

Full Paper

Ubiquitous Computing Services Discovery and Execution Using a Novel Intelligent Web Services Algorithm

Okkyung Choi¹ and SangYong Han^{2,*}

¹ Yonsei Graduate School of Information, #303-6, New Millennium hall, 134, Shinchon-dong, Seodaemun-gu, Seoul, 120-749, Korea; E-mail: okchoi20@gmail.com

² School of Computer Science and Engineering, Chung-Ang University 221, Heukseok-dong, Dongjak-gu, Seoul 156-756, Korea; E-mail: hansy@cau.ac.kr

* Author to whom correspondence should be addressed. hansy@cau.ac.kr

Received: 8 June 2007 / Accepted: 16 July 2007 / Published: 20 July 2007

Abstract: Ubiquitous Computing makes it possible to determine in real time the location and situations of service requesters in a web service environment as it enables access to computers at any time and in any place. Though research on various aspects of ubiquitous commerce is progressing at enterprises and research centers, both domestically and overseas, analysis of a customer's personal preferences based on semantic web and rule based services using semantics is not currently being conducted. This paper proposes a Ubiquitous Computing Services System that enables a rule based search as well as semantics based search to support the fact that the electronic space and the physical space can be combined into one and the real time search for web services and the construction of efficient web services thus become possible.

Keywords: Ubiquitous Computing Services, Semantic Web Services, Quality of Services, Rule based Services, Intelligent Web Services Algorithm

1. Introduction

The development of various ubiquitous computing service technologies with a new technological system which enables "the computing environment any time, any place and for anything" in which human centered interface application technologies are added on top of conventional web service technology is progressing. This research is aimed at establishing the fundamentals to implement in the future a Ubiquitous Computing Architecture by developing an intelligent algorithm which can

integratedly manage and connectively operate individual applications which are composed of diverse platforms and components, in accordance with semantic web standards and web service standards, using intelligent web service technology. A Ubiquitous Computing Services System was also developed on the basis of a differentiated technology to complement the problem that most web service related research in the past was insufficient in suggesting the general and overall direction as it tended to focus on methodological aspects.

As potential breakthrough solutions to the problems of the current system we have InfoSleuth [1], which has adopted the brokering method and Larks [2], applying the matchmaking method. InfoSleuth [1], which uses broker agents, is an agent based information searching system where the broker takes charge of both discovering and relaying services for efficient web service searching. As a downside, bottlenecks can occur during data transmission since the system is centered on the broker. On the other hand, Larks [2], which is based on the dynamic matchmaking method developed by Carnegie Mellon University's Robotics Institute, uses LARKS language for service advertisements and requests to enable more flexible and efficient matching, but no mention is made of the QoS functions.

The proposed system solves the problematic point of existing UDDI by providing an Extended UDDI Search Module and can also provide a more exact examination of users' demands and reflect the demands in web searches by applying a new QoS Based Matchmaking Algorithm to perform quality evaluations. The proposed system also makes possible provision of a more credible service to users by reflecting the private preferences of respective service demanders through a Rule Based Search in which the services results generated from the Matchmaking engine and the users' preferences are reflected. This system is different from the existing Semantic Web Services Systems in the following aspects:

First, the meaning of the search keyword(s) used within the domain is defined accurately by constructing an ontology server. Additionally, based on the constructed ontology, similar words are searched for and interpreted.

Second, an extended UDDI search module is provided in order to solve problems with the existing UDDI. For this purpose, a matchmaking engine using the QoS based matchmaking algorithm was designed in this research.

Third, rule-based searching is executed in order to reflect the users' preferences. Rule-based searching is a method for converting rules in the user profile registry and each returned Web Services result is mapped into Jess script through XSLT.

Fourth, ranking is executed for more accurate information searches. The newly developed ranking algorithm was applied to rank search results and provide more accurate and reliable information.

Finally, Intelligent Web Services Integrated System was developed along with a step-by-step design method for efficient Web Services search and composition. Its efficiency and accuracy are verified by comparing it with the existing systems.

This paper is organized in the following order. In section 2 there will be an overview of the Semantic Web Services and Ubiquitous Computing Services. This is also analyzed and compared with current systems. In Section 3, the notion of an Intelligent Web Services Algorithm is suggested and in Section 4, the architecture and implementation of an improved Ubiquitous Computing Services System is explained, along with the functions, characteristics of its modules and execution results. It is also

compared with current systems. Conclusions are provided in the final section, along with plans for further studies.

2. Related Work

2.1 Semantic Web Services

Clustering Semantic Web Services, which is the method purported for providing services for the users' convenience in the semantic web environment, is composed of SOAP, WSDL, UDDI and DAML-S. SOAP, used for calling web services, is a quantitative protocol for sending data that is structuralized and standardized using XML. However, it is also a type of message format protocol, which is quite extensive and can actually be applied in many other fields other than that of a quantitative protocol. WSDL, used for web services description, contains information on the interface provided by the web services and the type of data that is to be used through the services. Despite many other related standards such as WSFL or RDF, presently WSDL plays the major role and is highly capable of extensively describing any type of sub-network or messaging protocol. UDDI is a directory service that can be used by an individual or company for registering and searching web services. In other words, UDDI allows one to register the service he/she is to provide on the web and the service user to find the service he/she wants from the registry. DAML-S (Semantic Markup for Web Service) is a key component for implementing Semantic Web Services, and at the same time, the ontology for web services. Thus, within the Semantic Web Services, the DAML supporting agent reads the service descriptions in WSDL format, sends the information to DAML and then DAML connects to the appropriate ontology and the information is provided through the search engine. The advantage of this is that the service consumers can search for and link to providers of the desired web service through any search engine supporting such a framework and in addition personalization related information and negotiation abilities are allowable. In result, the agent is capable of referring to the consumer and provider information and perceiving the conditions in order to automatically settle contracts through negotiation.

2.2 Ubiquitous Computing Services

"Ubiquitous" is a Latin originated word which means 'the god exists everywhere'. Ubiquitous computing or the abbreviated expression 'Ubicom' has come to mean an information and telecommunication technology taken for granted, which became part of our daily lives while we are unaware. Ubiquitous computing has brought about a considerable change in the field of commercial customer service. In the past, web services were composed of portal sites where users could search for necessary information and mainstream price comparing sites, but currently, customized web services which provide information taking into account the users' personal requirements are emerging constantly. Hence, the concept of "ubiquitous commercial deal" appeared, which not only provides the service(s) which fit the context of particular customers at any time, in any place and through any device, but also infers the shopping and purchase information of customers using sensors. In particular "trigger marketing" systems which conduct immediate marketing for customers, are acting as the core infrastructures for ubiquitous commerce which stresses privatization of internet connection tools [11].

2.3 Comparison with the current system

2.3.1 Comparison of the system based on the brokering and Matchmaking methods

Representative semantic web models [1,2,3,4] use the brokering or matchmaking methods in order to provide efficient Web Services. However, because there is no established definition and classification of ontology, queries are not perceived accurately enough and no consideration is made of the Web Services' QoS factors. Furthermore, these Web Services do not provide more specified rankings nor do they execute searches based on the user's preferences. The strong and weak points of the current Web Service models are as indicated in Table 1 below.

Table 1. Strong and Weak Points of Current Models.

Model	Strong Points	Weak Points
[1]	-Capable of flexible and efficient matching using LARKS language for service advertisement and request. -Supports matching for both phrases and queries using five different types of matching filters.	-Semantic method practically impossible.
[2]	-Multi broker agent plays role of meson. -Results provided through cooperation between brokers by applying peer-to-peer system method.	-Does not use standardized service description language.
[3]	-Semantic web system using OWL-S based broker. -Suggests new OWL-S to clarify broker function.	-Bottlenecks may occur when transmitting data.
[4]	-Semantic matching done with DAML-S. -Service grading method using matching algorithm.	-Accurate service search difficult because specific ranking not provided within the same level/grade.

The new system suggested in this study contains the following improved or differentiated aspects. First of all, the matchmaking engine composed of DAML-S/UDDI Translator and Reasoner suggests a more specific methodology to allow semantic searching and ranking using DAML-S, as well as current UDDI searching. In addition, QoS function factors have been added to improve the existing matchmaking algorithm to provide more accurate and reliable Web Services.

2.3.2 Comparison of the system based on Personalized Methods

In [14] the idea of personalized preference is applied to a web environment, the Culture Finder concept is suggested and the meta-data for the respective web pages is written by the Culture Finder. Subsequently, priority order scores are calculated for the respective pages based on the user profile by performing semantic searches. In order to work efficiently, Culture Finder uses five major techniques: a machine learning technique for generating user-profiles from user search behavior and meta-data repository, an efficient semantic search system for semantic web agents, query analysis for representing query and query results, personalized ranking methods to provide suitable search results to users, upper ontology for generating meta-dara. Though Culture Finder conducts personalized semantic web searches and reflecting personal preferences for the determination of priority order by the search, it is

lacking the capability of expressing facts in which various properties are combined because it relies only on an ontology method. To supplement this weak point of [14], it seems that in addition a search with an analysis of various situations by utilizing rule based search methods in which inference rules and service extraction rules are reflected is needed.

In [15] the ontology using rules based on space objects and the situation information model is suggested, which designs a new situation information model that can commonly recognize situations with multiple applications without being subordinated to a specific application for situation recognition technology. It defines the context information according to a context-aware process and designs the knowledge of the domain as well as applications using ontology and rules. The domain spatial ontology and application knowledge are represented using the spatial object model and the rules of expanded ontology, respectively. The expression of abundant spatial ontology represents the context information about distance between objects and adjacent object as well as the location of the object. The proposed context information model which is able to exhibit various spatial contexts and recognizes complex spatial contexts through the existing GIS. Work [15] defines its model as to be one which can recognize complex space situations, raising the possibility that the situation information is defined according to the situation recognition process, and domain knowledge and application knowledge are designed by ontology and rules. However, although in [15] the definition of the architecture of the situation information treating system exists, neither a system operation screen nor a specific utilization example exploiting the architecture are suggested. The research reported in [16], an integrated technology of semantic web and web services into a personalized semantic web information search system, seems to intend to meet the demands of next generation users. However, the superiority of the system in this work, is not assured in that system development situation, real applied examples and research techniques are not referred to, and only the system design method is described.

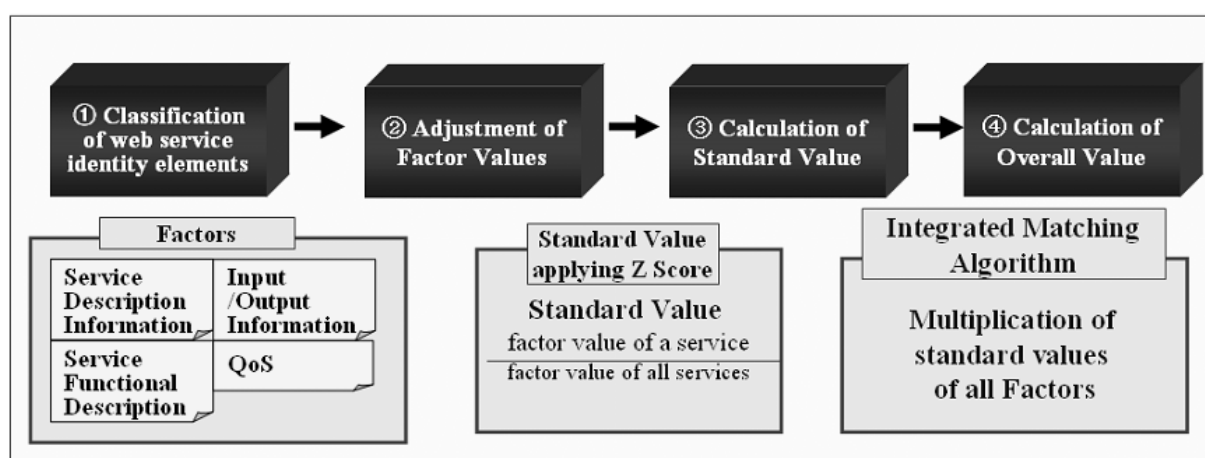
This study suggests a rule based search method in which ontology and rules are combined to supplement the faults of the existing research, designs a search engine using the suggested method and established an operation system for the search engine. By doing this, our study intends to solve the problem that the existing web service models are inclined toward theoretical aspect and the research is conducted mainly on this segment. In addition to it, the suggested method can provide more exact and credible information to the users, as it actively reflects personalized preference in web service results by performing an analysis of various possible situations and searching with inference and service extraction rules.

3. Intelligent Web Services Algorithm

Matchmaking is a process of finding a service provider that satisfies the service requester's requests [7]. Matchmaking is executed based on whether the web service request and web service advertisement match or not [7]. The match between requests and advertisements is determined based on whether the service input and output among the functional description match or not. The matchmaking system must support input and output through the repository and enable service browsing, correction and cancellation [7]. Ultimately the exact correspondence between service request and service advertised and the reflection on QoS elements should be achieved for an efficient Semantic Web Services search. Though some matchmaking systems were proposed in past research, they have some problematic points like those mentioned below.

First, the past research conducted matching by stressing certain partial elements as they failed to treat the identity elements in web services comprehensively. For [4], which conducts a matchmaking algorithm based on DAML-S, an exact web service search is unavailable as it performs web service matching using only input and output information. Second, the QoS matching mechanisms used in the past research have the problem that even high total QoS marks cannot guarantee providing good service in response to demands, as one particular QoS element decides the entire similarity. For example, a service for which the reliability or accessibility is very low but the price is very high could get high marks and selected as the result service in preference to other services. Conclusively, this research proposes solutions for the problematic points of the matchmaking algorithms suggested in the prior research. Integrated Matching Algorithm procedures are as shown in Figure 1 below.

Figure 1. Integrated Matching Algorithm Process.



First the web service identity elements were classified by item and algorithms for the respective items were proposed and next, the integrated matching algorithm was described. That is, this research classified the web service identity elements like in Table 2 for efficient matching.

Table 2. Matching Factors of Web Services.

Message	Factor	Description
Service Description information	Service Name	Describe the introduction of Service
	Service Description	Describe Service Characteristic and Description
Input/Output information	Input/Output Name	Matching between Input/Output Name
	Input/Output Data type	Matching between Input/Output Data type
	Input/Output Constraint	Matching between Input/Output Constraint
Service Functional Description	Service Functional Description	Describe the Service Functional information using (input, output, precondition, effect) Function
QoS	QoS	Availability, Response Time, Throughput, Reliability, Accessibility, Price

Second, the reference values of the respective QoS elements were calculated after the respective QoS elements values were adjusted. Afterwards, all the QoS marks were counted and put into the Integrated Matching Algorithm for application. The algorithm measuring method for the respective identity elements are as shown in Table 3.

Table 3. The Algorithm of Web Services Matching Factors.

Classification	Algorithm
Service Description information	$N[name(s)] = \frac{name(s)}{100}$ $N[desc(s)] = \frac{desc(s)}{100}$
Input/Output information	$Similarity[Input(s)] = \sqrt{sim(i_name)*i_type(s)*i_const(s)}$ $Similarity[Output(s)] = \sqrt{sim(o_name)*o_type(s)*o_const(s)}$
Service Functional Description	$Function(S_{level}) = sim(Input(s), Output(s))$ $0 \leq Function(S_{level}) \leq 1$
QoS	$\sum(QoS) = N(A_u) * N(\frac{Th_s}{T_r}) * N(R_s) * N(Ac_s) * N(P_s)$

The respective identity elements` values were standardized and formalized to a value between 0-1. The entire similarity value which is composed of the sum of the respective identity elements is like one given by the formula (1) below:

$$S_{total} = N[name(s)] + N[desc(s)] + Similarity[Input(s)] + Similarity[Output(s)] + Function(S_{level}) + \sum(QoS) \quad (1)$$

The respective correspondence elements were standardized and normalized to a value between 0 to 1. The entire similarity value resulting from the sum of the respective correspondence elements can be expressed as formula (1) and the detailed description of the algorithms of the mentioned definitions are as given below.

3.1 Service Description information

The most basic informations in a web service query are service name and service description information. For example, when a user of web service make a query with the service name "mobile phone sale" and the service description information is "a mobile phone sale service which permits paying by credit cards", the web service should find the appropriate service for the user by analyzing the similarity between the accessible service name and service description information, which are provided by character strings. This study calculated the similarity between service names and service description information by using TF-IDF (Term Frequency-Inverse Document Frequency Weighting) where service name is titled as $name(s)$ and service description information is titled as $desc(s)$. TF-IDF is an algorithm which searches for representative terms in documents by simultaneously applying inverse document frequency with term frequency, according to the judgment that the solution by existing method which only allows the technique to search for the terms which are used in highest frequency in most numbered documents in finding out document characteristic, has low correspondence between user demand and search result [12,13].

Here, $name(s)$ and $desc(s)$ have the value between 0 to 100. When these values are normalized to the value between 0 to 1, they can be expressed as below.

■ Definition 1

When the service name is $name(s)$ and the Service Description is $desc(s)$, normalizing results of $N[name(s)]$, $N[desc(s)]$ are as indicated below:

$$N[name(s)] = \frac{name(s)}{100} \quad N[desc(s)] = \frac{desc(s)}{100}$$

3.2 Input/Output information

In this study, the correspondence of input and output information is judged by the criteria of Input/Output Name, Input/Output Datatype, Input/Output Constraint and the similarity is measured by TF-IDF. The input and output information values can be expressed as in the formulas below:

■ Definition 2

When the service input information is $Input(s)$ and service output information is $Output(s)$, the similarity of its result is as indicated below:

$$Similarity[Input(s)] = \sqrt{sim(i_name) * i_type(s) * i_const(s)}$$

$$Similarity[Output(s)] = \sqrt{sim(o_name) * o_type(s) * o_const(s)}$$

3.3 Service Functional Description information

Whether or not a correspondence between service query and service advertisement is judged to exist relying on the correspondence of input and output in the functional aspect of service operation. In other words, if the elements from one side can be substituted with the elements from the other side when the two sides of the input of service query and the input of service advertisement are compared, it is regarded that the inputs of the two sides correspond. Viewing the opposition, if the elements from the two sides can be mutually satisfied when the output of service query and the output of service advertisement are compared, the outputs of the two sides are regarded to correspond. When both of the elements of input and output correspond in the result, the web service can treat the service query appropriately and provide a satisfactory service result to the user.

The correspondence of the functional expression information of web service is judged by the correspondence of $Input(s)$ which indicates input information and $Output(s)$, which indicates output information, as the criteria. The values on the functional expression information of web service can be expressed as the formulas below.

■ Definition 3

When the service input information is $Input(s)$ and service output information is $Output(s)$, the similarity of Service Functional Description, $Function(S_{level})$ is as indicated below.

$$Function(S_{level}) = sim(Input(s), Output(s))$$

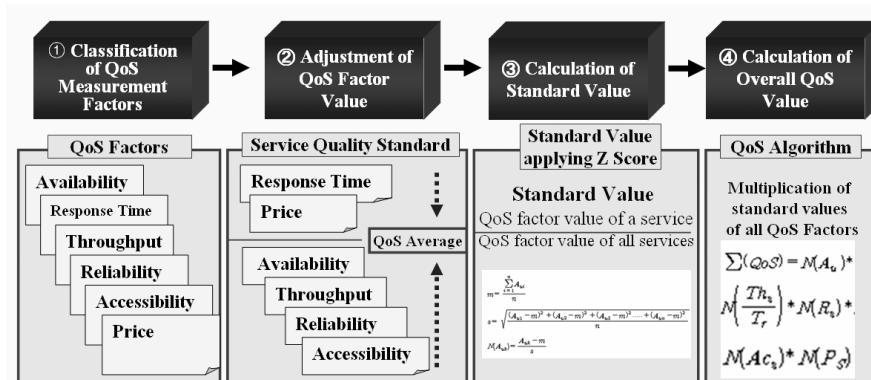
$$0 \leq Function(S_{level}) \leq 1$$

3.4 QoS Measurement Algorithm

For more accurate and reliable web service results, the provided matching method must reflect QoS factors such as time, price, reliability and performance. The current systems such as Larks [2] or Infosleuth [1] do not apply these factors in their matchmaking methods. Though METEOS [3] developed by the University of Georgia does reflect QoS factors in matchmaking, no adjustments are made with regards to each individual QoS factor. Instead, the overall QoS similarity is determined by only one of the factors. Thus, very relevant and meaningful web services could be evaluated lower than their actual ranking.

For such reasons, this study classifies quality evaluation factors and suggests separate algorithms for each factor. After applying algorithms to each QoS measurement factor value, the adjusted value and standard value are calculated. Then, the QoS score is calculated by multiplying all the QoS measurement factors. The QoS measurement procedures are as shown in Figure 2 below.

Figure 2. QoS Measurement Process.



The QoS measurement algorithm based on the sum of standard values (Definition 4) of each QoS factor is as indicated below.

■ Definition 4

The QoS measurement algorithm, the multiplication of the standard values of each QoS factor including Availability (A_{ij}), Response Time (T_r), Throughput (Th_s), Reliability (R_s), Accessibility (R_s) and price (P_s) is as indicated below.

$$\sum(QoS) = N(A_{ij}) * N(\frac{Th_s}{T_r}) * N(R_s) * N(Ac_s) * N(P_s)$$

4. Ubiquitous Computing Services System

As a solution to the problems of the current Web Services, the present study suggests Ubiquitous Computing Services System based on a Matchmaking Engine and a Rule-based Search Engine. Section 4 will describe the overall structure of the suggested system, the functions of each module and the evaluation of its system.

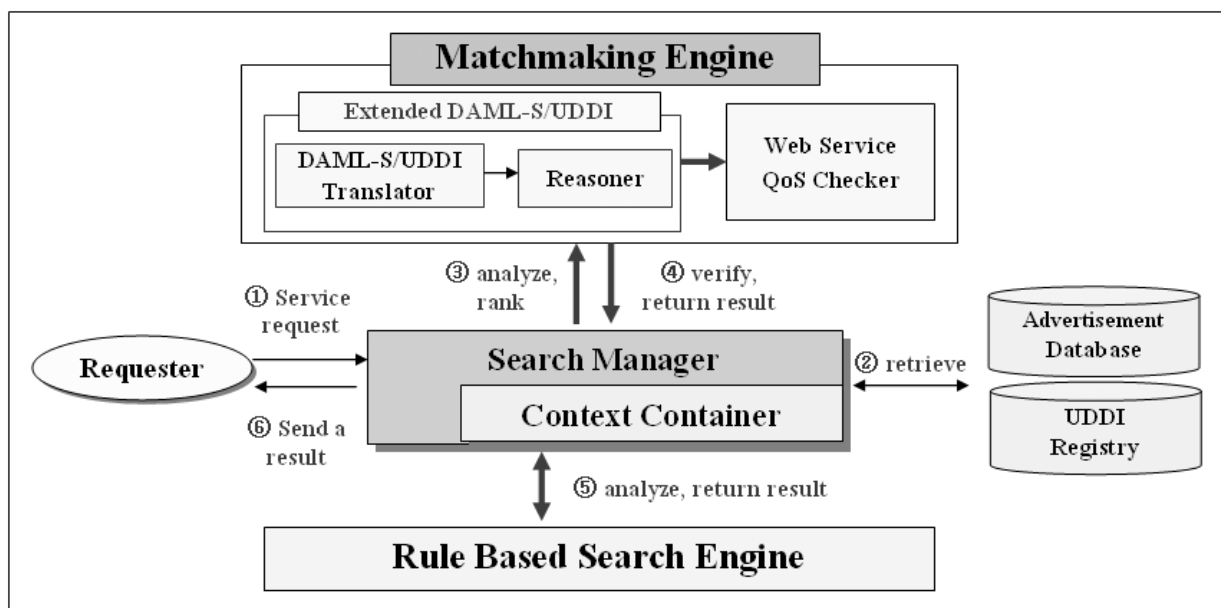
4.1 Scenario

John who is driving in a car, wants to find the most famous pasta restaurant in Manhattan. As the precondition, the restaurant should be located within 10 miles from his current location. With this situation and conditions, the analysis on the user should be conducted through the user information profiles in the user information registry first to satisfy the user's requirement. By analysis of the user information profiles using the privatized agent, the information that John ordinarily orders meals at restaurants within a 60\$~100\$ price range and prefers restaurants which provide various services relative to the food price is gathered. Then, the web service system finds the restaurant best fitting these criteria and provides the information to John by comparing the fact that John prefers restaurants in the 60\$~100\$ price range providing diverse services relative to the price, which is found out through the user profiles, and the service search result analyzed by the Matchmaking Engine. The web service also analyzes the location of John and guides him to a restaurant which is within the radius of 10 miles from his location. As above, it is necessary to find out the information which best fits the request of the service requester by analysis on the service search results and the user information profiles. For this, the technology to determine the user's preferences using a rule based search engine is needed and the service for context awareness and location awareness on the user should be provided simultaneously.

4.2 System Architecture

Matchmaking describes the processes by which parties that are interested in participating in an exchange of economic value are put in contact with potential counterparts [5]. Figure 3 illustrates the process of returning optimal results to a service requester using a Matchmaking and Rule-based Search Engines.

Figure 3. System Architecture.



Step 1: First, the requester submits a request to the Search Manager.

Step 2: The Search Manager determines the current context and location information of the user using the Context Container and request answers from the Advertisement Database and UDDI Registry on the basis of the results. The Context Container provides the information on the context or the environment of a service requester, that is, the context based privatized information. Finally, the Search Manager retrieves all the advertisements from the Advertisement Database and UDDI Registry.

Step 3: Search Manager sends it to DAML-S/UDDI Translator that constructs a UDDI service description. Then Reasoner computes the match level and ranks the results.

Step 4: Search Manager sends results to the Web Service QoS checker using the QoS measurement algorithm to determine if it is verified or not.

Step 5: the system conducts a rule based search which compares the service result produced by the Matchmaking Engine and the user rules taken from the user profile registry and thereupon selects the appropriate information for service.

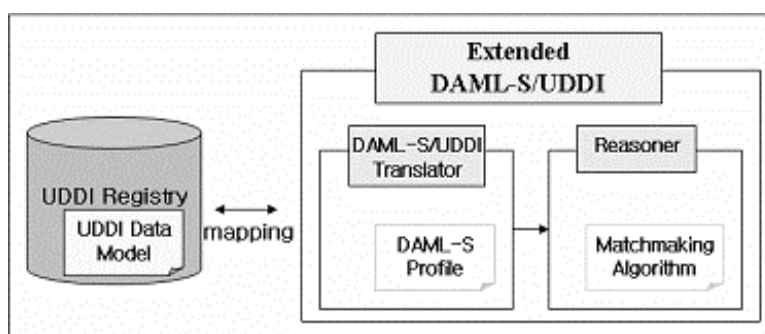
Step 6: Finally, Search Manager sends the final results to the requester.

4.3 Functions of Each Module

4.3.1 Matchmaking Engine

Figure 4 illustrates the system structure of the Matchmaking Engine. DAML-S/UDDI Translator stores DAML-S Profile expression in UDDI Registry.

Figure 4. System Architecture of the Matchmaking Engine.

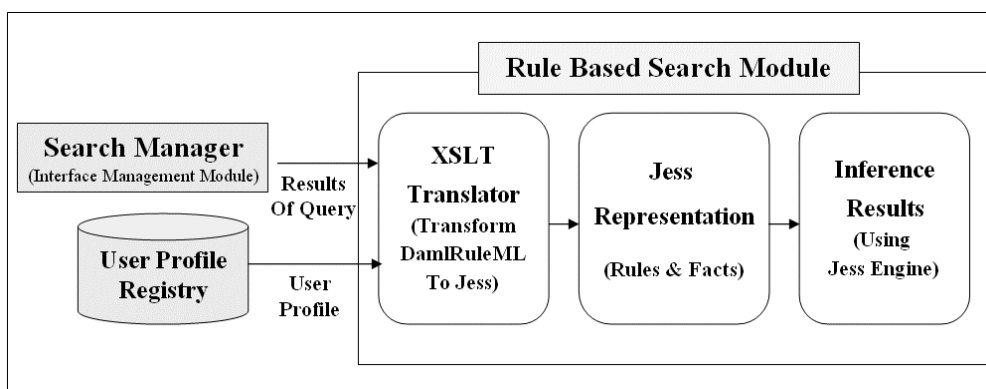


Web service requesters use the UDDI request Port to execute the questionnaire sequence which the UDDI registry provides. However this questionnaire mechanism has the fault that it can not execute the semantic information of web service advertisements and the capability description which are provided by DAML-S Profile. DAML-S/UDDI Translator solves the above problem by executing the extension of UDDI Registry through the mapping of UDDI Registry and DAML-S Profile. In DAML-S/UDDI Translator not only the input and output functions for web services are implemented, but also the web service search which uses the function of pre-condition and post-effect is available, as it supports capability functionality extensively on top of the existing UDDI method. The Reasoner takes the role of efficiently matching web service requests and web service advertisements using the Matchmaking Algorithm (Chapter 2) and provides prioritization.

4.3.2 Rule Based Search Module

The module executes rule based searches to compare produced results with the user's rules drawn from the user profile registry and selects only the appropriate service information. In this research, SweetJess [6] is used for executing rule based searches. SweetJess executes reasoning after expressing rules with DAML+OIL and converting them into JESS's rule language. With SweetJess, the rules indicated in the user information profile and the returned web service results are converted to Jess script through XSLT. Here the Jess engine [10] uses XSLT to convert rules expressed by RuleML to a format that can be used by the existing automatic reasoner and then draws the results through forward inference [6]. Figure 5 is the system architecture of the Rule Based Search Module based applied methodologies.

Figure 5. System Architecture of Rule Based Search Module.



4.4 Implementation Method

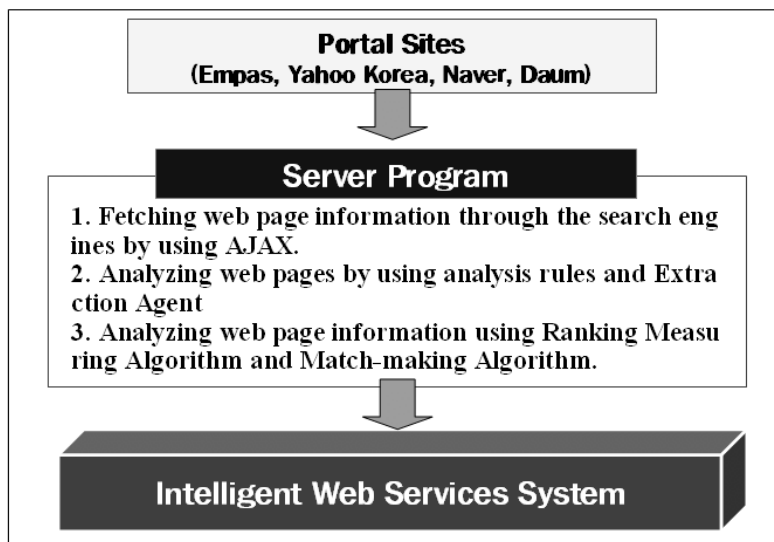
The intelligent web service integrated system implemented in this study is a system that fetches the information based on keywords, keyword explanations, price keywords, QoS information (network speed, credibility, availability, accessibility) and enterprise related items from the databases of the four major domestic portal sites (Naver, Yahoo Korea, Daum, Empas) and displays the information.

This system includes in its internal structure all the items suggested in this study, that is, the Match-Making Algorithm, the Korean character (Hangeul) using search engine and the Ontology Server. With this implementation method, the real data which were being provided at the major portal sites were fetched and the information from the data were processed to fit the intelligent web service concept. In this system, a 'server program' which can contact the portal sites of Overture, Daum, Naver, Yahoo Korea and Empas and fetch keywords, price information and QoS information from the sites, is operating continually and this server contacts web pages accompanying the search engine programs and fetches web pages. Subsequently the system analyzes the fetched web pages and stores the analysis result in the database. In the process of web page analysis the system only takes highly related information using the Extraction Agent suggested in this study.

At this time the system utilizes 'analysis rules' according to the respective portals. The 'analysis rules' according to the respective portal sites are made to be modified in correspondence with the portal sites as the portal sites tend to modify the web pages of search results. Finally the system carries out

analysis on the web pages by applying the the Ranking Measuring Algorithm and the Match-making Algorithm, and subsequently displays the analysis results. Figure 6 shows the implementation method of the system.

Figure 6. System Implementation method.



4.5 Execution Results

Figure 7 shows the service search screen illustrating the system in which web service site information, keyword price information and advertisement sponsor information can be searched, for matchmaking between the service requester and the service provider.

Figure 7. User Interface Screen (English Version 1.0).

Intelligent Web Services System

Service Name: Keyword Shop

Service Description: Search Price & Advertisement of Keyword

Input Name: Keyword

Input DataType: Text

Input Datatype Constraint: 키워드 * 키워드

Site Constraint: NAVER Naver.com Daum.net YAHOO! Yahoo.co.kr Empas.com

Input Name: price

Input DataType: integer

Input Datatype Constraint: 500000~1000000 * 가격

Input Name: frequency

Input DataType: integer

Input Datatype Constraint: 350~ * 빈도수

QoS

System Complete Response:	90	%
Throughput:	80	%
Availability:	75	%
Reliability:	80	%
Accessibility:	90	%

Search

The information in the search here are ones taken from the real data of four internet portal sites in Korea (Empas, Yahoo Korea, Naver, Daum) and processed, mainly on the basis of keywords, price information, demand frequencies and QoS information.

Figure 8. Search Result Screen (English Version 1.0).

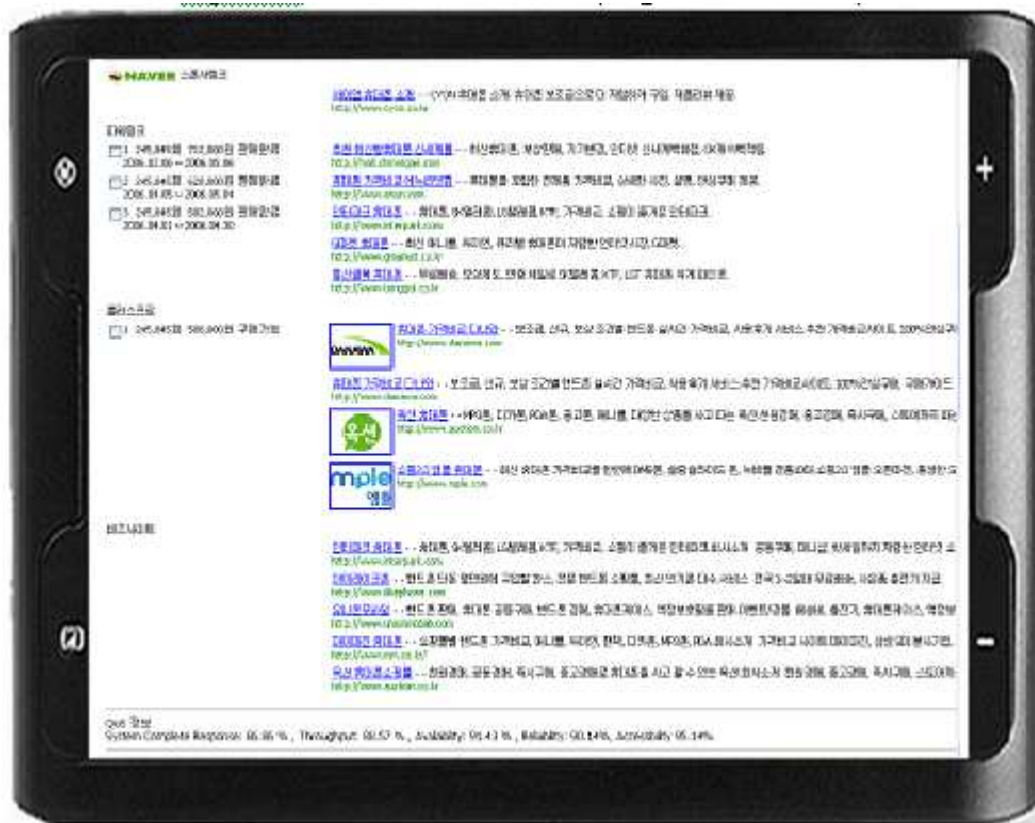


Figure 8 is the result screen obtained from the Ubiquitous Computing Services System after the user input the search keywords and sent the query. Having applied the Rule based Search Algorithm and the QoS Matchmaking Algorithm, the result differentiates from that of the current system.

4.6 An Experimental Evaluation

In this research the experiment was conducted after credible statistics were secured by pre-operation for 7 months. For the analysis on the performance of proposed system, the real data of NateOn, an ordinary internet portal site was taken and used for the comparison. The data described around NateOn is the result directly taken from the site, not via the server program of the proposed system. As experimental measured items, the service explanation information, input and output information, quality consistency rate for the collection of the keywords which users had used for searching, from February 26th, 2006 to Aug 12th, 2006 was used as standard, and the backup data was made into a statistical numeric by a server program developed with UNIX as standardized test marks do not exist as of now. The result of the experiment was as shown in Table 4 and Figure 9, below.

Table 4. Total matching ratio.

Similarity Score	Proposed System				Nate On
	Empas	Yahoo	Naver	Daum	
Service Description information	55.7	74.4	72.6	70.2	39.8
Input/Output information	48.1	18.7	65.5	40.9	39.2
Service Functional Description information	55.8	50.7	74.3	60.0	42.3
QoS information	89.1	89.1	87.2	89.1	82.5

Figure 9. Total matching ratio.

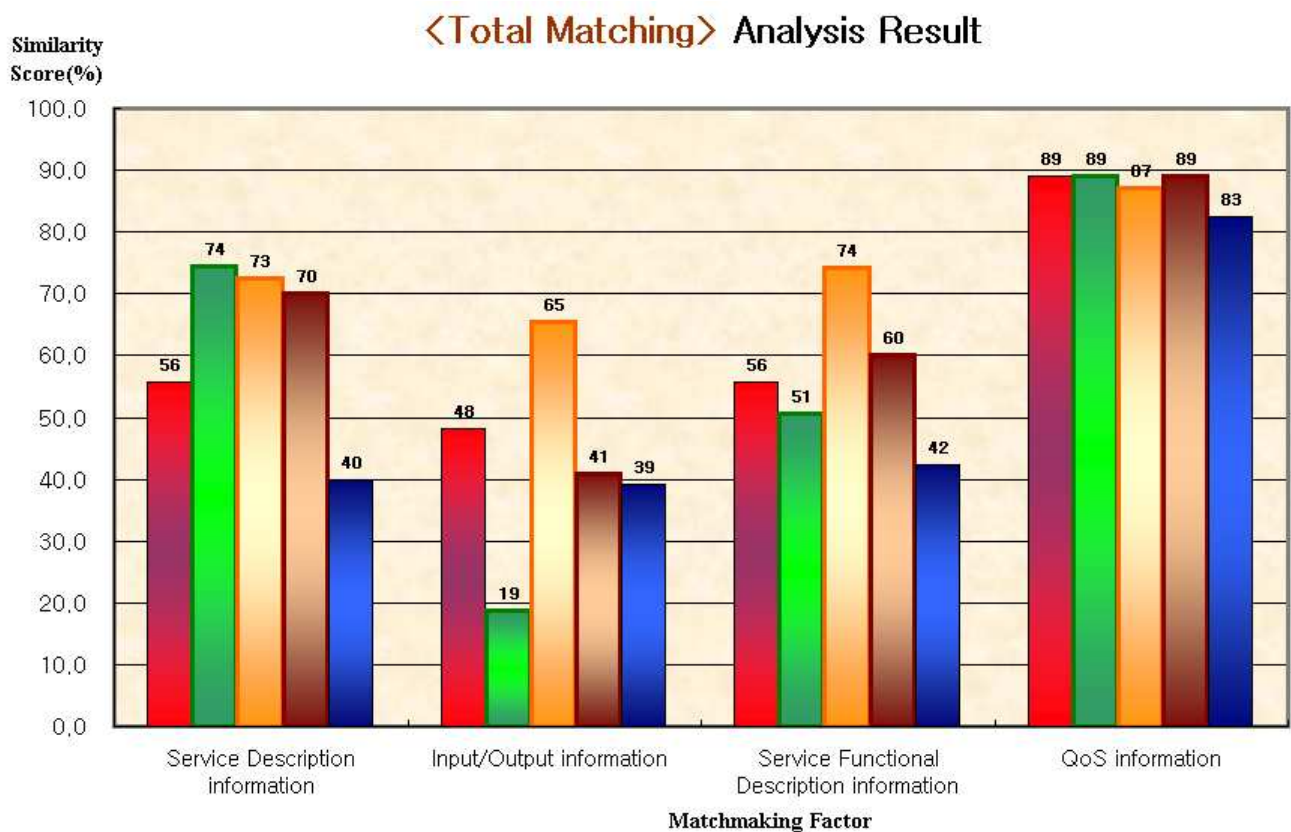


Figure 10. Matching ratio of Service Description Information.

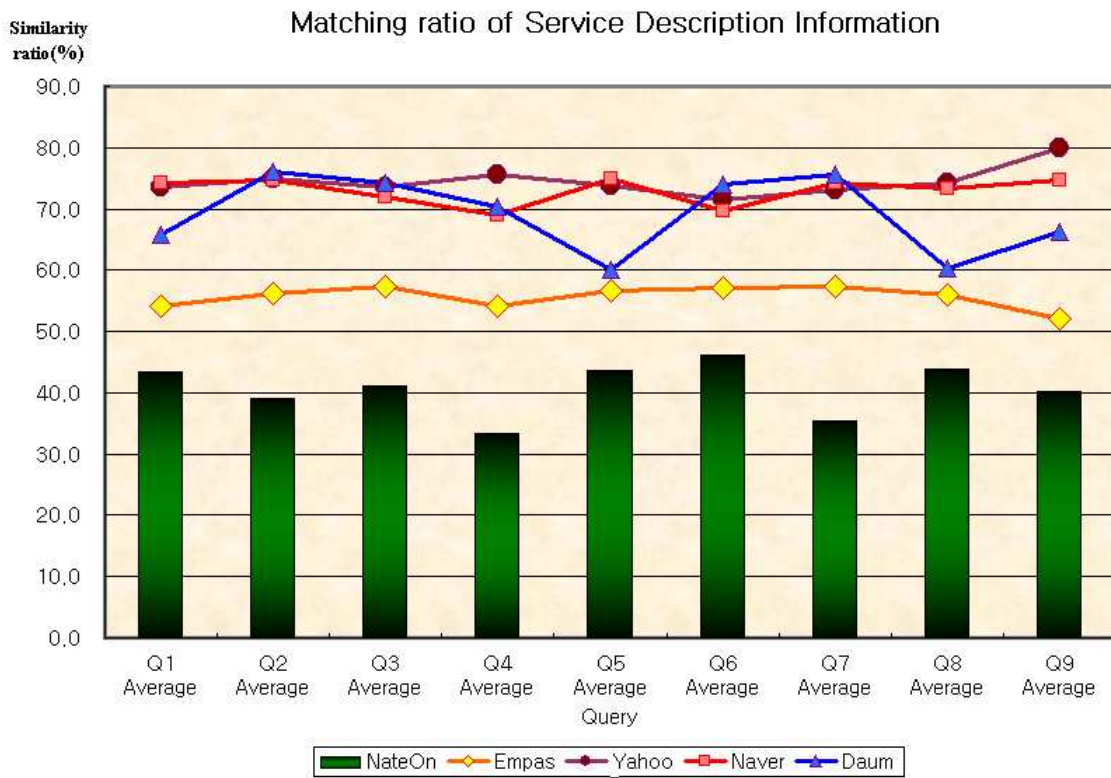


Figure 11. Matching ratio of Service Functional Description.

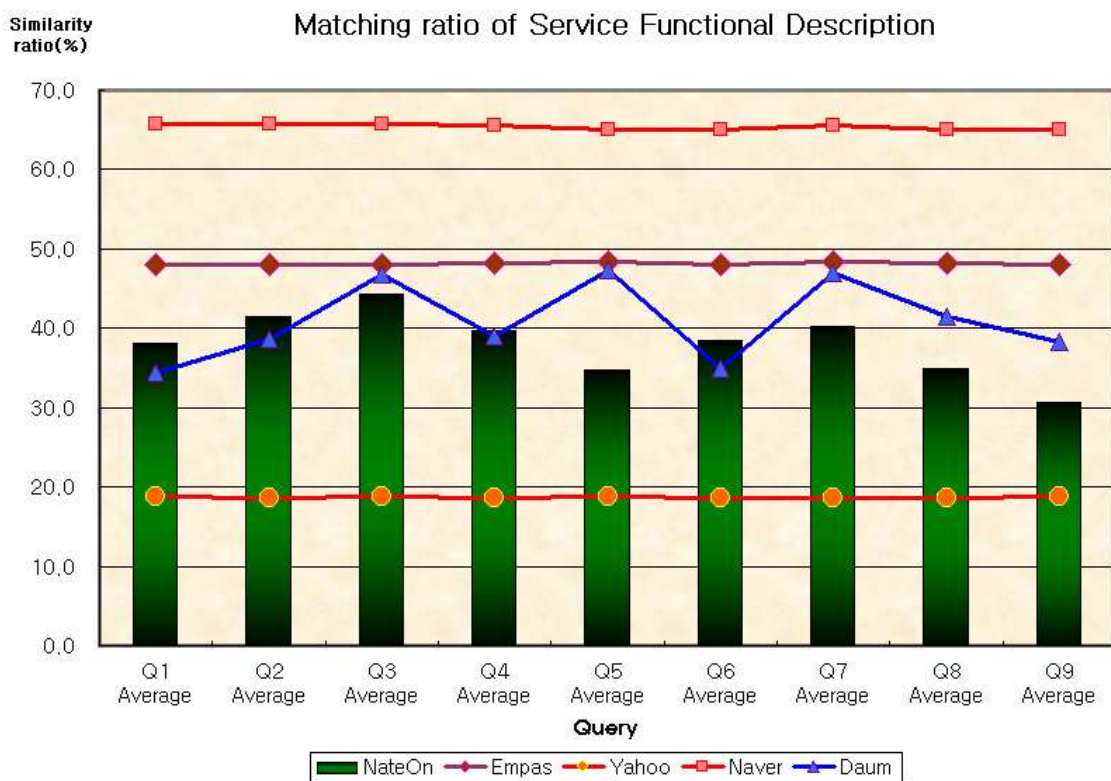


Figure 12. Matching ratio of Input/Output information.

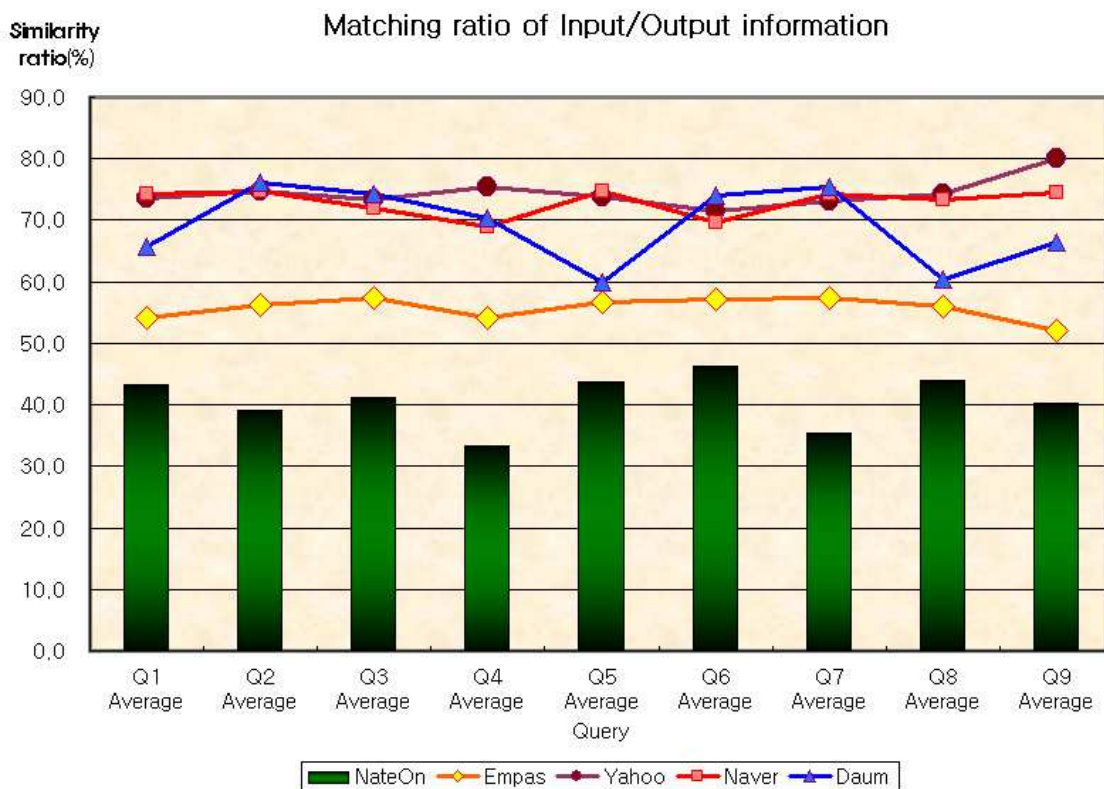
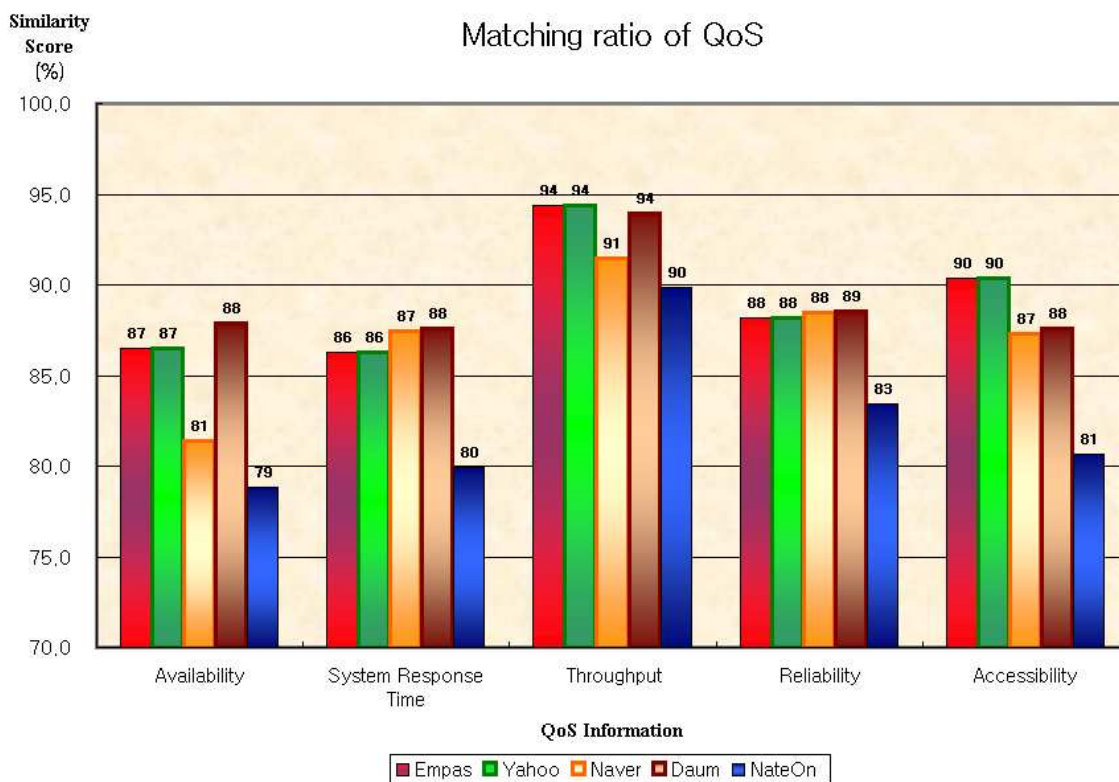


Figure 13. Matching ratio QoS(Quality of Services).



From the experimental result, it is found that the similarity marks of the "Matchmaking Factor" of the portal sites, Empas, Yahoo Korea, Naver, Daum, which is proposed in this research, is enhanced higher compared to the similarity marks of other sites.

5. Conclusion and Future Studies

"Ubiquitous" enables a computing environment which is accessible from any place, at any time while it also enables capturing of the user's location and provision of an optimal service. The ubiquitous computing services system suggested in this paper, in which "context awareness" and "location awareness" is available, enables searching and treating of information that users want, not being bound by the limitations of physical space. This paper also applied semantic web technology to web services and enabled the web to perform ubiquitous computing service composition and execution. The proposed system makes efficient matching between service requesters and providers possible through a system based on QoS. The proposed system also makes examining users' request and reflecting it to the service possible by the Rule Based Search Engine in which users' preferences are reflected. Finally, a quantitative experimental evaluation measuring the rate that service features such as information, input and output information, service function expression information and quality information corresponded to the users' demands based on the keyword collection selected by users, was conducted using a server program which was developed by us. In the result it was identified that the similarity marks of the proposed system was improved by 15-20% compared to that of ordinary internet sites. As a future research task, the adding of screen interface which can mirror demanding items by users and limitation conditions in more detailed divisions and the design of a methodology which can examine the efficiency and exactness of the proposed system objectively is needed.

6. Acknowledgment

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the Chung-Ang University HNRC-ITRC (Home Network Research Center) support program supervised by the IITA-Institute of Information Technology Assessment.

References and Notes

1. Nodine, M.; Bohrer, W.; Ngu, A. Semantic multibroking over dynamic heterogeneous data sources in InfoSleuth. *Proc. 15th Int. Conf. Data Eng. Sydney*; IEEE Comp. Soc. Press, **1999**, 358-365.
2. Sycara, K.; Widoff, S.; Klusch, M.; Lu, J. LARKS:Dynamic Matchmaking Among Heterogeneous Software Agents in cyberspace. *Auton. Agents Multi-Agent Syst.* **2002**, 5, 173-203.
3. Sycara, K.; Paolucci, M.; Soudry, J.; Srinivasan, N. Dynamic Discovery and Coordination of Agent-Based Semantic Web Services. *IEEE Internet Comput.* **2004**, 8, 66-73.
4. Paolucci, M.; Kawamura, T.; Payne, T.; Sycara, K. Semantic matching of web services capabilities, In *Proc. 1st Int. Semant. Web Conf. (ISWC)*, **2002**, 333-347.

5. David, T.; Claudio, B.; Javier, G. C. A Semantic Web Approach to Service Description for Matchmaking of Services. *Proceedings of the International Semantic Web Working Symposium (SWWS)*, **2001**, 19.
6. Grosz, B.; Gandhi, M.; Finin, T.; SweetJess: Translating DamlRuleML to Jess. *International Workshop on Rule Markup Languages for Business Rules on the Semantic Web in conjunction with ISWC2002*, <http://citeseer.ist.psu.edu/grosz02sweetjess.html>, **2002**.
7. Okkyung, C.; Sangyong, H.; Ajith, A. Semantic Matchmaking Services Model for the intelligent Web Services. *Int. Conf. Comput. Sci. Appl.* IEE Press: UK, **2006**, 146-148.
8. Okkyung, C.; SeongHwan, M.; Sangyong, H.; Ajith, A. Intelligent Web Services System based on Matchmaking Algorithm. *WSEAS Trans. Circ. Syst.* **2006**, 5, 1166-1172.
9. Okkyung, C.; Sangyong, H.; Ajith, A. Integration of Semantic Data Using a Novel Web Based Information Query System. *Int. J. Web Serv. Prac.* **2005**, 1, 21-29.
10. Jess, the Rule Engine for the Java™ Platform, <http://www.jessrules.com>, **2005**.
11. Sangdon, C. Ubiquitous Computing Technology and Service Trend. *Proc. Korea Inf. Process. Soc.* **2003**, 10, 5-10.
12. Hayes, J.H.; Dekhtyar, A.; Sundaram, S.K. Advancing Candidate Link Generation for Requirements Tracing: The Study of Methods. *IEEE Trans. SE*, **2003**, 32, 4-19.
13. Jin-Seob, S.; Chang-Hoon, L. 1999. Automatic Classification of Documents Using Word Correlation. *Korea Inf. Process. J. - A* **1999**, 6, 2422-2430.
14. Je-Min, K.; Young-Tack, P. 2006. Personalized Search Service in Semantic Webdata. *Korea Inf. Process. J. - B* **2006**, 13, 533-540.
15. Mi, P.; Keun-Ho, R. Context Information Model using Ontologies and Rules Based on Spatial Object. *Korea Inf. Process. J. - D*, **2006**, 13, 789-795.
16. Haibo, Y.; Mine, T.; Amamiya, M. An architecture for personal semantic Web information retrieval system integrating Web services and Web contents. *ICWS 2005*, **2005**, 329-336.
17. Jesse, J. G. *Ajax: A New Approach to Web Applications*; available at <http://www.adaptivepath.com/publications/essays/archives/000385.php>, **2005**.