Abstract

This project is real time ERP system for farmers. In day to day life the number of agents are involve between farmer and consumers. The aim of our project is to bridge between farmers and clients who want to buy and sell their product. Farmers can sale their products directly to end consumer i.e. Institution, Group Co-operative Societies, Citizen Associations or any other group buyers & also farmer can save lot of money via group buying of Agri products on a fair price. “Krushi Connect” is an initiative to facilitate the producers to sell directly to consumers, thereby eliminating middlemen who prolong the entire process. This project is beneficial for farmers who sell their Agri products in low cost so they can gain more profit. Also we generate graph for market rates of different products. Using mobile application, farmer can register their information and can directly communicate with seller by viewing current market rates of particular from website so he will gain more profit for their efforts. We are providing simple and reliable user interface to farmers to easily interact with our system. Clients can view current updates generated by farmers on website. Automatic notifications are delivered through SMS system to registered farmers. The facility of uploading image of their farm area, expected rates are provided by farmers for consumers.
1. Introduction
Today most of the farmers of villages are getting more advanced technology of farming. As per the survey many farmer uses the internet to check the weather forecasting. There are many faculties given by the government of India to farmer to use advanced technology they also given internet facility in minimum expenses. The growing trend in software architecture is to build platform-independent software components, called Web services that are available in the distributed environment of the Internet. Applications are to be assembled from a set of appropriate Web services and no longer be written manually. Seamless composition of Web services has enormous potential in streamlining business-to-business or enterprise application integration.
The main research focus of Web services is to achieve interoperability between distributed and heterogeneous applications. Therefore, flexible composition of Web services in order to fulfill the requirements of the tasks is one of the most important objectives in this research field. To date, however, many people have considered the service composition to be an ad hoc, time-consuming, and error-prone process involving repetitive low-level programming. To remedy these problems, in recent years, a plethora of research work and products on Web service composition (WSC) problems have appeared. In addition, the Web service research community has hosted competitive programs to solicit algorithms and software’s to discover pertinent Web services and compose them to make value-added functionality. We first show how a WSC problem typically forms by illustrating a small size problem.

2. Survey Review
1.  
   E. Michael Maximilien proposes a multivalent approach that naturally provides a solution to the selection problem. This approach is based on an architecture and programming model in which agents represent applications and services. The agents support considerations of semantics and quality of service (QoS). They interact and share information, in essence creating an ecosystem of collaborative service providers and consumers. Consequently, our approach enables applications to be dynamically configured at runtime in a manner that continually adapts to the preferences of the participants.

2.  
   Munindar P. Singh develops a multiagent framework based on ontology for QoS and a new model of trust. The ontology provides a basis for providers to advertise their offerings, for consumers to express their preferences, and for ratings of services to be gathered and shared. The ratings are essential, because they give an empirical basis for the selection of services. The ratings are quality-specific and are obtained via automatic monitoring or, if appropriate, user input.

3.  
   Sheila A. McIlraith proposes the mark-up of Web services in the DAML family of Semantic Web mark-up languages. This mark-up enables a wide variety of agent technologies for automated Web service discovery, execution, composition, and interoperation. The authors present one such technology for automated Web service composition.

4.  
   Dongwon Lee proposes an AI planning-based framework that enables the automatic composition of Web services, and explores the following issues. First, we formulate the Web service composition problem in terms of AI planning and network optimization problems to investigate its complexity in detail. Second, we analyze publicly available Web service sets using network analysis techniques. Third, we develop a novel Web service benchmark tool called WSBen. Fourth, we develop a novel AI planning-based heuristic Web service composition algorithm named WSPR.
3. Study Of Existing Web Services

In this section, we will observe public Web services and the ICEBE05 test sets and reveal their network features by means of complex network properties. For this purpose, we first define Web service networks in a novel way, as it is then easy to investigate the real-world Web service structure.

![Web service networks](image)

Figure 1: Web service networks.

A set of Web services forms a network (or directed graph). There are different kinds of models to determine nodes and edges of the network depending on the granularity level: Web service level (coarse granularity), operation level, and parameter level (fine granularity) models. Fig. 1 illustrates that three WSDL files can be converted into a bipartite graph structure that consists of three distinct kinds of nodes (parameter, operation, and Web service node) and directed edges between bipartite nodes (operation nodes and parameter nodes). An edge incident from a parameter node to an operation node suggests that the parameter is one of the inputs of the corresponding operation. Conversely, an edge incident from an operation node to a parameter node implies that the parameter is one of the outputs of the corresponding operation. The graph in Fig. 1 has three Web services, labelled WS1, WS2, and WS3. WS1 has two operations, Op11 and Op12. WS2 and WS3 have one operation, Op21 and Op31, respectively. The graph also displays 11 parameters, labelled A through K. According to the node granularity, we can project the upper graph into three different Web service networks.

4. Framework

To evaluate our trust model and hypothesis of self-adjusting trust, we created a framework that augments a typical SOA with agents. The principal idea is to install software agents between service consumers and each service that they consume. These consumer service agents expose the same interface as the service. However, they augment the service interface with agent-specific methods. An example of such a method is setWsPolicy() which allows consumers to communicate their QoS preferences. By exposing the same interface as the service these agents are able to transparently and dynamically select the actual service implementation by considering the service.
consumer's quality needs. The consumer communicates its needs via the augmented agent interface. Service method invocations are done via the service agent who in turn monitors and forwards all calls to the selected service. A high-level view of the architecture; the details and run-time operation are described at length. Briefly an SOA application makes use of some service which has many implementations by different providers. Instead of selecting the implementation directly, the application uses a service agent which exposes the same interface as the service and selects, on the consumer's behalf, the implementation which best matches the consumer's policy.

In addition, the service agents participate in common agencies where they share their quality opinions on the selected service implementations. An agency is simply a rendezvous node on the network where quality opinions are shared and aggregated. The agents share a conceptualization of quality in the form of ontology. The ontology is partitioned into three parts. The upper QoS ontology contains basic definitions for all qualities, including modelling relationships between qualities. The middle QoS ontology extends the upper ontology and defines qualities that are applicable across different domains. Lower QoS ontologies are defined for specific domains by extending qualities in the middle ontology or creating new ones from the upper ontology. Service agent behaviours for quality monitoring can also be attached to the ontology and dynamically bootstrapped in the agents.

Maximilien and Singh give an overview of the upper and middle QoS ontology as well as discussing examples of lower ontology qualities and example usages of the framework. We implemented this architecture in the Web Service Agent Framework (WSAF) and used simulation experiments on simple services as an initial evaluation. The initial results showed that the service agents are able to accurately select service implementations according to the consumer's preferences and adjust the selection as service implementations' quality degrades. In Section 4, we evaluate an approach to enable the service agents to adjust their service selection when a well-behaved service implementation starts degrading its exposed qualities and then again provides good qualities (or vice versa).

Figure 2: Architecture Diagram
According to a simple market research, there are large domestic supermarket chains located around every city in Finland and every corner in Helsinki. However, the selection of commodities in these different supermarkets is almost same, and there are only few Asian food items on the shelf. In addition, only three large scale Asian super markets are located in Helsinki and they are in the same place in the city center. According to information received from friends, a lot of Finnish people like Asian diet. Thus, the goal of this project is to provide a platform for more and more Finnish people to become familiar with Asian food and purchase it online, and to develop an independent online shopping system based on ASP.NET and SQL Server.

5. Conclusion
Thus we are studying to developed the project which is intend to create a hybrid system for product sales for new and old products on the common platform. It creates the authentication login and user account for both seller and purchaser and the system is user friendly so that all the users can communicate in correct order. We are providing simple and reliable user interface to farmers to easily interact with our system. Clients can view current updates generated by farmers on website. Automatic notifications are delivered through SMS system to registered farmers. The facility of uploading image of their farm area, expected rates are provided by farmers for consumers.

6. References