	0.00	100.00	100.00	Max	0.00	100.00	100.00

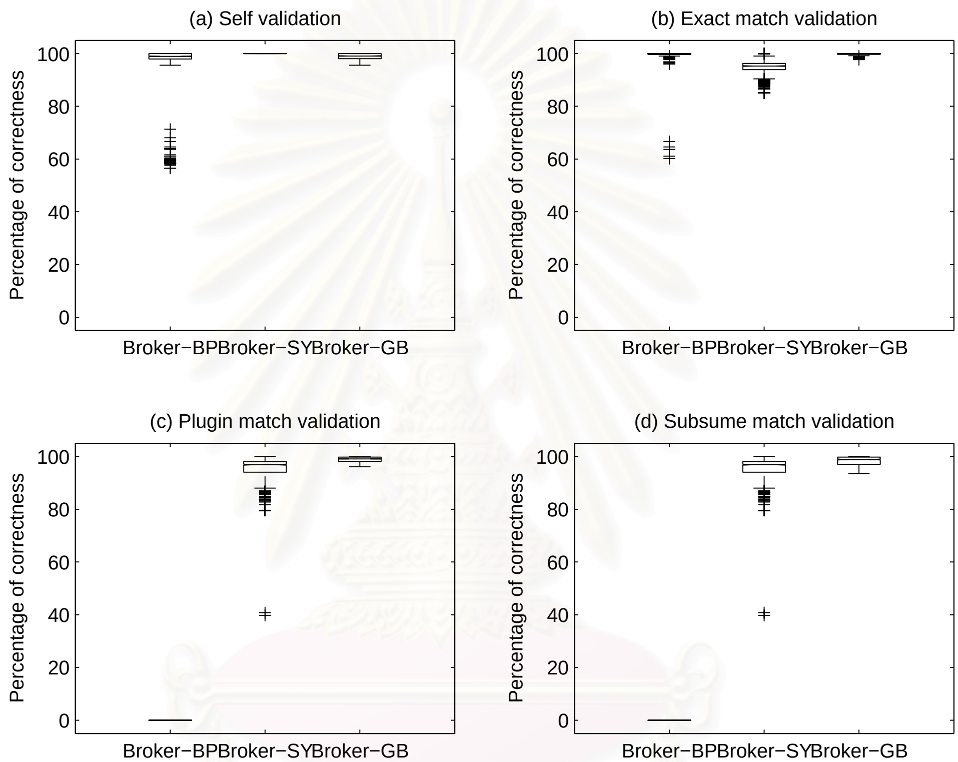

Figure 4.31: Box-plots of percentages of accuracy results from service validation algorithms

The *self validation testings* were experimented on the following samples.

> Let any t^{th} sample of an agent intention, $\beta^i(t)$, represent the t^{th} intention blueprint of $\mathcal{G}^i$, where $i = 0, 3, 4$. The *self validation testings* of this agent intention were conducted via $\Gamma(\beta^i(t), \pi^j(t))$, where $\pi^j(t)$ denoted the t^{th} intention blueprint of $\mathcal{G}^j$, $j = 0, 3, 4$, and $j = i$.

As can be seen in Table 4.9(a), the mean value of accuracy percentage from self validation testings performed by $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ are 98.73%, 99.07%, and 100.00%, respectively. Moreover, the median values acquired from these brokers are nearly 100%, indicated by the boxes in Figure 4.31(a).

2. **Exact match testings**

The *exact match testings* were experimented on the following samples.

> Let any t^{th} sample of an agent intention, denoted by $\beta^i(t)$, represent the t^{th} intention blueprint of $\mathcal{G}^i$, where $i = 4, 5$. The *exact match validation testings* of this agent intention were conducted via $\Gamma(\beta^i(t), \pi^j(t))$, where $\pi^j(t)$ denoted the t^{th} intention blueprint t^{th} of $\mathcal{G}^j$, $j = 4, 5$, and $j \neq i$.

Both Table 4.9(b) and Figure 4.31(b) provide percentages of accuracy from exact match testings. The average accuracy percentages obtained from $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ are 100.00%, 100.00%, and 95.25%, respectively.

3. **Plug-in match testings**

The *plug-in match testings* were experimented on the following samples.

> Let any t^{th} sample of an agent intention, denoted by $\beta^i(t)$, represent the t^{th} intention blueprint of $\mathcal{G}^i$, where $i = 4, 5$. The *plug-in match validation*

> *testings* of this agent intention were conducted via $\Gamma(\beta^i(t), \pi^j(t))$, where $\pi^j(t)$ denoted the t^{th} intention blueprint of $\mathcal{G}^j$, $j = 0$.

As illustrated in Table 4.9(c), the average accuracy percentages obtained from plug-in match testings conducted by $\mathcal{B}roker^{GB}$ and $\mathcal{B}roker^{SY}$ are 99.07% and 96.90%, respecitvely. It would be appropriate to say that these percentages of accuracy reach 100%. On the contrary, the average accuracy percentage acquired from $\mathcal{B}roker^{BP}$ is 0.00%.

4. **Subsume match testings**

The *subsume match testings* were experimented on the following samples.

> Let any t^{th} sample of an agent intention, taking the form $\beta^i(t)$, represent the t^{th} intention blueprint of $\mathcal{G}^i$, where $i = 0$. The *subsume match validation testings* of this agent intention were conducted via $\Gamma(\beta^i(t), \pi^j(t))$, where $\pi^j(t)$ denoted the t^{th} intention blueprint of $\mathcal{G}^j$, $j = 4, 5$.

On the average, accuracy percentages acquired from subsume match testings performed by $\mathcal{B}roker^{GB}$ and $\mathcal{B}roker^{SY}$ are 98.84% and 96.90%, respecitvely. As shown in Table 4.9(d), these accuracy percentages are almost 90.00%. By contrast, the percentage of accuracy obtained from $\mathcal{B}roker^{BP}$ is 0.00%.

These results indicate that both $\mathcal{B}roker^{GB}$ and $\mathcal{B}roker^{SY}$, performing extended scalable capability over inequivalent-λ, can be attributed to the limitation of conventional back-propagation approach, conducted via $\mathcal{B}roker^{BP}$. It would appear that the values of ϑ^{GB}, obtained from matchmaking process performed by $\mathcal{B}roker^{BP}$, are the highest score for every test aspect. In addition, the values of ϑ^{SY} attained from $\mathcal{B}roker^{SY}$ are more than 90% on average. These results point out that the degree of determinism acheived from both $\mathcal{B}roker^{SY}$ and $\mathcal{B}roker^{GB}$ are conceiveable closed to 100%.

4.6.2 Evaluation of Quality of Validation Process over Inequivalent Decision Criteria

The quality assessments of coarse-grain validation processes are carried out based on the results from service validation algorithms with the exact match, plug-in match, and subsume match. Conversely, self validation results are not included in these assessments. Evaluation measurements employ **recall** and **precision**. Section 4.6.2 and Section 4.6.2 reveal the results from recall and precision concerns, respectively.

In order to figure out recall and precision values obtained from each broker, the set of either relevant services or retrieved services for each testing situations must be indicated. A **set of relevant services** is named as $\tilde{\Pi}$, whereas a **set of candidate services** is previously defined as $\tilde{\Pi}$. An example of set $\tilde{\Pi}$ and set $\tilde{\tau}$ are illustrated as Table 4.10. In this table, $\tilde{\Pi}$ represents the results from $\mathcal{B}roker^{BL}$ and $\tilde{\tau}$ represents the results from others. There are two aspects to compare the quality of coarse-grain validation process obtained from $\mathcal{B}roker^{ND}$, $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{SY}$, and $\mathcal{B}roker^{GB}$ with *baseline results* conducted by $\mathcal{B}roker^{BL}$.

1. Comparison between the quality of coarse-grain validation process performed by $\mathcal{B}roker^{BL}$ and $\mathcal{B}roker^{ND}$ are illustrated. In the other words, the advantage of a deterministic behavioral specification approach over nondeterministic behavioral specification approach are evaluated.

2. Comparison between the quality of coarse-grain validation process performed by $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{SY}$, and $\mathcal{B}roker^{GB}$ are also illustrated. These three brokers were conducted with coarse-grain validation process based on the a deterministic behavioral specification approach. The objective of this aspects is to underpin the advantages of the proposed validation algorithm employed by $\mathcal{B}roker^{SY}$ and $\mathcal{B}roker^{GB}$ over $\mathcal{B}roker^{BP}$ algorithm with fixed number of training attributes.

4.6.2.1 Evaluation of Retrieved Results

Retrieved results in the set $\tilde{\tau}$s obtained from $\mathcal{B}roker^{ND}$, $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$ and $\mathcal{B}roker^{SY}$ are compared with the set $\tilde{\Pi}$s obtained from $\mathcal{B}roker^{BL}$ according to different values of σ.

1. **Comparisons between deterministic and nondeterministic behavioral specification approach**

   As shown in Table 4.10, there is no distinction in the set $\tilde{\tau}$s obtained from $\mathcal{B}roker^{ND}$ regarding different σ. Conversely, the set $\tilde{\tau}$s acquired from $\mathcal{B}roker^{BL}$, which conducts baseline process using deterministic behavioral specifications, are quite different in many aspects regarding different σ. The comparisons between deterministic and nondeterministic behavioral specification approach are as follows:

   - With regard to $\sigma = 30\%$, all services are elements in the set $\tilde{\tau}$s acquired from $\mathcal{B}roker^{BL}$. This is because all available services providing the same functionality yield the compatibility degree higher than 30%.
   - However, for higher values of σ, the set $\tilde{\tau}$s obtained from $\mathcal{B}roker^{ND}$ and $\tilde{\Pi}$s from $\mathcal{B}roker^{BL}$ are entirely different in many respects. In the majority of cases, the set $\tilde{\tau}$s appear to be the superset of $\tilde{\Pi}$s. These evidences show that some irrelevant services are retrieved from $\mathcal{B}roker^{ND}$.

   Because $\mathcal{B}roker^{ND}$ performs a service matchmaking algorithm on a nondeterministic behavioral specification, the behavioral similarity is **overlooked**. Thus, the values of σ are not concerned by this broker.

2. **Comparisons between different brokers relying on deterministic behavioral specification approach**

   The comparison for the set of retrieved results are as follows:

Table 4.10: Retrieved results from all $\mathcal{B}roker$s

(a) Retrieved results with $\sigma = 30\%$			
β	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$
$\mathcal{B}roker^{BL}$	$\{\mathcal{G}^4, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^4\}$
$\mathcal{B}roker^{ND}$	$\{\mathcal{G}^4, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^4\}$
$\mathcal{B}roker^{BP}$	$\{\mathcal{G}^4, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^4\}$
$\mathcal{B}roker^{SY}$	$\{\mathcal{G}^4, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^4\}$
$\mathcal{B}roker^{GB}$	$\{\mathcal{G}^4, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^4\}$

(b) Retrieved results with $\sigma = 70\%$			
β	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$
$\mathcal{B}roker^{BL}$	$\{\mathcal{G}^4, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^4\}$
$\mathcal{B}roker^{ND}$	$\{\mathcal{G}^4, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^4\}$
$\mathcal{B}roker^{BP}$	$\varnothing$	$\{\mathcal{G}^5\}$	$\{\mathcal{G}^4\}$
$\mathcal{B}roker^{SY}$	$\{\mathcal{G}^4, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^4\}$
$\mathcal{B}roker^{GB}$	$\{\mathcal{G}^4, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^4\}$

(c) Retrieved results with $\sigma = 90\%$			
β	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$
$\mathcal{B}roker^{BL}$	$\{\mathcal{G}^4\}$	$\{\mathcal{G}^0\}$	$\varnothing$
$\mathcal{B}roker^{ND}$	$\{\mathcal{G}^4, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^5\}$	$\{\mathcal{G}^0, \mathcal{G}^4\}$
$\mathcal{B}roker^{BP}$	$\varnothing$	$\varnothing$	$\varnothing$
$\mathcal{B}roker^{SY}$	$\{\mathcal{G}^4\}$	$\{\mathcal{G}^0\}$	$\varnothing$
$\mathcal{B}roker^{GB}$	$\{\mathcal{G}^4\}$	$\{\mathcal{G}^0\}$	$\varnothing$

- **Comparisons between $\mathcal{B}roker^{BL}$ and $\mathcal{B}roker^{BP}$**

  According to β of group $\mathcal{G}^0$, the set $\tilde{\tau}$s acquired from $\mathcal{B}roker^{BP}$ are all empty sets. By contrast, the set $\tilde{\Pi}$s acquired from $\mathcal{B}roker^{BL}$ contain candidate services. Moreover, there are others situations that the set $\tilde{\tau}$s become subsets of set $\tilde{\Pi}$s. For example, in case of $\sigma = 70\%$ and β of group $\mathcal{G}^3$, the set $\tilde{\Pi}$ is $\{\mathcal{G}^0, G^5\}$, whereas $\tilde{\tau}$ is $\{\mathcal{G}^5\}$.

- **Comparisons between $\mathcal{B}roker^{BL}$, $\mathcal{B}roker^{SY}$, and $\mathcal{B}roker^{GB}$**

  The set $\tilde{\tau}$s, which are acquired from both $\mathcal{B}roker^{GB}$ and $\mathcal{B}roker^{SY}$, are precisely the same as $\tilde{\Pi}$s, which are acquired from $\mathcal{B}roker^{BL}$ for every test subject.

These results suggest that some relevant services are unnoticed by $\mathcal{B}roker^{BP}$, while the proposed algorithms perform as well as what is specified in the set $\tilde{\Pi}$s of $\mathcal{B}roker^{BL}$.

4.6.2.2 Evaluation of Recall Results

Table 4.11 shows the comparative results of recall values obtaining from $\mathcal{B}roker^{ND}$, $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$. Box-plots of these evaluation results are illustrated in Figure 4.32. As can be seen, the average recall value of $\mathcal{B}roker^{ND}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ is approximately to 1.00. Nonetheless, the recall value of $\mathcal{B}roker^{BP}$ is almost three times lower than that of the rest.

From these results, it can be inferred that some services are overlooked by $\mathcal{B}roker^{BP}$ due to certain cases of service matchmaking, plug-in match, and subsume match being considered in this experiments are beyond the capability of $\mathcal{B}roker^{BP}$.

Table 4.11: Results of recall degree from service validation algorithm

(a) $\mathcal{B}roker^{ND}$			
threshold	Mean	Median	Std.
$\sigma = 30\%$	1.00	1.00	0.00
$\sigma = 70\%$	1.00	1.00	0.00
$\sigma = 90\%$	1.00	1.00	0.00
(b) $\mathcal{B}roker^{BP}$			
threshold	Mean	Median	Std.
$\sigma = 30\%$	0.34	0.50	0.09
$\sigma = 70\%$	0.33	0.00	0.03
$\sigma = 90\%$	0.33	0.00	0.03
(c) $\mathcal{B}roker^{SY}$			
threshold	Mean	Median	Std.
$\sigma = 30\%$	0.99	1.00	0.02
$\sigma = 70\%$	1.00	1.00	0.00
$\sigma = 90\%$	0.92	1.00	0.18
(d) $\mathcal{B}roker^{GB}$			
threshold	Mean	Median	Std.
$\sigma = 30\%$	1.00	1.00	0.00
$\sigma = 70\%$	1.00	1.00	0.00
$\sigma = 90\%$	0.91	1.00	0.22

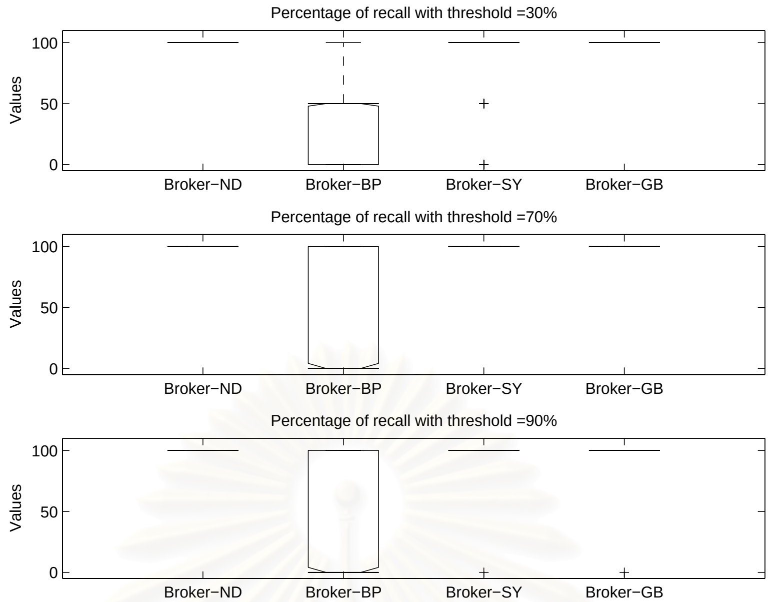

Figure 4.32: Box-plots degree of recall results from service validation algorithm

4.6.2.3 Evaluation of Precision Results

The comparison of precision values obtaining from $\mathcal{B}roker^{ND}$, $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ are presenting in Table 4.12. The box-plots of these evaluation results are illustrated in Figure 4.33. As can be seen, the average precision value of $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$ is close to 1.00. On the other hand, the precision value of $\mathcal{B}roker^{ND}$ reversely decreases in comparison with σ.

These precision values indicate that relatively few irrelevant services are selected by $\mathcal{B}roker^{BP}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$. These precision values are indicative of some irrelevant services being considered as relevant services by $\mathcal{B}roker^{ND}$.

4.6.2.4 Evaluation of Service Matchmaking Accuracy Result

In this section, the expectation percentage over compatible degrees, previously defined as ζ, were evaluated. As revealed in Table 4.13, the average values of ζ acquired from $\mathcal{B}roker^{BL}$ are higher than σ in every aspect. This growth benefits from the efficiency

Table 4.12: Results of precision degree from service validation algorithm

(a) $\mathcal{B}roker^{ND}$			
threshold	Mean	Median	Std.
$\sigma = 30\%$	0.92	1.00	0.85
$\sigma = 70\%$	0.33	0.50	0.13
$\sigma = 90\%$	0.33	0.50	0.13
(b) $\mathcal{B}roker^{BP}$			
threshold	Mean	Median	Std.
$\sigma = 30\%$	0.99	1.00	0.52
$\sigma = 70\%$	1.00	1.00	0.00
$\sigma = 90\%$	1.00	1.00	0.00
(c) $\mathcal{B}roker^{SY}$			
threshold	Mean	Median	Std.
$\sigma = 30\%$	0.92	1.00	0.85
$\sigma = 70\%$	1.00	1.00	0.00
$\sigma = 90\%$	1.00	1.00	0.00
(d) $\mathcal{B}roker^{GB}$			
threshold	Mean	Median	Std.
$\sigma = 30\%$	1.00	1.00	0.85
$\sigma = 70\%$	1.00	1.00	0.00
$\sigma = 90\%$	1.00	1.00	0.00

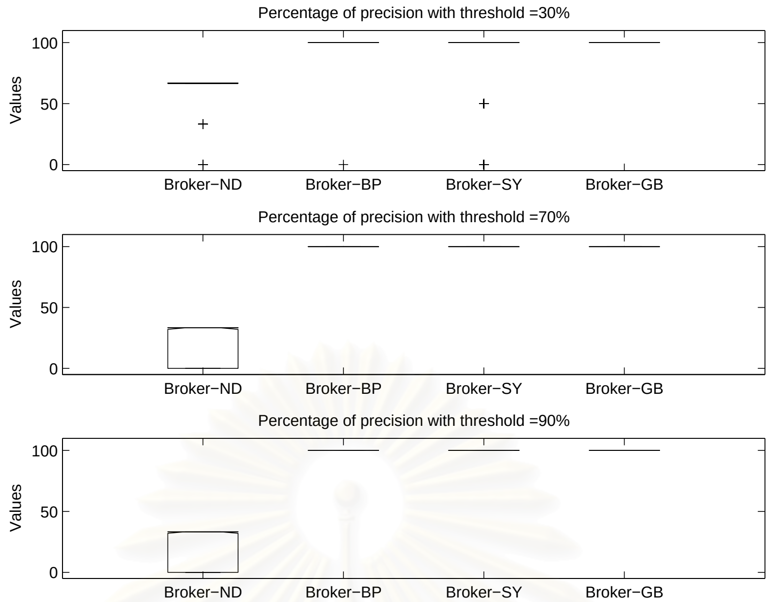

Figure 4.33: Box-plots degree of precision results from service validation algorithm

of coarse-grain validation process using deterministic behavioral specifications. The services, which are potential risks to cause the expectation percentage over accuracy results dropping lower that σ, should be eliminated from the coarse-grain validation.

For instance, $\mathcal{B}roker^{BL}$ yields 94.43% and 94.43%, on average expectation percentage over accuracy results from $\mathcal{G}^0$ and $\mathcal{G}^3$ having $\sigma = 30\%$. These results are higher than the results acquired from the testing with either $\sigma = 70\%$ or $\sigma = 90\%$. Thus, the values of ζ confirm that $\mathcal{B}roker^{BL}$ could perform higher than what was expected ($\sigma = 70\%$ and $\sigma = 90\%$) in these two groups.

Nevertheless, not all requirements should be guaranteed by $\mathcal{B}roker^{BL}$ to reach equal or higher ζ than those specified by σ. For example, $\mathcal{B}roker^{BL}$ performing requirement matchmaking in $\mathcal{G}^4$ yields a non-candidate service with $\sigma = 90\%$. The scores reported from this broker are the maximum expectation scores which the agent will achieve while continue selecting the services amid lacking of availabile appropriate services. The result of the above example is 57.41%.

Figure 4.34 depicts the comparison between different brokers relying on a deterministic behavioral specification approach based on the comparative statistics in Table 4.13.

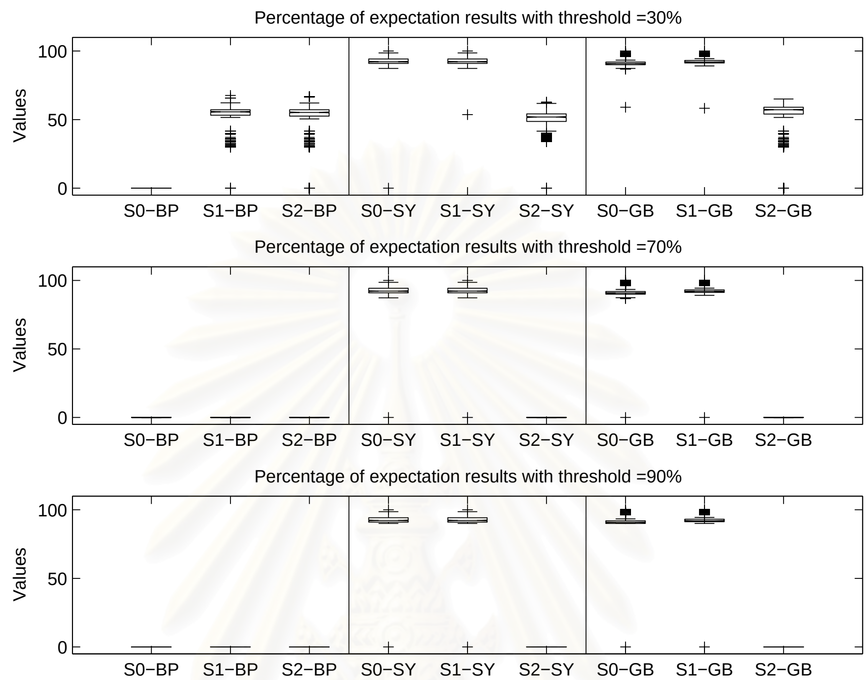

Figure 4.34: Box-plots of expectation results of three different deterministic behavioral specification approaches

Further analysis of the evaluation results entails a few comparative aspects as follows:

1. **Comparison between $\mathcal{B}roker^{BL}$ and $\mathcal{B}roker^{BP}$**

   As shown in Table 4.13, the values of ζ acquired from $\mathcal{B}roker^{BP}$ are dissimilar to what is acquired from $\mathcal{B}roker^{BL}$ in many respects. These discrepancies can be explained as follows:

   - With regard to $\sigma = 30\%$, the values of ζ obtained from $\mathcal{B}roker^{BP}$ and $\mathcal{B}roker^{BL}$ were different. Eventhough the results were all nonzeros, but the values of ζ obtained from $\mathcal{B}roker^{BP}$ were lower than those of $\mathcal{B}roker^{BL}$.

Table 4.13: Expectation results of all $\mathcal{B}roker$s

(a) Expectation results with $\sigma = 30\%$									
	Mean			Median			Std		
Requirement	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$
$\mathcal{B}roker^{BL}$	94.43	94.43	50.13	93.87	93.87	57.41	2.35	2.39	18.14
$\mathcal{B}roker^{BP}$	0.00	49.24	48.79	0.00	55.67	55.21	0.00	16.71	16.95
$\mathcal{B}roker^{GB}$	94.64	94.06	51.90	93.16	92.28	52.32	6.36	5.00	4.32
$\mathcal{B}roker^{SY}$	91.83	92.88	49.83	90.86	92.01	57.18	3.42	2.99	18.04
(b) Expectation results with $\sigma = 70\%$									
	Mean			Median			Std		
Requirement	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$
$\mathcal{B}roker^{BL}$	94.31	94.31	0.00	93.87	93.87	0.00	4.57	4.57	0.00
$\mathcal{B}roker^{BP}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\mathcal{B}roker^{GB}$	94.64	94.06	0.00	93.16	92.28	0.00	6.36	5.00	0.00
$\mathcal{B}roker^{SY}$	91.71	92.77	0.00	90.86	92.01	0.00	5.14	4.88	0.00
(c) Expectation results with $\sigma = 90\%$									
	Mean			Median			Std		
Requirement	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$	$\mathcal{G}^0$	$\mathcal{G}^3$	$\mathcal{G}^4$
$\mathcal{B}roker^{BL}$	94.31	94.31	0.00	93.87	93.87	0.00	4.57	4.57	0.00
$\mathcal{B}roker^{BP}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\mathcal{B}roker^{GB}$	90.88	85.50	0.00	93.16	92.28	0.00	20.05	28.29	0.00
$\mathcal{B}roker^{SY}$	70.13	90.62	0.00	90.86	92.01	0.00	39.97	15.03	0.00

For higher σ, however, the values of ζ obtained from $\mathcal{B}roker^{BP}$ were zeros, whereas those of $\mathcal{B}roker^{BL}$ were not. This is because $\mathcal{B}roker^{BP}$ did not consider the plug-in match, thereby some relevant services are unnoticed by this broker, while $\mathcal{B}roker^{BL}$ could still manage to find some available services.

- Similarly, $\mathcal{B}roker^{BP}$ did not devote its consideration to the case of subsume match. For requirements of $\mathcal{G}^0$, this broker was unable to determine the values of ζ for the available services. Accordingly, the results obtained from this broker dropped to 0.00%.

2. **Comparison of $\mathcal{B}roker^{BL}$, $\mathcal{B}roker^{GB}$, and $\mathcal{B}roker^{SY}$**

   In case of $\mathcal{B}roker^{GB}$, there was no distinction among the set of retrieval results that compared $\mathcal{B}roker^{BL}$ and $\mathcal{B}roker^{GB}$ statistics as shown in Table 4.10. Hence, the values of ζ obtained from $\mathcal{B}roker^{GB}$, for any requirement, were almost the same as those from $\mathcal{B}roker^{BL}$. On the other hand, the average of ζ from $\mathcal{B}roker^{SY}$ with $\sigma = 90\%$ were distinctively different from those from $\mathcal{B}roker^{BL}$. This means that some relevant service could be overlooked by $\mathcal{B}roker^{BP}$ due to underestimated compatibility between β and π acquired from Equation 3.11.

4.7 Discussions

The MA has emerged as realization of attempts to enable the adaptation under dynamic and unpredictable environments. This is, in part, because different hosts contain different resources. In order to model such an intelligent MA that can adapt itself to any foreign environments, all types of possible scenarios must be taken into account which unfortunately leads to an imploded MA. Furthermore, it is difficult to anticipate all possible situations that the MA must encounter during design time, and the prior knowledge will not be suitable for the new situation. Thus, what interesting is the external adaptation mechanism contributing toward the uncertainty dynamic environment.

To utilize an adaptation mechanism, one of the fundamental principles is to rely on service discovery technology. The MA contacts the service broker to ask for recommendation with an agent intention specification encompassed with user's profile and the environment of the current host's context. Upon receipt such *an agent intention specification*, the service broker performs service discovery mechanism, by means of *matchmaking* process, to make a recommendation of the suitable services based upon *a service advertisement*. Because of the exact match between an agent intention specification and a service advertisement is an ideal case, for some powerful brokers, the matchmaking process returns similarity degree, in which states the qualitative trustworthy degree of an agent over some the recommended services. This qualitative degree can be used as a decision criterion for selecting the most suitable service before any executions will commence.

A concise summary of convention $\mathcal{B}roker$ service discovery approaches, conducted over both nondeterministic and deterministic behavioral specification, were compared with the proposed approaches. $\mathcal{B}roker$s' capabilities are summarized in Table 4.14. The symbol $-$ indicates the unconcerning capability, whereas the symbol $\surd$ indicates the highly concerning capability of the designated approach. Details of the comparisons will be categorically elucidated in the sections that follow.

4.7.1 Nondeterministic Behavioral Specification

The instance of this approach has been illustrated as $\mathcal{B}roker^{ND}$. Some evidences show that this kind of service discovery are practical enough for devising an adaptation mechanism for the MA operating in uncertain environments. One of the main arguments is that, in all subjects of testings, $\tilde{\tau}$s acquired from $\mathcal{B}roker^{ND}$ were not empty sets. This

Table 4.14: Capabilities of all $\mathcal{B}roker$s

Approach	$\mathcal{B}roker^{ND}$	$\mathcal{B}roker^{WB}$	$\mathcal{B}roker^{BP}$	$\mathcal{B}roker^{GB}$	$\mathcal{B}roker^{SY}$
(1) Functional assessment	default	default	default	default	default
(2) Behavioral assessment	-	$\surd$	$\surd$	$\surd$	$\surd$
(3) Scalability over incomplete declared intention	undefined	$-$	$\surd$	$\surd$	$\surd$
(4) Scalability over inequivalent decision criterion	undefined	$-$	$-$	$\surd$	$\surd$
(5) Size of behavioral specification	-	$\alpha \times \mathcal{N}$	$(S \times \mathcal{N})$	$(S \times \mathcal{N})$	$2 \times \mathcal{N}$
(6) Comparison complexity	-	$\mathcal{O}(\mathcal{M})$	$\mathcal{O}(\mathcal{M})$	$\mathcal{O}(\mathcal{M})$	$\mathcal{O}(\mathcal{N})$

is due to the fact that no decision criteria is required to specified an intention in nondeterministic fashion, as well as the problems causing from the discrepancies between β and π are overlooked. The size of behavioral specification in an intention component is zero since there is no explicit specification about the MA's behavior. Even though this kind of specification is compact and practical for conducting over the uncertain environments, the values of ζ achieved from $\mathcal{B}roker^{ND}$ are low. These negative evidences indicate that this nondeterministic behavioral specification model requires some forms of deterministic behavioral representation.

4.7.2 Deterministic White Box Specification

The deterministic white box specification, noted previously as a clear-sighted approach, is announced as a practical approach for use in closed systems. The examples in this dissertation refer to it as $\mathcal{B}roker^{WB}$ and $\mathcal{B}roker^{BL}$. In many phases, $\mathcal{B}roker^{BL}$ is employed in various controlled experiments as an ideal cases. All situations which the MA will encounter must be all known to make use of the *controlled results.* On the other hand, $\mathcal{B}roker^{WB}$ employs a typical form of $\mathcal{I}^{WB}$ which will be discussed below.

Normally, in the matchmaking process, the process commences on case comparison between the request intention specifications and the collected service advertisements. The most suitable service will be recommended as a result to the requester. It is noticed that this approach satisfies equivalent matchmaking between intention specifications and service advertisements. Typically, this approach is restricted to uncertain environments which are notably unpredictable. In addition, the required resources are often not all known or adequately characterized in advance, and may change with time. It is, therefore, impossible to specify all situations the MA will encounter. This leads to an impractical use under uncertain environments.

The size of intention component for an MA employed $\mathcal{I}^{WB}$ as its specification are approximately equal to $\alpha \times \mathcal{N}$, where $\mathcal{N}$ is a finite number for each domain of knowledge. In case of a **complete behavioral specification**, the value of α is equal to m, i.e., the size of both β and π are $\mathcal{M} \times \mathcal{N}$. By contrast, for an **incomplete behavioral specification**, whose partial situations are specified, the variable α is the number of declared situations, i.e., the size of β and π is $\alpha \times \mathcal{N}$ and $\mathcal{M} \times \mathcal{N}$, respectively. The *detailed validation assessment* is applied for $\mathcal{I}^{WB}$, and the time for comparison is given below.

For any i^{th} tuple in matrix δ of a scaled-β, $1 \leq i \leq \alpha$ and $\alpha \leq m$,

- 1^{st} tuple in matrix δ^{β}, it takes m iterations for searching the *similar* tuple in matrix δ of a scaled-π, 1-pairs of elements is compared.
- 2^{nd} tuple in matrix δ^{β}, it takes $m-1$ iterations for searching the *similar* tuple in matrix δ of a scaled-π, 1-pairs of elements is compared.

- 3^{rd} tuple in matrix δ^{β}, it takes $m-2$ iterations for searching the *similar* tuple in matrix δ of a scaled-π, 1-pairs of elements is compared.
- ...
- α^{th} tuple in matrix δ^{β}, it takes $m-\alpha-1$ iterations for searching the *similar* tuple in matrix δ of a scaled-π, 1-pairs of elements is compared.

The comparison will be performed over $\Omega^{\beta}(i)$ is compared with $\Omega^{\pi}(k)$, where the k^{th} index is a position of the *similar* tuple, corresponding to i^{th} tuple in δ^{β}, in matrix δ of a scaled-π. The comparison can not be performed sequentially since the number of α^{β} and α^{π} are not equal. The comparison time can be quantified as

$$T = \mathcal{M} + (\mathcal{M}-1) + (\mathcal{M}2) + \ldots + (\mathcal{M}-\alpha-1)$$

The comparison complexity can be written as $\mathcal{O}(\mathcal{M})$.

4.7.3 Deterministic Gray Box Specification

The back-propagation learning algorithm, successfully implemented in machine learning research, is chosen as the induction learning algorithms. This approach is conducted by two brokers as

1. $\mathcal{B}roker^{BP}$ performed matchmaking algorithm based on $\mathcal{I}^{GB}$ without any progress was made on scalable capability over inequivalent-$\mathcal{N}$. Even though the new or unknown decision making situation can be assessed, it puts a limit on induction capability over the fixed size of attributes feeding in a leaning step. This means *scalability over an incomplete declared intention* is encompassed to $\mathcal{I}^{GB}$ employed by $\mathcal{B}roker^{BP}$, but *scalability over an inequivalent decision criteria* is not covered. The results from experiments are suggested as, in some of testing subjects, there are

some empty sets $\tilde{\tau}$s causing from the interruptions of conducting service validation algorithm.

Typically, the size of intention component for MA employed $\mathcal{I}^{GB}$ as its specification are approximately equal to $S \times \mathcal{N}$. A number of overall attribute values, $\mathcal{N}$, is a finite number for each knowledge domain. The *detailed validation assessment* is applied for this specification type. The comparison time is given as following.

> For any i^{th} tuple in matrix δ of a scaled-β, $1 \leq i \leq m$, it takes 1 iteration for searching the *similar* tuple in matrix δ of a scaled-π. For each tuple, 1-pairs of elements is compared. In other words, $\Omega^{\beta}(i)$ is compared with $\Omega^{\pi}(i)$.

After applied matrix operations for extracting all possible situations, the comparison can be performed sequentially case-by-case comparison. The value of m can be asserted from the finite common set of attribute values respecting to $\mathcal{N}$. The comparison complexity can be written as $\mathcal{O}(\mathcal{M})$.

2. $\mathcal{B}roker^{GB}$ performed matchmaking algorithm based on $\mathcal{I}^{GB}$ with further its process on extending such a scalable capability over inequivalent-$\mathcal{N}$. According to the evidences reporting as the execution results acquiring from $\mathcal{B}roker^{GB}$, it can be said that there are the significant aspects toward the attempt of extended the capability of a service validation algorithm to cope with an inequivalent of decision criteria. The contribution of this effort is presented a significant value in the percent of correction over the plug-in and subsume match comparing with the results obtained from $\mathcal{B}roker^{BP}$. Furthermore, the highest values of recall are achieved from $\mathcal{B}roker^{GB}$. This indicates that our proposed model is enabling the scalability capability to both an agent intention specification and a service advertisement performing under uncertain environments. Besides, the highest value of precision

can be achieved from $\mathcal{B}roker^{GB}$. This indicates that only the services which behave as the behavior, specified in an agent intention specification, are selected. Hence, we can ensure that the value of ζ will be higher than the σ. In case of no service is classified into a relevant set, the returned results are the maximum expectation value of accuracy results of the available services. Thus, these results suggest that the deterministic behavioral specification approach enable percents of accuracy results ensure of the services provided by $\mathcal{B}roker^{GB}$.

In order to cope with uncertain environments, $\mathcal{B}roker^{GB}$, employing its intention in form of $\mathcal{I}^{GB}$, makes some further steps in conducting an extended-$\mathcal{I}^{GB}$ elicitation algorithm. The size of β and π in form of $\mathcal{I}^{GB}$ are changed from an $S \times \lambda$ matrix to an $S \times \mathcal{N}$ matrix, where $\mathcal{N} = max(\lambda^{\beta}, \lambda^{\pi})$.

Hence, the size of intention component for MA employed $\mathcal{I}^{GB}$ as its specification are approximately equal to $S \times \mathcal{N}$. The validation assessment of this type of specification is conducted over a *detailed validation assessment*. The comparison time is presented below.

> For any i^{th} tuple in matrix δ of a scaled-β, $1 \leq i \leq \mathcal{M}$, it takes 1 iteration for searching the *similar* tuple in matrix δ of a scaled-π. For each tuple, 1-pairs of elements is compared, i.e., $\Omega^{\beta}(i)$ is compared with $\Omega^{\pi}(i)$.

With similar to $\mathcal{B}roker^{BP}$, the comparison can be performed sequentially case-by-case. In contrary with $\mathcal{B}roker^{BP}$, the value of m can be asserted from the set of attribute values respecting to $\mathcal{N}$. The comparison complexity takes a form as $\mathcal{O}(\mathcal{M})$. Moreover, the experimental results indicate that ϑ^{BP} conducting over self validation, exact match, plug-in match, and subsume match are 99.07%, 100.00%, 99.07%, and 98.84% on average, respectively. It can be concluded that, for noncongestion network, the MA employing the $\mathcal{I}^{GB}$ and performing the essential processes conducting by $\mathcal{B}roker^{GB}$, is potent enough to enable intelligent adaptation capabilities of the agent to accommodate the user's purpose.

4.7.4 A Synopsis of Deterministic Intention Specification

The synopsis of deterministic intention specification is nominated with the objective to model a compact form of MA. The estimation model of compatibility assessment are proposed. The evaluation of this estimation model is carried out with the help of statistical hypothesis testing. By the way of illustration $\mathcal{B}roker^{SY}$ is conducted as an instance of the propose approach. The size of intention component for MA employed $\mathcal{I}^{SY}$ as its specification are approximately equal to $2 \times \mathcal{N}$. The validation assessment of this type of specification is conducted over a *rough validation assessment* and the comparison time takes the form as below:

> For any j^{th} concerning attribute value, it takes 1 iteration for searching the information about j^{th} concerning attribute value. There are 2-pair of elements are compared, $\mathcal{E}^{\beta}(j)$ is compared with $\mathcal{E}^{\pi}(j)$, and $\mathcal{P}^{\beta}(j)$ is compared with $\mathcal{P}^{\pi}(j)$.

This broker takes $2 \times \mathcal{N}$ times for all comparison, i.e., the comparison complexity is $\mathcal{O}(\mathcal{N})$. This information suggest that, in case of congestion networks, the light-weighted MA can be composed in form of $\mathcal{I}^{SY}$. The experimental results indicate that ϑ^{SY} conducting over self validation, exact match, plug-in match, and subsume match are 100.00%, 95.25%, 96.90%, and 96.90% on average, respectively. These evidences support that the MA employed $\mathcal{I}^{SY}$ as its intention component and the vital processes conducting by $\mathcal{B}roker^{SY}$, permits the MA to be potent enough to accomplish its tasks within the acceptable users' satisfaction over uncentain environments.

CHAPTER V

CONCLUSION

Currently, service discovery technology integrates researches on several communities, such as reuse of Component-Based Software Engineering (CBSE), information retrieval of natural language service description, etc. It is apparent that modeling intention component of the MA with loosely or lack of concern on MA behavior often leads to low qualitative results assessed by the matchmaking process. Similarly, overlooking of incorporating scalable capabilities into specifications also brings about low qualitative values.

This study offers some insight to the behavioral specification in dynamic and unpredictable environments, in particular, under an incomplete declared intention setting. One precondition to establish a noble model is the agent's actions that must be transformed into bipolar attitude scales (favorable-unfavorable), thereby matching process can be performed. Some essence of research findings are summarized below.

First, a deterministic gray box intention specification is proposed as a detailed encapsulated form of agent behavior. The merits of elaborating a deterministic gray box specification in the form of matrix representation contribute to a deterministic service discovery as "how to access the service" under uncertain environment. This notion is practical enough to be incorporated as a part of the conventional table-based specification approach. The logical form of a behavioral model refines the service capabilities using state transition diagrams, whereas the physical form is represented in matrix form.

The matching process can thus be conducted by matrix manipulation procedures. Because matrix representation is closely related to attribute value fields, each matrix can be technically transformed to attribute field. All available matrix manipulation operations help facilitate a practical matchmaking process toward the proposed specification.

Second, the synopsis of deterministic intention specification is nominated. The proposed model compacts the deterministic agent intention specification to be suitable for applying to the MA. Similar to a deterministic gray box intention specification, the synopsis of deterministic intention specification also represents in the form of matrix. Variation on representing service specification is introduced as the visualized model.

The overall results indicate that both I^{SY} and I^{GB}, proposed in this dissertation, equip the proposed MA to be potent enough to accomplish its missions with a wide array of target hosts within the acceptable users' satisfaction. For high available bandwidth, I^{GB} is suggested to be the suitable form of MA's intention component to enable a precision matchmaking results. On the other hand, for low available bandwidth, I^{SY} is recommended as a compact form of MA's intention component to accommodate the network congestion problem. Implementation of a sample application based on the reference architecture was demonstrate and proved to be satisfactory.

One caveat of the proposed abstract model lies in its scope which is restricted to pairwise matchmaking. It would thus be of interest to further the investigation on classification of service and intention specifications which will in turn enable a more practical matchmaking process. Moreover, enhancing scalability capability to an agent can be further studied with the help of imputation technique to fill in missing specified attributes.

References

[1] Lange, D. B. Mobile objects and mobile agents: the future of distributed computing? Proceedings of the 12th European Conference on Object-Oriented Programming. pp. 1-12, Brussels, Belgium, July 20-24, 1998.

[2] Mäkelä, M. Adaptability for seamless roaming using software agent. Proceedings of the XIII International Symposium on Services and Local Access). pp. 3:2-3:2, Stockholm, Sweden, June 18-23, 2000.

[3] Magedanz, T., Rothermel, K., and Krause, S. Intelligent agents: an emerging technology for next generation communication. Proceedings of the 15th Annual Joint Conference of the IEEE Computer Societies: Networking the Next Generation. San Francisco, U.S.A., March 24-28, 1996.

[4] Sanneck, H., Berger, M., and Bauer, B. Application of agent technology to next generation wireless/mobile networks. Proceedings of the Second World Wireless Research Forum. Helsinki, Finland, May 10-11, 2001.

[5] Wooldridge, M. Practical Reasoning with Procedural Knowledge. Proceedings of the International Conference on Formal and Applied Practical Reasoning. pp. 663-678, Bonn, Germany, June 3-7, 1996.

[6] Rao, A. S. and Georgeff, M. An abstract architecture for rational agents. Proceedings of Knowledge Representation and Reasoning. pp. 439-449, Cambridge, MA, October 25-29, 1992.

[7] Rao, A. S. and Georgeff, M. BDI agents: from theory to practice. Proceedings of the First International Conference on Multi-Agent Systems (ICMAS). pp. 312-319, San Francisco, California, June 12-14, 1995.

[8] Braubach, L., Pokahr, A., Lamersdorf, W., and Moldt, D. Goal representation

for BDI agent systems. Proceedings of the Second International Workshop on Programming Multiagent Systems: Languages and Tools. pp. 9-20, New York, NY, July 20, 2004.

[9] Bernstein, A. and Klein, M. Discovering services: towards high-precision service retrieval. Proceedings of the CAISE 2002 Workshop on Web Services, E-Business, and the Semantic Web: Foundations, Models, Architecture, Engineering and Applications. pp. 260-276, Toronto, Canada, May 27-28, 2002.

[10] Wang, Y. and Stroulia, E. Semantic structure matching for assessing web-service similarity. Proceedings of the First International Conference on Service Oriented Computing. pp. 194-207, Trento, Italy, December 15-18, 2003.

[11] van Splunter, S., Wijngaards, N., Brazier, F., and Richards, D. Automated component-based configuration: promises and fallacies. Proceedings of the Fourth Symposium on Adaptive Agents and Multi-Agents System. pp. 130-135, Leeds, UK, March 29-31, 2004.

[12] Brazier, F. M.T., Overeinder, B. J., van Steen, M., and Wijngaards, N. J.E. Agent factory: generative migration of mobile agents in heterogeneous environments. Proceedings of the 2002 ACM Symposium on Applied Computing. pp. 101-106, Madrid, Spain, March 11-14, 2002.

[13] Morel, B. and Alexander, P. Automating component adaptation for reuse. Proceedings of the 18th IEEE International Conference on Automated Software Engineering. pp. 142-151, Montreal, Canada, October 6-10, 2003.

[14] Benjamin, V. R. et al. IBROW3: an intelligent brokering service for knowledge-component reuse on the world-wide web. Proceedings of the Workshop on Applications of Ontologies and Problem-Solving Methods. pp. 25-30, Brighton, UK, August 24-25, 1998.

[15] van der Aalst, W. M. P., Don't go with the flow: Web services composition standard exposed. IEEE Intelligent Systems. pp. 72-85, vol.18 No.1, Jan/Feb 2003.

[16] Gottschalk, K., Graham, S., Kreger, H., and Snell, J. Introduction to Web services architecture. IBM Systems Journal. vol.41 No.2, pp. 170-177, 2002.

[17] Markopoulos, P., Eggen, B., Aarts, E. H. L., and Crowley, J. L. Ambient intelligence: Proceedings of Second European Symposium, EUSAI 2004. Lecture Notes in Computer Science, Springer, vol.3295, 2004.

[18] Jini Network Technology. Sun Microsystems, Inc. http://www.sun.com/software/-jini/. March 2007.

[19] Richard, G. G. Service advertisement and discovery: enabling universal device cooperation. IEEE Internet Computing. vol.4 No.5, pp. 18-26, 2000.

[20] Catalog of Corba/IIOP specifications. http://www.omg.org/technology/documents/-corba_spec_catalog.htm. / Object Management Group. 2004.

[21] UDDI technical paper. http://www.uddi.org/pubs/Iru_UDDI_Technical_White-_Paper.pdf. The OASIS standards consortium, September, 2000.

[22] Web Services Description Language WSDL. http://www.w3.org/TR/wsdl, February, 2007.

[23] Haav, H.-M. and Lubi, T.-L. A survey of concept-based information retrieval tools on the web. Advances in Databases and Information Systems: Proceedings of 5th East-European Conference. vol.2 pp. 29-41, Vilnius, Lithuania, September 25-28, 2001.

[24] DAML+OIL website. http://www.daml.org/language. 2004.

[25] Abowd, G. D. et al. Towards a better understanding of context and context-awareness. Proceedings of the First International Symposium on Handheld

and Ubiquitous Computing. pp. 304-307, Karlsruhe, Germany, September 27-29, 1999.

[26] Mendoza, D. R. Using ontologies in context-aware services platforms. Masters thesis. University of Twente, November 2003.

[27] Broens, T. Context-aware, ontology based, semantic service discovery. Masters thesis. University of Twente, July 2004.

[28] Doulkeridis, C., Valavanis E., and Vazirgiannis, M. Towards a context-aware service directory. Proceedings of the 4th International Workshop on Technologies for E-Services. vol.2819 pp. 54-65, Berlin, Germany, September 8, 2003.

[29] Pappas, A., Hailes, S., and Giaffreda, R. A design model for context-aware services based on primitive contexts. Proceedings of the First International Workshop on Advanced Context Modeling, Reasoning And Management (UBICOMP). Nottingham, UK, September 7, 2004.

[30] Zaremski, A. M. and Wing, J. M. Specification matching of software components. ACM Transaction on Software Engineering Methodology. vol.6 No.4, pp. 333-369, 1997.

[31] Sycara, K., Klusch, M., Widoff S., and Lu, J.G. Dynamic service matchmaking among agents in open information environments. Journal ACM SIGMOD Record. vol.28 No.1, pp. 47-53, 1999.

[32] Sycara, K., Widoff, S., Klusch, M., and Lu, J. LARKS: dynamic matchmaking among heterogeneous software agents in cyberspace. Autonomous Agents and Multi-Agent Systems, vol.5 No.3, pp. 173-203, September 2002.

[33] Nakkrasae, S., Sophatsathit P., and Edwards, Jr., W. R. Fuzzy subtractive clustering based indexing approach for software components classification.

International Journal of Computer & Information Science. vol.5 No.1, pp. 63-72, March 2004.

[34] Broens, T., Pokraev, S., van Sinderen, M., Dockhorn Costa, P., and Koolwaaij, J. Context-aware, ontology-based, service discovery. Proceedings of the Second European Symposium on Ambient Intelligence. pp. 72-83, Eindhoven, The Netherlands, November 8-11, 2004.

[35] Li, L. and Horrocks, I. A software framework for matchmaking based on semantic web technology. Proceedings of the 12th International World Wide Web Conference, pp. 331-339, Budapest, Hungary, May 20-24, 2003.

[36] Brandt, R. and Reiser, H. Dynamic adaptation of mobile agents in heterogeneous environments. Journal of Communications and Networks. vol.3 No.4, pp. 307-316, December 2001.

[37] Brazier, F. M.T., van Splunter, S., and Wijngaards, N. J.E. Knowledge-based design and evolutionary design: adversaries or compatriots? Proceedings of the Intelligent Agents in Design Workshop. pp. 35-38, Cambridge, UK, July 15-17, 2002.

[38] Kashyap, V. and Sheth, A. Semantic and schematic similarities between database objects:a context-based approach. The International Journal on Very Large Data Bases. vol.5 No.4, pp. 276-304, 1996.

[39] Dickinson, I. and Wooldridge, M. An initial response to the OAS'03 challenge problem. Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent System, Workshops on Ontologies in Agent System. pp. 63-70, Melbourne, Australia, July 15, 2003.

[40] Jennings, N. R. and Wooldridge, M. Applications of intelligent agents. Agent

Technology: Foundations, Applications, and Market. pp. 3-28, 1st Edition, Springer-Verlag, 1998.

[41] Gomez-Perez, A. and Lozano, A. Impact of software components characteristics above decision-making factors. Proceedings of the International Workshop on Component-Based Software Engineering. pp. 781-788, Florida U.S.A., May 19-20, 2002.

[42] Fishbein, M. and Azjen I. Belief, attitude, intention, and behavior: an introduction to theory and research. 1st Edition, Addison-Wesley Publishing, 1975.

[43] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning internal representations by error propagation. Parallel distributed processing: explorations in the microstructure of cognition. 1st Edition, Bradford Books/MIT Press, 1986.

[44] Luan, X.C., Peng, Y., and Finin, T. Quantitative sgent service matching. Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence. pp. 334-340, Beijing, China, September 20-24, 2004.

[45] Martin, C.J. Introduction to Languages and the Theory of Computation. 2nd Edition, McGraw-Hill Higher Education, 1997.

Biography

Name: Miss Dussadee PRASERTTITIPONG.
Date of Birth: 23^{rd} April, 1973.
Educations:

- Ph.D. Candidate in Computer Science, Department of Mathematics, Chulalongkorn University, Thailand, (June 2002 - April 2007).
- Visiting Ph.D. researcher in School of Computer Science and Software Engineering at Monash University, Melbourne, Australia, (November 2004 - April 2005).
- M.Sc. Program in Computer Science, Chiang Mai University, Thailand, (June 1995 - September 1999).
- B.Sc. Program in Computer Science, Chiang Mai University, Thailand, (June 1991 - March 1995).

Publication papers:

- Praserttitipong D. and Sophatsathit P. A Framework for Deterministic Intention Specification of an Agent toward an Incomplete Declared Intention. *Proceedings of the* 6^{th} *IEEE International Conference on Computer and Information Science*, Melbourne, Australia, 2007. Accepted and scheduled to appear in July 2007.

Work:

- Lecturer, Computer Science Department, Faculty of Science, Chiang Mai University, (October 1998 - current).
- Official Staff, Faculty of Science, Chiang Mai University, (November 1998 - September 2002).

Scholarship: Office of the Commission on Higher Education, Thailand. (June 2002 - May 2005).