早稲田大学審査学位論文
博士（人間科学）

# An Adaptively and Dynamically Reconfigurable Service Model in Ubiquitous Cloud Environments

# ユビキタスクラウド環境における状況に応じて動的再構築可能なサービスモデル

２０１２年７月

早稲田大学大学院　人間科学研究科
朱　依水
ZHU, Yishui

研究指導教員：　金　群　教授

# Abstract

Services are expected to be a promising way for people to use information and computing resources in the emerging ubiquitous network society and cloud computing environments. In this study, we propose a metaphoric concept called Flowable Service, which is abstract as adaptively and dynamically reconfigurable service model. It is defined as a logical stream that organizes and provides circumjacent services in such a way that they are perceived by individuals as those naturally embedded in their surrounding environments.

We present our view on how some of these problems, such as flexibility, portability and interoperability of services, can be solved using the flowable services to make seamless integration of diverse services in an intuitive “flowable” way possible, thus achieving maximum satisfaction of both service providers and consumers while decreasing the delivery cost of services. Recognizing the importance of the awareness of each individual’s context for smooth and accurate provision of services, we further propose a human-centric framework for context-aware flowable services that harnesses users’ contexts to enable pro-active support of human activities in a flexible and natural way.

On the other hand, we propose an Adaptively Emerging Mechanism (AEM) to help people reduce this selection burden by an interdisciplinary approach. This AEM is applied and integrated into the Flowable Service Model (FSM) which has been proposed and developed in our previous study. We consider the user’s feedback information is a pivotal factor for AEM, which contains the user’s satisfaction degree after using the services. At the same time, we assume that these factors, such as the service cost, matching precision, responding time, personal and the social context information, etc., are essential parts of the optimizing process for the selection of ambient services. By analyzing the result of AEM simulation, we can expect: (1) substantially improving the selection process for LOW feedback users; (2) no negative impact on the selection process for MEDIUM or HIGH feedback users; and (3) enhancing the rationality for services selection.

Owing to context-awareness approach and AEM algorithm, the framework is able to find and integrate the circumjacent services that are considered to be needed by each individual, and create an ambient service environment tailored to his/her particular needs, proclivities and characteristics.

**Keywords** - flowable services; organizing human context; seamless integration of services; context-aware

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **FSM** | Flowable Service Model |
| **AEM** | Adaptively Emerging Mechanism |
| **HIP** | Human Information Processing |

# Chapter 1 Introduction

Cloud Computing has become conspicuous research topic from 2006. Our study goes with this flow from 2008. During these years Cloud Computing has changed the world by degrees. The power is obvious to people. At the meantime the problems concealed in the power perceived by people gradually. We focus on some problems to which we should find solution above all depending on the importance. Our contributions are from multi approaches in different research fields.

## 1.1 Background

In spite of the emergence of new services and technical solutions to benefit from Services Computing, some technology and non-technology issues always remain a big concern. Some of them are:

*how to make much more profits in service-based commerce;*

*how to reflect the cost of service providers by a fair price system;*

*how to (re-)configure resources to enlarge services, how to customize services to improve the customers' satisfaction;*

*how to increase the efficiency in different application fields, and others (e.g. open problems mentioned in (Benatallah, Perrin, Rabhi, & Godart, 2006)).*

To challenge those, we should deliberate on a new kind of interdisciplinary and integrated approach. Our study aims to create a flowable service model for seamless integration of services to seek maximum satisfaction of both service providers and consumers while decreasing the delivery cost of services. The model emphasizes on capturing users' needs to best provide required services and improve their satisfaction. We introduce basic measures for our proposed model. In addition, we conceive initial ideas on mediation of flow services which are inspired by Human Information Processing (HIP).

## 1.2 Purpose of This Study

Flowable Services Model (FSM) is seamlessly integrate the services to seek maximum satisfaction of both service providers and consumers, while decreasing the delivery cost of services in open cloud environments. The model emphasizes on capturing users' needs to best provide required services and improve their satisfaction with harnessing user contexts. We address the issues to harness user contexts, described as follows.

*How cloud-related problems are solved with contexts*

In (Abowd et al., 1999), researchers provide a widely accepted definition of context:

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between the user and application, including the user and applications selves.

In our study, user context information including three factors: human, nature and culture. In a certain situation, these context factors act as different weighting. The more contexts information which can be captured, the more accurate result can be predicted.

*How contexts in today's services are captured and applied*

RFID tag and sensor technologies are applied to acquire some physical information such as user moving. But how to acquire user's culture context in real situation is a key problem left unsolved.

## 1.3 Dissertation Organization

This paper is organized as follows. Chapter 2 describes related works concerning the context approach and a variety of related models. Chapter 3 explains the definition of our proposed FSM and its user-oriented architecture on user-oriented backgrounds and metaphors of human thinking for FSM. Chapter 4 gives a context aware system architecture Chapter 5 introduces AEM algorithm, which is applied into FSM-based adaptively and dynamically reconfigurable service system, including a designed simulation and application scenario. Finally, in Chapter 6, we summarize this study, and discuss future work.

# Chapter 2 Related Work

## 2.1 Service and Ubiquitous Computing

### *Traditional Concepts in Relation to Service*

We introduce our study from the explaining of the word ***service***. Referred to the Oxford Dictionary and Longman Dictionary, the old English (denoting religious devotion or a form of liturgy), from Old French *servise* or Latin *servitium* means ‘slavery’, from *servus* means ‘slave’. The early sense of the verb (mid 19th century) was ‘be of service to, provide with a service’. Service means “a system supplying a public need such as transport, communications, or utilities such as electricity”, “a particular type of help or work that is provided by a business to customers, but not one that involves producing goods.” (service, n.d.a; service, n.d.b)

In ICT field, ***service*** appeared from 1990’s, as an intermediate component named middleware, which lives inside a server and ties one or more connectors to exactly one Engine. The Service element is rarely customized by users, as the default implementation is simple and sufficient: Service interface. A "Service" is a collection of one or more "Connectors" that share a single "Container" application service provider ASP Web service

***Services Computing*** is a new computing paradigm that marries a variety of theories and technologies of computer science, cognitive science, communication science, networking, society science and other disciplines. A special session at the 3rd International Conference on Internet Computing held in Las Vegas, USA, in June 2002, first used “Web Services Computing” to label a combination of business services with computing technologies (IC2002, n.d.). Soon afterwards, in November 2003, IEEE

Computer Society (CS) officially approved Technical Committee for Services Computing (Technical Committee for Services Computing, n.d.).

Today, Services Computing is widely employed and being developed in academia, education, health, and especially in business. Google[1], Amazon.com[2], Salesforce.com[3], and ZOHO[4] are only several well-known examples providing widely-used Web services. Plenty of services are provided to the end user spanning to a variety of cultures, resources and technical solutions in a wide variety of domains. Their combination produces rich user experience and brings benefits of new value-added services. As the trend to proliferation of services remains strong, the service world in the future is likely to be one of great richness and diversity.

***Ubiquitous computing***

In (G. Bell & Dourish, 2007; Weiser, 1991), ubiquitous computing provides an environment in which we can use service anywhere and anytime. According to the development of context aware technology and sensor devices, users accept ubiquitous environment. But more and more high level users' needs generated.

***Services Provision Studies***

There are a number of platforms, architectures and applications that provide Services Computing. The research work in (Lopez, Innocenti, & Busquets, 2008) addresses the problem of assigning services to ambulances. An auction mechanism, which is a well-known market mechanism to distribute tasks to different agents, is applied in this system based on trust to select an ambulance for emergent patient transportation. The trust model is able to provide a value about the driver's abilities. And the winner

[1] http://www.google.com/
[2] http://www.amazon.com/
[3] http://www.salesforce.com/?ir=0
[4] http://www.zoho.com

determination algorithm uses the values of the modified bids to assign a task to an ambulance team. This research implements a multi-agent system that provides a computational framework for coordination. In our study, the multi-agent approach is used not only for emergency but also for usual situation.

In (Overbeek, Klievink, & Janssen, 2009), a conceptual modeling language called ORM (object-role modeling) is created, which is used to describe the ontology for the integrated service delivery. They use event-driven SOA to decentralize intelligence to orchestrate services across a network of public agencies, without compromising the individual agencies' autonomy. This system aims to serve for the government, which is of a highly fragmented structure, with many local and national agencies performing their own specific part of a more general task. So it is possible to use events to coordinate demand-driven services across a network of organizations by developing an event-driven SOA.

SemaPlorer described in (Schenk, Saathoff, Staab, & Scherp, 2009) is an easy application for real-time use. It connects with a large Flickr data set converted to RDF. Amazon's Elastic Computing Cloud (EC2) and Simple Storage Service are applied as its basic infrastructure. Different kinds of semantic data, including the location, time, person, tags, complexity of queries, achievements and experiences, are combined to provide blended browsing and querying about areas of interest in this system. Comparing with (Overbeek et al., 2009) and (Schenk et al., 2009), our research aims to provide a flowable service based on both static RDF information and real-time context information. Every change of contexts would be an event to drive the system update providing a flow of services.

A national semantic publishing network and portal for health information is introduced in (Suominen, Hyvönen, Viljanen, & Hukka, 2009). The system consists of a centralized content infrastructure, a distributed semantic content creation channel, and an intelligent semantic portal aggregating and presenting the contents from intuitive and health promoting end-user perspectives. Research work in (Delgado, Ros, & Amparo Vila, 2009) shows a Tagged World to identify a user’s behavior and to produce alarms for dangerous situations. And Allia in (Ratsimor, Chakraborty, Joshi, Finin, & Yesha, 2004) uses peer-to-peer caching based and policy-driven agent-service discovery framework which facilitates cross-platform service discovery in ad-hoc environments. Our study can provide result to user as a recommendation which cannot be offered by (Suominen et al., 2009). Our system is also aware of the culture context information, which is not mentioned in (Delgado et al., 2009). At the same time, the service discovery method is different from (Ratsimor et al., 2004). The active and passive discoveries can make the service sharing process more smoothly.

## 2.2 Cloud Computing

Since the term "Cloud Computing" was coined in 2006, there have been many cloud-related controversies that have been of concern (G. Bell, Hey, & Szalay, 2009; Nelson, 2009). We classified them into the following three groups:

### *2.2.1 Cloudy Concept vs. Attractive Marketing*

A majority of the Cloud Computing definitions (I. Foster, Yong Zhao, Raicu, & Lu, 2008), including on-demand, pay-by-use model, virtualized and dynamically-scalable, have the characteristics of Cloud Computing. However, all of the definitions are descriptive and can change on a daily basis. Although there are no clear examples of successful scientific applications of clouds (G. Bell et al., 2009), business long-term marketing for large profits, cost-effective storage for data sharing, and high-speed computing for system maintenance attract many individuals with good business sense who incorporate these technologies into the field.

### *2.2.2 Novel Diverse Computing Types vs. Old Internet Techniques*

Some of practitioners and researchers believe that Cloud Computing is nothing new because it uses existing Internet techniques for concepts, approaches, and practices. However, in others' opinions, everything that pertains to Cloud Computing is new, because Cloud Computing applies novel diverse computing types, such as SaaS (Software as a Service), PaaS (Platform as a Service), utility computing, web services in the cloud, MSP (Managed Service Providers), service commerce platforms, and Internet integration. Furthermore, these computing types are based on new payments, new deployments and new updated/maintained mechanisms.

### *2.2.3 Single Open Cloud Manifesto vs. Multiple Business Giants*

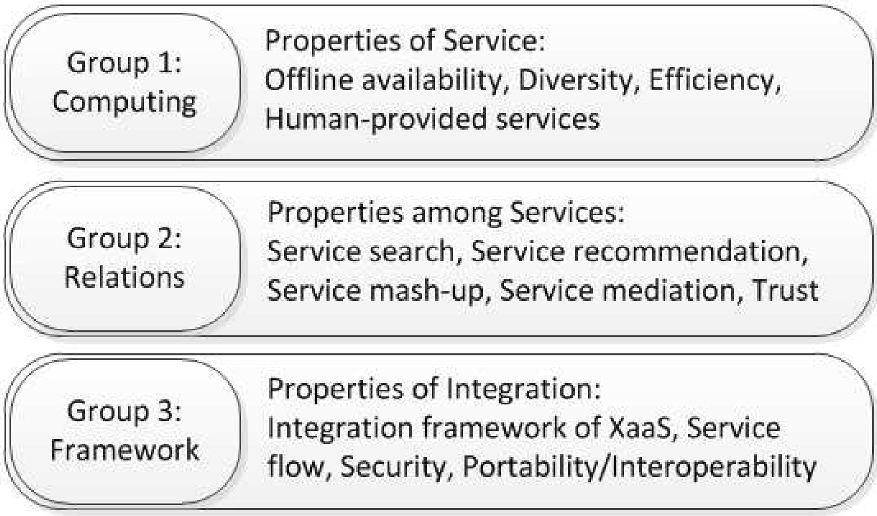
"Open Cloud" (or "Blue Skies")—the third of three types of Cloud Computing scenarios—is presented in (Nelson, 2009), where open standards, open interfaces, and open-source software are used to enable thousands of different organizations to link their infrastructures into a single, global cloud. The open cloud manifesto mentioned in (B. Worthen, March 2009) seems to not provide fair opportunities for all of the potential business participants. For developing Cloud Computing, the issues of protocol, security, and openness must be addressed. Commercial competitions also must be considered.

Although this young multidiscipline developed rapidly over recent years, many gaps between theory and practice have been exposed. On the basis of our survey, we summarize the open issues of Services Computing and classify them into three groups (see Fig. 2-1).

Group 1 is related to the Services Computing process.

Group 2 describes problems that involve relationships among services.

Group 3 concerns the whole environment of Services Computing, such as integration or security.



Group 1: Computing

Properties of Service:
Offline availability, Diversity, Efficiency, Human-provided services

Group 2: Relations

Properties among Services:
Service search, Service recommendation, Service mash-up, Service mediation, Trust

Group 3: Framework

Properties of Integration:
Integration framework of XaaS, Service flow, Security, Portability/Interoperability

**Figure 2-1 Classified Services Computing Issues**

As shown in Figure 2-1, Group 1 is a large set of issues that consider services such as offline availability and resource diversity. We believe that issues that are related to human needs deserve more consideration within a Services Computing paradigm (therefore, the human is an essential element of the flowable service model that we propose in this study). One of the attempts to reach this goal is in the Human-Provided Services (HPS) framework, which allows people to manage their interactions and to seamlessly integrate their capabilities into Web-scale workflows as services (Schall, Hong-Linh Truong, & Dustdar, 2008). To study these issues with greater depth, agent-based modeling approaches applied to the study of complex adaptive systems can be considered. Such approaches are used to study social systems, such as how local interactions among agents generate emergent social structures and patterns of behavior that are larger and more global (Sansores & Pavón, 2006).

Because of the growing number and variety of services, their efficient discovery from a large-scale collection of services (Group 2) has become especially important for the Services Computing field. Usually, for service provision (i.e., retrieval), information retrieval methods that are applied to proprietary XML formats (service descriptions in the registry database) are used. The UDDI (Universal Description, Discovery and Integration) data entries, WSDL (Web Services Description Language) definitions of services and structural features in the service descriptions are often considered (Lee, Lee, & Kim, 2008). However, validating, governing, or securing Web services cannot be accomplished either with UDDI Business Registries or Web-based search engines for service discovery (Al-Masri & Mahmoud, 2008). (Grigori, Corrales, & Bouzeghoub, 2008) pointed out that such approaches are insufficient for discovering relevant components, and they proposed service discovery that was based on the specification of service behavior. Another important issue in this group, which partially covers the discovery, is service composition. Composing Web services has gained considerable attention in Service-Oriented Computing and includes, for example, the dynamic discovery, interaction and coordination of agent-based semantic Web services. (Tang, Zheng, & Jin, 2008) considered Web services as “active entities (service agents) that are distributed over the internet” and proposed a multi-agent negotiation approach to fix the ineffectiveness that was caused by conventional approaches. Other studies focus on service discovery to facilitate web service outsourcing (Crasso, Mateos, Zunino, & Campo, ), and some studies focus on service composition (in an automated way), using adductive event calculus (Kirci Ozorhan, Kuban, & Cicekli, 2010).

Software as a Service (SaaS) and Platform as a Service (PaaS) have been gaining momentum in recent years and have extensions such as XaaS. These services are delivered over the Internet and are charged on a per-use basis; the result is a software application or resource provision, in essence. Because it includes business data and logics, which are usually required to be integrated with other applications/resources that are deployed by a SaaS subscriber, integration (as in Group 3) becomes one of the important issues in most SaaS adoptions. The key functional and non-functional SaaS integration requirements from an industry practitioner's point of view are provided in (Sun, Zhang, Chen, Zhang, & Liang, 2007).

The proposed flowable service model in our study tackles issues and problems from all of these three groups, with a specific focus on issues of service mediation and flow, portability and interoperability, and considering humans to be an essential element of the Services Computing.

## 2.3 Other Related Approaches

### *2.3.1 Agent Tools*

**Table 2-1 Abstract of Agent Tools**

| **Name** | **Jess**[5] | **JADE**[6] | **Voyager**[7] | **agentTool III**[8] | **JACK**[9] |
|---|---|---|---|---|---|
| **Language** | Java | Java | - | Java | Java |
| **Open Source** | N | Y | N | N | N |
| **Multi-Agents** | - | Y | - | Y | - |
| **Services Support** | - | Y | - | - | - |
| **Mobile support** | - | Y | Y | - | - |
| **cross-platform** | Y | Y | - | Y | Y |
| **Price** | no cost for academic use | Free | Not Free | - | Not Free |
| **Others** | small, light, and one of the fastest rule engines | Agent | - | - | Autonomous systems development platform.<br>a proven track record |

We use the JADE tool to simulate this model. Comparing with other tools, only JADE has the Services Support gateway function.

### *2.3.2 Context-Aware Concepts*

Context is used to challenge various issues in ubiquitous environments, such as network selection (Nguyen-Vuong, Agoulmine, & Ghamri-Doudane, 2008), service composition (Bastida, Nieto, & Tola, 2008), access control (Kulkarni & Tripathi, 2008),

[5] http://www.jessrules.com/
[6] http://jade.tilab.com/
[7] http://www.recursionsw.com/
[8] http://agenttool.cis.ksu.edu/
[9] http://aosgrp.com/index.htm

and service integration (Hwang, Liang, & Wang, 2007). Context provides flexibility and adaptation to services. For example, context-awareness in role-based access control models, as proposed in (Kulkarni & Tripathi, 2008), facilitates dynamism of context-based constraint specifications, and personalized permissions for role members. Knowledge of user context can ensure cost-effective messaging in heterogeneous wireless networks (Hwang et al., 2007). Furthermore, social context information has been used to recommend similar users for social network services (De Meo, Nocera, Terracina, & Ursino, 2011).

Currently, modeling and implementing a single situational/contextual model that is shared by all users or building models for every individual user are commonly used approaches; these approaches are either costly or entail a number of problems, such as complexity and inconsistency. To avoid these problems, (Feng, Teng, & Tan, 2009) conceived a shared situation awareness model that supplies each user with agents for focused and customized decision support according to the user's context. (Coutaz, Crowley, Dobson, & Garlan, 2005) proposed a conceptual framework for context-aware computing, including ontological and architectural foundations that structure the system adaptation process in a unified way. By their proposed framework, the context is modeled as a directed state graph, where the nodes denote contexts and the edges denote the conditions for changes in contexts. The runtime infrastructure is responsible for the acquisition of situation and context, the instantiation of entities, roles and relations of the context model, and adaptation between the infrastructure and the application. Many researchers attempt to formalize context models with ontology. For example, (T. Gu, Pung, & Zhang, 2005)proposed such a model, which was based on the OWL inside the SOCAM (Service-Oriented Context-Aware Middleware) architecture, to efficiently support the acquisition, discovery and interpretation of various contexts for building context-aware services.

In our approach, we also harness context to realize accurate and intuitive service provision with a flow of services. We provide both abstraction and implementation layers by taking a large number of services in the flow into account. However, to achieve such a service flux, we must consider context as a “part of a process of interacting with an ever-changing environment that is composed of reconfigurable, migratory, distributed and multi-scale resources,” because it is “not simply the state of a predefined environment with a fixed set of interaction resources” (Coutaz et al., 2005). Context requires a thorough consideration of its organization and dynamics in a holistic and context-as-process way. This goal is extremely challenging, and we attempt to approximate it—if not in this specific work, in future achievements of the same project—in the context of flowable services.

Many context models have been proposed and developed, according to (Hoareau & Satoh, 2009). For example, there is a key-value model, a markup model, an object-oriented model, a logic-based model, and ontology and situation logic. Because location is one of the most typical contextual pieces of information, the location context involves special models: geometric models, symbolic models and hybrid models. In our study, user context information, such as human context, nature context and culture context, is represented by different context models. Context modeling can be considered to be an active database system. We also consider culture context to be an active database system for context matching processing. Because traditional database management systems are passive, all of the operations are issued by users. Our study requires context awareness, which automatically performs some services in response to certain changes in the real world, once the appropriate conditions are satisfied.

A context-awareness module was introduced in (Wohltorf, Cissee, & Rieger, 2005). It combines three sub-modules: the personalization module, the location-based services module, and the device and network independence module. This work introduced an agent-based Serviceware Framework to assist service providers in developing innovative services. Our FSM is also an agent-based system. However, the framework not only assists the service providers' side but also supports the users' side. Research work performed in (Jiang, Hu, & Wang, 2010) used a multi-agent system for providing mobile services with a competitive and collaborative mechanism. In our research, we use every type of service and competitive or collaborative mechanism to ensure that services are delivered as a flow.

To adapt the multimedia context-rich application into Services Computing, a distributed information repository for autonomic context-aware MANETs was developed in (Macedo, Dos Santos, Nogueira, & Pujolle, 2009). The middleware instantiates an InP (Information Plane) over MANETs, which was adapted to the characteristics of the wireless medium and the resource restrictions of mobile devices; these changes made it possible to compose more complex services by incorporating context awareness. The RuleManager of (Bonino da Silva Santos,Luiz Olavo, Vink, & van Wijnen, 2007) is externally accessed via the IManageRule interface and provides social rules. When a third party wants to register a rule, it sends, to the RuleManager, the rule that is responsible for parsing, validating and storing the incoming rule, and the new rule can be saved into the social rule DB.

### *2.3.3 Services Selection*

In (Mobedpour & Ding, 2011), they introduced the Quality of Service (QoS)-based web service selection systems, for which inexperienced end users are not the focal point of the design. Most of the systems assume that users could formulate their QoS requirements easily and accurately by using the provided query languages. To emphasize more on the user-centered design of the service selection system, their study considered a more expressive and flexible way for non-expert users to define their QoS queries, together with the user support on formulating queries and understanding services in the registry. Their work greatly enhanced selection model.

We share the same concern stated in (Yu & Bouguettaya, 2012), that is, a key issue in Services Computing is selecting service providers with the best user desired quality. As multiple service providers may compete to offer the same functionality with different quality of service, the existing service selection approaches mostly rely on computing a predefined objective function. When multiple quality criteria are considered, users are required to express their preference over different quality attributes as numeric weights. They developed the so-called service skyline, and a set of service skyline computation techniques that return a set of most interested service providers. These providers are non-dominant in all user interested quality attributes. Our method is different from this study. We import the user’s feedback value as an important factor to regulate the output service flow.

(Shtykh & Jin, 2011) discussed a human-centric integrated approach for web information search and sharing incorporating the important user-centric elements, namely a user’s individual context and ‘social’ factor, which are realized with collaborative contributions and co-evaluations, into web information search.

The reputation attribute of QoS is investigated and considered in (Wang, Sun, Zou, & Yang, 2011). It is very important for users to obtain reliable services in service selection. Their reputation measure has three phases (i.e. feedback checking, feedback adjustment and malicious feedback detection) to enhance the accuracy. In this study, the feedback ratings are adjusted with different user feedback preferences by calculating the feedback similarity. And malicious feedback ratings are detected by adopting cumulative sum method. Their user survey focuses on feedback ratings between different users. In our study, we consider the distribution of feedback as an important factor.

(San-Yih Hwang, Ee-Peng Lim, Chien-Hsiang Lee, & Cheng-Hung Chen, 2008) tried to deal with the dynamic web service selection problem. It aims to invoke web services at runtime so as to successfully orchestrate the services. They observed that both the composite and constituent services often constrain the sequences of invoking their operations. As a result, they assigned each state of execution an aggregated reliability to measure the probability that the given state will lead to a successful execution in the context where each web service may fail with a certain probability. In our study, to orchestrate a composite web service, we propose two strategies to select web services that are likely to successfully complete the execution of a given sequence of operations.

Service-oriented systems invoke a number of available web services. Users need to specify different preferences and constraints, and service selection can be performed dynamically at runtime. (Ardagna & Pernici, 2007) introduces a new modeling approach to the web service selection problem that is particularly effective for large processes and when QoS constraints are severe. This model has three features: (1) the web service selection problem is formalized as a mixed integer linear programming problem, (2) the loops peeling is adopted in the optimization, and (3) the constraints posed by stateful web services are considered.

In (Skoutas, Sacharidis, Simitsis, & Sellis, 2010), the authors proposed a methodology to rank the relevant services for a given request, by introducing objective measures based on dominance relationships defined among the services. And they investigated the methods for clustering the relevant services in a way that reveals and reflects the different trade-offs between the matched parameters.

### *2.3.4 Services Composition*

In (El Hadad, Manouvrier, & Rukoz, 2010), the authors mentioned an important topic – services composition. It is capable to recursively construct composite services, which are provided by different organizations and offer diverse functionalities. However, the selection for each user's needs was left as a big challenge. They addressed the issue of selecting and composing services not only according to their functional requirements but also to their transactional properties and QoS characteristics. Their selection algorithm that satisfies the user's preferences, is expressed as the weights based on the QoS criteria and as the risk levels that define semantically the transactional requirements.

Another research group presented a fully decentralized service composition framework called SpiderNet in (Xiaohui Gu, & Nahrstedt, K., 2006). It provides the statistical multi-constrained QoS assurances and load balancing for service composition. One more research has been done based on the study of relationships of SOs' states and services in (Feng, Y., Cao, J., Sun, Y., Wu, W., Chen, C., & Ma, J., 2010). The researchers provide an algorithm to compose SOs' services, which can avoid generating the identified unqualified composite services.

### *2.3.5 Miscellaneous Selection Approaches*

In the research work presented in (Maamar et al., 2011), the discovery of web service is improved by social networks. When the web services are provided in or through social networks, they can identify their peers, *(1)* with which they'd like to work; *(2)* those that can replace them in the case of failure; *(3)* those that compete against them for selection. In the six steps for social web services development, they decomposed the profile into five categories: preconditions, inputs, outputs, effects, and QoS. And they established the degree of similarity (DS) which is split into three clusters between the web services. The three clusters are weak-similarity cluster, average-similarity cluster, and strong-similarity cluster. We refer to this social management mechanism as grouping services.

For data clustering, the clustering stability methods on model selection techniques have been investigated in (Shamir & Tishby, 2010). They focus on the behavior of clustering stability using the k-means clustering. They gave an exact characterization of the distribution to which suitably scaled measures of instability converge have been defined, based on a sample drawn from any distribution in a satisfying mild regularity conditions. They showed that the clustering stability does not 'break down' even for arbitrarily large samples, at least for the k-means framework. And they identified the factors which eventually determine the behavior of clustering stability. This research helps us to make a choice of appropriate services, especially make appropriate services grouping and service selection.

Work given in (Olvera-López, Carrasco-Ochoa, & Trinidad,José,Francisco,Martínez, 2010) is about prototype selection and (Rudoy & Zelnik-Manor, 2012) is concerning about view selection. Prototype selection is an important task for classifiers since through this selection process the time for classification or training could be reduced. The authors proposed a new fast prototype selection method for large datasets, based on clustering, which selects border prototypes and some interior prototypes in (Olvera-López et al., 2010). It reported that the experimental results showed the performance of the method and compared accuracy and runtimes against other prototype selection methods. They automated a process by evaluating the quality of a view, captured by every single camera, for which a human producer anually selects the best view in (Rudoy & Zelnik-Manor, 2012). They regard human actions as three-dimensional shapes induced by their silhouettes in the space-time volume. Resting on the features which are evaluated based on the features of the space-time shape, two view quality approaches have been proposed. Their experiments showed that the proposed view selections could provide intuitive results which match common conventions. In our study, we also consider human factors to regulate a selection process.

In (Ahn & Ramakrishna, 2010), the authors present an efficient diversity preserving selection (DPS) technique for multi-objective evolutionary algorithms (MEAs). It aims to preserve the diversity of non-dominated solutions to the problems with scaled objectives. Their core mechanism selects a group (of individuals) that is statistically furthest from the worst group, instead of just concentrating on the best individuals, as in the truncation selection. Their experiments demonstrated that DPS significantly improved the diversity of non-dominated solutions for badly-scaling problems, while at the same time it exhibits an acceptable proximity performance.

In (Q. Gu & Lago, 2011), the service-oriented software engineering (SOSE) methodologies have been proposed and practiced. Service-oriented computing can be expected to effectively deliver software services in a dynamic environment. Most of SOSE methodologies share common features but are presented for different purposes, ranging from project management to system modernization, and from business analysis to technical solutions development. It is very difficult for a company to decide which methodology would fit best for its specific needs. (Q. Gu & Lago, 2011) introduced a feature analysis approach and devised a framework for comparing the existing SOA methodologies.

Both of the research works given by (Pyshkin & Kuznetsov, 2010) and (Klyuev & Yokoyama, 2010) are about web searching. (Pyshkin & Kuznetsov, 2010) introduced approaches used by web search tools to interact with users. They examine interactive interfaces that use textual queries, tag-focused navigation, hyperlink navigation, visual features, etc. with respect to different kinds of information resources. The implementation of a visual-interface based on the concepts of query token network and the WordNet ontology has been described. The work presented in (Klyuev & Yokoyama, 2010) assumed that the top ranked pages returned are relevant to the user's query. They made the search process more convenient for users. They found the most important synonyms and hypernyms for the terms of the user query, utilizing Japanese WordNet. And by combining the aforementioned terms together, a new expanded query is then submitted to the search engine. This web searching process is easier for beginners.

# Chapter 3 Adaptively and Dynamically Reconfigurable Service Model

## 3.1 Basic ideas

### *3.1.1 Computational Consciousness*

In (James, 1984), all the consciousness of human has motility. It can express into some important forms: (1) emotion expression; (2) instinct; (3) intentional action. A user's consciousness can be reflected from a user's expression, action, etc. And also, personal consciousness has the change from a moment to the next moment continuously in time. Although the habit of user can be sensed by a sensor device as user’s movement records, the personal consciousness cannot be recognized only by the movement. Because the habit actions should reduce some required conscious attention then other actions.

Although we had considered predicting user’s consciousness from one's action and then providing suitable service flow before, because there are so many kinds of user’s actions and we can not recognize user’s personal consciousness only by the actions, so it is much more difficult for us to predict the results. All we can use are the three approaches: the user's expression changing; the user's instinctive behaviors and the user’s habit actions. It is unreasonable to provide services flow according to user’s stream of consciousness. But, it is reasonable to do according to a part of one’s stream of consciousness by those three approaches.

### *3.1.2 Human Consciousness and Human Behaviors*

Some research (Liaw, 2002) showed that there had a significantly positive correlation between students' computer attitudes and Web attitudes. It implied that computer attitudes and Web attitudes could provide concurrent validity to each other. (Thatcher, Wretschko, & Fridjhon, 2008) explored the theoretical and practical overlap between online procrastination, problematic Internet use, and flow on the Internet. (Maia & Cleeremans, 2005) emphasized the relevance of these models to understanding consciousness. Interestingly, the models we review have striking similarities to others directly aimed at implementing 'global workspace theory'. The top-down attention and consciousness can have opposing effects (Cohen & Dennett, 2011; Koch & Tsuchiya, 2007; Kouider, de Gardelle, Sackur, & Dupoux, 2010; McCreery, Kathleen Krach, Schrader, & Boone, 2012; Sánchez-Pi, Carbó, & Molina, 2012; Vandekerckhove & Panksepp, 2009). Also another studies focused on these vastly underestimated primary forms of consciousness which may be foundational for all forms of higher 'knowing consciousness' (Cohen & Dennett, 2011; Koch & Tsuchiya, 2007; Kouider et al., 2010; McCreery et al., 2012; Sánchez-Pi et al., 2012; Vandekerckhove & Panksepp, 2009). (Cohen & Dennett, 2011; Koch & Tsuchiya, 2007; Kouider et al., 2010; McCreery et al., 2012; Sánchez-Pi et al., 2012; Vandekerckhove & Panksepp, 2009) explained 'phenomenal' consciousness and 'access' consciousness. A 'perfect experiment' illustrates this point, highlighting the unbreachable boundaries of the scientific study of consciousness in (Cohen & Dennett, 2011; Koch & Tsuchiya, 2007; Kouider et al., 2010; McCreery et al., 2012; Sánchez-Pi et al., 2012; Vandekerckhove & Panksepp, 2009).

In game design process, the virtual self of player in a virtual environment is an important role. (Cohen & Dennett, 2011; Koch & Tsuchiya, 2007; Kouider et al., 2010; McCreery et al., 2012; Sánchez-Pi et al., 2012; Vandekerckhove & Panksepp, 2009) To better understand this relationship, in-world behavior was recorded and then analyzed using a behavioral assessment checklist. Results suggested a relationship between personality and behavior within the domain of agreeableness. In Knowledge-based systems (KBS) (Cohen & Dennett, 2011; Koch & Tsuchiya, 2007; Kouider et al., 2010; McCreery et al., 2012; Sánchez-Pi et al., 2012; Vandekerckhove & Panksepp, 2009), the architecture and representation formalisms are the groundwork of today's systems. They evaluate the context-aware information system from the user's point of view, and analyze knowledge derived from expertise behaviors in specific areas.

### *3.1.3 From Stream of Consciousness to Flowable Services*

We simply imagine that there is a model best satisfying human with one's needs. We call this as Flowable Service Model (FSM) (Zhu, Shtykh, & Jin, 2012). It provides flowable services to the users following with their stream of consciousness. It is known to all that Stream of Consciousness is a concept of the psychology, which William James, the American psychologist, used in the 1890s first. Human's consciousness is not be realized by arrangement of static segments in the mind, but a view of dynamic images about some ideas flow. Our interdisciplinary study started from reviewing this basic concept. We supposed FSM can make user feel more satisfaction. It sounds impossible but we surely believe it is possible. This paper focuses on finding the assurance of the psychological theories support. And we also consider the practicality of FSM by using Markup languages. It can be considered to represent the complex information for computational processing of human’s consciousness as a kind of user context information.

## 3.2 Flowable Service Model: Definition

A flowable service is a logical stream that organizes and provides circumjacent services in such a way that they are perceived by individuals as those naturally embedded in their surrounding environments. A flow of service is a metaphor for a subconsciously controlled navigation that guides the user through fulfillment of a flowable service process that fits the user's context and situation and runs smoothly with unbroken continuity in an unobtrusive and supportive way. An FSM puts a large emphasis on knowing the users for the purpose of intuitively providing the required services and, thus, increasing the level of satisfaction of the user.

The flowable service concept is our endeavor to challenge problems such as service heterogeneity and interoperability of services. It is an adaptively and dynamically reconfigurable service model. Its primary purpose is to realize seamless integration and provision of diverse services in an intuitive “flowable” way, thus achieving maximum satisfaction of both service providers and consumers while decreasing the delivery cost of services.

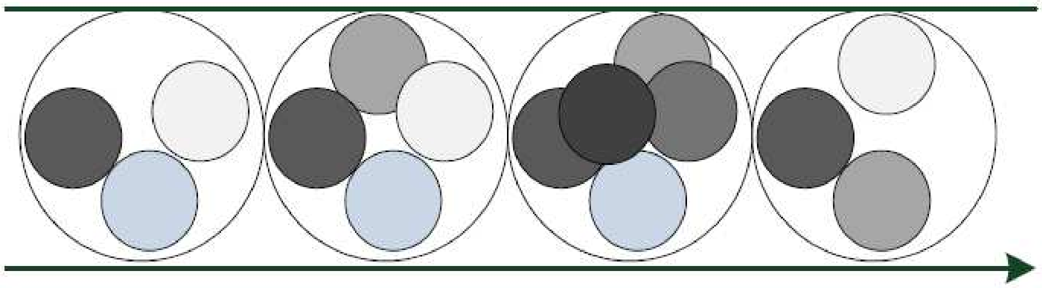


**Figure 3-1 Flow of Services**

Figure 3-1 shows a sketch of a flow of services. We can consider the conventional way of realizing Services Computing (through, for instance, composition approach) as one-service process, where services can be a combination of many services, but they are separated by time or distance, or any other physical barrier, and from the user's perspective are different (sometimes unrelated) services. In contrast, our proposed flow of service eliminates these boundaries happening continually and being rather "personal" for a user, as described above. In the figure, it is presented as a flow of services (large circles) combined from smaller services matching the user's need and context at the time.

To introduce the model, we first show its two important characteristics: flowability and constitutional similarity. Then, we discuss its fundamental components (Zhu, Shtykh, Jin, & Ma, 2009).

## 3.3 Flowable Service Model: Characteristics

### *3.3.1 Flowability*

*Flowability* is a characteristic of a service that effectively combines diverse services to maximally satisfy a user's needs and provides the perception, to the user, of using one service through all of the user's daily activities. This characteristic ensures that the user is provided with the services he/she needs regardless of the activity that the user is engaged in. Furthermore, it ensures hiding all of the connections of service discovery and combination, thus introducing a flow of services that is interpreted by the user as one integral service. Hence, although the flowability must be realized within certain technology solutions, this property is tightly related to how a user interprets, and perceives a set of services-therefore, a close consideration of the user is essential to realizing it.

### *3.3.2 Constitutional Similarity*

Another characteristic of the FSM is *constitutional similarity*. A constitutionally similar object is exactly or approximately similar to a part of itself, in its constitution. There are three constituents in the proposed model: service, user and mediator. These three elements compose one mesh through mediators, and the whole system is composed of a number of such meshes. No matter how many meshes it includes, the picture of the whole system does not change. The mesh consists of User, Mediator and Service, as shown in Figure 3-2 Therefore, we can say it is “roughly” constitutionally similar. In addition, because the proposed model conducts close consideration of a user, the resources that a service provides are not limited to the hardware and machine-provided resources and can include human-provided service resources as well.

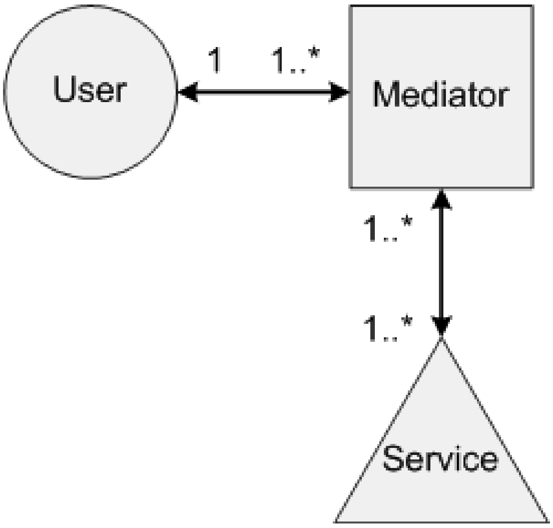


**Figure 3-2 Basic Components and Their Relationship**

## 3.4 Human Information Processing and User Factors

As shown in Figure 3-3, the conceptual composition contains Users and Services, a Context Capturer, a Service Controller and Organizer, and a Mediator Complex with Mediators and Meta-Mediators. The Mediator Complex usually consists of several mediators, which facilitate user-service interactions, and the Meta-Mediator's role is to coordinate and organize the Mediators in order to help them work efficiently. The complex solves such important tasks as interoperability and discovery of services. All of its activities can be sorted into two groups: *active* and *pro-active*. When a user makes a request to the system, the Mediator Complex, collaborating together with the Service Controller and Organizer and the Context Capturer, assembles the services that are pertinent to the user's needs. For further details, see (Y. Zhu et al., 2009).

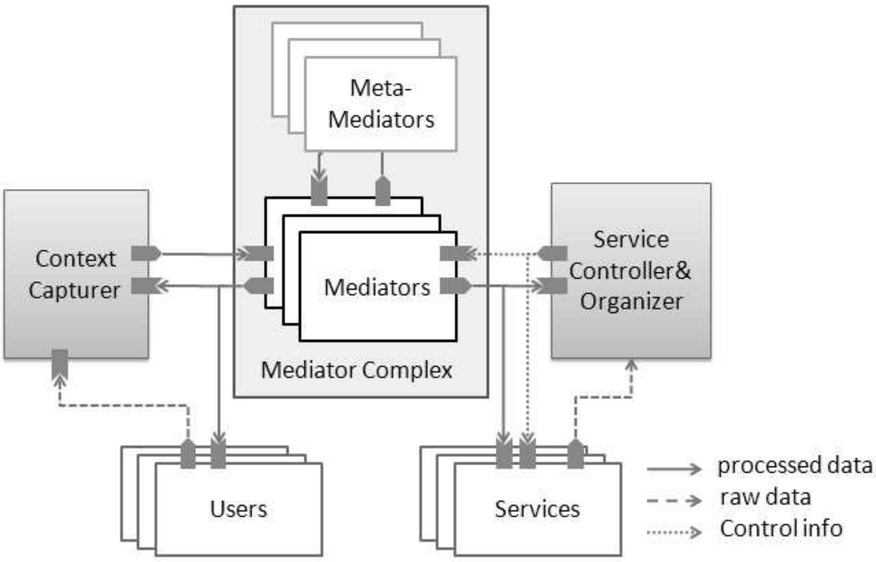


**Figure 3-3 Conceptual Composition of an FSM**

In our proposed FSM, the analogy of human information processing has been accounted for, as shown in Figure 3-4. On the user side (Figure 3-4 (a)), the Needs Capturer emulates Human Information Processing (HIP) for consciousness, as shown in

(Minsky, 2007). On the basis of this construct, a user's needs are inferred and provided to the mediator. Note that, here, we do not attempt to provide a complete solution for the needs inference but instead indicate an abstract method in order to show that needs must be extracted and passed to the mediator. To infer the user's needs, the method can be substituted by any other method that is capable of making the inferences.

Similarly, on the service side (Figure 3-4 (b)), there are many service resources, from which those that are suitable to the concrete user's needs must be chosen. The Service Controller & Organizer assists the mediator in this task; it monitors all of the available services and, when requested, provides only usable services. When a request with user-needs information reaches the Mediator Complex, five activities occur in the complex, as shown in Figure 3-4 (b):

*Receive* – Send a request to the Service Controller for services and obtain the available services set.

*Identify* – Check the trust value among members of this set. If the Identify process cannot satisfy the trust threshold, then replace it with another maximally similar service obtained through the Service Controller.

*Match* – Mash-up the chosen service's matching user's needs.

*Integrate* – Integrate matched services into one flow of services.

*Output* – Configure the environment for the specific service and provide it to the user.

In this way, we attempt to find the best match between a specific user's needs and the services that are available to the user; this task is an important emerging task of Services Computing that will become even more serious because of the fast proliferation of services that pervade a large number of environments.

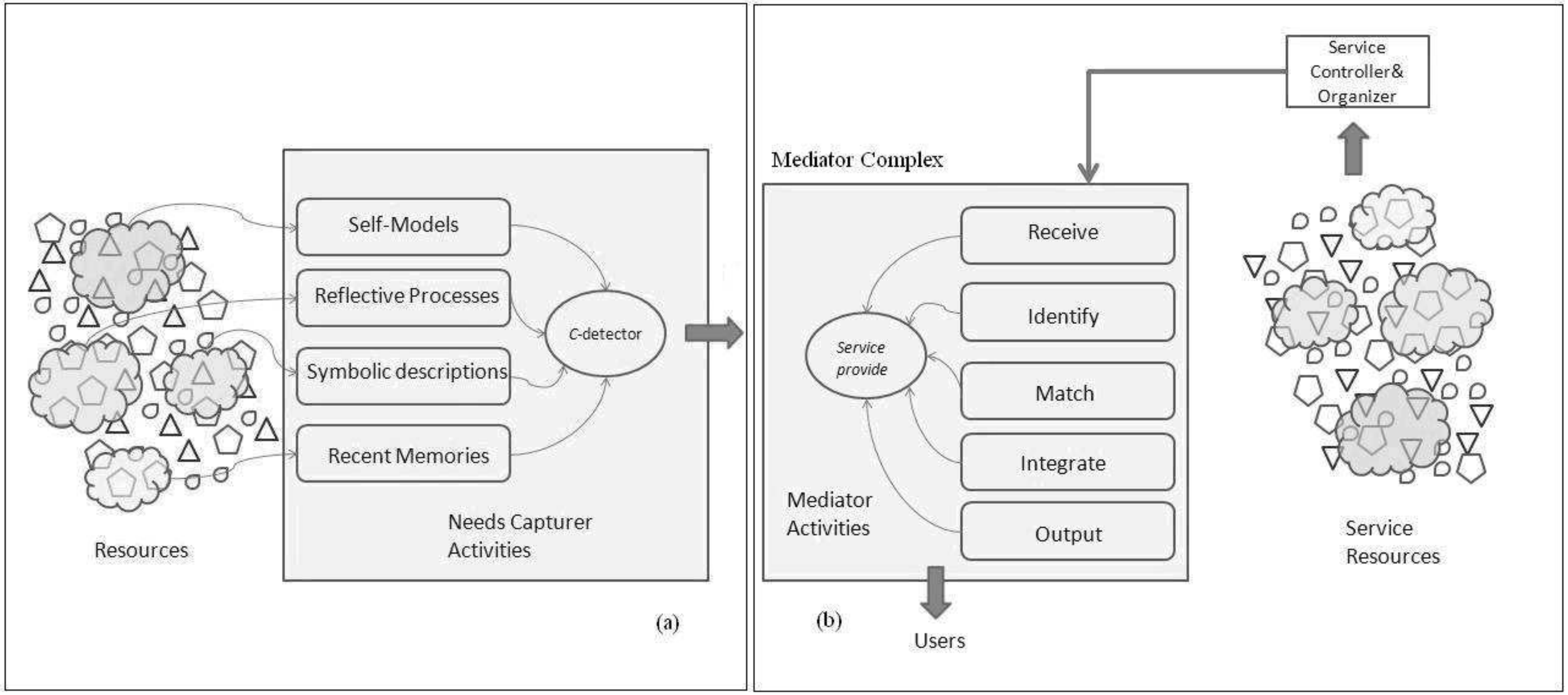


**Figure 3-4 Needs Inference (a) and Service Provision (b)**

Human information processing is a very complex process that integrates all of the structures of the human brain. According to James (2003), a user’s expressions and actions can reflect the user’s stream of consciousness. Consciousness is similar to a motor. Movements that are consequent to cerebro-mental changes can be enumerated as the following:

*(1) expression of emotion;*

*(2) instinctive or impulsive performances; and*

*(3) voluntary deeds.*

In this study, we consider a human’s **expression**, **action**, **behavior** and **habit** to be *User Factors*.

**Expression**: What shows on a human’s face, representing his/her feelings or emotions, such as a sweet smile or a sad expression.

**Action**: A human performance (or activity) that occurred naturally (rather than by the human’s intent).

**Behavior:** A human performance (or activity) that is produced by the human’s intent and/or with a certain purpose (rather than by nature). According to Behavior (n.d.), human behavior can be common, unusual, acceptable, or unacceptable. Furthermore, humans evaluate the acceptability of behavior using social norms and regulate behavior by means of social control.

**Habit:** Habits are routines of behavior that are repeated regularly. Habits tend to occur subconsciously, without directly thinking consciously about them (Habit, n.d.).

## 3.5 Metaphors of *Human Thinking* for FSM

From the varieties of user-centeredness aspects (Iivari & Iivari, 2006), our approach focuses on user interests, preferences and other personal-related information. According to the records of user activities, log mining and processing for what catches a user's interests or habits can be accomplished. The relevant activities can be divided into conscious activities and unconscious activities. Conscious activities can be recorded by a user's inputs or operations, according to specific, clear requirements. Unconscious activities can be recognized by some sensors, which could record physical information, eye gazing, or even the inflexions of the voice. User participation is an indispensable part of human-computer interaction processing. In our approach, user participation is one of the important sources of feedback information and can support building the whole services flow. User personalization and user participation are incorporated in this model as context elements or user needs elements, as shown in Figure 3-5. User personalization crosses different systems by the user identification approach, which was proposed in (Carmagnola & Cena, 2009). In our Framework, user modeling of personalization is different because it focuses on capturing a user's interests or activities.

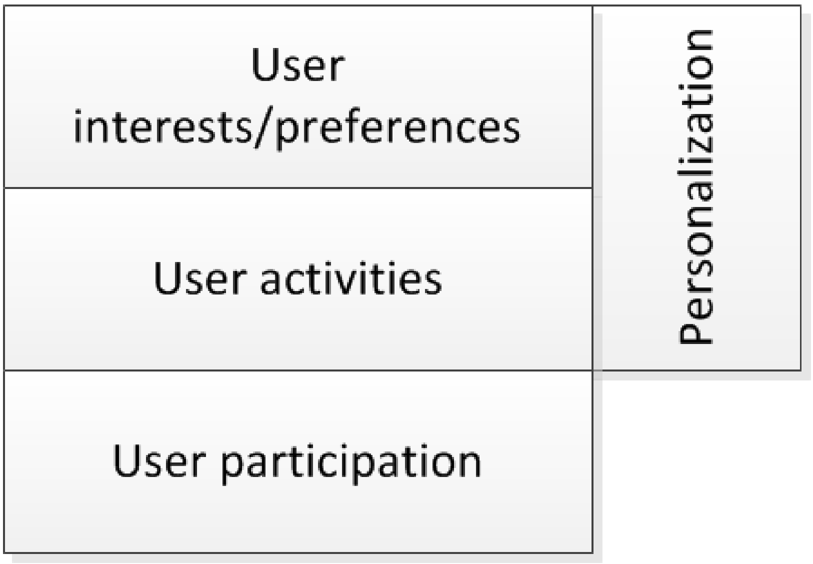


**Figure 3-5 User-Centeredness Aspects**

According to the Metaphors of human thinking for usability inspection and design, which are mentioned in (Frøkjær & Hornbæk, 2008), we listed the MOT (Metaphors of human thinking) for an FSM (Table 1.).

Metaphor M1: One’s habits are routines of behavior that are repeated regularly. Habits can be recognized by the system through learning the habit regular pattern.

Metaphor M2: The stream of consciousness is an activity of the human brain. Flowable service aims to provide a result stream for a user according to his/her consciousness flow.

Metaphor M3: Dynamics of human thinking implies the possibility of multiple results. All types of services are applied, including human-provided services.

Metaphor M4: Human needs are utterances, like splashes over water. Explicit needs and implicit needs are shown alternately or at the same time. Service flows can also be provided as an utterance.

Metaphor M5: Because needs or contexts can be changed at any time, there is no specific input expression for the system. Thus, understanding or interpreting human thinking accurately is important.

## Table 3-1 Metaphors of human thinking for FSM

| Metaphor | Metaphor of Human Thinking | Implications for Flowable services | Key Questions/Examples/Scenarios |
|---|---|---|---|
| M1 | Habit formation is like a landscape eroded by water. | Support of existing habits: recognize and memorize one's habit, or recommend users. | Are existing habits recognized?<br>(e.g., the user usually takes a cup of tea not coffee after a meal)<br>Can effective new habits be recommended?<br>(e.g., the user wants to change from smoking to not smoking)<br>Is the system predictable?<br>Can a cluster/group recommendation be supported?<br>(e.g., healthcare with fitness or diet)<br>Value comparison (e.g., shuttle or rental)<br>Special situation (e.g., alarm)<br>Emergence response (e.g., fall sick)<br>Marketing analysis |
| M2 | Thinking as a stream of thought. | A user's thinking should be recognized or pre-recognized by smoothly continuing services. | Does the system help users to resume interrupted tasks?<br>(e.g., the user reads the newspaper on the train commuting, but before a special exam, the user would like to review for an exam)<br>Can real-time information be collected? |

| | | | |
|---|---|---|---|
| | | | (e.g., weather or traffic or Stock index) |
| M3 | Awareness as a jumping octopus. | Support users' needs associations with not only Machine-provided services but also<br>Human-provided services. | Can the Human-search be accepted?<br>Trust<br>Security<br>Availability (NOT out of time) |
| M4 | Utterances as splashes over water. | Support both explicit needs and implicit needs.<br>Support to complete partial requirements from users. | Does the system provide diversity services?<br>Are alternative services for the same users' needs available?<br>(e.g., transfer navigation) |
| M5 | Knowing as a site of buildings. | Users should not have to rely on accurate expression for the input of a system. | Can the system be used without knowing every detail of it?<br>Can the system be able to get the context automatically? (e.g., sensor of temperature)<br>Can the system accumulate the experiences?<br>(e.g., when the same requirement comes, the services respond rapidly.) |

# Chapter 4 Context Aware System Architecture

User contexts are essential to making our proposed FSM work in the real world. In this section, we discuss how to harness user contexts to enable FSMs. After describing the classification of the contexts, we introduce the context capturer components for an FSM. We further conceive numerical measures for this model.

## 4.1 Applying Contexts

In this study, we classify contexts into three categories (Zhu, Shtykh, & Jin, 2010).

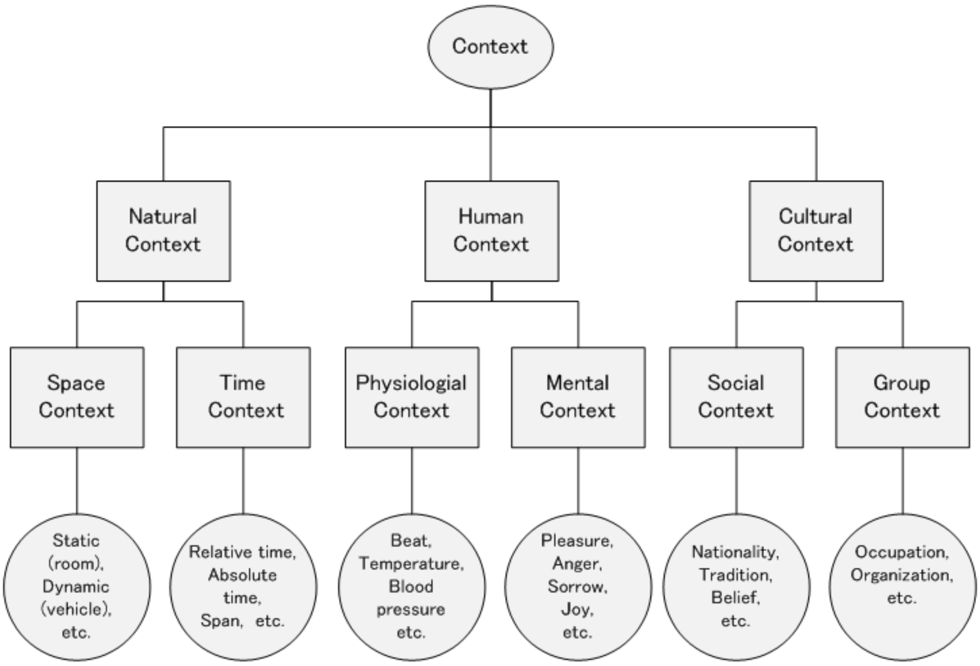

Context
Natural Context
Human Context
Cultural Context
Space Context
Time Context
Physiologial Context
Mental Context
Social Context
Group Context
Static (room), Dynamic (vehicle), etc.
Relative time, Absolute time, Span, etc.
Beat, Temperature, Blood pressure etc.
Pleasure, Anger, Sorrow, Joy, etc.
Nationality, Tradition, Belief, etc.
Occupation, Organization, etc.

**Figure 4-1 Categories of Context Factors**

We classified contexts into three categories, as shown in Figure 4-1.

**Human context** is about the human self. There are *physiological contexts* and *mental contexts*. The physiological contexts, such as the heartbeat, body temperature and blood pressure, can be checked by simple instruments, and the resulting values represent whether a user's body (i.e., health) is in good or poor condition. Mental contexts, for example, pleasure, anger, sorrow and joy, can reflect a user’s innermost status. It is difficult to capture these contexts by instruments directly. However, mental status can be illustrated by an emotional expression on one’s face or by a gesture or by body language. We believe that the expression on a face or a gesture (the action of a hand or finger or the movement of the body, e.g., nodding) can be tracked by chips and transformed into signals to be processed. Both the physiological contexts and the mental contexts are important elements for FSM-based systems.

**Nature context** includes the WHEN (*time context*) and WHERE (*space context*) information. Timestamp is a format of absolute time that records the trigger time of events or the end point of a change. The relative time is used as a stopwatch. Space contexts, such as humidity, temperature and location data obtained by a GPS, or some physical information about a certain space, are also important. For a vehicle, if it is moving, the velocity is also one of the space contexts.

**Cultural context** represents the group attributes of human relationship information. *Group contexts*, such as one’s occupation and organization information, may have some effects on the communication between a user and the people around him/her. For example, a visually impaired person communicates in his/her own way, while other people cannot understand the meaning well unless they learn the communication methods or body signs of the visually impaired people. For a larger human group, *Social contexts* cannot be ignored. The nationality, traditions and beliefs are a few examples in this category. As an automatic ordering service in a restaurant, the menu provided to muslins in the Islamic world is different from a general menu.

To bring these contexts into the model on a component level (as shown in Figure 4-2), the Context Capturer consists of a User/Service Data Aggregator, which is an interface for various services and devices that interact with a user. The Context Capturer collects contextual/situational information about a user and his/her services, detects context changes and collects snapshots. The Concept Organizer is responsible for analysis, grouping and construction of the structure of user context models that are stored in a Context Base, which is used by Mediators from the Mediator Complex to discover and compose flowable services that match the user's current context and situation (R. Y. Shtykh, Zhu, & Jin, 2009). This step is performed by accomplishing the following:

*(1) matching the characteristics of circumjacent available services with the user's current context;*

*(2) for better tuning, providing information about inferred user preferences to the services.*

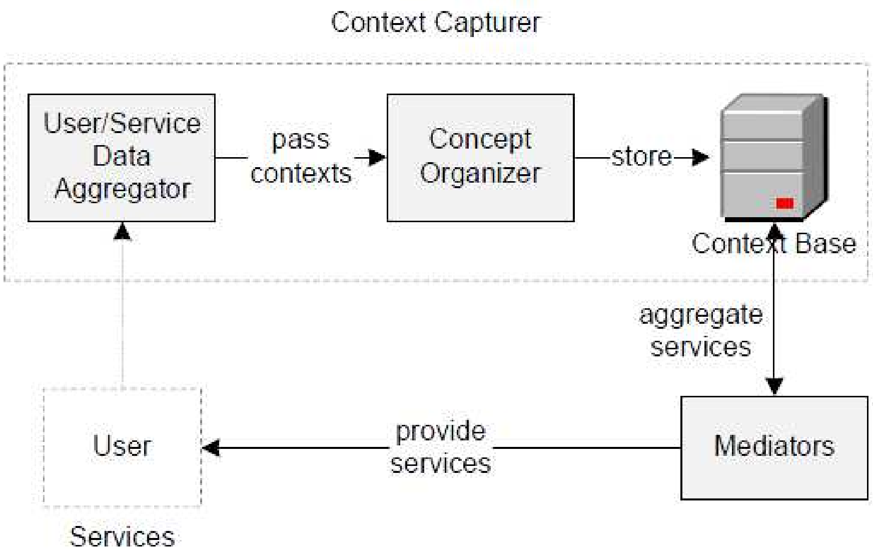


**Figure 4-2 Context Capturer: Components**

In contrast to fragmentary contexts (Shtykh & Jin, 2008), we can obtain such contexts by observing only a part of the user activities. The proposed system must have complete control of the services that are requested (or could be requested or needed) by a user.

This type of control is especially challenging today, because of the lack of an adequate infrastructure to support a ubiquitous environment.

The greatest challenge for Mediators here is not only to guess the context dependency of the user preferences in a specific situation but also to find the services that can be seamlessly integrated according to the user's context, because not all circumjacent objects can be contexts. As (Winograd, 2001) argues, "something is context because of the way that it is used in an interpretation and not because of its inherent properties." Otherwise (if it cannot be used for the interpretation of a target object), the context is only a physical setting.

## 4.2 Measures for the Model

A service can be evaluated by a set of parameters. We assume that, from a user's perspective (or a user-oriented service perspective), a service can be evaluated as follows:

$$\Phi_o = \prod_{i=1}^{n} N_i D_i \Big/ \sum_{i=1}^{n} C_i \qquad \text{Eq. (4-1)}$$

where $\Phi_o$ is an evaluation of a one-service process. Having $n$ services, $N_i$ is the needs of a specific user to be satisfied by service $i$ and $D_i$ is the satisfaction degree (how much one single service satisfies a user's needs). $C_i$ is the cost of the service process, which is summarized when the user changes from one service to another.

For the flowable services model, the same measure can be expressed as follows:

$$\Phi_f = ND/C \qquad \text{Eq. (4-2)}$$

because it is virtually considered to be one service, and the needs, satisfaction and costs are considered with regard to the model. Certainly, the costs are growing, because users' needs and their satisfaction from provided services change, but they are hidden in the flow. Furthermore, owing to such a holistic approach, the costs *C* in the model can be largely reduced, compared with a one-service model. Such a reduction is ensured, for example, by the Mediator Complex controlling the available services and ensuring the users' satisfaction by the timely provision of services that pertain to their needs. A basic example of such a reduction involves providing the service that was already demanded by a user in the same or similar context before. Because the flowable service is assumed to *know* about these user-service interactions, the costs are likely to be reduced. In addition, the flowable nature of the proposed model is expected to contribute to better user satisfaction. Therefore, $\Phi_f > \Phi_o$ can be expected.

## 4.3 System Layers

An FSM-based system can be designed in three layers, as shown in Figure 4-3. The bottom layer is the network of resources, and the middle layer is the network of services. A flowable service is mashed-up in the top layer. The resources include the data, applications, consults and hardware. All of these resources build up a resources network environment by special communication protocols, standard APIs and some middleware. This resources environment is the basis for the network of services.

We consider three types of services, which are included in the middle layer:

- **SaaS/PaaS (Software as a Service/Platform as a Service)**, which is a software/platform component that is described, e.g., via WSDL (Web Services Description Language) for a web service, and is capable of being accessed via standard network protocols such as SOAP (Simple Object Access Protocol) over HTTP (Broberg, 2002, May 03) or RPC (Remote Procedure Call) or REST

(REpresentational State Transfer). Once a web service is deployed, other applications or web services can discover and invoke the deployed service.

- **HaaS (Hardware as a Service, or hard equipment service)**, which is a type of virtualization, such as the *network disk*.
- **haaS (human as a Service) or HPS (human-provided service)**, which enables humans to publish their skills and capabilities as a service, such as *cyber manhunt* or *mass communication* or *expert team*. The role of humans in Cloud Computing is not limited to consuming services; services can also be provided by human actors. The haaS or HPS unifies humans and services because a service can be provided by a human actor or implemented as a software service (Schall & Dustdar, n.d.).

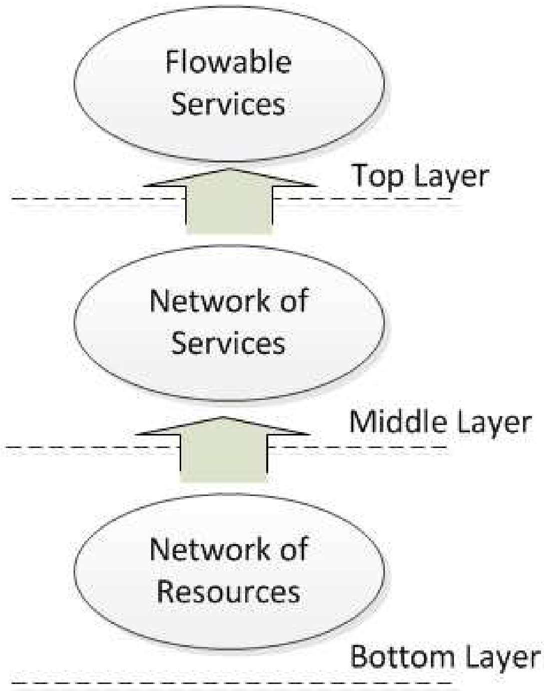


**Figure 4-3 Three Layers of an FSM-Based System**

In Figure 4-3, every resource reforms into services by *servicelization* processing, with open UDDI or WSDL or any other standards and protocols. Our research is *flowlization* processing, in other words, the FSM mashes up certain services to meet a user's needs in a flow style and provides the service flow to end users seamlessly.

## 4.4 System Architecture

The architecture of an FSM-based system is depicted in Figure 4-4. This type of system can be divided into three subsystems, and our research efforts are concentrated in the three directions.

### *4.4.1 Situated Behaviors Capturing Subsystem*

As explained above, concerning the user factors, this subsystem is designed and implemented to catch a user's habit by a certain routine of behaviors, which is done repeatedly and regularly over a period of time. User information composed of user factors is the input into the system. The Abstract module puts user information into an abstract and uniform structure, and then sends it to the Behaviors/Habits Learning module and Behaviors/Habits Matching module. In the Behaviors/Habits Learning module, a user's atom actions, sequential behaviors, or long-time habits are recorded into the user's Personal Behaviors/Habits Database. The system uses these records to match the real-time user information that fits the user needs.

### *4.4.2 Rules/Contexts Matching Subsystem*

All of the nature context data and physiological context data of the human contexts are logged by various types of sensors, RFIDs or some other chips. The other data, which are related to cultural contexts, are pre-saved in a Social Rules Database. This database is composed of XML editable format files. A user's intrinsic properties about culture contexts are saved as the user's profile, and can be edited by the user. When the intrinsic properties are compared with the Social Rules Database, the temporary contexts are selected out. The temporary contexts are compounded with the user’s behaviors/habits information at the User Contexts/Needs Specification module.

### *4.4.3 Services Synthesizing Subsystem*

As mentioned before, three types of services, i.e., SaaS/PaaS, HaaS and haaS, exist in the service space. These services use the same method to register, via the Services Identification module. The Services Identification module allots a unique identifier to the service, and performs a simple mapping of different usages. The mapping information is the source of the Services Recommendation module. The Services Recommendation module, according to a user’s needs, searches mapping information and recommends suitable services. Another input branch starts from the User Needs/Contexts module. Fixed user needs information is an input to the Services Selection module, where recommendation information, feedback information and contexts/needs information are considered selecting conditions. Most suitable services are selected from the services space, and then, the services are sent to the Services Mash-up module. With mash-up processing, all of the selected services are made into a composite, appearing to be one service, seamlessly, regardless of whether they are provided by the same service provider. Moreover, the feedback information of the used flowable services is saved into the Services Feedback Database. The feedback information is saved transparently for users. By the feedback information, the Services Mash-up module can review the user’s needs and then re-mash-up the services.

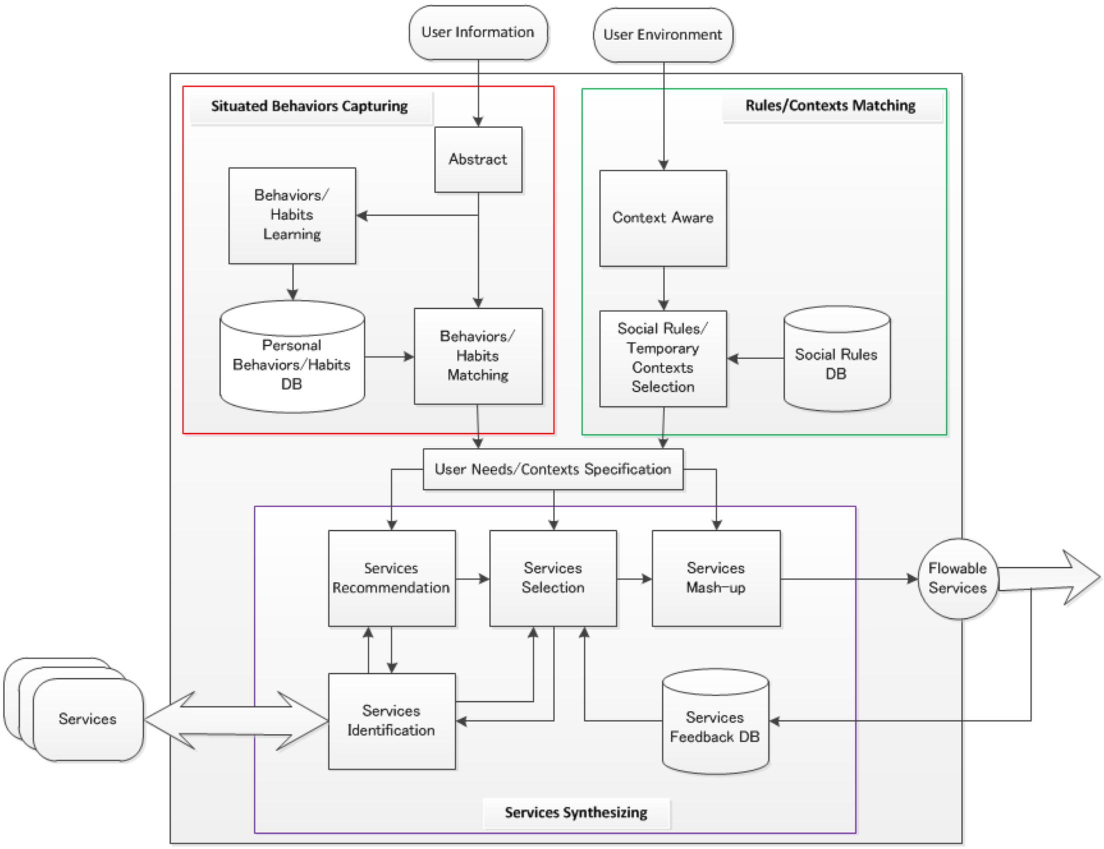
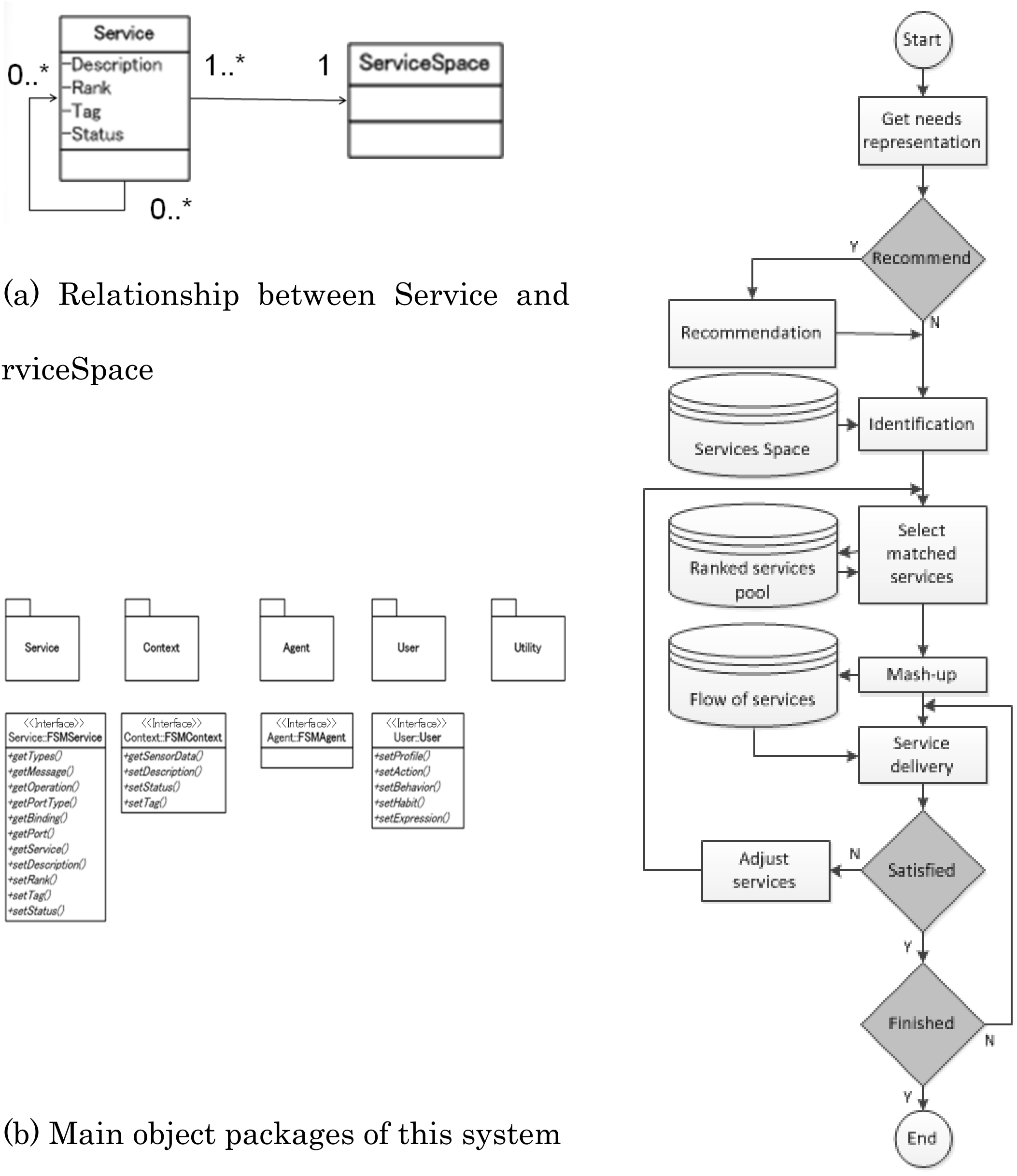


**Figure 4-4 Basic Architecture of an FSM-Based System**



(a) Relationship between Service and ServiceSpace

(b) Main object packages of this system

(c) Work flow of the services provision

**Figure 4-5 An Instance of the Service Space and Working Flow**

The data structures and working flows from the service selection, from mash-up to delivery, are shown in Figure 4-5. In our system, a services space is a container of services. This space contains one or multiple services. One service has attributes, such as Description, Rank, Tag and Status. Additionally, one service can be referred by another service or by multiple services, as shown in Figure 4-5 (a). Figure 4-5 (b) illustrates the main packages of the system. Figure 4-5 (c) displays the detailed working flow, which is described as follows. The User Needs/Contexts Specification module outputs a formatted user needs in a computable representation. Then, the system checks the formatted needs as to whether the recommendation is required. If the user needs the system to recommend services, the Services Recommendation module is called to perform this processing. If not, we go to the Services Identification module directly. The identification module pre-processes services with unique IDs and special mapping mechanisms from Services Space. Then, matched services are selected and ranked by the similarity of user needs. The temporary results are saved in the Ranked services pool. Then, in the Mash-up module, the flow of services is produced to be saved and delivered. By checking the user’s feedback information, if the adjustment of the services flow is necessary, the system goes back to the services selection again. If the feedback shows that a user accepts the provision, the system performs these processes until the checking of this flow completion is true.

# Chapter 5 FSM-Based Adaptively and Dynamically Reconfigurable Service System

## 5.1 Adaptively Emerging Mechanism

Cloud Computing brought great advantages in terms of its cost control, reliability, elasticity, etc. Services, as the elements in Cloud Computing, attracted more attentions in recent years. According to (Hilbert & López, 2011), the Cloud Computing's information processing capacities are quickly growing at clearly exponential rates. Consequently, the numerous ambient services can be highly expected. In the meanwhile, more and more people, including researchers, enterprise users, and general users, must acquire new capabilities to build and master services (I. a. b. Foster, 2005). By this means, we think it becomes necessary to support people on how to make a proper selection of ambient services quickly, and how to make optimal selection of ambient services economically.

As we know, that it is not enough to utilize and assess Cloud Computing just from the perspective of businesses/enterprises profitability or technical evolution. We have proposed a new framework of Flowable Service Model (FSM) in our previous work, and tried to gain a solid understanding of a novel concept from a human-centric view in this compelling new trend. Under the framework, users do not need to know which company or organization provides the services they are having, since the basic concept of FSM is to synthesize and mash-up numerous different services that are been running by different companies or organizations or other people. In this study, we newly propose and develop an Adaptively Emerging Mechanism for the selection of ambient services which is applied and integrated into this model.

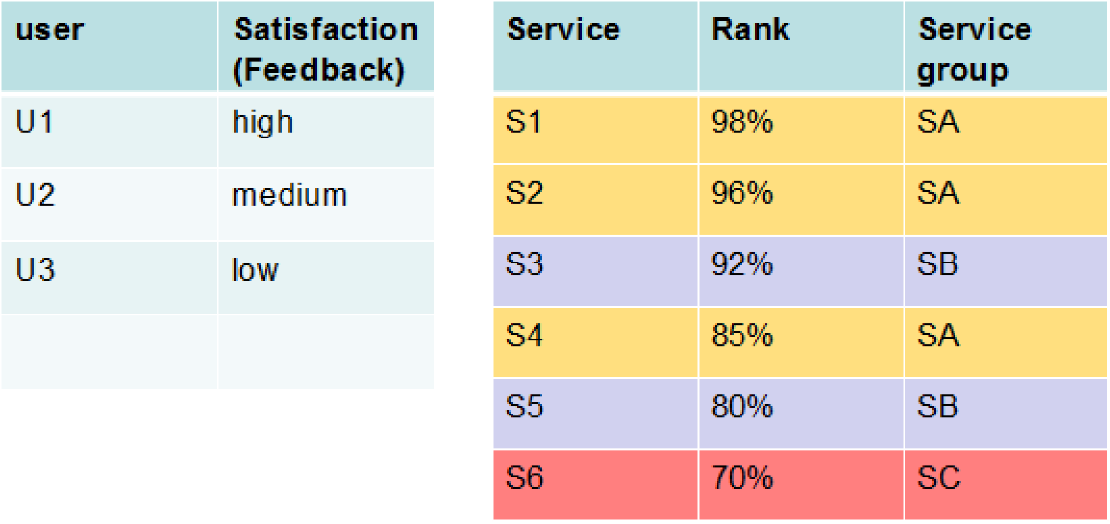
The pivotal factor in AEM is the feedback of user satisfaction. We assume that the satisfaction information can be captured and obtained by some means such as context-aware sensors in the process of users' responses. The user, who is very particular about the exploited services, often rejects the result which is provided by FSM. We consider this kind of particular user is in the LOW feedback users group. We take the other three user groups as MEDIUM, HIGH, RANDOM by their distribution of feedback values.

To arrange these similar but different numerous services, we set five dimensions for grouping services:

- *service cost,*
- *matching result precision,*
- *service responding time,*
- *personal context information, and*
- *social context information.*

The first three dimensions are relevant to the service provider. The last two dimensions depend on the user and the using situation. The dimensions for grouping services are not limited into this five ones. Further consideration of all-inclusive dimensions shall be more effective for grouping services.

## 5.2 AEM Algorithm



| user | Satisfaction (Feedback) |
|---|---|
| U1 | high |
| U2 | medium |
| U3 | low |
| | |

| Service | Rank | Service group |
|---|---|---|
| S1 | 98% | SA |
| S2 | 96% | SA |
| S3 | 92% | SB |
| S4 | 85% | SA |
| S5 | 80% | SB |
| S6 | 70% | SC |

**Figure 5-1 Example Data for AEM**

An example data for this regulated selection process is shown in Figure 5-1. There are four users, from U1 to U4, belonging to Random, Low, Medium, High feedback types separately. The right side table of Fig. 3 is a result set, which are ranked by the similarity of a user's needs and the services. U3, a student, who wants to cut down the expenses for service usage, his personal context information shows he needs economy choice. The services are classified by the dimension – service cost. These services that are well-matched by cost are in a same service group, SA, Group A in yellow line, SB, Group B in blue line, or SC, Group C in red line. If the student U3, rejected S1 in Group A, it can be inferred that both S2 and S4 in Group A, in yellow line, could NOT satisfy U3. Then S3 is pumped to U3 by skipping the S2.

Figure 5-2 shows the algorithm of Adaptive Emerging Mechanism in pseudo code.

```
Step1:    Initialization
              Needs ← user's needs
              $S$ ← all available services
Step2:    Selection & Ranking
              $\overline{S}_{(needs)}$=Match(Needs, $S$);
              $Q_{(similarity,needs)}$ =Rank( $\overline{S}_{(needs)}$ by the Similarity);
Step3:    Grouping Services
              Dimension=GenerateDimension($\overline{S}_{(needs)}$);
              for each Service in $Q_{(similarity,needs)}$
                  $\overline{Q}_{(grouping,similarity,needs)}$=Grouping(Dimension);
              end for
Step4:    Outputting
          Provide service s to user
Step5:    Getting Feedback
              if Feedback == REJECT then
                  Regulate($\overline{Q}_{(grouping,similarity,needs)}$, s.Dimension);
                  goto Step4
              end if
```

**Figure 5-2 Adaptively Emerging Mechanism Algorithm**

The description is given as follows.

$S$ , is a services set, which contains all available services.

$\overline{S}_{(needs)}$ , is a sub services set, which contains the services matched with users' needs. $\overline{S}_{(needs)} \subset S$

$Q_{(similarity,needs)}$ , is a queue which ranks the element services in $\overline{S}_{(needs)}$ by the similarity. The similarity is decided by services and user's needs. In the traditional method, services are provided to a user one by one according to this ranking. The ranking is static.

$\overline{Q}_{(grouping,similarity,needs)}$ , added the grouping information of services. And AEM can regulate the ranking based on the user feedback information. The ranking here is dynamic and adjustable.

As shown in Figure 5-2, Step 1, 2, and 4 compose a usual selection process (matching query and outputting result). But in AEM, we further add the Steps 3 and 5. In Step 3, the preliminary results are dynamically grouped by *Dimensions* and *weights*, which will be discussed below. In Step 5, the user's satisfaction feedback is used to adjust the results link. It skips the result(s) which are in the same group that the user has rejected one.

## 5.3 Design of AEM Simulation

When we search for information or services, the search engine lists a long result set. Artificial checks are needed to filter the entries in the result set one by one. But mostly, the filtered information has some certain similarities, which can be processed automatically. In this study, we newly propose and develop Adaptively Emerging Mechanism (AEM), which aims to help users save their time and cost in a certain degree.

The basic idea of AEM is to dynamically group the ambient services based on a variety of available context information, and thus adaptively select the candidate components that constitute the current flowable service based on the feedback information of user satisfaction and the cost of a service. AEM works in the service selection process, which matches the query with the available resources and adaptively adjusts the result based on the feedback information. The main process is the same as a usual selection method which only focuses on matching the query conditions with the available resources. In addition, AEM adds a complement process which can dynamically adjust the result by the user's satisfaction feedback.

### *5.3.1 Emerging Resources*

There are many useful resources that can be used and included in the provision of services. Users get satisfied by a service mashing-up or synthesizing these resources. In order to find a best or better selection for every user, we utilize the brain working model presented in (Minsky, 2007), and extend it to build the proposed mediator's model.

From the perspective of users, Context Capturer (Y. Zhu et al., 2009) emulates the Human Information Processing (HIP) for consciousness as shown in (Minsky, 2007). Based on this, a user's needs are inferred and provided to the mediator. Note that here we do not try to give the complete solution for needs inference, but indicate an abstract method for it to show that needs have to be extracted and passed to the mediator. In order to infer the user's needs, the method can be substituted by any other one capable to do so. Emerging Resources are extended from this kind of "resource". It is a complex of Agents and Services. One agent can be integrated with one or plural services into an emerging resource. It not only has the features of agents: autonomous, reactive, proactive, but also has the features of services, for example, performing business processes more efficiently and effectively

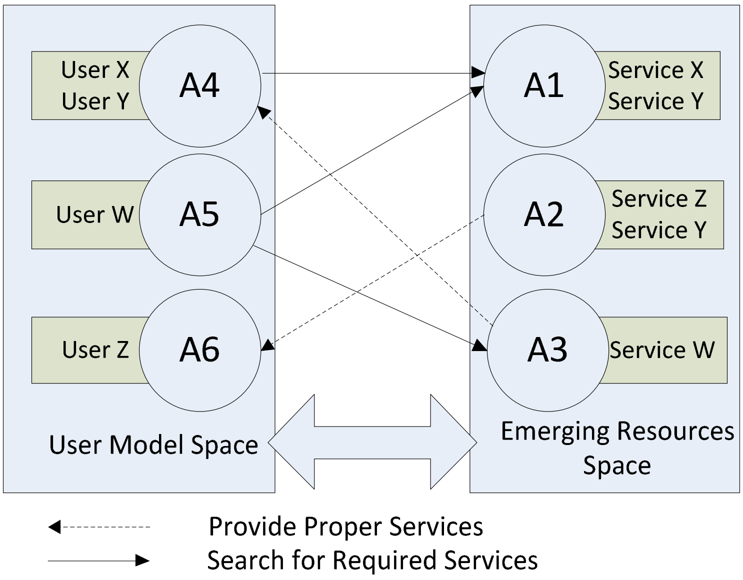


**Figure 5-3 Emerging Resources Space and User Models Space**

As shown in Figure 5-3, emerging resources space is shown in the right side box. There are many agents list as A1, A2, A3, etc. One agent can compound with one or more services. In the left side box, the individual user model or the group user model, or the machine user model compound with agents composed into user model space. Between the spaces two types of dialogue exist. One is providing proper services, and the other is searching for required services. These dialogues are established between agents and use the language of agents, such as ACL (Agent Communication Language).

### *5.3.2 Feedback Distributions*

Feedback may be obtained or captured by sensors, such as an eye-tracking device, or other devices, such as camera. We define the feedback value scope is between 0 and

100. The user satisfactory level may be set according to the real application situation. In this simulation experiment, the satisfactory level is set to be 50. That means the user is dissatisfactory if the feedback value is lower than 50.

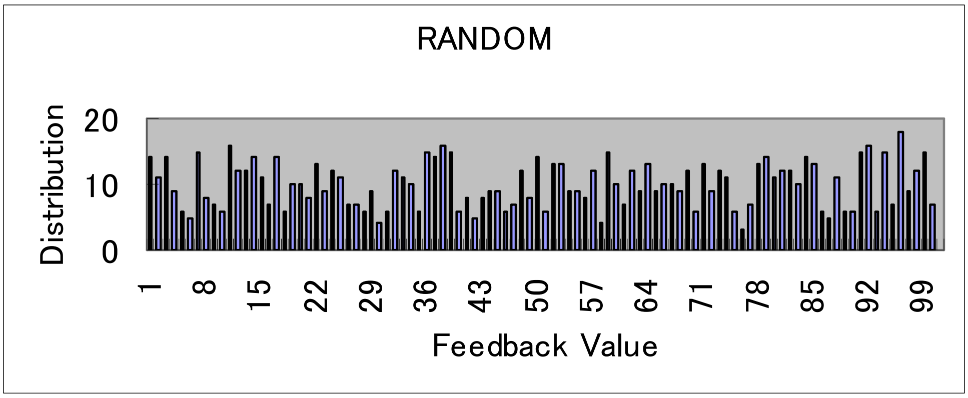
In our simulation, we created a feedback generator which can produce four kinds of feedback distributions shown as follows.

1) RANDOM: in the Uniform distribution. As shown in Figure 5-4 (a), the feedback generator produces 1000 feedbacks randomly, which should be limited within the assumed value scope.

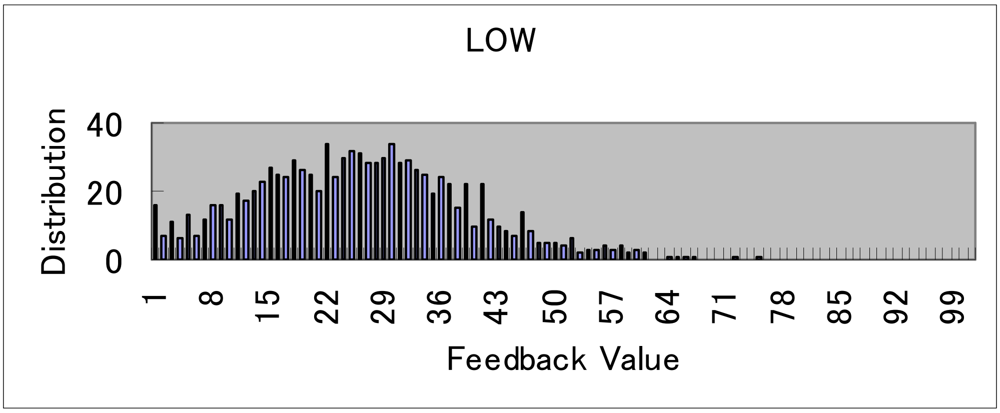
2) LOW: in the Gaussian distribution with the standard deviation equals to 15, and the average value equal to 25. The result is shown in Figure 5-4 (b).

3) MEDIUM: the same Gaussian distribution with the standard deviation equal to 15, and the average value equal to 50. The result is shown in Figure 5-4 (c).

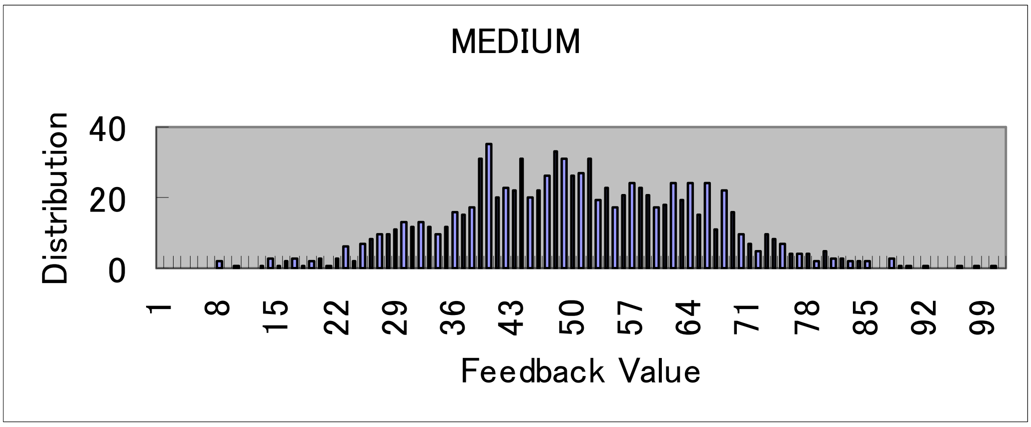
4) HIGH: the same Gaussian distribution with standard deviation equals to 15 and the average value equal to 75. The result is shown in Figure 5-4 (d).



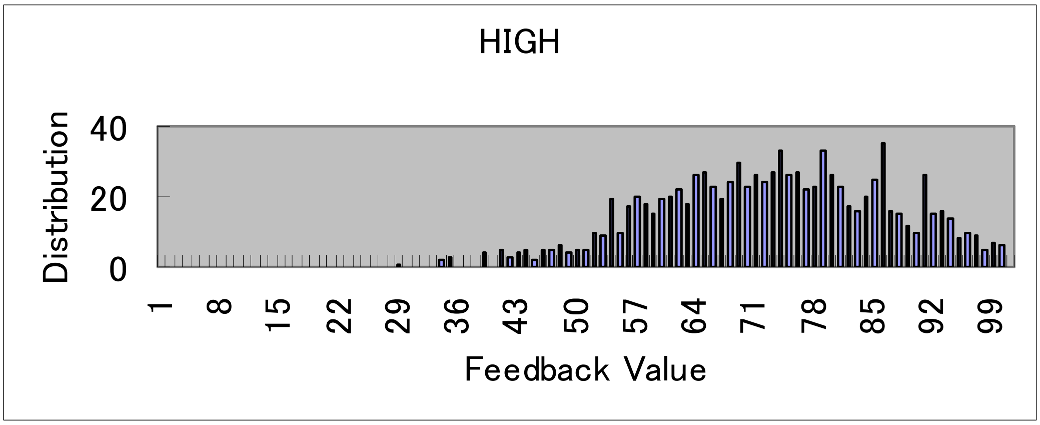
**(a)**



(b)



(c)



(d)

**Figure 5-4 Feedback Distributions**

The four feedback distributions represent four types of users. RANDOM means the user accepts a service with a good satisfactory level randomly, HIGH means the user can accept a service with satisfaction easily, MEDIUM means the user can accept a service with satisfaction averagely, LOW means the user can accept a service with satisfaction difficultly. We have conducted the simulation of AEM applied into these four types of users.

### *5.3.3 Regulated Selection*

In order to regulate the selection results by the feedback value, we dynamically group those similar but different services based on a set of five dimensions. Services are grouped according to: the service cost, the matching result precision, responding time, and personal and social context information with a weight of different values.

*Service cost* represents as the price for every service usage. The service provider independently determines the price of a service based on the laws of the market. For cost-saving users, this dimension gives a higher weight for selection.

*Matching result precision* means a scale of precision, just as a number 1.00 is more accurate than 1. In a certain use case, users need to consider not only the value but also the result precision. For precision-priority users, this dimension works better.

*Service responding* time is considered as a period of time that starts at a new query and ends at a given result. For time-saving users, this dimension takes an important role for the selecting process. This is dynamic. The same service could have different responding time per usage.

*Personal context information* is considered as a user priority. By personality, every user has a special selecting pattern. A certain brand, or a certain expressive style or a famous service provider, all of these factors can be decisive ones.

*Social context information* is given from a social perspective. If there is not enough personal context information, the social context can make a supplement by assuming that the users in a same social group have a similar selection tendency

Dimensions = {d1, d2, d3, d4, d5} Eq. (5-1)

Weight = {w1, w2, w3, w4, w5} Eq. (5-2)

Grouping = |Dimensions|*|Weight| Eq. (5-3)

Services can be grouped by *Grouping* value applied with k-means or other clustering method.

## 5.4 Simulation Result and Analysis

We design the AEM simulation by using four types of Feedback distributions. The number of services is assumed 400 in this simulation.

Total services are counted to 400. At the same time, the AEM applied services are counted to 100 including in the total services. The grouping scope is considered into multiple groups from 2 to 11.

The simulation interface is showed in Figure 5-5.

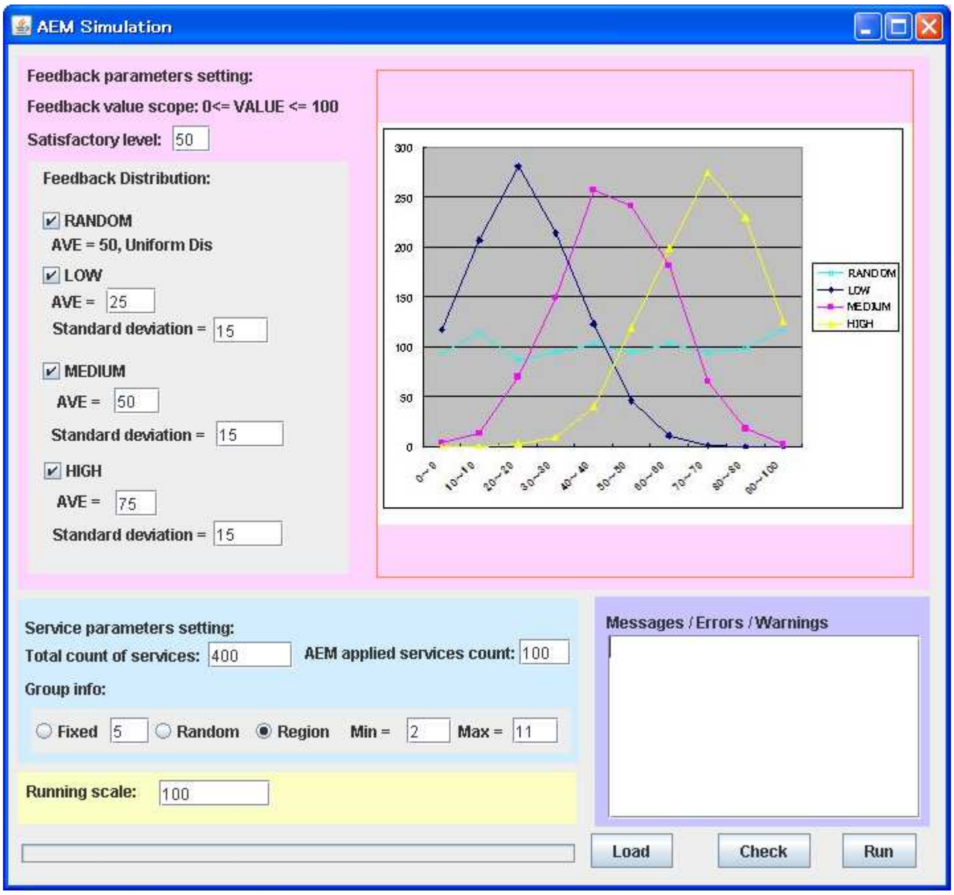
We considered a measurement evaluation for AEM, shown in Eq. (5-4).

$$Saving = \sum_{i=1}^{MAXTOTAL} \left( \frac{TimesCount(i)}{TOTALTIMES} * \sum_{j=0}^{MAXSKIP} \left( \frac{TimesCount(j)}{TimesCount(i)} * \frac{Skipsteps}{Totalsteps} \right) \right)$$

Eq. (5-4)

where *Saving* stands for a whole evaluation. How many skipped steps can be counted by *TimesCount()* function. When running the simulation, the service group number is changed gradually. The more the skip steps there are, the higher percentage of *skipsteps/totalstep* can be. The value of *Saving* is a relatively evaluation. Although the real cost of selection can not be represented by *Saving*, which is based on the skipped steps, the selection processing time/step can be decreased greatly.

*Step* is a service number which counted beginning at the first when a service is provide and ending when a user accepts one service. The *Totalsteps* represents it. *Skipsteps* is a skipped service number which applied in AEM.



AEM Simulation

Feedback parameters setting:
Feedback value scope: 0<= VALUE <= 100
Satisfactory level: 50

Feedback Distribution:

☑ RANDOM
AVE = 50, Uniform Dis

☑ LOW
AVE = 25
Standard deviation = 15

☑ MEDIUM
AVE = 50
Standard deviation = 15

☑ HIGH
AVE = 75
Standard deviation = 15

Service parameters setting:
Total count of services: 400 AEM applied services count: 100
Group info:
○ Fixed 5 ○ Random ● Region Min = 2 Max = 11

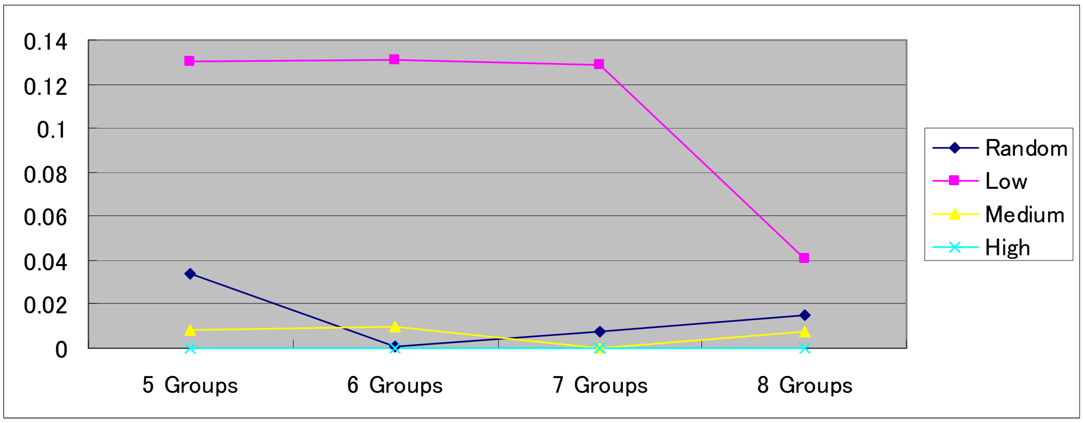
Running scale: 100

Messages / Errors / Warnings

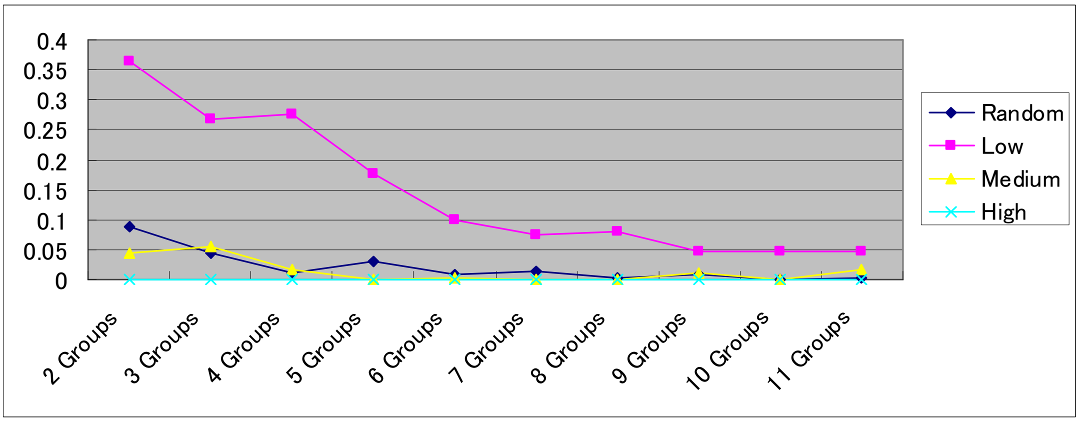
**Figure 5-5 AEM Simulation Interface**

The simulation result is showed in Figure 5-6. In the different grouping case, the AEM saves higher percentage in LOW feedback users than in others.

In Fig.6, the horizontal axis is grouping information. The vertical axis is saving value. Fig.6 (a) means the 400 services are divided into 5 groups, 6 groups, 7 groups and 8 groups separately. Fig.6 (b) means the 400 services are divided into 2 groups to 11 groups separately.



**(a)**



**(b)**

**Figure 5-6 AEM Simulation Results**

From this AEM simulation result, some of interesting findings can be listed below:

(1) Substantially improving selection process for LOW feedback users;

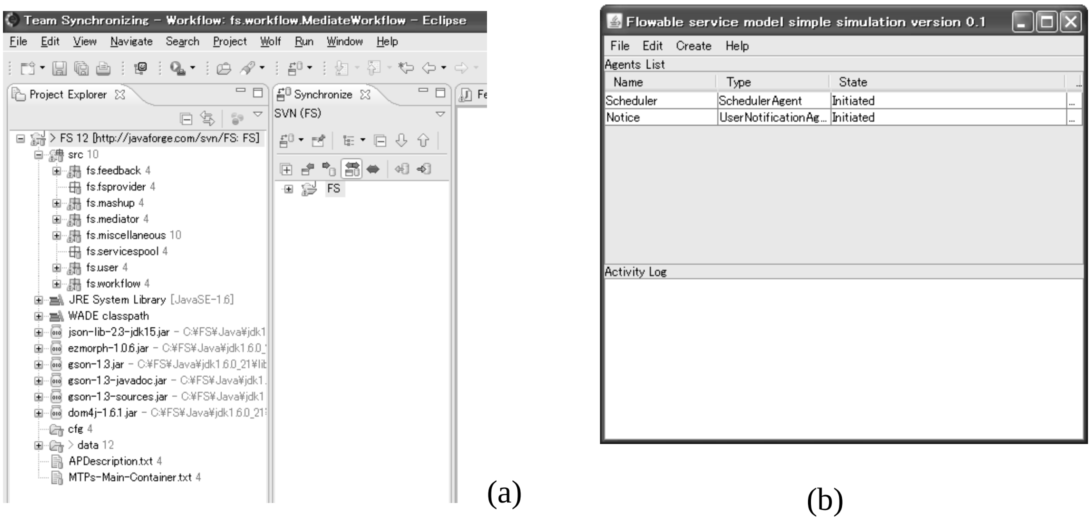
(2) No negative impact on the selection process for MEDIUM or HIGH feedback users;

(3) Enhancing the rationality for services selection.

From the result line, we can see some point up and down with a little value difference. We are going to do larger scale simulation by increase the services number and running times. We also consider the grouping number of services can impact the result precision within a certain number of services. On the other hand, the measurement evaluation *saving* for AEM, which mentioned above, should be improved.

## 5.5 Application Scenario

We have developed a simulated implementation of an FSM-based prototype system by using the Eclipse IDE and a free open source project hosting site (javaforge.com) for the SVN, as shown in Figure 5-7 (a). Figure 5-7 (b) is an initialized interface. In the main menu, different mediators can be configured and started.


Team Synchronizing – Workflow: fs.workflow.MediateWorkflow – Eclipse
File Edit View Navigate Search Project Wolf Run Window Help
Project Explorer
Synchronize
SVN (FS)
FS
Flowable service model simple simulation version 0.1
File Edit Create Help
Agents List
Activity Log

```
> FS 12 [http://javaforge.com/svn/FS: FS]
  src 10
    fs.feedback 4
    fs.fsprovider 4
    fs.mashup 4
    fs.mediator 4
    fs.miscellaneous 10
    fs.servicespool 4
    fs.user 4
    fs.workflow 4
  JRE System Library [JavaSE-1.6]
  WADE classpath
  json-lib-2.3-jdk15.jar - C:¥FS¥Java¥jdk1
  ezmorph-1.0.6.jar - C:¥FS¥Java¥jdk1.6.0_
  gson-1.3.jar - C:¥FS¥Java¥jdk1.6.0_21¥lib
  gson-1.3-javadoc.jar - C:¥FS¥Java¥jdk1.
  gson-1.3-sources.jar - C:¥FS¥Java¥jdk1
  dom4j-1.6.1.jar - C:¥FS¥Java¥jdk1.6.0_21
  cfg 4
  > data 12
  APDescription.txt 4
  MTPs-Main-Container.txt 4
```

(a)

| Name | Type | State |
|---|---|---|
| Scheduler | SchedulerAgent | Initiated |
| Notice | UserNotificationAg... | Initiated |

(b)

**Figure 5-7 Initialized Simulation Interface**

Figure 5-7 (a) shows the snapshot for a job hunting mediator in the application scenario. This mediator uses a user's needs as a Condition and calls other job hunting services, matching the results as a recommended job list that is shown in the bottom Recommendation area. On top of this results list, detail information about the most similar result is displayed in the top left area. The same as shown in Figure 5-8 (a) for the job hunting mediator; Figure 5-8 (b) shows the education mediator, and Figure 5-8 (g) shows the house finder mediator, which are some major roles in our applications scenario described below.

Li was a commerce engineer of an investment company with a high-level salary. He had many business experiences in reorganization and cross-border mergers and acquisitions. However, because of the economic recession, the company could not withstand the burden of the large human resource cost, and fell into a business crisis. Li had to find another new job. Then, Li registered with a job-hunting agency (Figure 5-8 (a) Job Hunting). According to the advice provided by the job-hunting agency, if Li had the National E-commerce Engineer Certificate and the TOEIC Score Certificate, Li would have an increased chance for a high-paying job. Thus, Li hoped to recharge himself quickly during a short period of time (Figure 5-8 (b) Education). At the same time, unemployment pressures and lifestyle related to disease, such as heart disease, metabolic syndrome and nephritis, troubled both his mind and body. He needed a treatment of pressure and building strength of his body with the National Physical Fitness club VIP card.

On the first day after he registered with the job-hunting agency, when he checked his email inbox, the job information messages also popped up, because the email account identification was open to the job hunting service (Figure 5-8 (c) Notice). The messages were the related job vacancies for him, depending on the conditions that he had registered. In addition, the training courses were recommended and provided to him. Li reviewed all of the jobs but had not yet decided on one or two with which he should do more contact. He selected a one-month Business English Writing online training course and began this program.

On the second day, he received a message form one of his SNS friends, Hu. Li did not meet Hu before. They kept in touch with each other on SNS space. Hu knew Li had lost his job from a blog article in his space (Figure 5-8 (c) Notice, Figure 5-8 (d) SNS). Then, Hu mailed Li to invite him to join a certain project of his company for half a year. Li replied to Hu to accept this short-term job and made an appointment for an interview.

On the third day, he was preparing for the interview. Because the project was about the computer business, Li quickly understood the basic concepts of the computer hardware. The vocational education service provided him a 90-min edition of a course on basic computer knowledge. Because the company was in another city, the travel service booked the train ticket for him and the weather forecast service prompted him that he had better to take an umbrella (Figure 5-8 (e) Movement Guider).

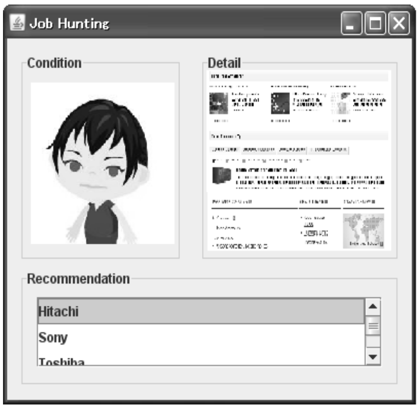
On the fourth day, he traveled to the company for an interview. On the bullet train, he read the materials from the new company to familiarize himself with it (Figure 5-8 (f) Contents Provider). When he stepped down from the train, the navigator showed him which communication media should be chosen, and during the period of the interview, the mobile phone changed the phone profiles from normal to silent automatically, because the navigator knew the address of that company and the mobile phone knew the interview time from the system.

Because of his good performance during the interview, Li obtained the job. However, before he took the new position in the branch company, he had to be trained for one month in the headquarters, which meant that he would commute for a long distance every day during the next month between the two cities. At this time, the system recommended that he rent a room on a monthly basis, with household electrical appliances and furniture. It was a little small, but it looked very comfortable for one person to stay for a short period. Thus, he rented the recommended room (Figure 5-8 (g) House Finder).

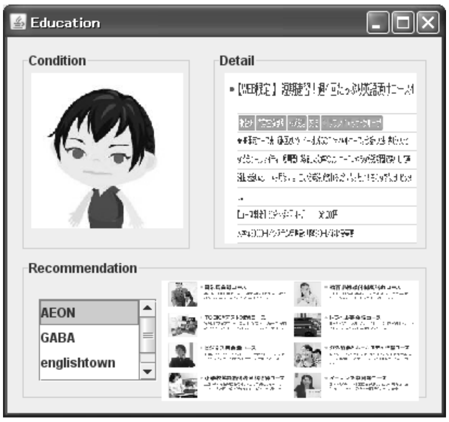
During this month, when his daughter's birthday arrived, he seemed to have no time to buy a present and go home for her birthday party, because he had to attend an important meeting on that day until late evening. As a result, he recorded a message for his daughter. The system delivered the message and a gift to his home, according to his daughter's favorite choices (Figure 5-8 (h) Scheduler).

After having a difficult but helpful train ride, he started his new job.

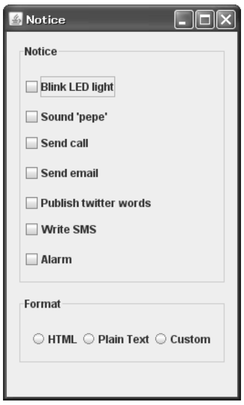
Our proposed FSM attempted to provide a unified way to deliver, seamlessly, the related services for the target user, by utilizing the available contexts and services.



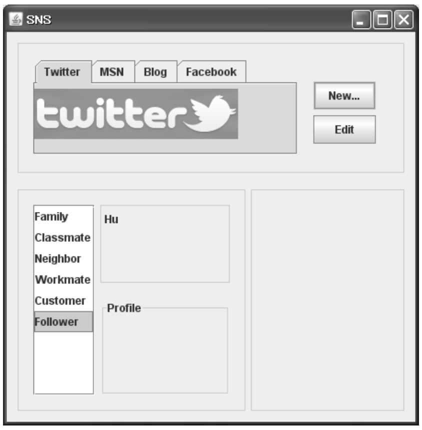
(a)



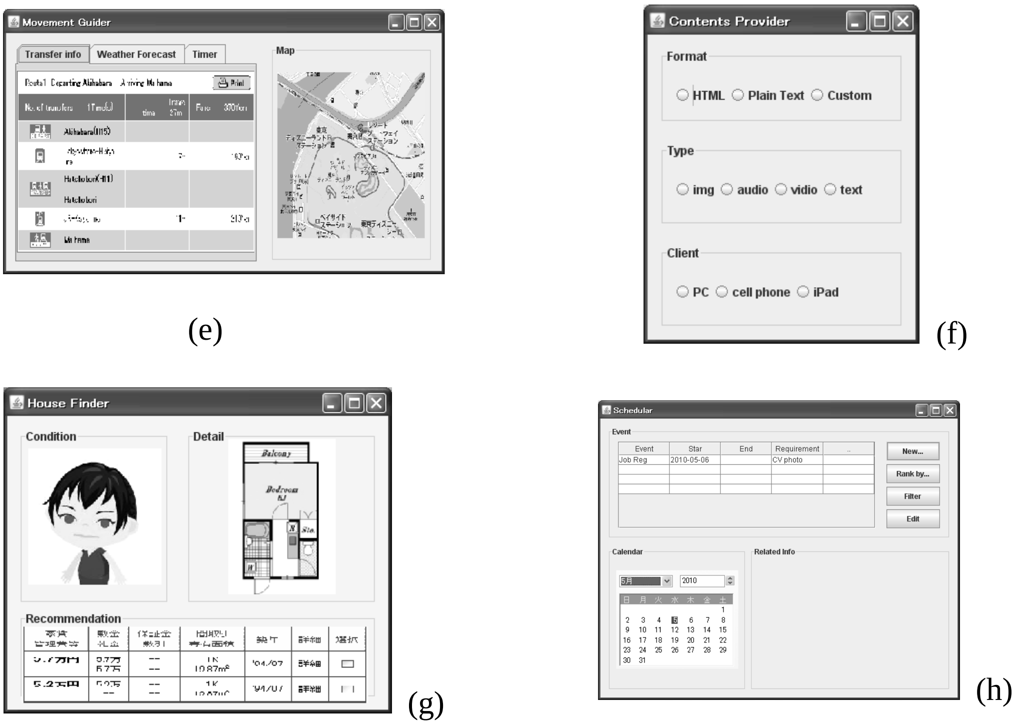
(b)



(c)



(d)



| Event | Star | End | Requirement | .. |
|---|---|---|---|---|
| Job Reg | 2010-05-06 |  | CV photo |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

(e)

(f)

(g)

(h)

**Figure 5-8 Snapshots for the Application Scenario**

## 5.6 Discussion

### *5.6.1 Markup Languages*

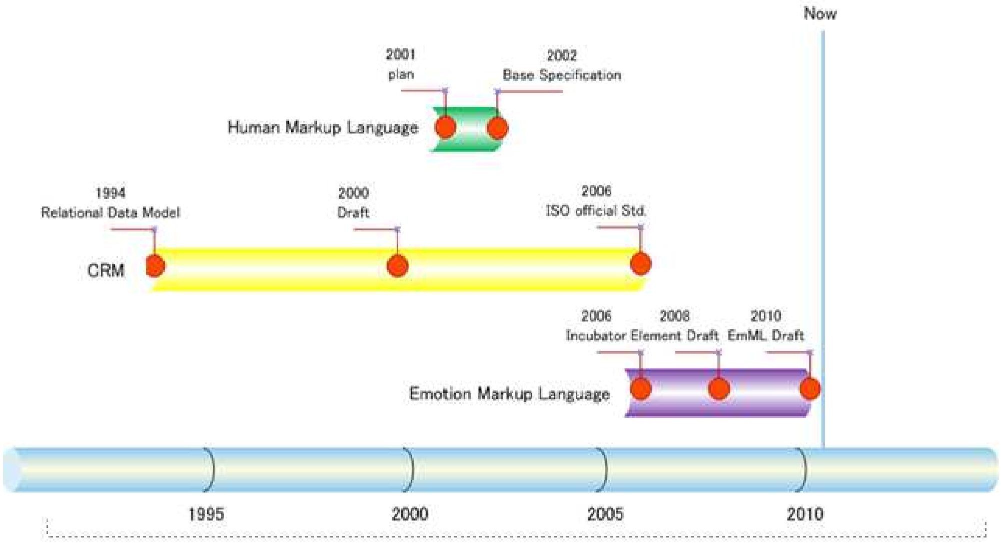


**Figure 5-9 Timeline of Markup Languages**

The CIDOC Conceptual Reference Model (CRM)[10] is generated in 1994. It is intended to promote a shared understanding of cultural heritage information by providing a common and extensible semantic framework that any cultural heritage information can be mapped to. Over a decade working, the CIDOC CRM became an official standard ISO 21127:2006.

Figure 5-9 shows the timeline of the CRM, Human Markup Language[11] and Emotion Markup Language[12]. Human Markup Language aims to reduce miscommunication through a standard framework of referents to descriptions of emotional states; enhance communication by enabling emotional states to be identified and used to query if requests and responses do not conform to predicted ranges for sequence and frequency

[10] http://www.cidoc-crm.org/
[11] http://xml.coverpages.org/humanML.html
[12] http://www.w3.org/TR/2009/WD-emotionml-20091029/

within a genre; create communication through authoring tools that use genre-based schema to organize sequences and frequencies of emotional expressions. Emotion Markup Language aims to allow a technological component to represent and process data, and to enable interoperability between different technological components processing the data. At present the CRM and Human Markup Language seems to stop their improvement. Some updated are on Emotion Markup Language.

To express the context information, we consider designing an XML-style Consciousness Markup language, just like human markup language and emotion markup language, to representative the human's consciousness succinctly so that it can be processed by our FSM. We have classified the contexts information of user in to three categories. The Consciousness Markup language can convey those contexts information to a same process module.

# Chapter 6 Conclusion

## 6.1 Summary

This study has focused on new services synthesizing model called FSM, which is a logical stream that organizes and provides circumjacent services in such a way that they are perceived by individuals as those naturally embedded in their surrounding environments. To support this context-aware model, we have applied and classified contexts into three categories: human context, nature context, and cultural context. We have further proposed and developed Adaptively Emerging Mechanism, in which a concept of Emerging Resource has been proposed. Our AEM simultaneously considers a user's needs and context information in a certain selection process. For the simulation, four types of feedback distributions, and five service grouping dimensions have been considered. AEM provides a regulated selection for ambient services and can be applied in different applications, such as service selection, information recommendation, and personalized prediction, and for marketing recommendation.

### *6.1.1 Contributions*

The thesis makes some original contributions:

Contribution 1: Flowable service

In this paper, we have presented a flowable service concept,

Contribution 2: Context aware approach

Contexts information has been classified into three groups: human factors, nature factors, and cultural factors. In the user needs acquisition processing, these contexts factors as needs elements are integrated with user attributes. Especially, the cultural factors are background information for users to enjoy a proper services flow which consider users' nationality, race, and traditions and so on.

Contribution 3: Adaptively Emerging Mechanism

Adaptively Emerging Mechanism (AEM) can help people reduce this selection burden by an interdisciplinary approach. We consider the user's feedback information is a pivotal factor for AEM, which contains the user's satisfaction degree after using the services. At the same time, we suppose the service cost, the service result precision, the responding time, the personal context information, and the social context information etc. as essential parts of the optimizing process for the selection of ambient services.

Contribution 4: Needs

We have tried to definite the needs representations, so that user immanent needs can be expressed with environment context information. Although this representation is an abstract of real user needs, the more context elements can be obtained, the better represent user needs. All kinds of sensors applied to the environment without catching users' attention, just as the air around the environment. User behaviors and user interests' logs are another crux portion of the needs. This could be recognized as predicted require. The feedback can improve the needs at the next time point.

Contribution 5: Service grouping

Service similarity and service group have been considered. Service similarity is a measurement for a service, which can be used to express the difference degree between a service and one user needs. On the other hand, service group is classified in the initial processing. Services which are in the same group, means they are of the similar function, the similar input, and may be the approximate cost for once using. Service similarity is used to rank a service link in order to output the most proper service to user. Service group would be applied with user feedback to remove approximate services in the link to realize higher response speed to user. Service group can be set as a relative static attribute of services. Similarity can be set as a temporary dynamic attribute of services. Feedback can be got implicitly by some sensors around users or the explicit input of users.

## 6.2 Future Work

Adaptively and dynamically reconfigurable service systems have been widely applied in many studies. This approach is a good method for providing flexible and synergic functions. In our prototype implementation, we made some agents manually for our application scenario. In our proposal, the agents should be made automatically according to a user’s needs. The Meta-mediator was expected to accomplish this task. The key points are the recognition of the feedback from the user in a real-time environment and making a distinction between the user’s needs and the user’s satisfaction. Through this simulated implementation of an FSM-based prototype system, we realized that several other issues should be further considered and challenged.

***Authentication*** **–** Validating the authenticity of something or someone. In an open cloud environment, it is not necessary to perform authentication separately for every instance. The authentication process is hidden by clouds and is transparent to the user, and the authentication is executed only once during the whole usage, in the ideal case. At present, we use only a password for authentication, but we are considering whether there is a better authentication method that could be applied in this FSM-based system. Biometrics, brain rhythms, or consciousness authentication could become choices.

***Authorization*** **–** The power or right to give orders or make decisions. Authorization is not only between the users but also between the users and services or among the services.

During the services mash-up processing, two or more services are synthesized into an integrated service. The data between these services should transfer data safely and quickly, without becoming lost, because authorization is necessary for every service in the system.

***System security and user privacy* –** Each system must consider the security and must protect the user’s privacy. Our proposed FSM is not an exception. We consider users in this FSM system to have their own services spaces (private services spaces), and at the same time, there are many open service spaces that serve different users (public services spaces). When a user comes into his/her own private services space, strict authentication is required, just as he/she goes home and uses a key to open the door. However, in the public services spaces, users share everything that they want to share, and enjoy others’ sharing, just to keep public social manners.

***Self-reliance* –** The final target of an FSM is to completely satisfy the users’ needs in order to create more profit for the providers and to decrease any wasted resources. Self-reliance can help an FSM to achieve these goals from three points of view: users can build their own user model, a system can arrange resources by itself, and resources substitutes can be found easily. In our study, self-reliance ensures that an FSM is independent in the cloud environment.

***Robust flow output*** – A flow of services should not be weak services, but instead must be physically strong enough to provide services to users continually. This outcome depends on the utility of the context information and the maturity of the user model. How to represent a user’s needs in a clearly calculable method remains a large research problem. Our proposal integrates user information with a large amount of context information to represent a user’s needs at a certain point in time.

## Acknowledgments

First of all, the great appreciation should go to my supervisor Professor Qun JIN, who was gently supporting and encouraging me with patience through the whole researching process. Without Prof. Jin's help, this study can not be accepted by the high level journals. Because of my poor communication ability, I can not catch the entire meanings of Prof Jin's instructions. He had to spend more time on me than on other students. I apologized for taking so much valuable time of him. Second, I should say thanks to my senior Roman SHTYKH. He helped me to finish my first conference paper, when the research was just in the beginning at 2008. And of course I should express many thanks to all the members in Networked Information Systems Laboratory.

During these years the e-learning courses in School of Human Sciences, Waseda University (e-school) provide me the chance to experience the contract lecture, which give me more thinking of education processing and applied this study in education scenario appropriately. And special thanks to Japanese government (Monbukagakusho: MEXT) scholarship for economical support over three and a half years.

Final acknowledgement should go out to my parents who helped me to overcome myself even when I have wanted to give up this study. The look of love is always in their eyes expecting me to go back home early. Even when my mother received the critical condition notice issued by a hospital, she concealed that in order to make me focus into study. After the surgical operation beyond the duration of anesthesia, my mother endured the pain without tears. But when I told her that I have received the first journal paper’s notice after waiting for more than one year, she was in tears. Researching needs the wise. I think I am not fit for doing research. It is shame for me to contribute this poor paper to my parents for their long time great patience and enormous sacrifice. Now, I just want to stay with them in the rest of the limited life, even only doing a little simple nursing.

# References

Abowd, G., Dey, A., Brown, P., Davies, N., Smith, M., & Steggles, P. (1999). In Gellersen H. (Ed.), *Towards a better understanding of context and context-awareness* Springer Berlin / Heidelberg. doi:10.1007/3-540-48157-5_29

Ahn, C., & Ramakrishna, R. (2010). *A diversity preserving selection in multiobjective evolutionary algorithms* Springer Netherlands. doi:10.1007/s10489-008-0140-0

Al-Masri, E., & Mahmoud, O. H. (2008). Discovering web services in search engines. *Internet Computing, IEEE, 12*(3), 74-77.

Ardagna, D., & Pernici, B. (2007). Adaptive service composition in flexible processes. *Software Engineering, IEEE Transactions on, 33*(6), 369-384.

B. Worthen. (March 2009). A cloud manifesto controversy. *The Wall Street Journal,* , http://blogs.wsj.com/digits/2009/03/28/a-cloud-manifestocontroversy/.

Bastida, L., Nieto, F. J., & Tola, R. (2008). Context-aware service composition: A methodology and a case study. *Proceedings of the 2nd International Workshop on Systems Development in SOA Environments,* Leipzig, Germany. 19-24. doi:http://doi.acm.org/10.1145/1370916.1370921

Bell, G., Hey, T., & Szalay, A. (2009). Computer science: Beyond the data deluge. *Science, 323*(5919), 1297-1298. doi:10.1126/science.1170411

Bell, G., & Dourish, P. (2007). Yesterday's tomorrows: Notes on ubiquitous computing's dominant vision. *Personal Ubiquitous Comput., 11*(2), 133-143. doi:10.1007/s00779-006-0071-x

Benatallah, B., Perrin, O., Rabhi, F., & Godart, C. (2006). In Zomaya A. (Ed.), *Web service computing: Overview and directions* Springer US. doi:10.1007/0-387-27705-6_17

Bonino da Silva Santos,Luiz Olavo, Vink, P., & van Wijnen, R. P. (2007). A service-oriented middleware for providing context awareness and notification. *Proceedings of the 2007 ACM/IFIP/USENIX International Conference on Middleware Companion*, Newport Beach, California. 6:1-6:2. doi:http://doi.acm.org/10.1145/1377943.1377950

Carmagnola, F., & Cena, F. (2009). User identification for cross-system personalisation. *Information Sciences, 179*(1-2), 16-32. doi:10.1016/j.ins.2008.08.022

Cohen, M. A., & Dennett, D. C. (2011). Consciousness cannot be separated from function. *Trends in Cognitive Sciences, 15*(8), 358-364. doi:10.1016/j.tics.2011.06.008

Coutaz, J., Crowley, J. L., Dobson, S., & Garlan, D. (2005). Context is key. *Commun.ACM, 48*(3), 49-53. doi:http://doi.acm.org/10.1145/1047671.1047703

Crasso, M., Mateos, C., Zunino, A., & Campo, M.EasySOC: Making web service outsourcing easier. *Information Sciences,* (0) doi:10.1016/j.ins.2010.01.013

De Meo, P., Nocera, A., Terracina, G., & Ursino, D. (2011). Recommendation of similar users, resources and social networks in a social internetworking scenario. *Information Sciences, 181*(7), 1285-1305. doi:10.1016/j.ins.2010.12.001

Delgado, M., Ros, M., & Amparo Vila, M. (2009). Correct behavior identification system in a tagged world. *Expert Systems with Applications, 36*(6), 9899-9906. doi:10.1016/j.eswa.2009.01.077

El Hadad, J., Manouvrier, M., & Rukoz, M. (2010). TQoS: Transactional and QoS-aware selection algorithm for automatic web service composition. *Services Computing, IEEE Transactions on, 3*(1), 73-85.

Feng, Y., Cao, J., Sun, Y., Wu, W., Chen, C., & Ma, J. (2010). Reliable and efficient service composition based on smart objects' state information. *Journal of Ambient Intelligence and Humanized Computing,* 1(3), 147-161. doi:10.1007/s12652-010-0017-z

Feng, Y., Teng, T., & Tan, A. (2009). Modelling situation awareness for context-aware decision support. *Expert Systems with Applications, 36*(1), 455-463. doi:10.1016/j.eswa.2007.09.061

Foster, I., Yong Zhao, Raicu, I., & Lu, S. (2008). Cloud computing and grid computing 360-degree compared. *Grid Computing Environments Workshop, 2008. GCE '08,* 1-10.

Foster, I. a. b. (2005). Service-oriented science. *Science, 308*(5723), 814-817. doi:10.1126/science.1110411

Frøkjær, E., & Hornbæk, K. (2008). Metaphors of human thinking for usability inspection and design. *ACM Trans.Comput.-Hum.Interact., 14*(4), 20:1-20:33. doi:http://doi.acm.org/10.1145/1314683.1314688

Grigori, D., Corrales, J. C., & Bouzeghoub, M. (2008). Behavioral matchmaking for service retrieval: Application to conversation protocols. *Information Systems, 33*(7-8), 681-698. doi:10.1016/j.is.2008.02.004

Gu, Q., & Lago, P. (2011). *Guiding the selection of service-oriented software engineering methodologies* Springer London. doi:10.1007/s11761-011-0080-0

Gu, T., Pung, H. K., & Zhang, D. Q. (2005). A service‐oriented middleware for building context‐aware services. *Journal of Network and Computer Applications, 28*(1), 1-18. doi:10.1016/j.jnca.2004.06.002

Hilbert, M., & López, P. (2011). The world's technological capacity to store, communicate, and compute information. *Science, 332*(6025), 60-65. doi:10.1126/science.1200970

Hoareau, C., & Satoh, I. (2009). Modeling and processing information for context-aware computing: A survey. *New Generation Computing, 27*(3), 177-196. doi:10.1007/s00354-009-0060-5

Hwang, R., Liang, B., & Wang, C. (2007). Development of a context-aware messaging system over heterogeneous wireless networks. *Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing,* Honolulu, Hawaii, USA. 457-462. doi:http://doi.acm.org/10.1145/1280940.1281038

IC2002. (n.d.). Retrieved from http://tab.computer.org/tcsc/conf-IC02.htm

Iivari, J., & Iivari, N. (2006). Varieties of user-centeredness. *System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on, , 8* 176a-176a.

James, W. (1984). *Psychology, briefer course*. Cambridge, Mass.: Harvard University Press.

Jiang, G., Hu, B., & Wang, Y. (2010). Agent-based simulation of competitive and collaborative mechanisms for mobile service chains. *Information Sciences, 180*(2), 225-240. doi:10.1016/j.ins.2009.09.014

Kirci Ozorhan, E., Kuban, E. K., & Cicekli, N. K. (2010). Automated composition of web services with the abductive event calculus. *Information Sciences, 180*(19), 3589-3613. doi:10.1016/j.ins.2010.05.040

Klyuev, V., & Yokoyama, A. (2010). *Web query expansion: A strategy utilising japanese WordNet* Future Technology Research Association International.

Koch, C., & Tsuchiya, N. (2007). Attention and consciousness: Two distinct brain processes. *Trends in Cognitive Sciences, 11*(1), 16-22. doi:10.1016/j.tics.2006.10.012

Kouider, S., de Gardelle, V., Sackur, J., & Dupoux, E. (2010). How rich is consciousness? the partial awareness hypothesis. *Trends in Cognitive Sciences, 14*(7), 301-307. doi:10.1016/j.tics.2010.04.006

Kulkarni, D., & Tripathi, A. (2008). Context-aware role-based access control in pervasive computing systems. *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies,* Estes Park, CO, USA. 113-122. doi:http://doi.acm.org/10.1145/1377836.1377854

Lee, K., Lee, K., & Kim, K. (2008). An efficient approach for service retrieval. *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication,* Suwon, Korea. 459-464. doi:http://doi.acm.org/10.1145/1352793.1352889

Liaw, S. (2002). An internet survey for perceptions of computers and the world wide web: Relationship, prediction, and difference. *Computers in Human Behavior, 18*(1), 17-35. doi:10.1016/S0747-5632(01)00032-2

Lopez, B., Innocenti, B., & Busquets, D. (2008). A multiagent system for coordinating ambulances for emergency medical services. *Intelligent Systems, IEEE, 23*(5), 50-57.

Maamar, Z., Santos, P., Wives, L., Badr, Y., Faci, N., & de Oliveira, J. P. M. (2011). Using social networks for web services discovery. *Internet Computing, IEEE, 15*(4), 48-54.

Macedo, D., Dos Santos, A., Nogueira, J. M. S., & Pujolle, G. (2009). A distributed information repository for autonomic context-aware MANETs. *Network and Service Management, IEEE Transactions on, 6*(1), 45-55.

Maia, T. V., & Cleeremans, A. (2005). Consciousness: Converging insights from connectionist modeling and neuroscience. *Trends in Cognitive Sciences, 9*(8), 397-404. doi:10.1016/j.tics.2005.06.016

McCreery, M. P., Kathleen Krach, S., Schrader, P. G., & Boone, R. (2012). Defining the virtual self: Personality, behavior, and the psychology of embodiment. *Computers in Human Behavior, 28*(3), 976-983. doi:10.1016/j.chb.2011.12.019

Minsky, M. (2007). *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind* Simon and Schuster.

Mobedpour, D., & Ding, C. (2011). *User-centered design of a QoS-based web service selection system* Springer London. doi:10.1007/s11761-011-0091-x

Nelson, M. R. (2009). Building an open cloud. *Science, 324*(5935), 1656-1657. doi:10.1126/science.1174225

Nguyen-Vuong, Q., Agoulmine, N., & Ghamri-Doudane, Y. (2008). A user-centric and context-aware solution to interface management and access network selection in heterogeneous wireless environments. *Computer Networks, 52*(18), 3358-3372. doi:10.1016/j.comnet.2008.09.002

Olvera-López, J.,Arturo, Carrasco-Ochoa, J.,Ariel, & Trinidad,José,Francisco,Martínez. (2010). A new fast prototype selection method based on clustering. *Pattern Analysis and Applications, 13*(2), 131-141.

Overbeek, S., Klievink, B., & Janssen, M. (2009). A flexible, event-driven, service-oriented architecture for orchestrating service delivery. *Intelligent Systems, IEEE, 24*(5), 31-41.

Pyshkin, E., & Kuznetsov, A. (2010). *Approaches for web search user interfaces: How to improve the search quality for various types of information* Future Technology Research Association International.

Ratsimor, O., Chakraborty, D., Joshi, A., Finin, T., & Yesha, Y. (2004). Service discovery in agent-based pervasive computing environments. *Mob.Netw.Appl., 9*(6), 679-692. doi:10.1145/1035715.1035728

Rudoy, D., & Zelnik-Manor, L. (2012). *Viewpoint selection for human actions* Springer Netherlands. doi:10.1007/s11263-011-0484-5

Sánchez-Pi, N., Carbó, J., & Molina, J. M. (2012). A knowledge-based system approach for a context-aware system. *Knowledge-Based Systems, 27*(0), 1-17. doi:10.1016/j.knosys.2011.08.017

Sansores, C., & Pavón, J. (2006). Agent-based modeling of social complex systems. In R. Marín, E. Onaindía, A. Bugarín & J. Santos (Eds.), *Current topics in artificial intelligence* (pp. 99-102) Springer Berlin / Heidelberg. doi:10.1007/11881216_11

San-Yih Hwang, Ee-Peng Lim, Chien-Hsiang Lee, & Cheng-Hung Chen. (2008). Dynamic web service selection for reliable web service composition. *Services Computing, IEEE Transactions on, 1*(2), 104-116.

Schall, D., Hong-Linh Truong, & Dustdar, S. (2008). Unifying human and software services in web-scale collaborations. *Internet Computing, IEEE, 12*(3), 62-68.

Schenk, S., Saathoff, C., Staab, S., & Scherp, A. (2009). SemaPlorer—Interactive semantic exploration of data and media based on a federated cloud infrastructure. *Web Semantics: Science, Services and Agents on the World Wide Web, 7*(4), 298-304. doi:10.1016/j.websem.2009.09.006

service. (n.d.a). *Longman dictionary*. Retrieved from http://www.ldoceonline.com/dictionary/service_1

service. (n.d.b). *Oxford dictionary*

. Retrieved from http://oxforddictionaries.com/definition/service

Shamir, O., & Tishby, N. (2010). *Stability and model selection in k-means clustering* Springer Netherlands. doi:10.1007/s10994-010-5177-8

Shtykh, R. Y., & Jin, Q. (2008). Capturing user contexts: Dynamic profiling for information seeking tasks. *Systems and Networks Communications, 2008. ICSNC '08. 3rd International Conference on,* 365-370.

Shtykh, R. Y., Zhu, Y., & Jin, Q. (2009). A context-aware framework for flowable services. *Multimedia and Ubiquitous Engineering, 2009. MUE '09. Third International Conference on*, 251-256.

Shtykh, R., & Jin, Q. (2011). *A human-centric integrated approach to web information search and sharing* Springer Berlin / Heidelberg. doi:10.1186/2192-1962-1-2

Skoutas, D., Sacharidis, D., Simitsis, A., & Sellis, T. (2010). Ranking and clustering web services using multicriteria dominance relationships. *Services Computing, IEEE Transactions on, 3*(3), 163-177.

Sun, W., Zhang, K., Chen, S., Zhang, X., & Liang, H. (2007). Software as a service: An integration perspective. In B. Krämer, K. Lin & P. Narasimhan (Eds.), *Service-oriented computing – ICSOC 2007* (pp. 558-569) Springer Berlin / Heidelberg. doi:10.1007/978-3-540-74974-5_52

Suominen, O., Hyvönen, E., Viljanen, K., & Hukka, E. (2009). HealthFinland—A national semantic publishing network and portal for health information. *Web Semantics: Science, Services and Agents on the World Wide Web, 7*(4), 287-297. doi:10.1016/j.websem.2009.09.003

Tang, J., Zheng, L., & Jin, Z. (2008). Web services composing by multiagent negotiation. *Journal of Systems Science and Complexity, 21*(4), 597-608. doi:10.1007/s11424-008-9137-3

Technical Committee for Services Computing. (n.d.). Retrieved from http://tab.computer.org/tcsc

Thatcher, A., Wretschko, G., & Fridjhon, P. (2008). Online flow experiences, problematic internet use and internet procrastination. *Computers in Human Behavior, 24*(5), 2236-2254. doi:10.1016/j.chb.2007.10.008

Vandekerckhove, M., & Panksepp, J. (2009). The flow of anoetic to noetic and autonoetic consciousness: A vision of unknowing (anoetic) and knowing (noetic) consciousness in the remembrance of things past and imagined futures. *Consciousness and Cognition, 18*(4), 1018-1028. doi:10.1016/j.concog.2009.08.002

Wang, S., Sun, Q., Zou, H., & Yang, F. (2011). Reputation measure approach of web service for service selection. *Software, IET, 5*(5), 466-473.

Weiser, M. (1991). The computer for the 21st century. *Scientific American, 265*(3), 94-104.

Winograd, T. (2001). Architectures for context. *Hum.-Comput.Interact., 16*(2), 401-419. doi:http://dx.doi.org/10.1207/S15327051HCI16234_18

Wohltorf, J., Cissee, R., & Rieger, A. (2005). BerlinTainment: An agent-based context-aware entertainment planning system. *Communications Magazine, IEEE, 43*(6), 102-109.

Xiaohui Gu, & Nahrstedt, K. (2006). Distributed multimedia service composition with statistical QoS assurances. *Multimedia, IEEE Transactions on,* 8(1), 141-151.

Yu, Q., & Bouguettaya, A. (2012). *Multi-attribute optimization in service selection* Springer Netherlands. doi:10.1007/s11280-011-0121-9

Zhu, Y., Shtykh, R. Y., Jin, Q., & Ma, J. (2009). A flowable service model for seamless integration of services. *Advances in Computer Science and Engineering,* 199-204.

Zhu, Y., Shtykh, R. Y., & Jin, Q. (2010). Provision of flowable services in cloud computing environments. *Future Information Technology (FutureTech), 2010 5th International Conference on,* 1-6.

Zhu, Y., Shtykh, R. Y., & Jin, Q (2012). A human-centric framework for context-aware flowable services in cloud computing environments. *Information Sciences,* (0) doi:10.1016/j.ins.2012.01.030