

Adaptive SOA with Interactive Monitoring Techniques and HPS

¹Kasara Venkata Ajay Reddy, ²Maddineni Kiran Chowdary,

³Dr.R.V.Krishnaiah

¹Department of CSE, DRK institute of science and technology Hyderabad, India

²Department of CSE, DRK college of engineering and technology Hyderabad, India

³Principal, Department of CSE, D.R.K Institute of science and technology, Hyderabad, India.

Abstract: *With the advent of distributed technologies and related Integrated Development Environments (IDEs), business collaborations across the world irrespective of the platforms in which applications were built. The underlying technology is Web Services which is based on XML standard and the architecture is SOA (Service Oriented Architecture). Many SOA frameworks came in to existence. However, their primary focus is discovering, composition of services in order to make flexible service oriented distributed enterprise applications. Such systems without human expertise may be incomplete. To overcome this problem, this paper proposes techniques to collaborate services based software with human experts thus forming a knowledge-intensive and people centric web besides sharing and reusing software components in distributed environment. To achieve this, we propose a flexible and adaptive approach which facilitates software services and humans to collaborate using Web Services and its important standards like SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description Discovery Integration). This kind of human centric enterprise web has far reaching implications and it can attract the human experts to participate in rendering their best possible services. Interaction monitoring techniques that are adaptive thus realizing standards based people-centric systems.*

Index Terms: *SOA, Web Services, human expertise provisioning, service adaptation*

I. INTRODUCTION

In the real world, systems are built on different platforms for various organizations. As the globalization led to business collaboration possibilities, there is great demand for large scale business collaborations. Fortunately, distributed technologies such as Web Services and architectural standards to foster them such as SOA are made available. This has led to an increasing interest in the research circles to focus on adaptive techniques for optimizing business collaborations. However, it is a challenging job as service based systems need the human expertise for optimization. The inherent complexity of such systems makes it difficult to adapt to new requirements, partners, expertise levels and so on. To have a seam less integration of software services and human experts web services technology is used as it can support loosely coupled business collaborations including HPS (Human Provided Services). Schall et al., in [8] proposed a framework that supports involvement of human beings in distributed environment. Shahaf and E. Horvitz [9] developed a document translation service that involves software components and also human experts. However, it proved to have inadequate quality. Afterwards, a new phenomena known as HPS came into existence. Human provided services are essential no matter how good the distributed technologies are. In other words, the complex business systems with collaborative business software components are incomplete without provisioning human experts somewhere in the system. Web services and SOA are the defacto standards that help in realizing such systems with human provisioning too. This has led to the development of socio-technical systems instead of pure technical systems having software components to collaborate each other sans humans interfere [5]. Such systems are loosely coupled in terms of technological components and but having social and human aspects that are tightly coupled. The aspects pertaining to technical include develop, deploy, discover and integrate services with even BPEL for flexible compositions. The social aspect in such systems [5] include humans, their relationships, expertise levels, styles of their behavior, attitudes and apprehensions that differ in degrees among human beings.

When distributed applications with HPS are considered, they are mostly dynamic in nature in terms of people with their skills, experiences; incentives in collaborations are ever changing or evolving in other words. This nature of HPS needs really flexible technologies and architectures. These expectations are reached and realized using Web Services and SOA respectively. They are in technical frameworks that allow flexible interactions among service components and human experts in flexible and adaptive fashion thus forming Mixed Service Oriented Systems. For these systems to work properly they depend on three essential fundamentals known as HPA, adaptive interaction models and adaptive service infrastructure. HPA does mean that certain services are provided by human experts. This is required because many applications run perfectly with input

from domain experts. As systems are to be compatible with changing requirements and also adaptive in nature, human interaction and expert input is essential wherever required. At the same time the interactions must be very flexible and adaptive in nature. The ever changing HPS and the level of quality and expertise demand such flexible software compositions provisioning human inputs. In order to realize these two fundamental aspects, the third one is required i.e. adaptive service infrastructure. This kind of infra should include service features and service behavior at runtime. The system with this infra has four phases namely monitoring phase, analysis phase, adapting phase and execution phase. All interactions in the system are annotated for better apprehension and adaptation at runtime. The monitoring phase observes service interactions and events when system is running. The analysis phase is to analyze service behavior, relations and provide ranking in terms of reliability and dependability using certain interaction metrics as discussed by Skopik et al., in [11]. Depending on the calculated ranking metrics, the relations among the clients and services are provided. Moreover the analysis is context – aware that is to make use of annotations at runtime. In the planning or adaptation phase rewards to the services are given and at the same time punishment may be given to services that do not perform as expected. This has impact on future participation of services and operations in the enterprise system. The adaptations and changes are made visible in the execution phase which actually keeps ball moving such that system provides desired services to all the stakeholders involved.

II. PRIOR WORK

Web services and SOA are standards driven. They enable heterogeneous applications to be integrated seamlessly. However, a human provided service that is possible to integrate with enterprise applications where machine to machine communication is realized is still in its inception. This is the area where new standards or protocols are required to come into picture. Towards this end efforts have been put forth by the popular vendors in terms of products such as Bpel4People [1] and WS-HumanTask. Major vendors such as SAP, Oracle, IBM, Adobe, and Active Endpoints collaborated in 2007 and published specifications for WS-HumanTask and BPEL4People in an attempt to standardize the human interaction with BPEL processes. As the name implies WS-HumanTask is the specification that provides proper definitions for human tasks, behavior and operations pertaining to such tasks. The service enabled human tasks thus emerged are controlled by a protocol with interoperability feature introduced. BPEL4People specification provides a well defined extension to WS-BPEL which already existed. First class citizen is WS-BPEL that with the new specification resolves the problem of human interaction with BPEL. Fig. 1 shows integration of human experts with workflow of BPEL.

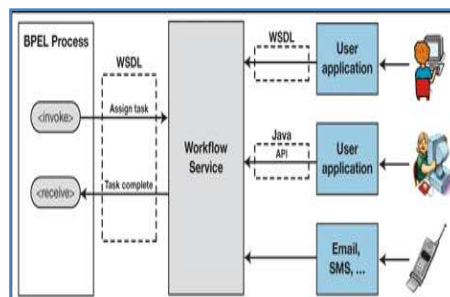


Fig.1 – Human Interaction Integration with BPEL processes [7]

As seen in fig. 1 there is workflow service that makes integration of BPEL process and human expertise in a single framework in distributed environment. In order to realize human interaction with BPEL it has to be aligned with the rich vision of web technology i.e., Web 2.0 that facilitates active contribution from people. HPS framework [8] came into existence to facilitate humans to involve and provide their expert services uniformly. In [9] a system was discussed that involves computer and humans in translation of documents without realizing true SBS (Service Based System). Improved flexibility in distributed systems is made possible with feedbacks from environmental conditions also making the enterprise application to adapt to the conditions. As discussed in [6] MAPE is a mechanism that achieves adaptability to such environmental conditions. This enables the systems to become more and more adaptable and flexible thus facilitating further preventing under-utilization of resources available. Many existing researches in the area of enterprise applications and distributed computing the human element are kept outside the loop. This prevents the rich features that human beings possess can't be realized as part of the framework of the distributed architectures. However, Crowdsourcing techniques as presented in [3] that introduces social element in the computing applications. This makes the enterprise applications a provision to have collaboration with human experts as part of the business processes. MAS (Multi Agent Systems) are introduced in [4] which makes use of social techniques. Technique that addresses the challenges in realization of self-configuration in dynamically is found in [12].

III. SOA WITH PERSONAL PROVISIONING

Human provided services as discussed in [1] can be integrated with SOA. Humans can directly interact with service oriented systems in order to share their expertise with system. However, when human beings are not willing to show their identity directly, avatars can be used in the web environments. The avatars actively participate and act on behalf of humans. Such complex SOA system has to address personal provisioning with various constraints pertaining to given context and the need based human expertise involvement can be realized. The conceptual overview is as shown in fig. 1

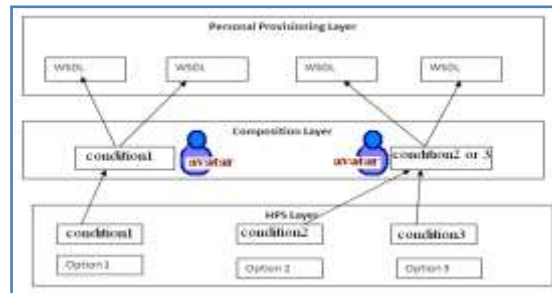


Fig. 2 – SOA with Personal Provisioning

HPS layer enables human provided services to be integrated with other layers. This layer is responsible to provide pre-defined data structures, human services, interfacing standards and other operations. However, the technologies used are very much similar to that of SBS (Software Based Services). This layer provides a web based interface through which human experts can have interaction with other layers. In other words, the HPS layer has applications used by human beings that are having standards driven interaction with rest of the enterprise application. As can be seen in fig. 1, human services are rendered and the avatars represent humans. In order to have this provision a script is required that creates corresponding avatars in the upper layer.

The composition layer lets humans make use of avatars to represent them in the interactions. Humans have to configure the avatars to suit their requirements. Though the avatar characteristics defined by users are static in nature initially, over a period of time they are personalized and reflect the true nature of humans. This is achieved by maintaining personal profiles and updating them periodically thus enabling the avatars to reflect humans social aspects such as service quality, interaction behavior and interests. Thus the initial static characteristics and the dynamic behaviors characterize avatars. The owners' ongoing activities and preferences cause the profile associated with avatar to be changed.

The personal provisioning layer has services who discover avatars based on the functional and non functional properties and related interfaces. A single service instance gets created for each and every client. The kind of communication is one to one. Software services can serve concurrent request while the humans can serve only single requests at any given point of time. However, personalization can be made to the clients with respect to the avatars they use frequently. This layer is responsible to have web services instances provided by various vendors. These services instances are personalized and one new instance may be created for each client. Listing 1 shows sample script used for service deployment.

```
def profile = profilePlgIn.connect(); //current profile
def Language=datatype.create("tconf.xsd","langTypeA") //imports
def Status=datatype.create("tconf.xsd","statType")
def i=callinterceptor.create() //interaction logging
i.hooks=[in:"RECEIVE", out : "PRE_STREAM"] //hooks on streams
i.code={ m -> ... } //logged message

def arrSrv=webservice.build { //interface definition
// create web service
  TranslationService(binding:"doc,lit", namespace="http://...") {
    interceptors+=i //attach interceptor
    docQueue = [:] //current document queue
    repEP = "" //reporting endpoint
    // create translateDoc operation, return doc refId
    translateDoc(docref:String, fromLang:Language,
    toLang:Language, response:int) {
    def refId = genId(docQueue) //new id for doc
```

```

//active pre-processing with checks
if (profile.checkTotalLoad() < LOAD THR)
docQueue.put(refId,docref)
return refId
}
getJobStatus(refId:int, response:Status){
return report(refId)
}
cancelJob(refId:int){
docQueue.remove(refId)
}
setAsyncReportEP(wsdl:String) {
repEP = wsdl
}
}
}
def srv=arrSrv[0] // only one service declared, take it
def h=host.create("somehost:8181") // import back-end host
srv.deployAt(h) // deploy service at remote back-end host

```

Listing 1 – Service deployment script

Listing 1 shows the service description script. Listing 2 given below shows WSDL excerpt related to document translation.

```

<?xml version="1.0" encoding="utf-8"?>
<definitions ...>
<types>
<schema elementFormDefault="qualified"
targetNamespace="http://socsoa.infosys.tuwien.ac.at/">
xmlns="http://www.w3.org/2001/XMLSchema">
<element name="translateDocRequ">
<complexType>
<sequence>
<element name="document" type="xsd:anyURI" />
<element name="fromLang" type="Language" />
<element name="toLang" type="Language" />
</sequence>
</complexType>
</element>
<simpleType name="Language">
<restriction base="xsd:string">
<enumeration value="German" />
<enumeration value="English" />
</restriction>
</simpleType>
...
</schema>
</types>
<message name="translateDocRequest">
<part name="parameters" element="xsd1:translateDocRequ"/>
</message>
...
<portType name="TSPortType">
<operation name="translateDoc">
<input message="tns:translateDocRequest" Action=.../>
<output message="tns:translateDocResponse" Action=.../>
</operation>
<operation name="getJobStatus"> ... </operation>
<operation name="cancelJob"> ... </operation>
...
</portType>
<binding type="tns:TSPortType" name="..."> ... </binding>
<service name="TranslationService"> ... </service>
</definitions>

```

Listing 2 – WSDL Excerpt on Document Translation

IV. Description and Discovery of Services

A distributed environment where components are to adapt new schemes must use configurable and flexible mechanisms for service discovery. The client gets dynamically compiled metadata before each request with respect to non functional and functional properties in order to update its view. The realization of this is done by using WSDL file of HPS. Discovery of a service is required for two reasons. First of discovery is required in order to find an avatar. Functional properties are used to discover avatar. Then they are clubbed with metrics pertaining to capability metrics that reveal how satisfied the clients of avatar. The search queries are carried out using SPARQL. The structures on which queries are performed are FOAF structures. Responsibility and reliability of avatars determine the project work.

V. IMPLEMENTATION OF SYSTEM AND ITS ADAPTION

The implementation of the proposed system is done to support service provisioning and also the expertise provided by human beings. The architectural framework is provided in fig. 3.

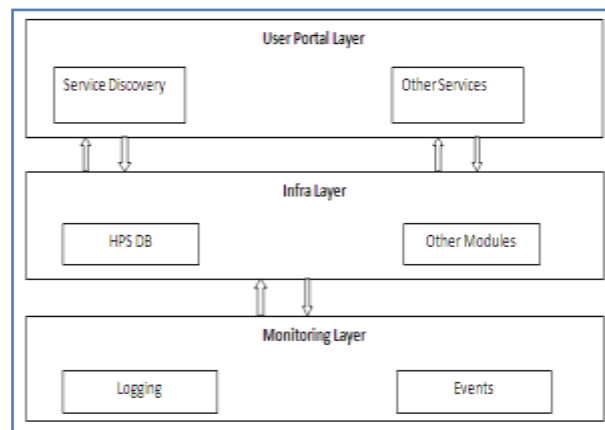


Fig. 3 – Architectural overview

As seen in fig. 3, there are three layers in the architecture. The uppermost layer has service discovery and other required services. Users can discover required services using service discovery module. All such interactions are logged for future retrieval and revisions. The infrastructure layer in the middle contains database of human provided services, other modules and hosting environment. The adaptation module verifies rules and takes necessary steps. The steps include adaptation of HPS and the registry pertaining to HPS. The monitoring layer is responsible to identify and process SOAP interactions and changes related to environment. This layer is responsible to trigger composite events and adaptation.

VI. Adaptation Strategies

Timely adaption of require services is an important activity. In the mixed service oriented system, it is essential. The study in this paper is on client – driven interventions, and provider driven interventions which are the reasons that tells the need for service adaptations. Client driven interactions are used to ensure that the unreliable services are not supported in client. This enables the system to replace unreliable services, services that take more time and miss deadlines, is replaced by services that are reliable. Service owners initiate provider driven interactions. This will protect them from clients with malicious intentions. For example, the requests coming from client and cause attacks like replay, denial of attack and so on. Adaptations could be intrusive and distinct. For instance they are behavior adaptations and interface adaptations. Interface adoptions indicate that certain service is updated or deleted from the distributed environment. Such adaptations may be made by system based on the ongoing experience of runtime operations. With respect to behavior adaptations, they are more intrusive and they are capable of changing the behavior of avatars. In [3] it is described how to perform modifications at run time without cause a service going offline. In listing1, a typical SOAP interaction example is given.

```

<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:types="http://socsoa.infosys.tuwien.ac.at/Type"
  xmlns:ts="http://socsoa.infosys.tuwien.ac.at/TS">
  
```

```

<soap:Header>
<types:timesent value="2010-09-01T15:13:21"/> <types:deadline value="2010-09-06T12:00:00"/>
<types:context tags="Web,Services,SOA,research,paper"/>
<wsa:MessageID>uuid:722B1240-...</wsa:MessageID>
<wsa:ReplyTo>http://socsoa....</wsa:ReplyTo>
<wsa:From>http://socsoa....</wsa:From>
<wsa:To>http://socsoa....</wsa:To>
<wsa:Action>http://socsoa....ac.at/Type/RFS</wsa:Action>
</soap:Header>
<soap:Body>
<!-- applied document translation request -->
<!-- schema details omitted -->
</soap:Body>
</soap:Envelope>
    
```

Listing 3 – SOAP interaction example

Listing 3 shows an interaction with SOAP. The SOAP is an application level communication protocol used in distributed applications. The temporal properties of SOAP calls are utilized in the system. Simple request response patterns are also implemented. This means that a request can be utilized by either avatar or software service. A request can also be rejected if necessary. In these kinds of interactions, dynamic behavior profiles are maintained in order to process request that come from various clients. Many behavioral metrics are context sensitive with respect to expertise area. The services provided by HPS through avatar are also can be ranked. However, these ranking changes over time as the expertise of human beings evolve over time.

VII. DISCUSSION

Evaluation of the proposed system is done in terms of scalability, performance and functional properties. With respect to adaption it is cyclic with some kind of looping that includes logging and analysis of interactions. The cost of such analysis is measure using the units discussed in [10]. The adaptation performance is provided in fig. 4 and 5.

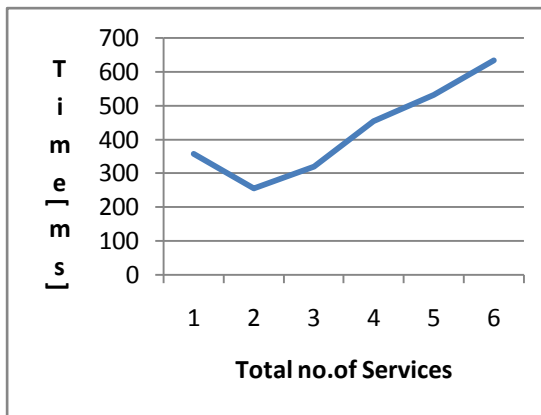


Fig. 4 – server interface update

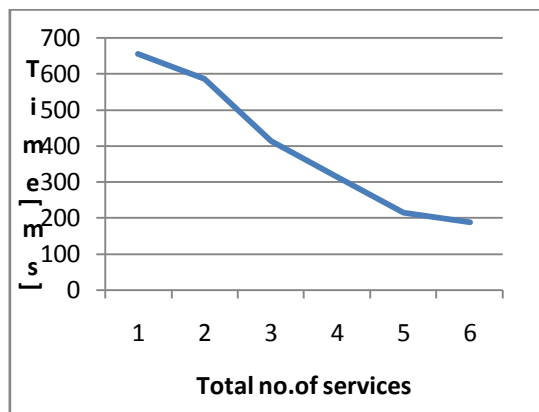


Fig. 5 – service closure update

As seen in fig. 4 and 5, service adaption details are presented. When number of services is increasing the time taken is increased. The time is taken in milliseconds indicates the average time that is required to perform redeployment of an altered WSDL and underplaying an operation of any service. When higher amount of services the time taken is increases while for less number of services the time taken is relatively less. Interface adaptations that need redeployment are 10 times slower than the closure exchanges.

VIII. SIMULATION SCENARIO

The proposed framework is tested using simulation. The simulation toolkit used for this purpose is “Repeat Symphony”. Human behavior is simulated with respect to interest drifts, working styles, expertise evolution over a period of time, and reliability. Simulation setup is created in environments with 25 clients and 5 avatars. The clients are having certain relations with avatars that represent human provided services. Time is set in such a way that clients send request periodically in 10 rounds interval. To process such request avatar needs 1 or 2 rounds. This way it is found that an avatar can serve only seven clients in average. Simulated agents are used to represent clients and service agents while testing the application after hosting the services. The proposed adaptations are also considered while evaluating the framework.

The experiential setup considers two expertise areas that are unique. 10 different tags are used to describe those areas. Avatars make use of 5 tags for having interest profiles. 3 different tags are used to annotate client requests. Each avatar is associated with clear a profile that is located in area A or B. The area A is known as white nodes while area B is known as black nodes. Initially avatars are not overloaded as they can match with the number of client requirements and expected expertise.

Running experiments caused some problems. These problems are due to human roles’ drifting skills, varying incentives, evolving expertise, specifications related to standards and so on. Here is an assumption that is avatars can shift between interests that cause them to have different levels of expertise. However, in general the deployed services that serve clients in the long run remain same. Only human expertise is evolved over a period of time and that is reflected in the profile of human associated with the corresponding avatar.

IX. Experimental Results

Experimental results revealed that the clients can change from one avatar to another avatar based on the service requirements and the HPS (Human Provided Service). This is because; the requirements of clients change over time. In the simple approach experiments adaptations are less while the aggressive approach has evidence to many adaptations. As the testing is done with simulation scenario, it has to be implemented in the real world in order to realize the framework in a live environment. The empirical results show that the system is promising and it can be used in real time applications by using the global standards being improved.




X. CONCLUSION

In this paper we fostered mixed service – oriented system that involves software components and also human provided services. The integration of such services into the architecture of SOA is the new trend that we fostered in this paper. When such environment is realized with human user’s involvement, many social dynamics such as expertise of the user, interests of user, personalization and many other implications pertaining to human users are to be taken care of. This warrants a systematic approach or framework that can address such implications well. In this paper our endeavor is to have a framework which seamlessly integrates the distributed software components through web services technology and human users’ involvement orchestrated well in the framework so as to enable the enterprise applications to have human expertise involvement apart from rich machine to machine interaction among diverse applications of related businesses. We aim, in the future, to improve the framework further by inventing new strategies for monitoring and adaptation.

Refernces

- [1] A. Agrawal et al. Ws-bpel extension for people (bpel4people), version 1.0., 2007.
- [2] M. Amend et al. Web services human task (ws-humantask), version 1.0., 2007.
- [3] D. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1):75, 2008.
- [4] V. Bryl and P. Giorgini. Self-configuring socio-technical systems: Redesign at runtime. *Int’l Trans. on Syst. Science and App.*, 2(1):31–40, 2006.
- [5] A. Cherns. The principles of sociotechnical design. *Human Relations*, 29(8):783–792, August 1976.
- [6] S. Dobson et al. A survey of autonomic communications. *ACM Trans. Auton. Adapt. Syst.*, 1(2):223–259, 2006.
- [7] Matjaz Juric, Doug H. Todd. SOA Web Services Journal: BPEL Processes and Human Workflow. 12-04-2006. <http://webservices.sys-con.com/read/204417.htm>
- [8] D. Schall, H.-L. Truong, and S. Dustdar. Unifying human and software services in web-scale collaborations. *Internet Comp.*, 12(3):62–68, 2008.
- [9] D. Shahaf and E. Horvitz. Generalized task markets for human and machine computation. In *AAAI*, 2010.
- [10] F. Skopik, D. Schall, and S. Dustdar. Modeling and mining of dynamic trust in complex service-oriented systems. *Information Systems*, 35:735–757, 2010.

- [11] F. Skopik, D. Schall, and S. Dustdar. Modeling and mining of dynamic trust in complex service-oriented systems. *Information Systems*, 35:735–757, 2010.
- [12] J. Zhang and R. J. Figueiredo. Autonomic feature selection for application classification. In *ICAC*, pages 43–52. IEEE, 2006.

	Kasara Venkata Ajay Reddy is student of DRK institute of science and technology, Hyderabad, AP, INDIA. He has received B.Tech Degree computer science and engineering, M.Tech Degree in computer science and engineering. His main research interest includes data mining. Cloud computing.
	Maddineni Kiran Chowdary is student of DRK college of engineering and technology, Hyderabad, AP, INDIA. He has received B.Tech Degree computer science and engineering, M.Tech Degree in computer science and engineering. His main research interest includes data mining. Software Engineering.
	Dr. R.V. Krishnaiah is working as Principal at DRK Institute of Science & Technology, Hyderabad, AP, and India. He has received M.Tech Degree EIE and CSE. His Main Research interest includes Data Mining, Software Engineering.