Ontological On-Line Service Processing for Integrating Financial Data

Chi-Chun Lo†, Ding-Yuan Cheng††

†Faculty of Institute of Information Management, National Chiao TungUniversity, Hsinchu, Taiwan
†† Ph.D. Student of Institute of Information Management, National Chiao TungUniversity, Hsinchu, Taiwan

Summary
Due to extensive employment of web services on the Internet, issues on intelligent management and selection of web services have attracted more and more attention. The complicated business process in an enterprise can be automated via the composition of a collection of web services. However, the web services to be composited and the sequence of these web services are pre-defined. This makes the applications of the web service in automation of business processes somehow limited.

The aim of this study is to propose an On-Line Service Processing (OLSP) framework which can invoke web service dynamically according to the service content and the expert rules. The proposed OLSP framework combines service-component design and heterogeneous data source integration features. Without modifying and rewriting extra program code, the component-based modeling can achieve the latest and diversity information on-line in an open service architecture. By means of the Service Component Architecture (SCA), a model which conforms with the Service Oriented Architecture (SOA), a web service system is developed to demonstrate the usefulness of the proposed framework. Finally, a case study of the financial health of enterprise is provided to illustrate how the proposed system operates.

Key words:
Service Component Architecture; Enterprise Service Bus; On-Line Service Processing

1. Introduction

In recent years, the current Web is not only a repository for static data but also further offers interfaces to Web-access services, ranging from simple dynamically generated pages for pure information provision to more complex services. Rapid progress in information technology has led to a fast-growing tremendous amount of data stored in databases, data warehouses, or other kinds of data repositories including the World Wide Web. The heterogeneous and diverse data source, how to integrate by Web services, are critical issue focus by industry and academic recently [1].

The growing number of Web services available related to a specified domain area becoming popular and familiar such as financial statement and stock price quote which could assist managers and investors in focusing their efforts and identifying expected opportunities and obstacles. Financial data provide historical information on to evaluate a corporation which could become critical sources for investigating achievements of a company. Another related and similar service is crisis news announcement Web service that informed the financial crisis or warning of one company. Financial statement provide the main source of information for all parties who are interested in the performances of a company; including its managements, creditors, equity investors, and others. They have not only different perspective in viewing financial statements but also diverse source data escaping company profiles.

Therefore, Current standards and methods for service compositions require statically binding the Web Service Description Language (WSDL) interface into compositions in Business Process Execution Language (BPEL). Information collection and acquisition have become easier when all the data inquiry and search becoming Web services, but the effort which is required to retrieve relevant knowledge from databases or information systems has become significantly greater. Service Oriented Architecture (SOA) makes it easier and quicker to tailor a portfolio to fit the desired style and preference of the investor who can customize difference indicators simultaneously and can scan on relative values, e.g., the socks with the highest relative strength, or on fundamental variable, like earning per share (EPS). On the other hand, the passing problem is the choosing of appropriate financial data service or company information selection from tremendous web services.

In this paper, we will describe an On-Line Service Processing (OLSP) framework for integrating SCA model with SOA features to choose the proper web services while one data detected by service was needed to be traced. Our main goals are simplifying the tasks of (a) finding information in XML format via service data methods, and (b) tackling heterogeneous data queries in accordance with relationships between services and attributes. We use a

Manuscript received June 5, 2010
Manuscript revised June 20, 2010

financial healthy case study to illustrate how the proposed system operates.

The remainder of this paper is organized as follows. Related works are given in Section 2. In section 3, we introduce the related domain knowledge of web service that describes OLSP framework. Section 4 addresses the proposed methodology. The main features of this work gives an example of representing and the through an example case study which is described in Section 5. In the last section, conclusion is presented.

2. Related Works

Before providing details of our proposed framework, we would describe four concepts currently being announced and used by developers and briefly review their positive and negative features. In SOA environment, designing a on-line service processing system with the ability of dynamic data-driven has to provide (i) service description and profile, (ii) service grounding, (iii) content of service. Necessary research background and relevant technology include as follows: (1) UDDI service registry, (2) Service Component Architecture (SCA)[2], and (3) Service Data Object (SDO)[3], (4) Enterprise Service Bus (ESB)[4]. We describe as follows.

2.1 UDDI service registry

The aim of UDDI service registry is to locate services automatically based on the functionalities of web services provider. The UDDI service registry is helpful to discover the description and grounding of web service with WSDL in XML format. We build UDDI environment which provides Business Entities, Service Entities, Binding Templates, and tModels to represent the detail of business and its services. Therefore, we choose jUDDI to be our UDDI repository to store the profile of financial web services, as its performance is better than JWSDP showed in previous study.

2.2 Service Component Architecture

Service component architecture (SCA) is a set of specifications which describe a model for building applications and systems using SOA by the composition and deployment of new and existing service component. SCA mainly focuses on the implementation method of component and emphasizes the correlation between the service-oriented components and the existing platform or component, and describes how to implement service-oriented component with the existing technology, platform and even the existing components in the service-oriented environment.

2.3 Service Data Object

Service data object (SDO) defines a unified, language independent programming model for handing data across multiple data resource types which could be relational database, XML file, web service or others. SDO is not only complemented with a data access services which handles data mediation as well as data introspection and data binding, but also offers many capabilities that are very useful for supporting integration with tools and other frameworks, such as: uniform data access across heterogeneous sources, dynamic data API and simple data introspection. SDO addresses the ability to build loosely coupled services and supports for marshalling the domain model into XML.

2.4 Enterprise Service Bus

An enterprise service bus (ESB) is an architecture that exploits web services, intelligent message forwarding, service routing, and transformation data format which provide message exchange patterns [4]. We use an open source system, Tuscany, which is an Enterprise Service Bus (ESB). ESB is not only a web server container but also can forward the message to other web services accordingly which are deployed on the other server

3. Problems

Web services are about the integration of application via Web. Content providers would construct a variety of application and service. After publishing service on UDDI repository by providers, Internet users could search these services according to some keywords. However, a variety of services and tremendous data behind the web server are difficult to integrate or discover useful information from these services.

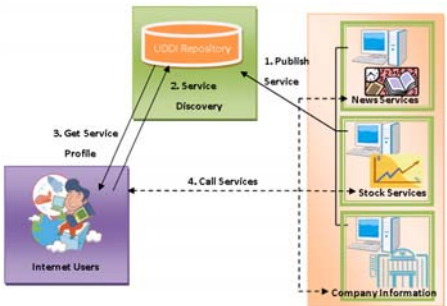

Fig. 1 The flow of Web Services Execution

Fig. 2 shows the overall flow of the web service execution scenario [5]. For example, there have three services: news service, stock price quote service, and company enquiry service. Users could read the latest news through news service daily. The firms would announce the company information, operation status or financial statement to investors. The investors can check the stock price by stock quote service. These services were designed and provided by difference providers. Hence, the most appropriate web service must be selected according to service contexts, and it could trigger the other specified service driven by the return data dynamically. Two of web service development problems are heterogeneous data source integration and forward the message to another appropriate service according to service contexts.

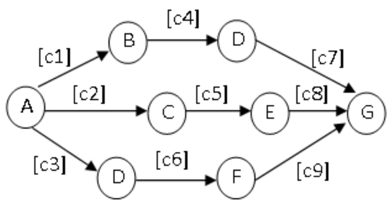

Fig. 2 Example of message routing to different other web service

The fig. 3 shows that an example of message routing from one service A to difference other web service. There exist two major problems in web services: First, is the heterogeneous data source integration. It is critical to provide discovery of related data source and infer their declarative relationship between different services [6]. To collaborate these services to seamlessly work together needs to know the semantic relationship and the interface. Therefore, converting data into exchangeable format and tackling data schema would increase the complexity and difficulty.

Second, is content based message forwarding. According to the basic architecture of Web services which is including service providers, service brokers and service requesters, if a complex service request is submitted, the broker should give an execution plan concerning multiple component services. To improve the effectiveness of the given plan, feedback information and historical service invocations are always necessary to be obtained by the agent.

4. Proposed Methodology

The goal of this research is to provide a framework that allows the users to be aware of their expectations according to their requests and the response of content of web services. From the user point of view, application layer can discover and match these appropriated sources and then be able to meaningfully compose them to serve their purposes. In the portal layer, each portal would provide one specified domain information on the Internet. Behind the portal layer, there are services which can support the functions that connect the difference application system into resource layer. Services are designed for alleviating the mismatch in conceptual models and semantic differences among the resources if they need to be integrated. Business Process Execution Language (BPEL) is used for flow control in previous study. On the other hand, from the machine point of view, there are three layers: data link layer, transformation layer, and physical layer. These not only includes a set of transformation functions which can convert data into a desirable form before their integration, but also supports the capability of machine readable format between different operation system or repository.

4.1 On-Line Service Processing (OLSP) framework

The proposed On-Line Service Processing (OLSP) framework depicts in Fig. 4. which descript an environment for service providers and users to represent their services and requests in a way that can automatically be interpreted by service agent and service router.

According to the different protocols, we can formulate the eight layers which represent the standards from machine point of view. For example, the data exchange can be flat file, xml file, or table in the physical layer. With respect to concept is the point of view from human. For example, the user concept is related with user that would discover and query the information. In agent concept, it enables service agent to find the web services with respect to the content of web service by considering service domain knowledge.

Consequently, service agents have a greater chance of precisely composting their required services. The proposed framework comprises of different layers including application layer, portal layer, service layer, resource layer, data link layer, transformation layer, and physical layer. To meet the above objectives, the proposed framework combines a number of methods such as a service data model, a semantic web mechanism, the service component architecture, and data access methods. The data source provider can easily publish a sharable data source and its associated schema over the internet. A data source, with a unique URL and URI represented as a service name, is treated as a unit. Each view or aspect of a data source can be explicitly declared as a service.

A service could be compared to traditional technology as a data cube in OLAP, or an entity in RDBMS. Each data cube or entity has attributes or properties and their associated values. An atomic service is a data cube (or entity) representing a view of one data source. Two or more atomic services can be logically or physically integrated to form a composite service by giving a clear description of their relationships such that each entity could be represented as a data service. Each attribute or property of the services could be modeled as an operation including input and output messages. An input can have none or more than one of the required parameters. An output message is able to accommodate multiple values.

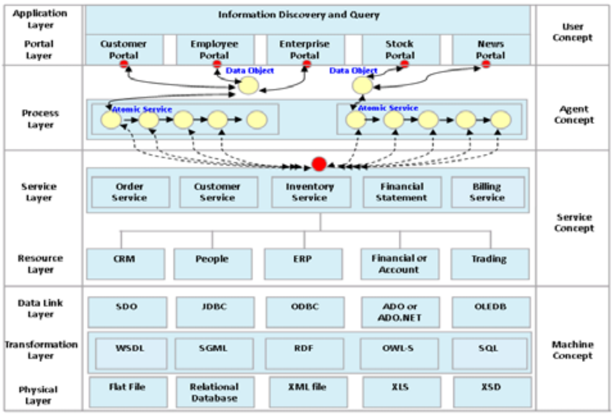

Fig. 3 The proposed On-Line Service Processing framework

The term OLSP covers, as its name suggests, applications that work with transactional or "atomic" service, the individual service contained within a database. OLSP applications usually just retrieve groups of services and present them to the end-user, for example, the list of company financial statement during one time period. These services typically use relational databases, with a fact or data table containing individual transactions linked to meta-data that financial data about company financial statement and stock price details. The OLSP is an acronym, standing for "On-Line Service Processing". The OLSP means many services can on-line serve to different people, and the abilities of OLSP usually involve the terms "composite", "composition & combination" and "on-line response". This, in itself, provides a communication mechanism between users and services, and it does distinguish it from the On-Line Analytical Processing (OLAP) which serves database administrator.

The OLSP presents the end user with information rather than just data. They make it easy for users to find trends or patterns trends between the services quickly, without the need for them to search through input "right" enquiry condition. Typically this analysis is driven by the need to answer business questions such as "How are these companies earning this season in European?" From these foundations, OLSP applications find facts was dynamically driven by data, allowing users to answer questions such as "What are our predicted the earning per share (EPS) according to the latest report in this season?" and "Show me the prediction of crisis company". OLSP application differ from OLAP application in the way that they retrieve data, the way they store data, the way that they analyze data and the way that they present data to the end-user. It is these fundamental differences that allow OLSP applications to answer more sophisticated questions within difference heterogeneous database, even across domain knowledge. We summarize the comparison between OLAP, OLTP, and OLSP on Table 1.

Table 1: The comparison of three types of database technology

Type / Characteristics	OLAP	OLTP	OLSP
Operation	Analyze	Update	Analyze, Update
Level of detail	Aggregate	Detail	Aggregate, Detail
Time	Historical, Current	Current	Historical, Current
Orientation	Attributes	Records	Data Object

4.2 OLSP Module

The five modules of the proposed OLSP architecture addressed the aforementioned problems illustrated in Fig. 5 The SDO data mediator is responsible for reading data from difference sources (database, flat file or proprietary data APIs), and then transforms them into data graph. Data graph that represent SDO data adapter input include all data content (e.g., attribute values). Those objects later serve as input parameter for the SDO query engine and XML generator. The service router can forward and send the query string to one or another web service. Moreover, the result of invoked web service could also be another data source, and then, the difference web service can invoked synchronous at the same time for one purpose or asynchronous for service composition.

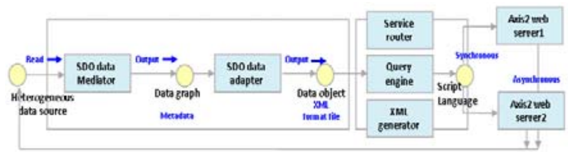

Fig. 4 OLAP modules in the proposed OLSP framework.

The five modules of our OLSP architecture that address the above-listed problems are illustrated in Figure 4. The SDO mediator reads data from difference sources (database, flat file or proprietary data APIs) for conversion into data graph. Data graph that represent SDO data adapter input include all data content (e.g., attribute values). Those objects later serve as input parameter for the SDO query engine and XML generator. The service router can forward and send the query string to one or another web service.

Moreover, the result of invoked web service could also be another data source, and then, the difference web service can invoked synchronous at the same time for one purpose or asynchronous for service composition.

4.2.1 Service data manipulation for heterogeneous data source integration

The SDO is a technology which is a language-independent representation of a data entity that can be passed between services. Within the SCA, the SDO provides common and simple APIs which manipulate data from heterogeneous sources including relational databases, XML data, or web services, etc.

4.2.2 Service component architecture for difference service computing

The SCA is an executable model which is increased productivity for the assembly of services into business solutions. Composition of services which could be easily invoked either synchronously or asynchronously is a key requirement of the SOA. New components which can be invoked a component asynchronously without additional coding could be wired existing applications easily. Here, we develop the stock price quote service, financial statement service, and news service in SCA modeling for the illustrated examples in our prototype system.

4.2.3 Content based router for service routing

For achieving goal of content based routing, we use the Synapse which is an open source project to be the routing server on Apache web server. Synapse provides a simple, lightweight and fully open source SOA infrastructure to assembly, managing composite applications, and message routing supported for binary, text file, and XML format. Synapse has support for HTTP, SOAP, SMTP, JMS, FTP and file system transport for message exchange using XSLS, XPath and XQuery. Hence, we can automatically binding the web services which are related ,and use the scripting language to define the routing rules according the content.

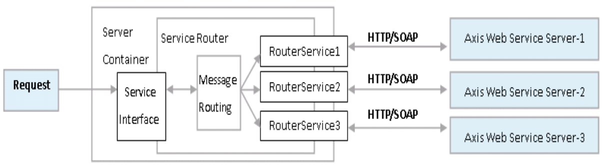

Fig. 5 Content based routing of message forwarding to the other services

5. Case Study and Implementation

5.1 System Architecture

In this section, a financial health is used as an example to demonstrate how the proposed framework works and to give readers the syntax of the proposed on-line service processing. For many investors, assessing the credit of investment targets and the financial healthy is a vital issue before investment. [7] aims at comparing five well-know feature selection methods used in bankruptcy prediction. Suppose that a financial health evaluation could be divided two stages. In the first stage, short-term item would check the current ratio (CR), net operation cycle (NOC), sales growth rate (SGR). In the second stage, long-term item would check the equity to assets ratio (ETA), long-term fund to fixed assets radio (LFTA), times interest earned (TIE). The Fig.6 shows an example of financial experts' decision-making process.

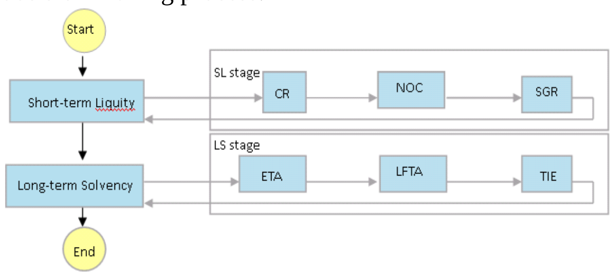

Fig. 6 Example of financial expert's decision-making process

Table 2: Problem categories of evaluating enterprise financial health

Short-term liquidity, SL	
A1	Current ratio, CR
A2	Net Operation Cycle, NOC
A3	Sales growth rate, SGR
Long-term solvency, LS	
B1	Equity to assets ratio, ETA
B2	Long-term fund to fixed assets ratio, LFTA
B3	Times interest earned, TIE

Here, we collect these financial statement data from the Taiwan's market observation post system (MOPS) and develop these services in our prototype system. Each field in data source will have an associated operation with optional input and out messages. In this case, there is a input parameter required which is "getCurrentRatio", and it return "getCurrentRatioResponse" by invoked "CurrentRatio" method. The fig. 7. shows part of a WSDL description of a web service about the financial statement which is liquidity. "Liquidity_ProviderService" is the interface for retrieving information of a company about liquidity data stored in a data source.

```
<definitions
xmlns:soap="http://schemas.xmlsoap.org/wsdl/
soap/" xmlns:tns="http://ws.netlab.edu/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://ws.netlab.edu/"
name="Liquidity_ProviderService">
...
```

Fig. 7 WSDL of Liquidity Service

In the previous study, many researchers identified financial ratios related to financial health evaluation based the literature and additional ratios which financial experts thought important. However, the meaning of these financial statements should be machine readable. Thus, we use the software called Protégé to create the financial ontology (FO).

We use a Java Expert Shell System (JESS) to do the inference and expert suggest. Moreover, we use a tool kit, which is called FuzzyJ, for developing our fuzzy inference engine in the prototype system. In the fuzzy classifier, it contains essential pre-defined knowledge for interpreting and classifying the information residing in web services, e.g., Good (良好), Normal (正常), Dangerous(有危險), Crisis(高度危險).

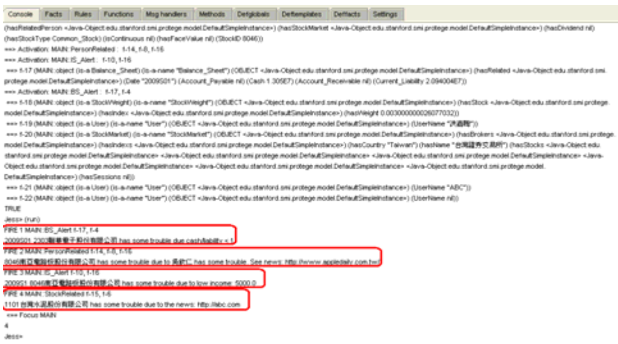

Fig. 8 Protégé cooperate with JESS

The results of data mining can store in database. The expert rule can provide to the JESS inference engine. The FuzzyJ and JESS can integrate within Protégé by using JESSTab.

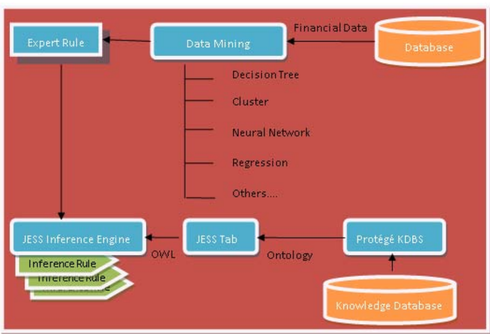

Fig. 9 Kernel module of our inference prototype system

It consists of primitive and composite fuzzy terms, modifier and quantification fuzzy terms, and fuzzy rules. Primitive terms are a set of atomic terms that represent a collection of raw data collecting by service broker from web service which is financial information and crisis news in our example. Composite terms are generated through the combination of primitive terms and fuzzy rules. Composite terms can also be represented in fuzzy rules, whenever heuristic associations between terms are required. The terms are used to model the probabilities of financial distress occurrences. The data schema of the fuzzy rule shows as Table 1.

Table 3: data schema of the fuzzy rule

Column	Data Type	Note
rid	int (10)	Rule id, Primary Key, auto_increment
name	varchar(50)	Rule Name
range_min	decimal(10,0)	Min Value of Fuzzy Set
range_max	decimal(10,0)	Max Value of Fuzzy Set
unit	varchar(20)	Term of Unit
firstTerm	varchar(100)	Reference to Fuzzy Set
secondTerm	varchar(100)	Reference to Fuzzy Set
thirdTerm	varchar(100)	Reference to Fuzzy Set

5.2 System Implementation

The financial data as one kind of web services is provided by various sources on the Internet. We designed the OLSP system can help investor find the latest financial information for their investment plans. Generally, users want to find financial data through investment services for web services. In our designed system, it will fetch the latest financial data from the web services, for examples: Taiwan Stock Exchange (TWSE) and Market Observation Post System (MOPS). The Fig. 9 shows the earn data that can fetch automatically by our service calling and binding.

Fig. 10 Fetch earn data automatically

We are applied the On-Line Service Processing to in our system. This study has developed several JESS rules for inferring relationships between individual companies. The individual rules are utilized as detailed below.

Rule 1: the news contains some key persons

```
(defrule MAIN::PersonRelated
        (object          (is-a          News)
(hasRelated ?obj) (NewsLink ?link))
        (object (is-a User) (OBJECT ?obj)
(UserName ?pName))
        (object         (is-a          Stock)
(isPartOfCompany                      ?name)
(hasRelatedPerson ?obj) (StockID ?sid))
```

```
=> (printout t ?sid ?name " has some trouble
due to " ?pName " has some trouble. See
news: " ?link crlf)
);;
```

Rule 2: news contains some StockID

```
 (defrule MAIN::StockRelated
        (object         (is-a         News)
(hasRelated ?obj) (NewsLink ?link))
        (object (is-a Stock) (OBJECT ?obj)
(isPartOfCompany ?name) (StockID ?sid))
=>      (printout t ?sid ?name " has some
trouble due to the news: " ?link crlf)
);;
```

Rule 3: income is lower than 1000

```
(defrule MAIN::IS_Alert
        (object  (is-a  Income_Statement)
(hasRelated                         ?obj)
(NetIncome ?income&:(< ?income 10000) )
(Date ?date))
        (object (is-a Stock) (OBJECT ?obj)
(isPartOfCompany ?name) (StockID ?sid))
=>
        (printout t ?date " " ?sid ?name "
has  some  trouble  due  to  low  income:
" ?income crlf)
);;
```

Rule 4: current liability is lower than 1

```
(defrule MAIN::BS_Alert
        (object   (is-a    Balance_Sheet)
(hasRelated    ?obj)     (Cash     ?cash)
(Current_Liability ?cl&:(< (/ ?cash ?cl)
1)) (Date ?date))
        (object (is-a Stock) (OBJECT ?obj)
(isPartOfCompany ?name) (StockID ?sid))
=> (printout t ?date " " ?sid ?name " has
some trouble due cash/liability < 1" crlf)
);;
```

Thus, the statement can be altered by a modifier, thereby making the statement a little more imprecise. In other words, the statements associated with quantification and modifier terms are represented in fuzzy rules for the purpose of reasoning. The fuzzy classifier extends the aforementioned rules and their combinations to provide powerful classifications on the data residing in services in order to produce informative declarations.

A fuzzy engine is used to drive the fuzzy classifier to carry out classification and evaluate the content of financial data and news for specified web services. Once the web service of getCurrentRatio is detected the value is higher than other, and the inference engine tell the signal "很危險". The getCurrentRatio service should go on check the other financial statement such as Current ratio (CR) and Net Operation Cycle (NOC). The routing server can automatically binding the correct service according to the service. We can binding the service on-line and fetch the latest data from Internet. These financial data providing by market observation post system and different web sites. The implementation result shows as Figure 10, and Figure 11.

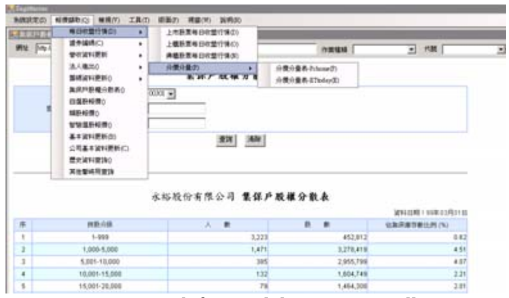

Fig. 11 Fetch financial data automatically

Fig. 12 Report the integrated information in our system

In our prototype system, we use 25-rules that are the expert rule and result of data-mining. We find out the companies that might have financial crisis. And we compared the warning-list announced by official. Fig. 13 illustrates that our system can not only find the warning companies but also find other potential companies.

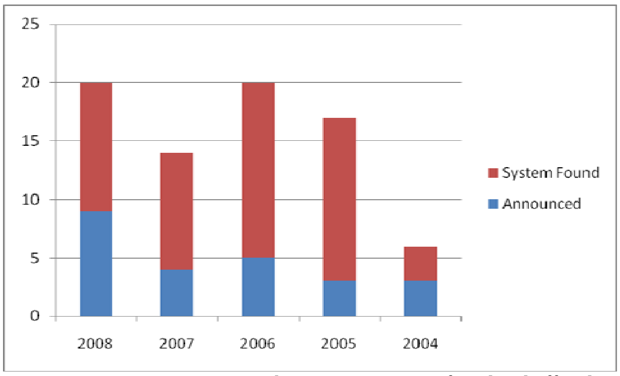

Fig. 13 Warning comparisons between our system found and official announced

5. Conclusion

In this paper, an On Line Service Processing (OLSP) framework is presented to integrate web services with heterogeneous data source and service content routing. A prototype system is further implemented in terms of Apache Tuscany with Java to justify the SCA model conforming with SOA. The study which is using Protégé to construct the domain ontology which is financial ontology (FO) employs OWL as the notation for representing knowledge to be stored in the ontology.

This paper provides an case of the financial health of enterprise in web service environment, combines the technical application of the SDO, SCA, and ESB, the system gives strong auxiliary utility to support users while they have some complex problems. Additionally, the proposed design architecture that can be continually maintained by factual knowledge providers rather than system developer would help establish a complete knowledge recommendation system. In the future work, the expert rule about other domain knowledge should provide a stored and modified mechanism such that the recommendation information can also detect and re-act the real world situation dynamically.

Acknowledgments

The research is supported by the National Science Council of Taiwan under the grant No. NSC 96-2416-H-009-008-MY3.

References

[1] Vernadat F.B., Interoperable enterprise systems: Principles, concepts, and methods, Annual Reviews in Control 31 137–145. (2007)

[2] Service Component Architecture (SCA), OASIS, Nov. 2006 at http://www.osoa.org/display/Main/Service+Component+Architecture+Specifications

[3] Service Data Object (SDO), *Open SOA Collaboration*

[4] Jiang J.C., Gao M., Enterprise Service Bus and an Open Source Implementation, *International Conference on Management Science and Engineering*, p.p.926-930, 2006

[5] Natis, Y. (2003). Service-oriented architecture scenario. Gartner Research Note AV-19-6751.

[6] Soo H.C., S.B. Jeong, Won Y.J., Hyun H.L, and Soo D.K. (2007), A Practical Framework for Dynamic Composition on Enterprise Service Bus, *IEEE International Conference on Services Computing 2007*.

[7] Shiue W., Li S.T., Chen K.J. (2008), "A frame knowledge system for managing financial decision knowledge, *Expert Systems with Applications*", Volume 35, Issue 3.

Chi-Chun Lo received the B.S. degree in Mathematics from National Central University, Chungli, Taiwan, in 1974, the M.S. degree in Computer Science from the Memphis State University, Memphis, TN, in 1978, and the Ph.D. degree in Computer Science from the Polytechnic University, Brooklyn, NY, in 1987. From 1981 to 1986, he was with the AT&T Bell Laboratories, Holmdel, NJ, as Member of Technical Staff. From 1986 to 1990, he worked for the Bell Communications Research as Member of Technical Staff. Since 1990, he has been an Associate Professor in the Institute of Information Management, National Chaio Tung University, Hsinchu City, Taiwan. Currently, he is the Director of the institute. His current major researches include network security, network management, network design algorithm, network architecture, and multimedia system.

Ding-Yuan Cheng received the B.S. degree in Information Management from National Tamkang University, Tam-sui, Taiwan, in 2003, the M.S. degree in Information Management from National ChaioTung University, Hsinchu, Taiwan, in 2004. Now, he is a Ph.D. student in institute of information management, National Chaio Tung University, Hsinchu City, Taiwan. His current major researches include web service, semantic web, Voice over IP, and network management.