A FRAMEWORK FOR HEALTH CARE BY USING MULTI-AGENT SYSTEM

1V. Revathi, 2Dr. G. Tholkappia Arasu

Address for Correspondence
1Assistant Professor/IT, Adhiyamaan College of Engineering, Hosur, Tamil Nadu, India
2Research Supervisor, Anna University - Chennai, Tamil Nadu, India

ABSTRACT
Several applications are built based on Service Oriented Architecture (SOA). Different atomic web services are incorporated and provide new composite web service. A new composite android application is developed with the help of multi-agents for medical assistance to patients. This application will register the interested patients to its system who are willing and can send alerts to mobile like their next check-up date or updating of medicines list so that they get proper treatments and stay healthy. This application will also help a patient to get help from ambulance service. When they are alone at home by just clicking on a button that will send the patients to address to the hospital and ambulance service. The patients can also view whether their doctor is available for a given date and book their appointments with him/her. They can also view the various treatments provided by the hospital and fees for each treatment and also the doctor fees. A new patient could register with the system through the hospital if they interested. This application will help a patient by correctly maintaining their regular check-up dates, and they may not miss any check-up.

Keyword: Service Oriented Architecture (SOA), Web services, Web service composition and multi-agent.

INTRODUCTION
Service Oriented Architecture (SOA) have different services, which is offered to clients by various service providers in a loosely coupled environment. In general, SOA [1-3] based application is executed as a composite web service that cites external web services. Agents do the processes and service invocations. So the developers can easily recreate the application by altering the process definition without changing the service implementation. Benefits of SOA applications are elasticity and reusability. The multi-agent is needed to generate automatic web service composition [4] and to adopt dynamic changes in the environment. For the web service composition, three agents are used. There are a User Interface (UI) agent, Selection agent and Negotiation agent. UI agent is used to taking the user requirement as input from the client. Selection agent is used to selecting the services from the set of atomic web services according to the user requirement. Negotiation agent is used to compositing the various selected web service into well as any poor people. These agents are also used to adopt dynamic changes in the environment. This framework will help a patient by correctly maintaining their regular check-up dates, and they may not miss any check-up. This framework build as mobile application service, this will minimize the human effort. This proposed work shrinks the turnaround time when compared with the traditional approach.

LITERATUREREVIEW
Rajeswariet al [5], proposed a plan, Deal with various challenges in QoS parameter for the composition of web service because it is difficult to identify. In this paper, they illustrated the related technology by analyzing QoS parameters based on conventional algorithms with composition model and compared the results. However, they consider QoS parameters, but not find agents for automatic web service composition.

Li, J., Li, B. [6], & Zhang, X., proposed an approach for, "The web services compositions build for dynamic environments will need to become more adaptable and adaptive to the unexpected event. This paper defines a way for web services composition which based on agent organization. The functions of three layers, classification of the agent, and agent model and agents design in this framework are introduced in details. It realizes a reliable and flexible web services composition using this framework". Here they consider only two agents for service composition, and they implemented in ontology web-based system.

Satoh, F., & Tokuda, T. [7], proposed, “application based on the Service-Oriented Architecture (SOA) consists of an assembly of services, which is referred to as a composite service. A composite service can be implemented from other composite services, and hence, the application could have a recursive structure. Securing an SOA application is an important non-functional requirement. However, specifying a security policy for a composite service is not easy because the policy should be consistent with the policies of the external services invoked in the composite process. Therefore, this paper proposes a security policy composition mechanism that uses the existing policies of the external services. Their contribution is defining the process-independent policy composition rules and providing a method for semi-automatically creating a security policy of the composite service. Their method supports two approaches to policy composition: top-down and bottom-up. Their study makes it possible to verify the consistency of the policies without increasing a developer’s workload, even if the composite service has a recursive structure". Here also they are not considering agents for automatic web service composition.

Kumar, Sandeep. [8], proposed, “the semantic web service composition process, the evaluation of negotiation-agreements resulting from the negotiation between the service requester and various service providers can be used for the selection of best service provider. The chapter presents a semantic web service composition model for the same purpose. A mathematical model providing multi-attribute negotiation-based service selection using evaluation of negotiation-agreements has also been proposed. A prototype system has been implemented based on the proposed service selection and composition models". Here also they consider only two agents for web service composition and they applied in semantic based web system.

Hwang, S. Y., Lim, E. P., Lee, C. H., & Chen, C. H. [9], this paper studies, " the dynamic Web service selection problem in a failure-prone environment, which aims to determine a subset of Web services to be invoked at runtime. So as to successfully
orchestrate a composite Web service. We observe that both the composite and constituent Web services often constrain the sequences of invoking their operations. Therefore, they proposed using a finite state machine to depict the permitted invocation sequences of Web service functions. We assign each state of execution an increasing reliability to measure the probability that the given state will lead to successful performance in the situation where each Web service may fail with some probability. It shows that the computation of aggregated reliabilities is equivalent to eigenvector computation and adopts the power method to derive efficiently aggregated reliabilities. In orchestrating a composite Web service, they proposed two strategies to select Web services that are likely to complete the execution of a given sequence of functions. A prototype that executes the proposed approach using BPEL for specifying the invocation order of a Web service is built and serves as a test bed for comparing the proposed strategies with other baseline Web service selection policies. In this paper, agents are not involved for automatic web service composition.

**WEB SERVICE**

Web Services are dynamic, modular, self-contained, distributed applications that can be described, published, located, or invoked through the network to create products and processes. These applications can be distributed, local, or Web-based. Web services are built on top of open standards such as Extensible Markup Language (XML), Transmission Control Protocol/Internet Protocol (TCP/IP), Hypertext Transfer Protocol (HTTP), Java, and Hypertext Markup Language (HTML). It is a collection of open protocols and standards used for exchanging data between web applications. Software applications written in different programming languages and running on different platforms can use web services to exchange data over networks like the Internet in a manner similar to interprocess communication on a single computer. This interoperability is due to the use of open standards.

A web service [10] have unique behavioral characteristics such as loosely coupled, coarse-grained, ability to be synchronous and asynchronous, supports Remote Procedure Calls (RPC) and supports document exchange. And also, it plays three major roles,

Service Provider:
It is the source of the web service. The service provider implements the service and creates it accessible on the Internet.

Service Consumer:
It represents any user of the web service. The requestor uses an existing web service by opening a network connection and sending an XML request.

Service Registry:
It represents a logically centralized directory of services. The registry provides a central place where developers can publish new services or find existing ones. Therefore, it serves as a centralized repository for companies and their services.

**WEB SERVICE COMPOSITION**

These days many enterprises bring out their applications on the internet. This kind of applications allows greater availability and efficiency for business people. There are many services present on the web, each one, taken alone and has limited functionality. In several cases, single web service is not sufficient for the client request. In that case, composite services [11-14] are starting to be used as a collection of web services combined to achieve the client request. This composite web service will consider as a simple service, even though it is composed of many web services. Prior to web service composition, atomic services should be discovered first and then selected based on user requirement for composition. In this traditional web service composition, the client got involved to find and select appropriate web services for service composition. Turnaround time is more in this traditional approach.

**MULTI-AGENT SYSTEM**

Agent can be defined as a component that, given a goal could act in the place of a user within its domain knowledge. Agent [15-17] represents the piece of software that possesses the belongings of social ability, autonomy, reactivity, temporal continuity, proactivity, and goal-orientedness. Multi-agent [18-24] system consists of some agents that can interact with each other. In these systems, the agents can coordinate, negotiate, and cooperate with each other. The interagent dependencies between these agents are deal with the process of negotiation. Here the negotiation denotes mutual agreement between the group of agents. In this multi-agent system, agents are acting on behalf of clients, and they have different motivation and goals. Some of the characteristics of multi-agent systems are each agent has capabilities for solving the problem, data are decentralized, and asynchronous computation takes place. This multi-agent system has capable of enhancing the performance of the system based on the following dimensions.

Reliability:
It provides graceful recovery of component failure.

Efficiency:
It develops the concurrency of computation and communication is kept minimal.

Responsiveness:
It can handle anomalies locally due to its modularity.

Reuse:
Different agent teams can reuse the functionally specific agents.

Robustness:
This system can able to tolerate uncertainty

Flexibility:
Agents with various abilities can adaptively organize to solve the problem.

Maintainability:
It is easier to maintain because of its modularity.

**FRAMEWORK FOR COMPOSED WEB SERVICE USING MULTI-AGENT**

In this framework, three agents are used to provide composite web service. This proposed work consists of following modules:

**Designing UI Agent:**
In this module, UI agent fetches the requirements from the client, and priorities the requirements based on the users preference and send back the result to the service selection agent.

**Designing Selection Agent:**
In this module, Selection agent selects the services from the available set of an atomic agent.
Designing Negotiation Agent:
In this module, negotiation agent communicates with selection agent and takes the selected service to create a sophisticated composite service. Here dynamic changes are accepted based on the negotiation principle. So it will not affect the system performance as well as it will minimize the turnaround time.

By using this proposed framework, going to create an application that will help the patient by correctly maintaining their regular check-up dates and they may not miss any check-up. Through this application, any patient can buy medicines by invoking pharmacy web services, and they can do payment by invoking online shipping web services. For all kind of payment again the client can invoke any bank web services. And also, this application can provide health care for all people through our composite web service. (Fig. 1.)

Fig. 1. Overview of Proposed Framework

RESULT
In Fig. 2., we compared our proposed agent-based web service composition with traditional web service composition, based on a turnaround time. By using proposed framework, agent-based composition method takes minimum turnaround time when compared with the traditional method that depicts on the below graph.

Fig. 2. Comparison of Web service composition with traditional method

CONCLUSION AND FUTUREWORK
In this paper, this mobile application is created for helping patients by using the multi-agent system. This application is implemented by using JADE tool. The advantage of JADE tool is to provide low administration, high performance, and interoperability. Monitored the composite web service turnaround time with the help of this JADE tool and the results are shown in the form of the graph. Here the above (Fig. 2.) depicts the comparison of turnaround time.

As a future work, the user can implement this proposed framework as a mobile application in cloud infrastructure. So the cloud-based web service is reliable as well as portable to the client.

REFERENCES
[19] Dr.G.Tholkappia Arasu and Ms. V.Revathi, "Orchestration of web services based on TQoS using...


