ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

An Adaptive Framework for Autonomic Person Identification System using Pervasive Computing

C.Sathya

Assistant Professor, Dept. of CSE, PSNA College of Engineering and Technology, Dindigul, TamilNadu, India.

ABSTRACT: Economic Globalization is transforming the global market into an interdependent and mutual influenced organic entity that requires interoperability to integrate diverse information systems to share knowledge and collaborate among organizations. Web services are the latest and perhaps hottest buzzword in web development world. The advantage of employing web service composition is to create and consume a value added service by composing simple and complex software components which are deployed at different locations in an autonomic manner. The key objective of our work is to design a framework for person identification system using web service architecture through ontology and to provide a breakthrough mechanism in event of identifying person regardless of place and time. We integrate autonomic capabilities like self Configuring and self Discovery using "Ontology". Ontology is knowledge representation of various domains that are used automatically to find out the relevant details about anything. Now-a-days the personal detail about an individual are digitized and is used for various activities in every field. Moreover there are many ways in which a person can disguise his true identity, so it is necessary to implement a powerful approach for person identification system. Our objective is though a person can fake his identity in many ways there are some features that make him unique. Service Oriented Architecture (SOA) decouples reusable functions and offers considerable flexibility in aligning IT functions. Web service is the main component of SOA. With the help of these features, a person can be easily traced in the society.

KEYWORDS: Web services; Ontology; Service Oriented Architecture

I. Introduction

Pervasive computing often synonymously called ubiquitous computing, is an emerging field of research that brings in revolutionary paradigms for computing models in the 21st century. Pervasive solutions enable anytime, anywhere information exchange and access to applications.

- Enable users to access business functionalities
- Enable streamlining and enhancement-heterogeneous enterprise application integration.

Therefore, flexible composition of web services in order to fulfil the requirements of the tasks is one of the most important objectives in the research field.

A. *Web Service:*

Web Service is an emerging paradigm in which loosely coupled software components are published, located and invoked on the web as a part of distributed applications. The main objective is to obtain interoperability between heterogeneous and distributed software components in an ad-hoc manner. [10] A web service is a way to expose some business functionality over the internet using SOAP protocol. According to W3C, a web service is a software system designed to support interoperable machine to machine interaction over a network. It has an interface that is described in a machine- processable format such as WSDL. Other systems interact with the web service in a manner prescribed by its interface using messages, which may be enclosed in a SOAP envelope, or follow a REST approach. These messages are typically conveyed using HTTP, and are normally comprised of XML in conjunction with other web-related standards.

B. *Web Service Composition:*

When a single web service does not satisfy the user need then providers turn to composite web services. Composite web service is the combinations of web services. The advantage of employing web service composition is to create and consume a value added service by composing simple and complex software components which are deployed at different locations in an autonomic manner. [4]

Web service composition lets developers to create application on top of service oriented computing native description, discovery and communication capabilities. Such applications are rapidly deployable and offer developers with reuse possibilities and users can seamlessly access to a variety of simple and complex services. Composition of web services has received much interest to support business-to-business or enterprise application integration. On the one side, the business world has developed a number of XML- based standards to formalize the specification of web services, their own composition and execution. This approach is primarily syntactical. Web service interfaces are like remote procedure call and the interaction protocols are manually written. On the other side, the semantic web community focuses in reasoning about web resources by explicitly declaring their preconditions and effects with terms precisely defined in ontology.

The perspective of system is, everything is implemented as a web service, which can be dynamically discovered and orchestrated using XML messaging in the network. This approach facilitates

- Interoperability by minimizing the requirements for shared understanding
- Enables just-in-time integration
- Reduces complexity by encapsulation and
- Enables interoperability of legacy applications

C. *Service Oriented Architecture:*

SOA is an architectural approach for defining, linking, and integrating reusable business services that have clear boundaries and are self-contained with their own functionalities. Within this type of architecture, we can orchestrate the business services in business processes. A service-oriented architecture (SOA) is the underlying structure supporting communications between services. SOA defines how two computing entities, such as programs, interact in such a way as to enable one entity to perform a unit of work on behalf of another entity. Service interactions are defined using a description language. Each interaction is self-contained and loosely coupled, so that each interaction is independent of any other interaction.[10]

Simple Object Access Protocol (SOAP) based Web services are becoming the most common implementation of SOA. However, there are non-Web services implementations of SOA that provide similar benefits. The protocol independence of SOA means that different consumers can communicate with the service in different ways. Ideally, there should be a management layer between the providers and consumers to ensure complete flexibility regarding implementation protocols.

An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

D. *Web Service Architecture:*

Web services are elements of distributed applications. The applications use the services by composing or putting them together. Architecture for service-based applications has three main parts:

- Service provider
- Service requestor
- Service registry

Providers publish or announce their services on registries, where the requestors find them and invoke them. Web Services architecture requires three fundamental operations

*Publish***:** Service providers publish services to a service broker.
Find: Service requesters find required services using a service broker.
Bind: Service requesters bind to them.

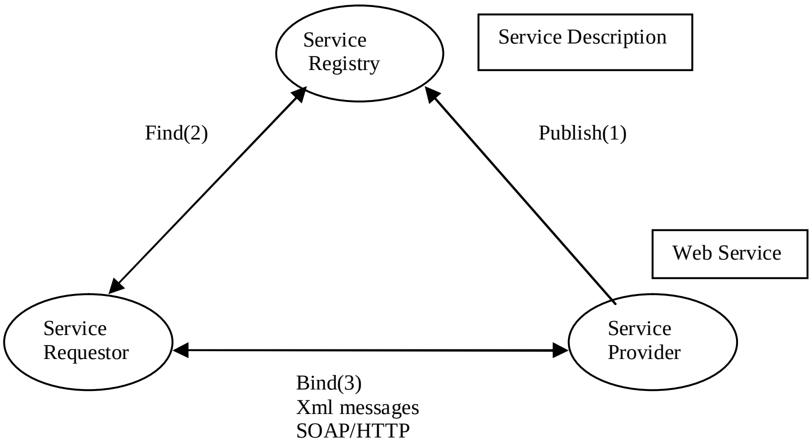

Fig. 1 Web Service Architecture

E. *Web Service Standards:*

Web services are registered and announced using the following services and protocols

1. XML
2. SOAP
3. UDDI
4. WSDL

1) *XML:* Extensible Markup Language (XML) is a mark-up language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML emphasize simplicity, generality, and usability over the Internet.

2) *SOAP:* The Simple Object Access Protocol (SOAP) defines an XML messaging protocol that allows applications to communicate. It defines what goes in as XML message and how to process it, rules for encoding instances of application defined data. SOAP is a simple, lightweight XML-based protocol for exchanging structured and type information on the web. The overall design goal of SOAP is to keep it as easy as possible and to provide a minimum of functionality.

3) *UDDI:* Universal Description, Discovery and Integration is a platform independent, Extensible Markup Language (XML)-based registry by which businesses worldwide can list themselves on the Internet, and a mechanism to register and locate web service application. UDDI has lead to automated discovery and the resulting execution of e-commerce transactions would result in an exceedingly liquid and frictionless environment for business.

II. Related work

Ontology bootstrapping process for web services was described by Aviv Segev et al [1]. Ontological bootstrapping aims at automatically generating concepts and their relations in the given domain and is a promising technique for ontology construction. The WSDL descriptor is evaluated using two methods, namely term frequency/inverse document frequency (TF/IDF) and web context generation. The result of two methods is integrated and evaluated using third method called concept evocation. The drawback here is designing and maintaining ontologies. The infrastructure for E-Government Web Services, which was described by Brahim Medaled et al [2], proposed that the framework for automatically composing e-government services is based on set of rules that check composability of services. Christof Ebert et al [3] explain guidelines for orchestrating web services with BPEL. The business process execution language supports modelling and executing business processes from both the user and system perspectives. Web service application developers can use BPEL to orchestrate service interactions in a global system view to manage individual interactions based on outside events. But proficient use of any BPEL framework requires knowledge of underlying technologies. After programmers write a BPEL program they must package it and deploy it in the selected container. The steps differ among products and require considerable effort to get right the first time. Agent based autonomic service composition was described by Hongxia Tong et al [4] in which a distributed algorithm for web service composition (DPAWSC) is presented. DPAWSC is based on the distributed decision making of autonomous service agents and addresses the distributed nature of web service composition. Here it involves multiple agents and it is difficult achieve coordination. One more drawback is prior knowledge about the services is needed. Philipp Leitner et al [5] presents a message based service framework that supports implementation of SOAs, enabling dynamic invocation of web services. This framework enables the application developers to create service clients which are not coupled to any service provider. The disadvantage here is clients have to find a service that they want to invoke. Steve Vinoski et al [6] explains different web service interaction models in practice. Swaroop Kalasapur et al [7], proposes novel service composition mechanism for pervasive computing. The proposed service composition mechanism models services as directed attributed graphs, maintains a repository of service graphs, and dynamically combines multiple basic services into complex services. Abdaladhem Albreshne et al [8] explain different technologies used for web service orchestration and composition. Dynamic Invocation of web services proposed by Tere G.M et al [9] explains a framework for client to dynamically invoke web services. This framework can increase the use and reliability of web services invocation in a dynamic, heterogeneous environment. But service registry accepts service information passively and it contains too much classifications and information.

III. System Design

A. *Existing System for Person Identification System:*

Current person identification systems are

- Finger Print analysis
- DNA analysis
- Facial Recognition
- Voice Recognition
- Card Identification

Drawback:

- Each system will be having only the particular details of a person but not the entire details. For example voters card, license, ration card

B. *Proposed System:*

There are many situations in which data has to be collected in an Ad-hoc manner. Any acquisition can be done but in our application we are acquitting fingerprint of any individual. Fingerprint image acquisition will be done at the location where the incident occurred and its image is sent as a MMS to our portal.

In our application we have chosen the location as a place where accident has happened. Finger print image acquisition will be done at the location where the incident occurred and its image is sent as a MMS to our portal.

In the server side, request handler automatically process the request and it invokes web services provided in regulating authority. By using ontology, a service tracker tracks the services from service providers through orchestration process. Orchestration describes automated arrangement, coordination and management of complex computer systems and services. As a result we get back the information about the victims and vehicles from service providers who maintain separate XML data store. Ex: Civil Supplies, RTO service, Passport Service, etc. The collected information will be sent as a mail with the help of push technology. The collected information contains personal information, License information, Criminals records (if any) against a person, and information about the vehicle.

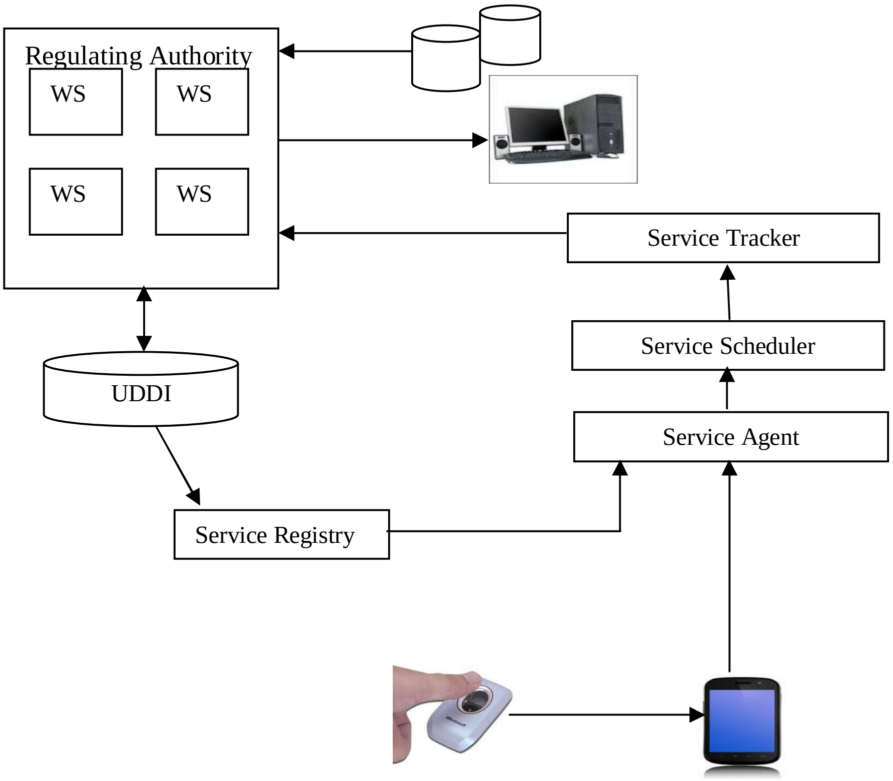

Fig. 2 Proposed System Architecture

C. *Module Description:*

The system is decomposed into four modules

1) *Service Agent:* Service agent acts as a proxy between the service consumer and the service provider. It provides list of services to the clients, which are retrieved from the repository of regulating authority(web services).

2) *Service Scheduler:* The service scheduler process the service requested by the client by accessing the dependency information. The dependency information is stored in stored in a XML data store. If the service requested is a simple service, which involves with a single service provider, it invokes the corresponding web service through a service tracker.

3) *Service Tracker:* The service tracker tracks the services from the service providers through the invoking web services with the help of ontology. It takes care of executing the request by constructing the SOAP message with proper parameters to invoke the requested services. Once the service is invoked it uses the push technology to notify the client about the reaction of event thereby provider's reactive service.

4) *Web Services:* Here we provide several web services through UDDI. The various service providers provide their web services and register it in UDDI. Ontology's are created by the regulating authority. This ontology's can help for dynamic orchestration if simple web services are compared in order to provide complex web services. The following services are designed for our system.

- Civil Supplies Service
- RTO Service
- Insurance Service

- Municipality Service
- Passport Service
- Police Service

D. *Ontology:*

The Semantic Web is a vision for the future of the web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the web. The semantic web will build on XML's ability to define customize tagging schemes and RDF's flexible approach to representing data. The first level above RDF required for semantic web is an ontology language what can formally describe the meaning of terminology used in web documents.

The OWL Web Ontology Web Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interoperability of web content than that supported by XML, RDF and RDF-Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL has three increasingly expressive sublanguage.

1) *OWL Lite:* It supports those users primarily needing a classification hierarchy and simple constraints.
2) *OWL DL:* It supports those users who want the maximum expressiveness while retaining computational completeness and decidability.
3) *OWL Full:* It is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees.

IV. CONCLUSION

The identity of any person is especially needed for various activities in every field such as road accidents, murders, thefts, etc. So it is desirable to combine basic resources in an ad-hoc manner. Thereby our framework identifies the particular person involved in the activities at the current location in pervasive manner.

REFERENCES.

1. Aviv Segev,Quan Z.Sheng, "Bootstraping Ontologies for web services", IEEE transactions on service computing, vol.5. no.1, January-March 2012
2. Brahim, Medjahed, Rezgui, Athman, Mourad, "Infrastructure for E-Government web Services", IEEE Internet Computing, Jan-Feb 2003
3. Christof Ebert, "Orchestrating Web Services with BPEL", IEEE Software, March-April 2008
4. Hongxia Tong, Jian Cao, Shensheng Zhang, and Minglu Li, "A Distributed Algorithm for Web Service Composition Based on Service Agent Model", IEEE Transactions on parallel and Distributed Systems, vol.22,No.12, December 2011
5. Philipp Leither, Florian Rosenberg and Schahram Dustdar "DAIOS Efficient Dynamic web Service Invocation", Distributed Systems Group, Technical University of Vienna
6. Steve Vinoski (May-June 2002), "Web Services Interaction Models-Current Practice", IEEE Internet Computing Paradigm, HPL-2002-130, Internet Encyclopedia.
7. Swaroop Kalasapur, Behrooz A. Shirazi, "Dynamic Service Composition in Pervasive Computing", IEEE Transactions on Parallel and Distributed Systems, vol. 18, No. 7, July 2007
8. Steve Vinoski, "Web Services Interaction Models-Current Practice", IEEE Internet Computing, May-June 2002.
9. Abdaladhem Albreshne, Patrik Furer, Jacques Pasquier, "Web Services Orchestration and composition", IEEE transactions, September 2009
10. Tere G.M, Jadhav B.T and Mudholkar R.R, "Dynamic Invocation of Web Services", Advances in Computational Research, ISSN:0975-3273 and E-ISSN:0975-9085, volume 4, Issue 1,2012,pp78-82