E-Portal Software Development Framework in Emerging Cloud Computing Environment

Dr.Enas H. Salih **Dr.A.K.R. Almazedi**

Al-Rafidain University College
Software Engineering Department

Abstract: *Cloud computing is an emerging new technology context that introduces promising environment for computer applications. This context put the enormous resource capabilities of the globe under the disposal of the development lifecycle of a software system. The new emerging concept of cloud computing is turning web service into eUtility that can be delivered over the internet to consumers who are normally software application developers and consumers. eUtility has announced the start of new eCommerce era where eUtility is metered due to its usage by consumers.*

This paper is presenting a framework to model the evolutional impact of eUtility on software engineering lifecycle and introduces E-Portal concepts as software integration points to the cloud.

Keywords: Cloud computing, E-Portal, Agent, Jade, SOAP, Web Services.

1. Introduction

Many definitions have been introduced to characterize what is the exact meaning of cloud computing like what Vaquero et al. [1] definition which states that “clouds are a large pool of easily usable and accessible virtualized resources, where resources is meant to be hardware, development platforms and/or software functionalities; these resources can be dynamically reconfigured to a variable loads and exploited by a pay-per-use model" [1,2]

Cloud computing is the future of revolutionary of web based inventions and innovations as in figure 1; the developed web technologies are representing a vast headings to reach this future soon.

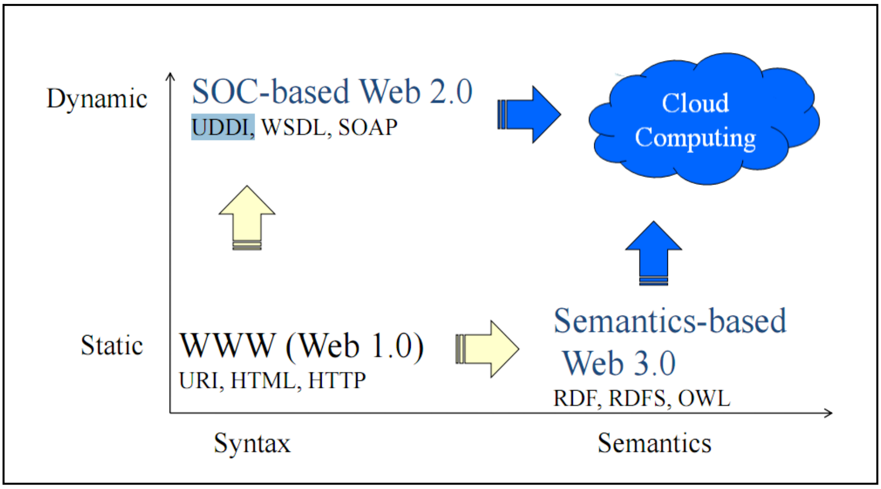

Figure 1: Cloud Computing is the future of Web Architecture

In cloud computing environment the Infrastructure, platforms and software are not a prerequisites for developing software product; this is what could computing is promising software developers and business organizations [1]

Cloud computing has different models; IaaS (Infrastructure as a Service) delivers computing environment infrastructure as Service; this is typically done by providing platform virtualization environment. The delivered infrastructure includes all resources demanded by the application requirements such as hardware and/or software resources. For example complicated GSM network can be delivered as a service to build mobile telephony application, the feasibility of using such a service rather building the GSM network on the ground can bring into the light the worthiness of being cloudy. [IBM report 2010]

The other cloud computing model is PaaS (Platform as a Service) through which the platform and/or solution stack is delivered as a service. PaaS facilitates deployment of applications without the cost and complexity of buying and managing the underlying hardware and software layers. [1,2,3]

SaaS (Software as a Service) is the most attractive and flexible cloud computing model due to its widespread usability and demand by applications developers.

The integration of desktop software development to cloud computing environment is still an ongoing project and has not been reaching standards. [3] this paper is focuses on a new software metric which is the E-Portal; this metric is an assessment factor to software readiness for the integration to web technologies and services. This paper is exploiting E-Portal concepts to reveal a new integration methodology between desktop application and services available on the cloud.

2. Outcomes of Cloud Computing[4]

Cloud Computing provides to organizations a new level of efficiency and economy to deliver IT services on demand:

2.1 IT Efficiency: Because Cloud Computing provides a way of deploying and accessing everything from single systems to huge amount of IT resources - on demand, in real time and at an affordable cost.

- **Reduce capital expense:** Cloud Computing makes possible for organizations to convert IT costs from capital expense to operating expense through technologies such as virtualization.

- **Cut the cost of running your Data Center:** Cloud Computing Improves IT infrastructure utilization rates and streamlines resource management.
- **Eliminate Over-provisioning:** Cloud Computing provides scaling on demand, which, when combined with utility prizing and pay-as-you-go, removes the need of over-provision to meet peak demands and removes the necessity of unused resources.
- **Shifts the risk of Under-provisioning:** Cloud Computing allows organizations to transfer the risk of mis-estimating workload to the Cloud Vendor. Companies can rapidly meet peak levels of their IT services. With the benefit of fulfilling SLAs and beaming potential new users.

2.2 Business Agility: Cloud Computing maximizes return using IT as a killer weapon through rapid time to market.

- **Accelerated cycle:** Cloud Computing gives a faster, more efficient way to develop new generation of applications or services. Provides faster development and testing cycles, reducing dramatically the time to market. Means that new business can be deployed in hours what used to take days, weeks, or even months.
- **Increase agility:** Cloud computing helps change like no other paradigms. It increases flexibility in service creation. Organizations can more easily deploy and manage business-critical applications than with traditional IT infrastructures. It brings you the elasticity of adapt, change and upgrade easily your IT services.
- **Fine grained billing model:** No up-front costs. The pay-as-you-go billing model offered by Cloud Computing vendors allow organizations to pay for the IT resources it actually uses; it does not have to maintain multiple sets of artificially high levels of resources to handle peak demands. It can reduce dramatically your IT costs.

3. Composite Computing Model (CCM)

Composite Computing Model is an architecture that uses a distributed, discovery-based execution environment to expose and manage a collection of service-oriented software assets. [4] client is requesting a web service list by communicating UDDI (Universal Description and Discovery Integration) registry. This registry holds XML based description of the published services

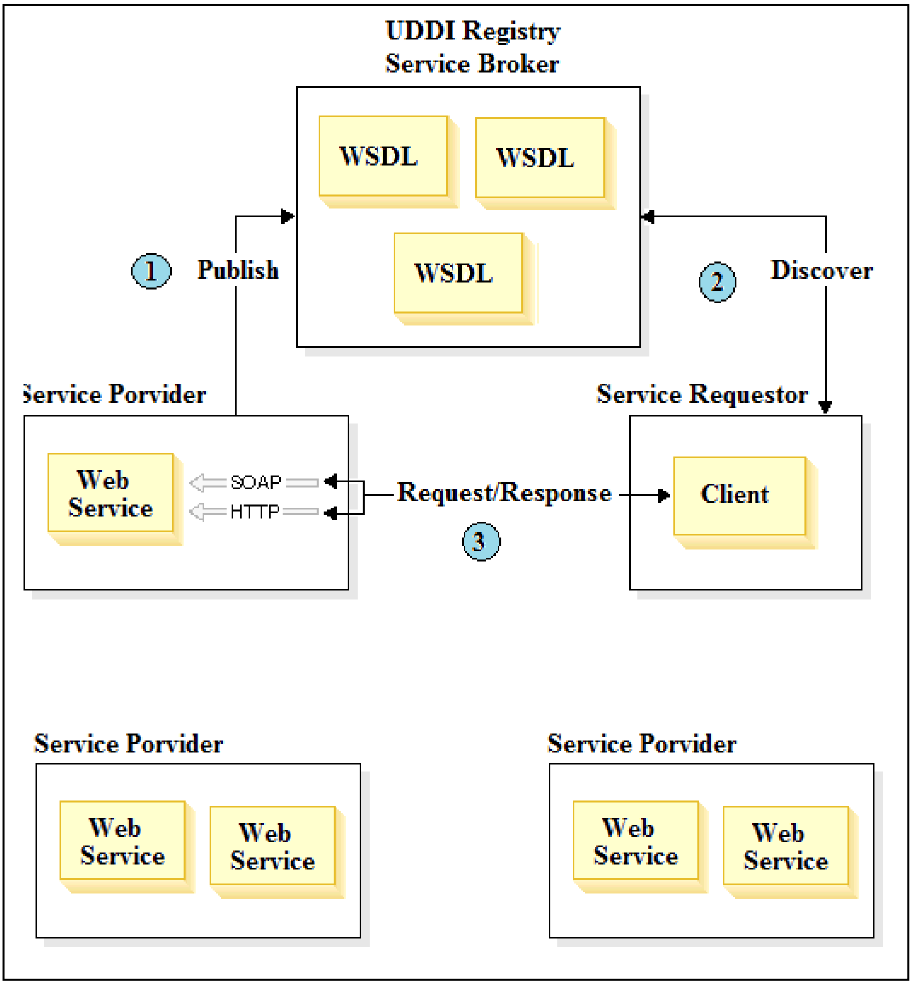

Figure 2: Client Interaction to Service Broker

4. Programming UDDI

Two APIs are described by the UDDI specification: the inquiry API and the Publishing API. They are accessed using the same techniques but use different XML documents, data structures, and access points. The inquiry API locates information about a business, the services a business offers, the specifications of those services, and information about what to do in a failure situation. Any read operation from a UDDI registry uses one of the inquiry API's messages. The inquiry API does not require authenticated access and is subsequently accessed using HTTP. [5,6]

The Publishing API is used to create, store, or update information located in a UDDI registry. All functions in this API require authenticated access to a UDDI registry; the UDDI registry must have a logon identity, and the security credentials for this identity must be passed as a parameter of the XML document for each UDDI invocation. Because publishing requires [1,6]

5. Java Agent Development Framework (JADE)

This paper has been deploying Java Intelligent Agent built using JADE environment, and due to the fact that this paper is not dedicated in any way to present Agent mechanism nor Agent programming techniques, thus this section is going to demonstrate only the monitoring and management tool used along the experiments conducted by this proposal. [7,8

JADE is the most widespread Agent-oriented middleware and it is a completely distributed middleware system with a flexible infrastructure allowing easy extension with add-on modules. The framework facilitates the development of complete Agent-based applications by means of a run-time environment implementing the life-cycle support features required by Agents, the core logic of Agents themselves, and of language features.[8] figure (3) presents Agent Management reference model used by JADE environment

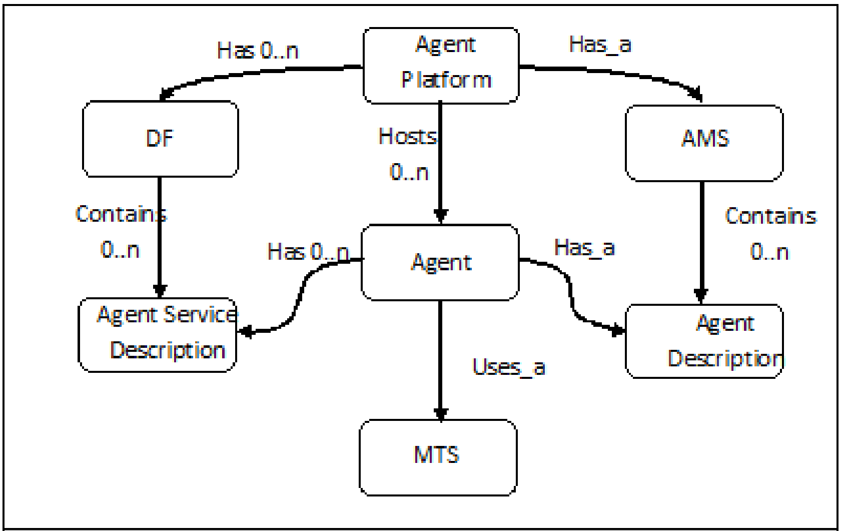

Figure 3 : Java Agent Development Platform Components

From figure (3) main components of the model can be briefly described as the following:

- **Agent Platform (AP):** Agent physical infrastructure in which Agents are deployed, this component includes machines, operating systems, FIPA agent management components, Agents.
- **Agent:** computational process that inhabits an AP and typically offers one or more computational services that can be published as a service description.
- **Directory Facilitator (DF):** the DF is an optional component of an AP providing yellow pages services to other agents. It maintains an accurate, complete and timely list of agents and must provide the most current information about agents in its directory on a non-discriminatory basis to all authorized agents. An AP may support any number of DFs which may register with one another to for federations. [7,8]

6. Software Development Integration Framework to the cloud

E-Portal is a measure of portability of software application to web technologies; in this manner Portal points are Integration gateways through which the invocation and deploying to external web services is conducted.

The framework introduced in this paper can be summarized by embedding portal endpoints to software development stages; these endpoints act as proxies to integrate web technologies such as Web services. The software development stages is abstracted in hierarchal XML nodes of PEP (Portal End Points). The portal point by itself is an XML representation to the inquiry demanded by this stage, and the demand is fulfilled by an Intelligent Agent who has the capabilities to parse the PEP inquiry and contact trusted venders for matching published services.

Eventually, the Agent is enriching the PEP hierarchy with detailed information about published web services (i.e., the most valuable information is the price and the name of the organization name).

The proposed framework is heavily using Java technologies and accompanied tools to experiment software components presented by this framework (i.e., Java Api for XML reqistry (JAXR), Simple API for XML (JAX) and Apache CXF to communicate web services). Figure (4)

PEP (Portal End Points) is a bundle of XML configuration, JAX-WS and JAX-RPC

and act a bidirectional integration points (cloud-to-application and application-to-cloud)

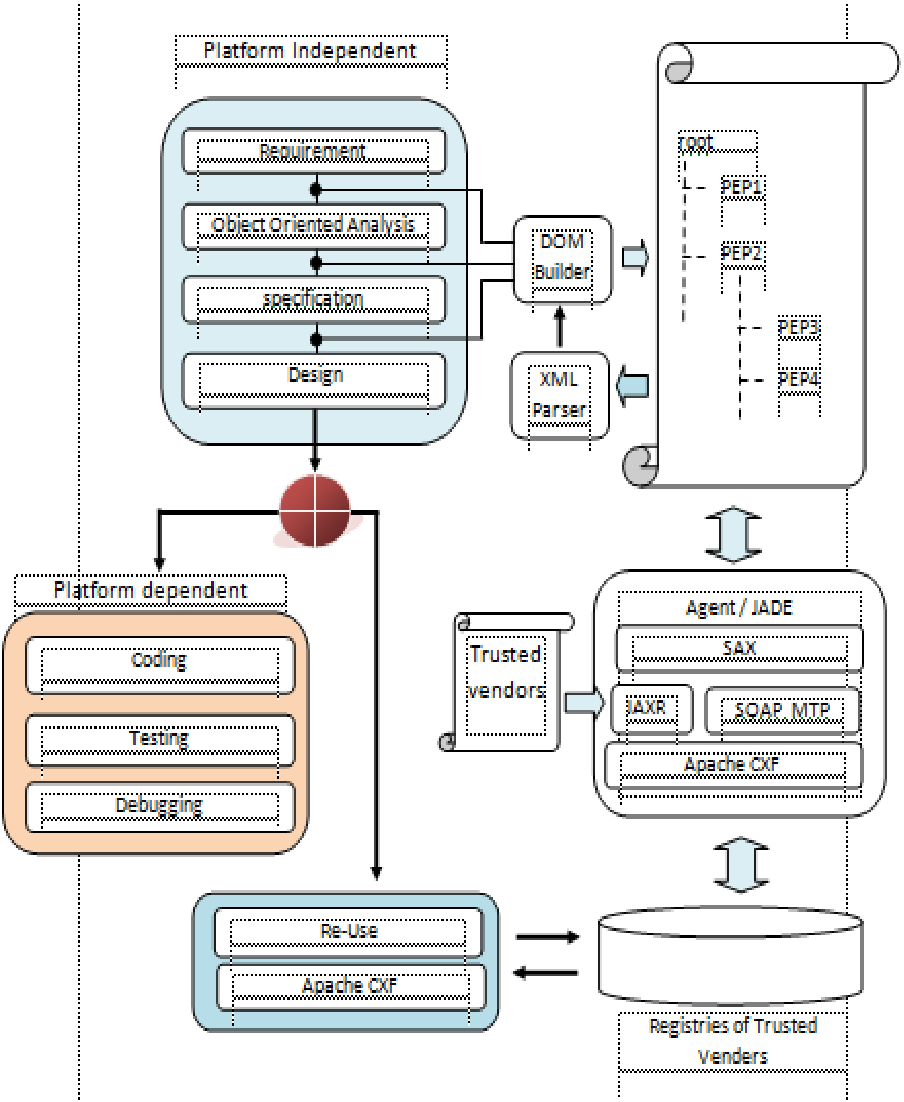

Figure 4: Integration Framework block diagram

Each software engineering stage is bond to at least one PEP through which cloud integration is established.

Along the software development a hierarchal XML file is constructed; this file contains leafs of PEPs each with represents a demand for an integration with the global cloud computing, the construction of the PEPs starts from the first stage and continue to tune up till the end of the development process. Figure 5 presents XML template of the PEP.

```
<PortalEndPoint>
    <ID>   </ID>
     <Requires>
                 <OrgName>
                       <ServiceName>
                              <url></url>
                              <cost></cost>
                        </ServiceName>
                 </OrgName>
           </Requires>
      <Provides>
              </Provides>
</PortalEndPoint>
```

Figure 5: Portal Enterprise Point XML Representaion

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <find_service businessKey="*" generic="1.0" xmlns="urn:uddi-
org:api">
      <name>UMLOptimizer </name>
    </find_service>
  </Body>
</Envelope>
```

Figure 6: Portal Enterprise Point XML Representaion

Requirement, analysis, specification and design status and requirements are to be described in XML token In this proposal the Software developer is responsible in selecting which service vendors he/she trusts and add them to a list of trusted web service vendors; the Agent is browsing UDDI registries of only trusted web service venders. Figure 6 presents an example to query UDDI registry for UML optimizer

Figure (7) presents the complete SOAP message template used to query UDDI registries of trusted vendors.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <find_service businessKey="***" generic="2.0"
xmlns="urn:uddi-org:api_v2">
      <findQualifiers>
        <findQualifier>***</findQualifier>
      </findQualifiers>
      <name>***</name>
      <categoryBag>
        <keyedReference tModelKey="***" keyName="***"
keyValue="***" />
      </categoryBag>
      <tModelBag>
        <tModelKey>***</tModelKey>
      </tModelBag>
    </find_service>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure 7: SOAP Message Template To Search for Specified Web Service

7. UDDI Registry Browser Agent

UDDI browser Agent is an autonomous Agent built using JAVA and JADE 4.2, the main functionality of this Agent is to automatically navigate through available trusted UDDI

Repositories for web services that matches certain constraints, the developer will be in charge by himself to list out trusted web sites that publish web services within the domain of interest for the developer (i.e., engineering, science, medical, humanitarian or others). The navigation process is initiated by the Agent by using JAXR to send SOAP messages over the internet to the inquiry URL of the UDDI repository publisher.

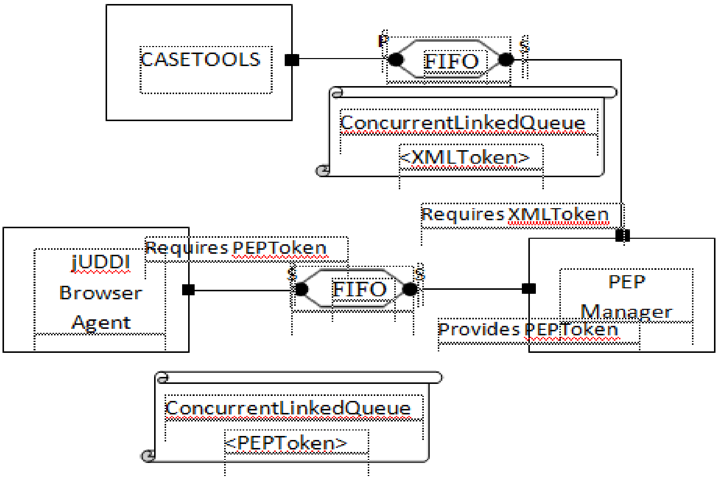

Figure 8: Conceptual View to PEP integration system

7.1 Agent Query Behavior

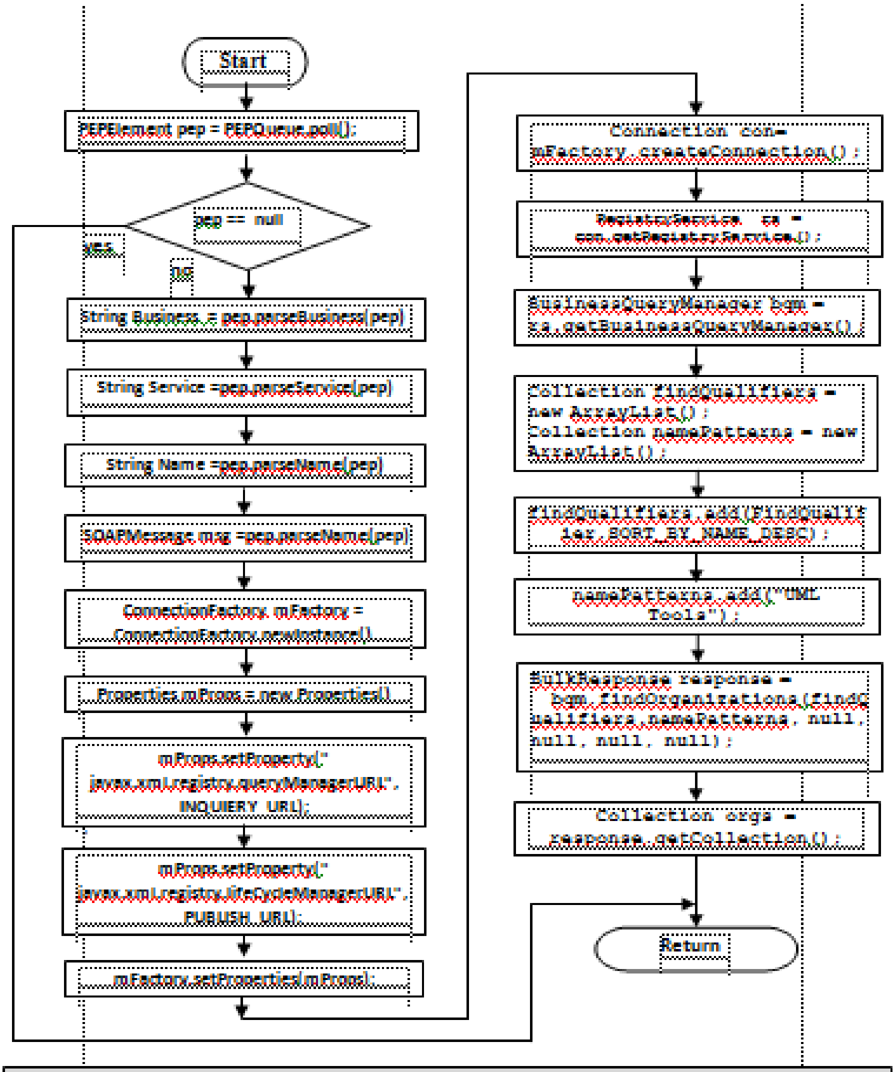
Start
PEPElement pep = PEPQueue.poll();
pep == null
yes
no
String Business = pep.parseBusiness(pep)
String Service =pep.parseService(pep)
String Name =pep.parseName(pep)
SOAPMessage msg =pep.parseName(pep)
ConnectionFactory mFactory = ConnectionFactory.newInstance()
Properties mProps = new Properties()
mProps.setProperty(" javax.xml.registry.queryManagerURL", INQUIERY_URL);
mProps.setProperty(" javax.xml.registry.lifeCycleManagerURL", PUBLISH_URL);
mFactory.setProperties(mProps);
Connection con= mFactory.createConnection();
RegistryService rs = con.getRegistryService();
BusinessQueryManager bqm = rs.getBusinessQueryManager();
Collection findQualifiers = new ArrayList();
Collection namePatterns = new ArrayList();
findQualifiers.add(FindQualifier.SORT_BY_NAME_DESC);
namePatterns.add("UML Tools");
BulkResponse response = bqm.findOrganizations(findQualifiers,namePatterns, null, null, null, null);
Collection orgs = response.getCollection();
Return

Figure 9: Agent Behavior Triggered by PEP Token Entered the Queue

7.2 Agent Update PEP Behavior

This behavior is responsible on updating PEP; the updating behavior is triggered when UDDI registry respond with at least one entry.

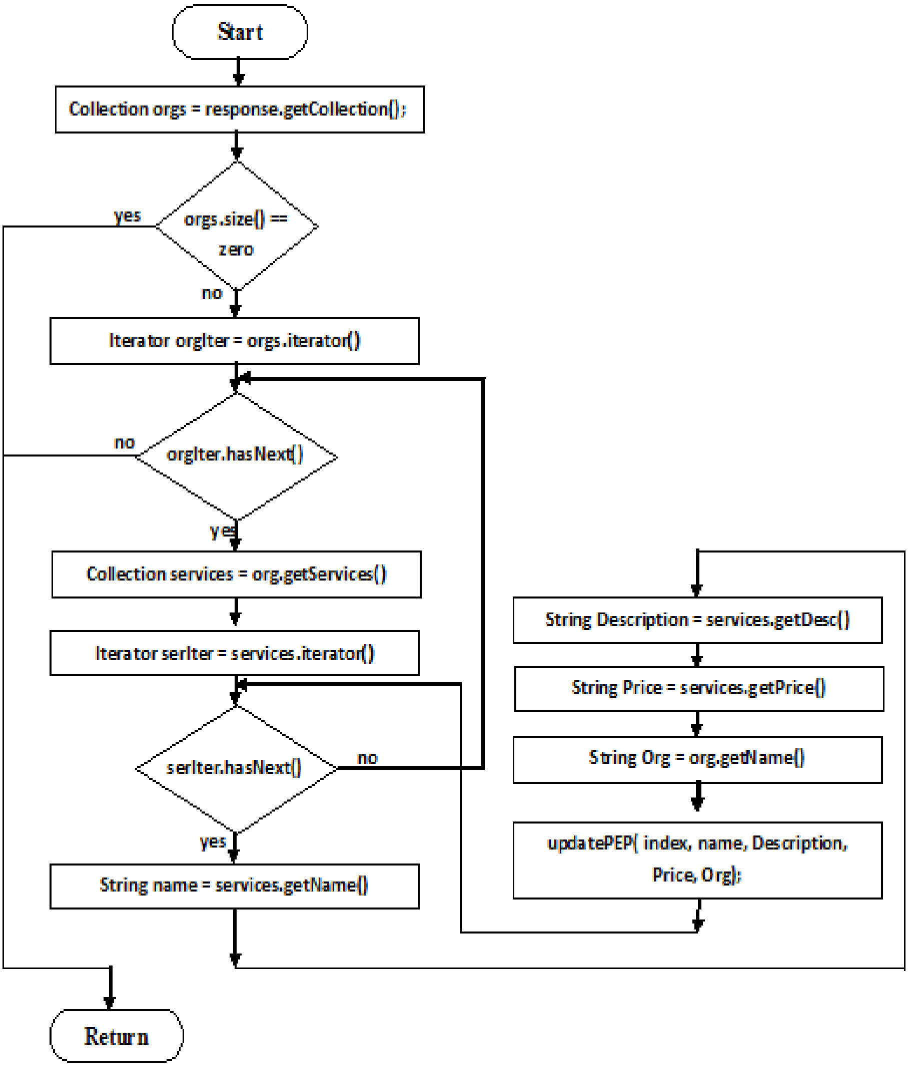
Start
Collection orgs = response.getCollection();
orgs.size() == zero
yes
no
Iterator orgIter = orgs.iterator()
orgIter.hasNext()
no
yes
Collection services = org.getServices()
Iterator serIter = services.iterator()
serIter.hasNext()
no
yes
String name = services.getName()
String Description = services.getDesc()
String Price = services.getPrice()
String Org = org.getName()
updatePEP(index, name, Description, Price, Org);
Return

Figure 10: Flowchart of Agent PEP Update Behavior

8. Discussion

In this paper and along many testing experiments done by JAVA software stubs built using NetBeans Ver. 7.2, Apache Axis 2 and JAXR library, most of the big UDDI registries have been found to be in a shut down state (e.g., IBM, SAP and Microsoft). Some venders moved to publish other registries schemes as in Microsoft who published BizTalk in the latest version on windows Azure.

Despite the shutdown of the free UDDI registries, the framework presented by this paper is valid to and ready to integrate jUDDI registries announced and a replacement for the UDDI.

9. Conclusions:

1- Domain specific software development reduces the efforts to discover web services over the cloud where web services on the cloud are categorized according to problem domains and the identity of the publishing organization.

2- Web service discovery methodologies are still ongoing projects and needs more efforts to be automatically conducted. The semantic of the query and UDDI entries is the crucial factor for accurate matching.

3- Migration of software products to the cloud is a configuration and management challenge rather that code compatibility issue. This proposal has introduced an integration methodology rather that starting over from scratch.

4- Cloud computing is a promising technology for the next generation of software development companies and software consumers at the same level.

5- Security and privacy issues are a milestone facing cloud computing; the database replica is what big companies cannot afford external hosting.

6- Expenses of software development cycle can be reduced by using published cloud computing services especially in software prototyping where many platforms and infrastructures have to be examined.

References:

1- Katarina Stanoevska-Slabeva, Thomas Wozniak, and Santi Ristol, "Grid and Cloud Computing, A Business Perspective on Technology and Applications", page 49, Springer, USA, 2010

2- Ira A. Fulton, "JICSIT/ITAIC 2011 keynote", Schools of Engineering, Arizona State University, USA, 2011.

3- Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley and Sons, US, 2011

4- David Chappell and Tyler Jewell, "Java Web Services", O'Reilly, USA, 2002

5- Richard Monson – Haefel,"J2EE WEB SERVER: XML, SOAP, WSDL, UDDI, WS-I, JAX-RPC, JAXR, SAAJ and JAXP", Addison-Wesley, USA, 2004

6- James McGovern, Sameer Tyagi, Michael Stevens and Sunil Mathew, "Java Web Services Architecture", Morgan Kaufmann Publishers, USA,2003.

7- Nicolaos Protogeros, "Agent and Web Service Technologies in Virtual Enterprises", Idea Group Inc, USA, 2008.

8- Fabio Luigi Bellifemine, Gioanni Caire, Dominic Greenweed, "Developing Multi-Agent using JADE", John Wiley, USA, 2007.

بنية تطوير البرامجيات الموافقة لتكنولوجيا الويب في بيئة الحوسبة السحابية الواعدة

م.د. ايناس هادي صالح أ.م.د. عبد الكريم رجب المزيدي
كلية الرافدين الجامعة – قسم علوم هندسة البرامجيات

المستخلص:

ان الحوسبة السحابية هي تكنولوجيا جديدة تقدم بيئة ناشئة لتطبيقات وبرامجيات الحاسوب. هذه البيئة الجديدة تضع مصادر وامكانيات الانترنيت الهائلة تحت تصرف دورة حياة بناء النظام البرمجي.

ان الفكرة الجديدة الواعدة للحوسبة السحابية تقوم بتحويل خدمات الويب الى وظائف الكترونية يمكن ارسالها عبر الانترنيت الى المستفيد والذي عادة هو مطور برامجيات او مستهلك لتلك الخدمات. الوظائف الالكترونية اعلنت بداية حقبقة جديدة من التجارة الالكترونية بسبب تسعير هذه الوظائف عند استخدامها من المستهلك.

هذا البحث يقدم بنية لنمذجة التأثير الهائل للوظائف الالكترونية على دورة حياة هندسة البرمجيات ويقدم مفاهيم جديدة لتكامل تطوير البرنامج مع الحوسبة السحابية.

Keywords: Cloud computing, E-Portal, Agent , Jade, SOAP, Web Services