Middle-East Journal of Scientific Research 21 (10): 1862-1885, 2014
ISSN 1990-9233
© IDOSI Publications, 2014
DOI: 10.5829/idosi.mejsr.2014.21.10.8258

Implementation of the Mobile Business Process Deployment Framework for Device Independence and Context-Aware Environments

[1]Saqib Ali, [2]Torab Torabi and [2]Hassan Ali

[1]Department of Information Systems,
Sultan Qaboos University, Muscat, Sultanate of Oman
[2]Department of Computer Science and Computer Engineering,
La Trobe University, Melbourne 3086, Australia

Abstract: A business process has always been vital to an enterprise. Mobile technologies provide real-time access to business process management (BPM) processes from remote locations. By extending existing processes to mobile workers, enterprises can significantly improve business responsiveness, increase worker productivity and increase processing times. Deployment of a business process to mobile devices in different context and in a device independence environment is a challenging task. The objective of this research is deployment of user business process to any mobile devices in any location and environment. Rapid growth in mobile technologies is the major obstacle is in business process deployment on mobile devices. In this research a mobile business process deployment framework has been developed and implemented that caters for both device independence and context-aware environments. In this research the focus is on how device independence and context-awareness can be integrated for mobile business processes deployment. A logistic company case study is adopted to implement and test the framework that uses location-aware methodologies and device specification for mobile business process deployment.

Key words: Business Processes · Location-Aware · Device Independence · Context-Aware

INTRODUCTION

The companies have been paying a great deal of attention to the potential of mobile communication technologies to redefine and extend the world of traditional E-Business by making its applications more available to mobile users. According to Swaminathan and Tayur [1], E-business is defined as "a business process that uses internet or other electronic medium as a channel to complete business transactions", whereas Mobile Business means the use of a mobile phone, or any mobile device, for business purposes.

Mobile Business is often described as the successor of electronic business and defined as the subset of it [2]. Usually some researchers assume that under mobile business any business operations performed by desktop computer can also be performed via wireless network. However mobile technology offers additional possibilities that are unique to the wireless world and cannot be performed via fixed network. For example, providing the mobile users with the services of location-aware and context-aware applications which cannot be performed with a fixed internet connection [3]. Summarizing several researches the mobile business can be defined as the exchange of goods, services and information using mobile devices [4].

The primary feature of mobile service is the ability to reach people regardless of their location and for mobile users' possibility to act from any location and at any time. Mobile Services are usually characterized by next four value drivers-mobility, reachability, localization and identification [5, 6]. As mobile devices constantly accompany its users, people can receive and send data regardless of place and time. They can also be reached by people at all times. This feature is especially useful to logistic and supply chain companies that regularly need to reach their mobile workforce to allocate various tasks to them.

Corresponding Author: Saqib Ali, Department of Information Systems, Sultan Qaboos University, Muscat, Sultanate of Oman.

A simple example of companies using mobile business for their current applications can be taken from the Scottish company "Justfone" [7]. It has developed a location based, work management and hub allocation system that save businesses time and money and also improve staff productivity and increase operational effectiveness. Another example of the system is an emergency plumbing service receiving a call from the customer with a burst pipe. Using this system the call center operator quickly determines the location of all plumbers within that geographic area and verifies their current status [8]. There are other examples use mobile technologies in their businesses such as Center Base, AppBuilder etc. [9, 10].

The success of the mobile applications is based not only on a new technology, but rather on its proper use [11]. Users buy technology to find solution to their problems. Wireless applications would be advantageous only in case, when they can be useful to their end users [12]. Defining your mobile audience is critical as we work in product teams, concentrating on deadlines and flowcharts and building wireless applications with new technology, it's easy to lose site of the end customer [13]. Mobile business value proposition originate from the fusion of the wireless technology with already available electronic business applications. On the other hand, the unique features of mobile business – mobility, localization give rise to emergence of completely new applications and business models.

Mobile Business Processes: The term "business process" was defined by numerous authors. This research follows the commonly used definition of Davenport [14] according to which a business process can be defined as "a specific ordering of work activities across time and place, with a beginning, an end and clearly identified inputs and outputs: a structure for action."

As the technology is changing very fast and the companies operate in complex environments that consist of thousands of processes. The business profit depends on efficient delivery of goods and services controlled by business process [15, 16]. So there is a need for the companies to make use of the technologies to make their product more profitable and services more efficient.

Many companies have been able to make their traditional business processes into mobile business processes [17, 18]. Mobile Business Processes can be based on these three assumptions: a. Uncertainty of Location b. Uncertainty of Location is externally determined and c. A corporation with external resources is needed in the execution of the process. The mobile business applications are changing the business process is through improved customer care. It is now easier than ever for any company representatives to aid clients in despite of their location.

Context-Aware Mobile Applications: Context means situational information, or as [19] states: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves."

Context-aware is a concept where the applications can discover and take advantage of contextual information such as user location, time of day, nearby people and devices and user activity. An example of context-aware applications is the scenario in which a consumer makes use of the mobile technologies in retail grocery supply chain through mobile shopping of electronically referenced grocery products. This kind of application is also beneficial to retailers, who by knowing the exact location of the consumer can allocate the workforce more accordingly and efficiently and can avoid out-of-stocks [20, 21]. For better understanding about context-aware mobile application, types of context used and model of context information, a detailed survey of context-aware mobile computing research was carried out by Guanling Chen and David Kotz [22].

One of the best mobility strategies is to adopt a mobile policy that is device independence. By adopting standards like HTML 5, developers can develop one application and can deploy on multiple devices. This would protect the organization technology investments and would provide the organization with the most flexibility in terms of dealing with new and unforeseen technology advancements. The mobile computing space is a rapidly changing environment, as evident in today's mobile marketplace. Currently, it was observed in the decline of the Blackberry Device, which was once unthinkable, the rise in popularity for the Google Android platform and the new emergence of new technologies like APPLE TV and Google TV. The mobility marketplace is extremely competitive and dynamitic, making it difficult to foresee and predict what device or platform is the next "BIG THING" [23]. The next sections will discuss the device independent technologies.

Device Independence: In today's world of rapidly changing technologies and the rise in the use of mobile devices has created a need for content adaptation. According to the author [24], the goal of device

independence is to develop ways for future web content and applications to be authored, generated, or adapted for a better user experience when delivered via many device types. Mobile devices are presenting new challenges and opportunities to companies looking to address the mobile market. Most pressing of these challenges is: how can I maximize the chance that my investment into a mobile solution will give me the return that my company needs? The answer: mitigate the risks of entering the mobile app space by lowering development costs, leveraging existing assets and providing a consistent user experience for your customers and clients [25]. This approach takes more time, cost and resources. To solve this problem, the device-independent approach is used to support different devices without the high cost.

Device Independence technologies can be divided into three different categories namely: - intermediate, client-side and server-side [26]. Intermediate approach can offer limited adaptation to the content delivery chain by having not changed the server that provides the content and the client that consumes it. This approach gives data-enabled phones access to web sites either omitting server's full resolution color images or changing it to the low resolution depending on the device display capabilities [26, 27]. In the client-side approach, the content adaptation can occur in the device itself. The advantage for this kind of approach is that the adaptation code has direct access to the device capabilities [26]. Server-side content adaptation approach offers maximum control over the delivered content including the ability to change content, navigation and style. In this approach the server is assumed to have sufficient information about the delivery context, including the delivery devices capabilities. As a result of device independence the content integrators does not need to develop content for an each single device and there content can be integrated to different devices without accumulating more resources and effort.

In order to present deployment framework design, a thorough research has been conducted on available mobile business processes, its context-awareness and device independence. The following section is going to discuss how mobile business process can be deployed and what deployment methods are currently available.

Deployment: The word "deployment" in terms of technology means, "Installing, setting up, testing and running" [28]. Therefore, deployment can be interpreted as a general process that has to be customized according to specific requirements or characteristics. Deployment of software, applications or a process is a complex task which covers all the activities that have to be carried out from the end of the development itself to the installation and maintenance of the application on the consumer devices. In [29, 30], the authors have compared certain types of deployment techniques in terms of their scale, complexity, expressiveness and barriers to first use. The deployment solutions being handled by these implementation techniques: manual, script, language and model-based deployment.

The automation of application or service deployment improves correctness, speed and documentation but, as different companies have experienced, it comes at an increased cost in development time and a steeper administrators' learning curve. In this research prototype implementation business processes are deployed through XML web services due to its importance and flexibility [31-34].

A web service is defined as "a collection of protocols and standards used for exchanging data between applications or systems" [35]. Web services can be considered as the emerging distributed middleware technology that uses a simple XML-based protocol to allow applications to exchange data across the web [36, 37]. At the core of the web service is the Simple Object Access Protocol (SOAP) an XML-based communication protocol for interacting with web services. The SOAP specification includes syntax to define messages, encode or serialize rules for data exchange and conventions for representing RPCs [38].

Web Services Description Language (WSDL) is used for describing the services available. It describes where the service is located, what operations are supported and the format of the messages to be exchanged based on how the service is invoked [38, 39]. On the other hand Business Process Execution Language (BPEL) is the language that is used to implement business processes in web services. It defines a notation for specifying business process behavior based on web services [40]. Certain technologies and models have been presented in recent years for the deployment of services and applications using mobile technology. One of the technologies being used recently by the companies for the deployment of services is over-the-air (OTA) deployment. OTA delivery of applications is important to support as it enables easy deployment and upgrades to applications, thereby reducing the disrupting effect which installations may have on mobile users [41, 42].

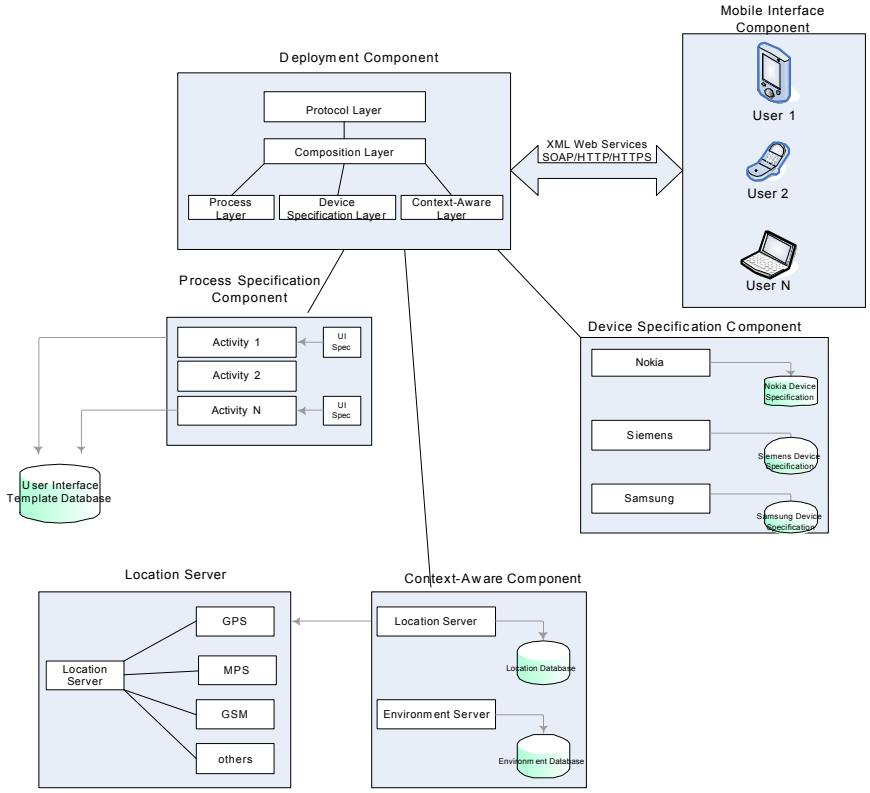


Fig. 1: Device Independence & Context-aware mobile process deployment framework

Another kind of deployment technique is "static deployment" where the user connects to the site of the application server through its mobile and then subscribes to a download operation. Then the user receives an SMS containing instructions for downloading and installing the application. The framework known as Smart Deployment Infrastructure (SDI) is designed to facilitate the installation of large distributed applications for any kind of user terminal. This framework is also presented for the context-aware deployment of applications to the mobile users [41]. The framework is implemented with middleware technologies like CORBA and SOAP which facilitates the development of large scale distributed applications. SDI offers automatic deployment of multi-component applications and provides a deployment solution to customize installation and to adapt to device capabilities [41, 43].

Deployment of a business process to a mobile device in a chosen context and in a device independence environment is a difficult task. Deploying an appropriate business process to a device can improve the effectiveness of the applications or services provided by the companies. The development of the framework is attempted to cater device independence and context-aware environment for mobile business process deployment. The following section presents a framework for mobile business process deployment.

The Framework: This research presents a novel framework that used in business process deployment to a mobile environment taking in notice the context of the user and its device specification. High level framework architecture is presented in Fig 1. Proposed framework is consists of five different main components, namely: a. Deployment component b. Process specification component c. Device Specification component d. Context-Aware component e. Mobile Interface component and followed client server architecture design.

The "***deployment component***" is the main component of the framework. This framework consists of two layers namely *Composition Layer* and *Deployment Layer*. The work for the composition layer is to compose a selected Business Process, setting the context of the deployment and customizing the process according to the device being used in the "Mobile Environment"; actually it is where the business process is going to be deployed. Whereas the deployment layer deploys the customized business process to the Mobile Environment using XML

Fig. 2: High level component diagram

web services and SOAP technologies. This framework adopted the same deployment techniques as discussed in section 1.4.

The second component in this framework is "***process specification component***". This is a traditional Business Process component consisting of activities, resources and user interfaces. User Interfaces are stored as XML documents in the User Interface (UI) Database associated with process specification component.

"*Device Specification component*" consists of different hardware configurations, screen resolution settings and other features for the mobile devices being currently used in the Mobile Environment.

"*Context-Aware component*" consists of many parameters but this research considers two parameters *"Location and Environment* "for the implementation. The location of the *Mobile User* is determined using different technologies like GPS, GSM, or MPS as shown in location server defined in the proposed architecture. The "Location Database" is used to store all the locations of the users and later can be used in process deployment.

"*Mobile Interface component*" consists of the N number of users using N number of mobile devices. The Mobile devices being used can be pocket pc's, smart phone's or PDA's etc.

In the development of the framework, design approach would be divided into two parts: The Framework Design and System Architecture. Framework's object oriented design properties includes it is generic, easy to automate, easy to reuse, flexible and scalable, easy to integrate, coordinate and synchronize between different components. The use of three-tier system architecture is the best way to show the interaction between the components and communication between the three tiers. Scenario based approach is used to explain the architecture so the framework can be described properly.

The Framework Design: This section will focus on the Object-Oriented Design for the Mobile Business Process Deployment Framework. The "High Level Component Diagram" is presented in Fig 2.

The components consist of logical parts of a complex class diagram. Components make it easier to understand the functionality through a high level of abstraction. For framework design there are nine active components. This section will cover a brief description of each of the components shown in Figure 2 above. This section also briefly discusses the functionalities and properties that each of the components possesses. The next section will explain each component in detail with the help of a low level class diagram. The brief explanation of each of the components is as follows:

Routing Engine: This component will take care of routing part of the framework. All the functionalities like displaying simple map, Geocoding, Reverse Geocoding and routing would be done by this component.

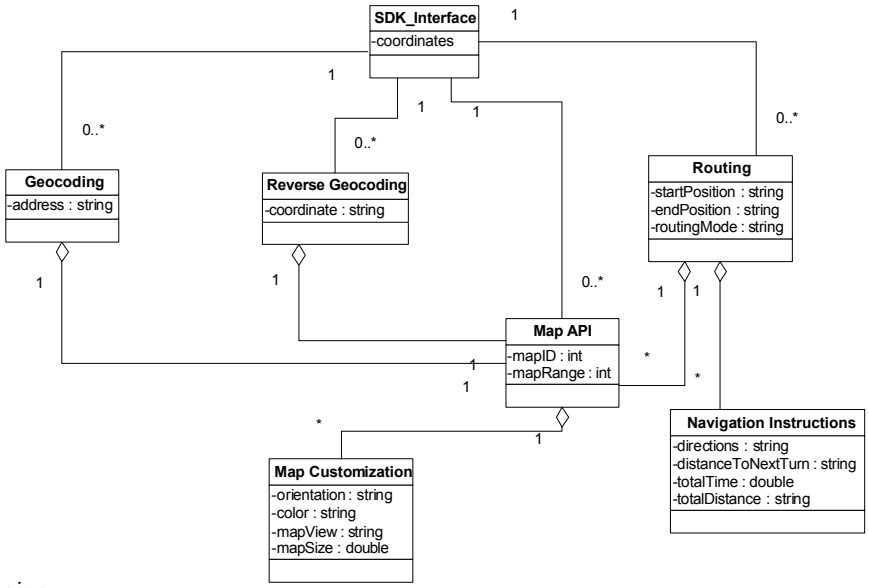


Fig. 3: Routing Engine

Process Specification: This is a traditional Business Process component consisting of activities, resources and user interfaces. All the specifications would take place in this component. All the specification for the Business Process along with its activities, resources and user interfaces are stored in process database.

Device Specification: It is a typical component. It consists of different hardware configurations, screen resolution settings and other features for the mobile devices being currently used in the Mobile Environment. In proposed framework, device specification would be stored as an XML documents and would be used at the time of deployment. Device database is used to store all the device specification for different mobile devices.

Context-Aware: It consists of many parameters but in framework implementation only two parameters "Location and Environment" are considered. This component will show how the location and environment parameter can be determined using different positioning and sensor technologies. Context database is used to store the context information of the user in the mobile environment.

Deployment: This is one of the main components of the proposed framework. This is the component where the actual deployment would take place. All the components would be integrated and coordinated before deploying to the mobile environment. All the Protocols and web services technologies would also reside in this component.

XML Parsing Component: All the XML parsing of the incoming responses, XML documents stored for Process Specification, Context and Device Specification would be done by this component.

Routing Engine: Routing Engine is used to provide maps (if required) to framework during deployment. Maps are important to show the user location which is one of the parameters of the context information. The class diagram for the component is shown in Fig 3.

Routing Engine is being taken from a Location Based Services Company iGoPlus for research purposes. As can be seen from the class diagram above, there is a SDK Interface that links the application with the routing engine. Through this engine proposed framework would be able to do the following functionalities:

- Display simple map with centerOnAddress or CenterOnPositon, where the map would be centered either by latitude longitude position or by address.
- Show Geocoded address on the Map.
- Do Reverse Geocoding and show the position of the points on the map.
- Display the route between two points on the map. This also includes displaying the Text Instructions required in navigating from start to end position.
- Do the map customization in terms of orientation, color, mapView and size.

Process Specification: This component takes care of the Business Process Specification along with its activities and resources. A Business Process can have many

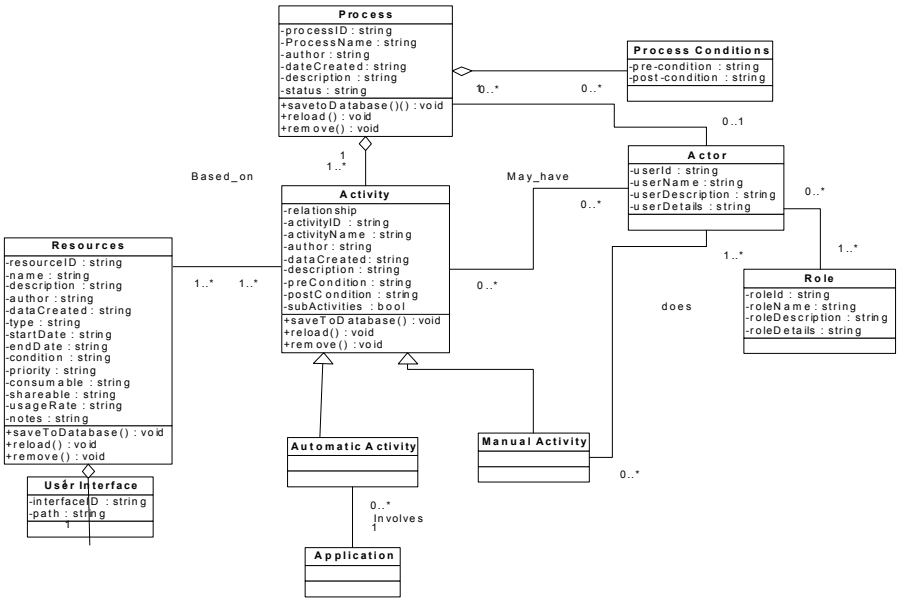


Fig. 4: Process specification

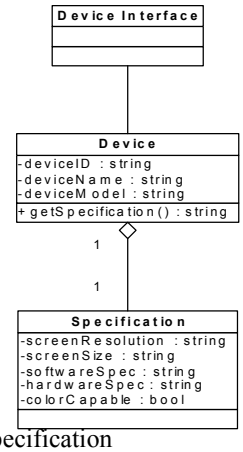


Fig. 5: Device specification

activities and each can have different resources. User interfaces are attached to each subsequent activity. Not every activity needs to have a user interface. User interfaces are associated to an activity during business process specification. The class diagram for Process Specification is shown in Fig 4.

A Business Process may have number of activities and each activity has some resources at its disposal. Like other activities of the specification component, resource as identifier, name, description, created by, type, its start date/time, end date/time, priority, condition, resource level and resource usage rate. Other characteristics of resources includes whether or not they are consumable or sharable resources. User interface is an important part of this component. In the proposed framework it is assumed that user interfaces would not be specified but only referenced. Its attributes include id and path where user interfaces are stored. It should be noted that not all activities would have user interfaces attached to it. The Process Specification is stored in the database. It can be seen from the diagram that there is one main super class "Database Management" which has all the subclasses from it, which are used in the system. The class "Database Management" contains all the attributes and the operations common to all the classes. This class would be containing an attribute like "dblink" that connects the user to the database and it also contains the attributes like query, result and table.

Device Specification: Having discussed the design for Routing Engine and Process Specification, this section will present the design for Device Specification. Device Specification is stored as XML documents in the database. Device Specification consists of software and hardware specification. Other components interact with this component through Device Interface. The representation of the device specification in the framework is shown in Fig 5.

As can be seen from the diagram above that the framework interacts to the device specification component through an interface. One device would have one specification and the specification would have attributes such as screen resolution, screen size, software specification, hardware specification and color capability.

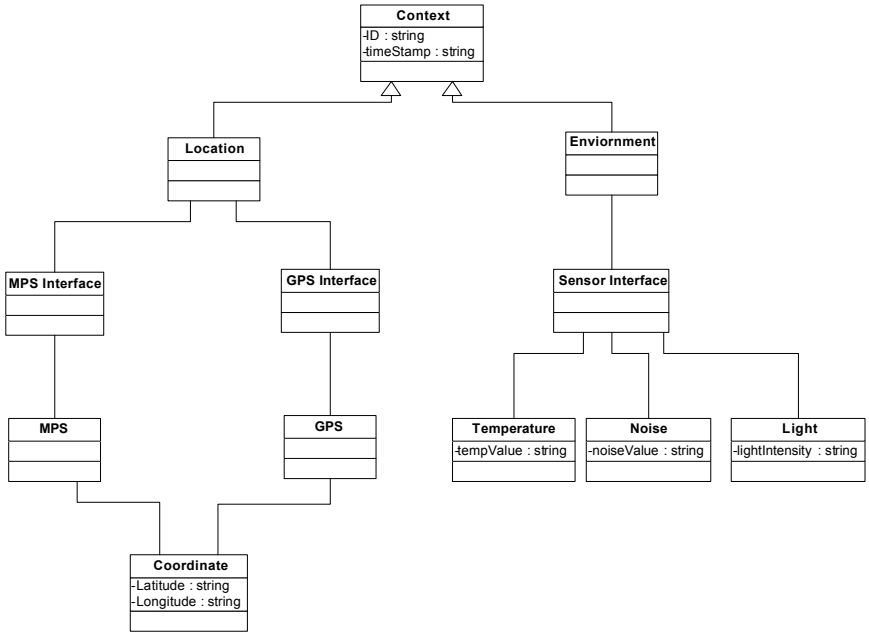


Fig. 6: Context-aware component

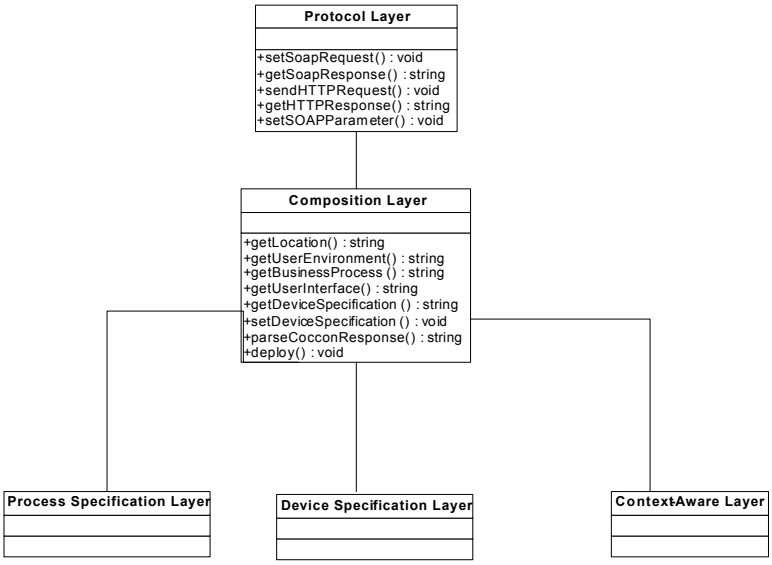


Fig. 7: Deployment component

Context-Aware: This section will be discussing the design for context-aware component. There are so many parameters for context-aware but in proposed framework only location and environment will be considered. The context information from this component would be stored as an XML documents in the database. The design for this component is done taking in view the working of commercial applications like on-board navigation systems, gadgets or sensors for light or noise. The way context parameters, location and environment can be determined is shown in Fig 6.

The framework will be determining location through two positioning technologies that are MPS and GPS. The position from these two technologies is in the form of Latitude and Longitude. Whereas Environment parameter can be determined through sensor interface as shown in Figure 6. Examples of attributes included in the Environment are temperature, noise and light.

Deployment: Deployment is the major component of the framework. Having specified Business Process, activities and resources, the deployment component

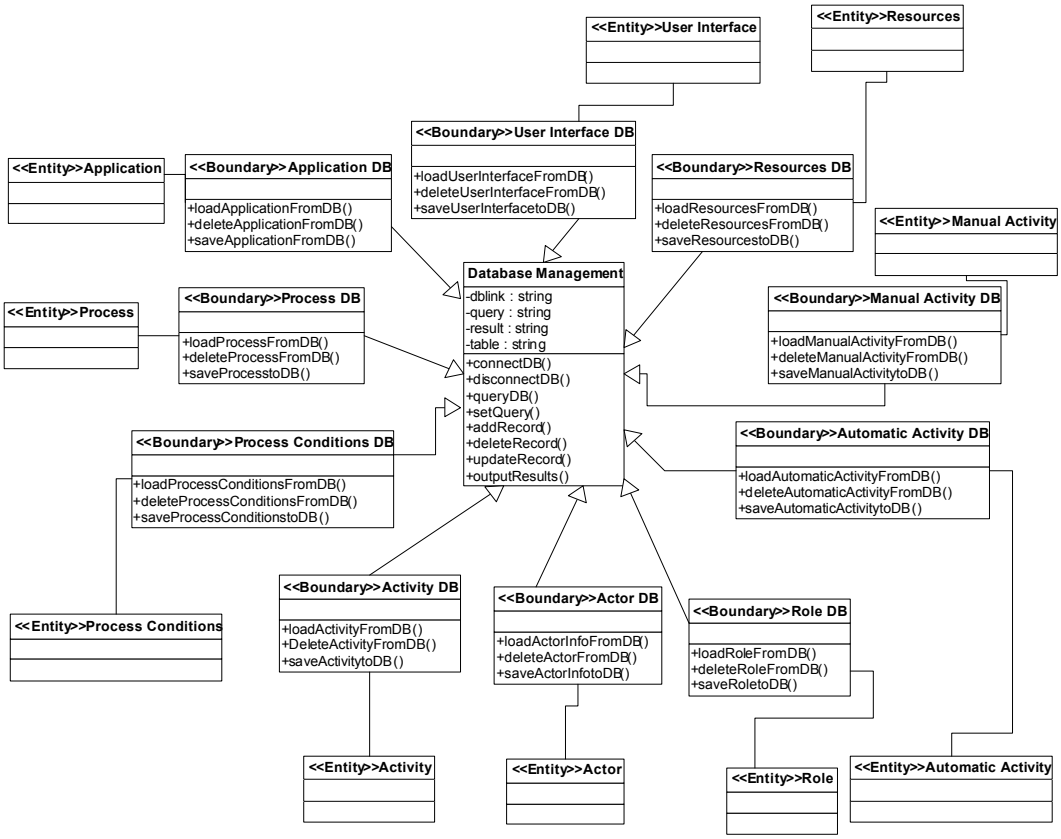


Fig. 8: Process specification database

brings context information and customized user interfaces together before deploying it to the mobile environment. The composition layer coordinates and integrates all three components and the routing engine component before the protocol layer can deploy it to the mobile device. The design and explanation for the functions of the classes in a deployment component is shown in Fig 7.

This component is the backbone for the deployment process. The composition layer coordinates and integrates all the three components together towards the deployment process. Examples of functions being performed by this layer are as follows:

- GetLocation():- Gets the location of the particular user from the Context-Aware Component. The response returned would be in terms of latitude and longitude.
- GetDeviceSpecification():-Gets the Device Specification for a particular device from the Device Specification Component.
- SetDeviceSpecification():- Sets the Device Specification for a particular mobile device in the Device Specification Component.

A Business Process can be deployed either by HTTP connection or by SOAP commands. The Framework caters for both protocols.

Process Specification Database: As discussed earlier, process specification will be stored in the database. The class diagram in Figure 8 shows inside the process specification database package. The next section will present process specification classes as database objects. Case study will show the example of using XML document of specified business process in the framework.

Process specification database package consist of one main super class called as “Database Management” which includes all the subclasses which are used in the system.

The class "Database Management" contains all the attributes and the operations common to all the classes. The class attribute “dblink” is used to connect the user to the database and it also contains other attributes like query, result and table. The class operations would be common to all the tables and can be seen from the class diagram presented in Fig 8. The information of the subclasses connected to the super class, can also be

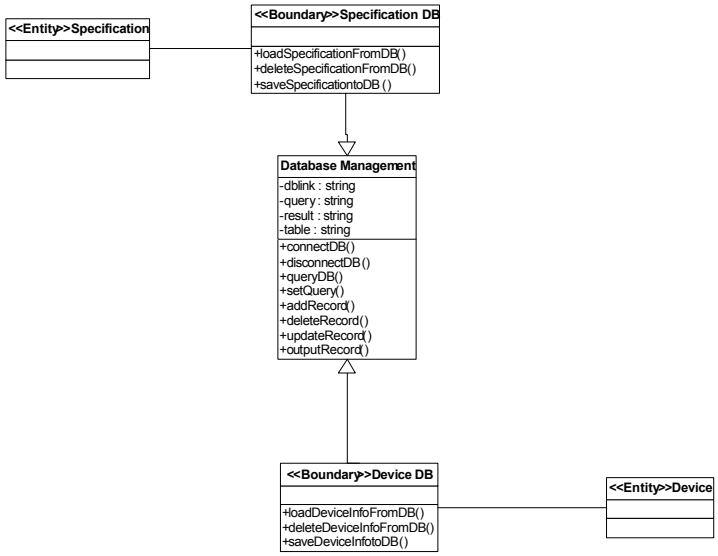


Fig. 9: Device database

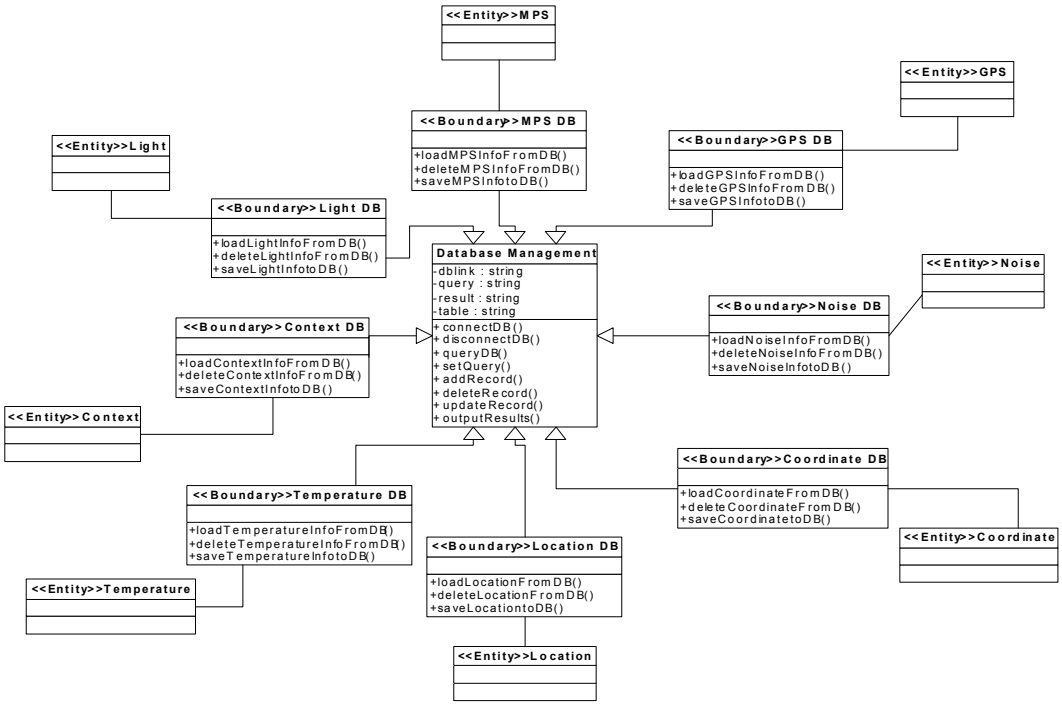


Fig. 10: Context database

retrieved from the way down shown in the class diagram. All the subclasses (as mentioned above) will be connected to the database and the information can be retrieved and stored from the individual tables as the load and store operations are given for each specific class. Each database class would then be connected with the associated classes. The main purpose of the database class in proposed design is to retrieve and store information from the associated class to the database. The detail design for following components is presented as follows:

Device Database: The class diagram for Device Database is shown in Fig 9.

Context Database: The class diagram for context database is presented in Fig 10.

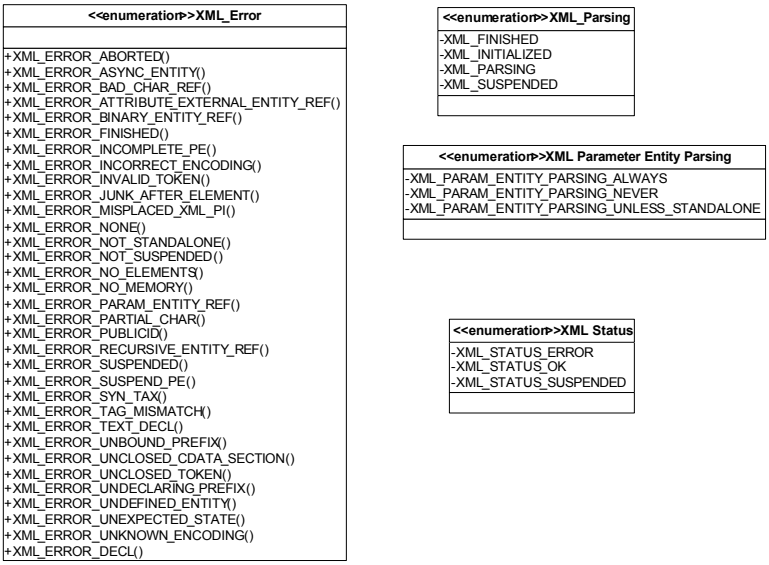


Fig. 11: XML parsing component

XML Parsing Component: This component takes care of XML parsing of incoming responses, Process Specification, context information and Device Specification documents. The XML classes for parsing, error, parameter entity parsing and status are shown in Fig 11.

System Architecture: The system architecture for the Mobile Business Process Deployment Framework is divided into three main parts client layer, business logic and database layer. The client layer interfaces between the system and the user. It consists of a mobile device running on client operating system. The Business Process is deployed to a mobile device in the shape of XML document. The communication between client tier and business logic tier is done through XML web services hosted on Web Server. The client tier consists of N number of users using N number of mobile devices.

Apache Cocoon Server is used in the framework to determine the software capabilities of a mobile device from a response being received from the client tier. ASP. Net local web server that is used to support XML web services and Simple Object Access Protocol (SOAP) as XML based protocol. SOAP is used to mark request and responses so that they can be transferred via protocols such as HTTP. These technologies set up the communication between the server and the client. Context Information and specification for mobile device are also stored as an XML document in the MySQL Database. Both of these components need to interact with MyODBC interface as shown in the architecture.

Deployment Engine Component consists of two layers Composition Layer and Protocol Layer. The work for the composition layer is to compose a selected Business Process, setting the context of the deployment and customizing the business process according to the device being used in the client tier. The protocol layer would set protocols for the deployment like setting up parameter for SOAP commands and setting up HTTP connections.

The database layer consists of database and Routing Engine. For communication between Business Logic Tier and MySQL, there is MyODBC interface that takes the request from the server and sends it to database which stores or retrieves XML document from the database. Map Interface in the Business Logic layer is used to retrieve maps. The System Architecture is shown in Fig 12.

Case Study: Case study approach is followed to implement the proposed framework. This case study is based on fictitious company named as “Independent Logistic System” or ILS in short. This case study is used for the purpose of proof of concept and implementation of proposed framework. In real life, ILS does not represent a specific company or its business process. Standard business process practices and strategies are adopted as they exist across the logistic industry.

Company Profile: ILS is a privately owned company dealing in server dispatch and fleet management. The company has different processes like order pickups,

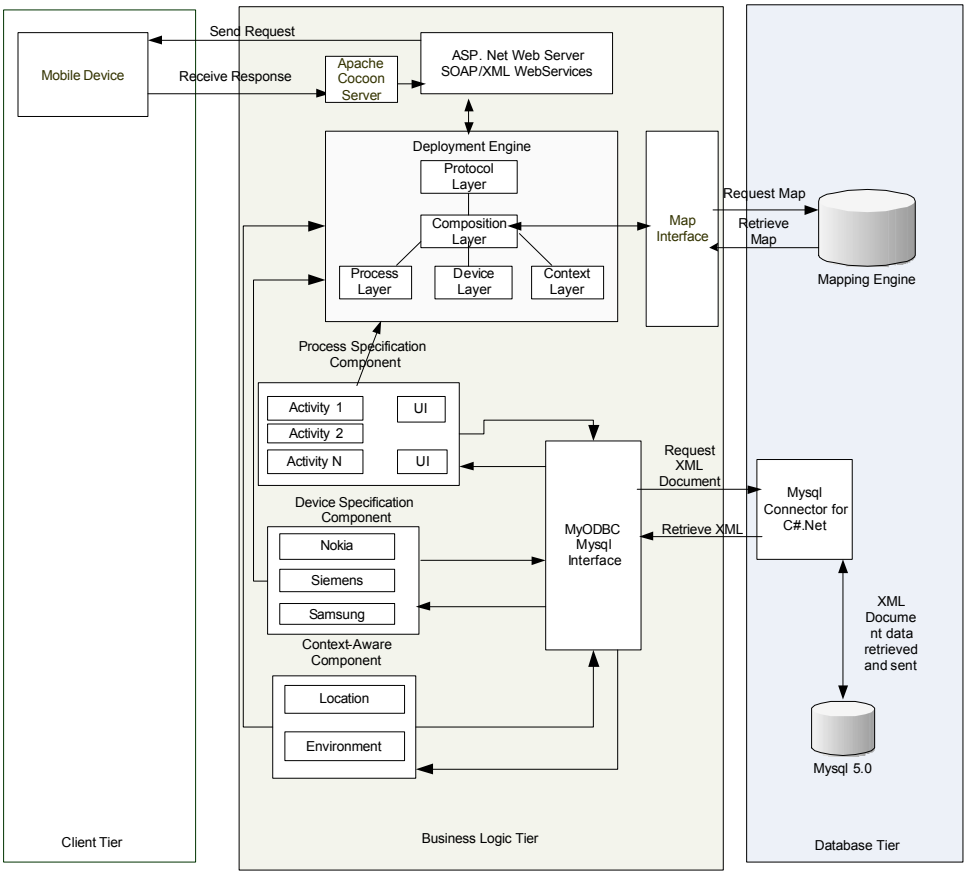


Fig. 12: Three tier architecture

order delivery and invoicing. All these processes are being processed at different locations. Each location may have different sets of processes and activities. The company has fleets that are being used to pick-up order from one location and deliver to another. The movement of the fleet is limited to the confined limits of the city or suburb. The application starts when a customer uses his/her mobile or web-based interface to login and request for the package to be dispatched by entering its pick-up and destination addresses and other related information. Then the dispatcher server would track the position of its fleet as well as their status (idle, busy or off-duty) and would determine the travel time between customer's pick-up location and the current location of the fleet shown with status as idle. The server sends the request to the fleet to accept or reject the job, on accepting the job the response comes to the server and the server then gives the details of the order to pick-up. If the user rejects the job then the server looks for the other fleet to pick-up the package.

If the distance of the idle fleet and the customer require a long delay before pick-up, then the server would automatically allocate the pick-up to the busy fleet. Based on conditions and parameters the best suitable fleet is chosen in regard to the proximity.

The whole purpose of this case study is to show, how a business process can be deployed to a mobile device (PDA, Smartphone or pocket PC) from the standalone server taking in regard the hardware specification of the device user is using. The whole process can also be initiated from the mobile user asking for the next available job. Context-aware has different set of parameters but in this case study, location is being taken as an example.

Overall objective of the case study is divided into primary and secondary goals, as discussed below:

Primary Goal:

- *Specifying a Business Process:* For this goal, a new business process (order pickup process) is specified and incorporate all the activities, resources and user interfaces within the business process.

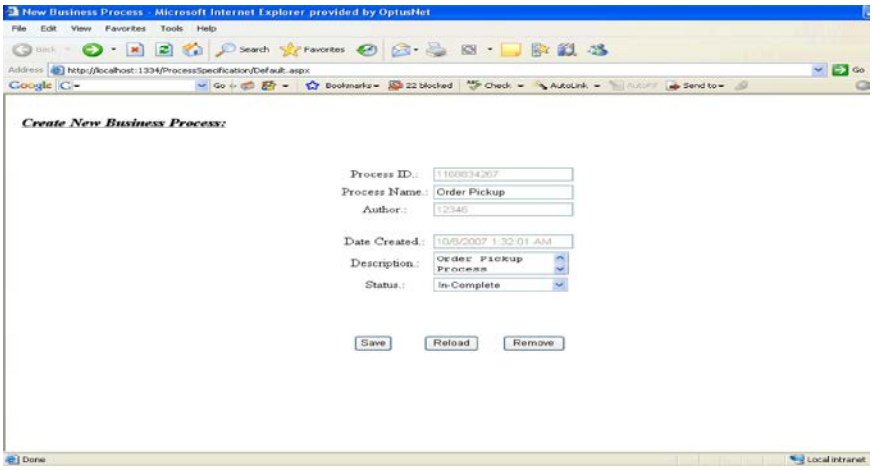


Fig. 13: Order pickup

- *Incorporating Device Specification:* For this goal, sequence is presented to show how device specifications for both mobile and non-mobile devices are stored and incorporated to be used in the developed application.
- *Determining and Incorporating Context information:* For this goal, sequence is presented to show how context information being determined from different positioning technologies can be stored, incorporated and coordinated towards the deployment of the business process.
- *Integration and Coordination:* For this goal, sequence is presented to show how the business process can be integrated and coordinated with specific device specification and context before the business process can be deployed.
- *Deployment:* For this goal, sequence is presented to show a coordinated and integrated business process can be deployed to a mobile or non-mobile environment.

Secondary Goal

The secondary goals include:

- Use of reusable components
- Follows reliable software engineering standards.
- Use of 3-tier scalable, flexible architecture that is being discussed in section 2.2.

The following sections will discuss how these objectives are going to achieve via the implementation of the case study. Section 4 presents a detailed evaluation of proposed framework implementation.

Specifying a Business Process: In this step, business process along with activities and resources are specified for an order pickup process. Business Process "Order Pick-up" is specified as shown in Figure 13.

The system will store all defined processes in the database in the form of XML documents. Later, these XML documents will be retrieved and reused for performing other operations in the application. The following part of XML schema is used to develop a business process document.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
atributeFormDefault="unqualified">
    <xs:element name="Process" type="ProcessType"/>
    <xs:complexType name="ProcessType">
        <xs:sequence>
            <xs:element name="process_id" type="xs:integer"/>
            <xs:element name="process_name" type="xs:string"/>
            <xs:element name="process_description" type="xs:string"/>
            <xs:element name="process_created_by" type="xs:string"/>
            <xs:element name="process_date_created" type="xs:string"/>
            <xs:element name="process_status" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

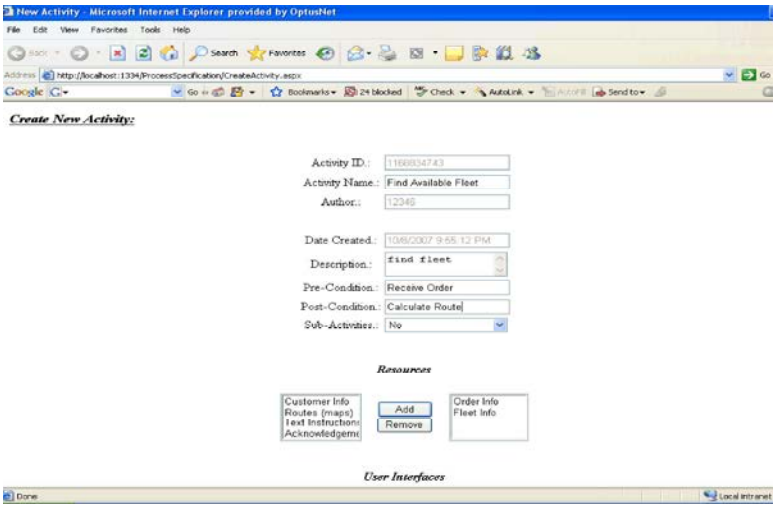


Fig. 14: Create new activity

Fig. 15: resource allocation within activity

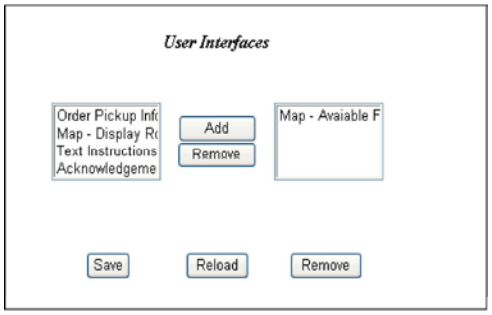


Fig. 16: User Interfaces

Like business process, each activity has its own attributes which include name, description, type, pre/post condition. Each activity has its own resources and user interfaces attached to it. An activity may or may not have sub-activities. This approach reuses the same activity template to define sub-activities. There can be number of activities for each Business Process. For example, in scenario "Order Pickup" would have activities: a. Receive Order b. Find available fleet c. Calculate Route that is the distance and time from the available fleet to the order pickup location d. Choose available fleet e. Forward detail to the user, this can include information such as text instructions, maps, order pickup details and other instructions f. Acknowledging order pickup, this acknowledgement would come from the fleet in implemented scenario. Created new activity is presented in Fig 14.

The activity "Find Available Fleet" has been specified with a specific pre-condition and post-condition. Pre-Condition being "Receive Order" activity that should be executed before specifying this activity. Post-Condition in this case "Calculate Route" activity that would be executed after "Find available fleet" activity. For a specific activity different kind of resources can be attached as shown in the Fig 15.

In this scenario, fleet info and order info resource has been attached to the "Find Available Fleet" activity (*Please Note: Resources displayed on the left are available resources and on the right are those that are attached to a specified activity*). Specification for resources would be discussed later in the section. Similarly different kind of user interface objects can also be attached to the same activity as represented in Fig 16.

In case "Map - Available Fleet" is being attached to the activity. In case study, it is assume that all the user interfaces would be saved in the local repository. These User Interfaces would only be attached or linked to an activity during its specification. A part of an XML document for "Find Available fleet" activity is presented as follows;

```
<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity id="1168834743" author="12346" name="Find Available Fleet" description="available fleet" type="activity"
  dateCreated="10/8/2007 9:55:12 PM" precondition="Receive Order" postcondition="Calculate Route"
  processSubactivities="No">
  <interfaces>
      <interface id="1164866283" name="Map-Available Fleet"/>
  </interfaces>
  <resources>
      <resource id="1162263736" name="Order Info"/>
      <resource id="1161827501" name="Fleet Info"/>
  </resources>
  </activity>
</activities>
```

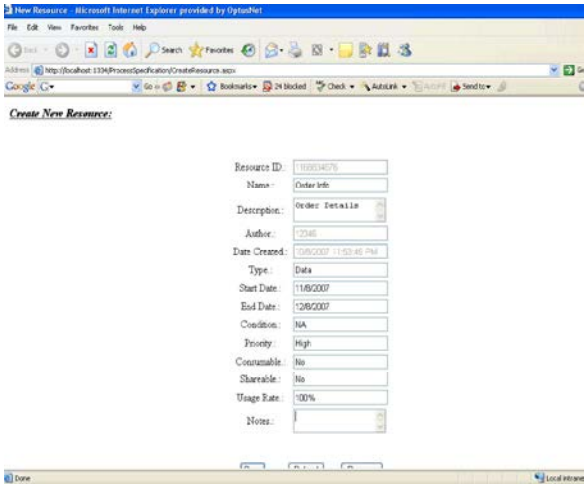


Fig. 17: Specification for resources

Resources in any business process as activities totally depend upon its resources. Each resource has its own name, description, created by, date created, start date/time, end date/time, type, whether they are consumable, shareable and/or have conditions on it. The use of these attributes entirely depends upon the resource type. All this information is very necessary in order to manage any resource. Specification for resource "Order Info" is in Fig 17.

All the attributes being used during the resource specification are already discussed and explained in the earlier sections. The XML Schema for resource is shown in Fig 18 below:

Device Specifications: Device Specification plays an important role in the deployment. Business Processes with different User Interfaces attached to it would be deployed to different devices with different screen resolutions. So in order to display a user interface correctly, it is required to know the screen size of the device that it is going to be deployed to. There are other properties or characteristics that are associated in devices specification like: a. Software Platform Characteristics b. Network Characteristics c. WAP Characteristics d. Browser Properties e. MMS Characteristics

In this research, only concern is with the screen resolution of the particular device. Device Specifications for various mobile devices would be stored as an XML document in MySQL database. Other way of determining device specification is through Cocoon Apache Server. In which the specification of the device can be determined only if request or response is being send by the device to the server. Cocoon Apache Server resides on the server. This section will show part of device specification for three different mobile devices including Nokia N-Gage, Pocket PC (PCC) and Siemens S5500. Part of device specification for Nokia N-Gage is shown in Fig 19.

As it can notice from the device specification, the screen resolution for the Nokia device is **176x208** that is helpful in process deployment**.** Now have a look at the device specification for Pocket PC (PPC) as shown in Fig 20.

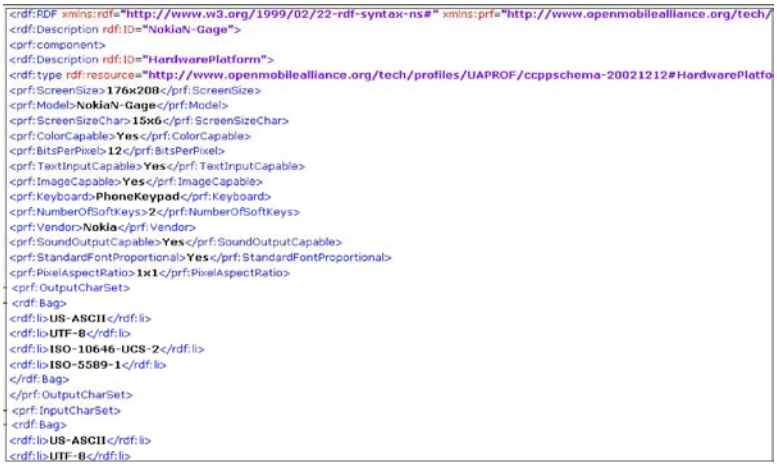
Screen Resolution for Pocket PC is shown as **240x320.** Part of device specification for Siemens S5500 is presented in Fig 21. The screen size for this particular device is shown as **101x80**.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
atributeFormDefault="unqualified">
    <xs:element name="Resources" type="ResourceType"/>
    <xs:complexType name="ResourceType">
        <xs:sequence>
            <xs:element name="resource_id" type="xs:integer"/>
            <xs:element name="resource_name" type="xs:string"/>
            <xs:element name="resource_description" type="xs:string"/>
            <xs:element name="resource_created_by" type="xs:string"/>
            <xs:element name="resource_date_created" type="xs:string"/>
            <xs:element name="resource_start_date" type="xs:string"/>
            <xs:element name="resource_end_date" type="xs:string"/>
            <xs:element name="resource_condition" type="xs:string"/>
            <xs:element name="resource_consumable" type="xs:string"/>
            <xs:element name="resource_shareable" type="xs:string"/>
            <xs:element name="resource_usage_rate" type="xs:string"/>
            <xs:element name="resource_notes" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```



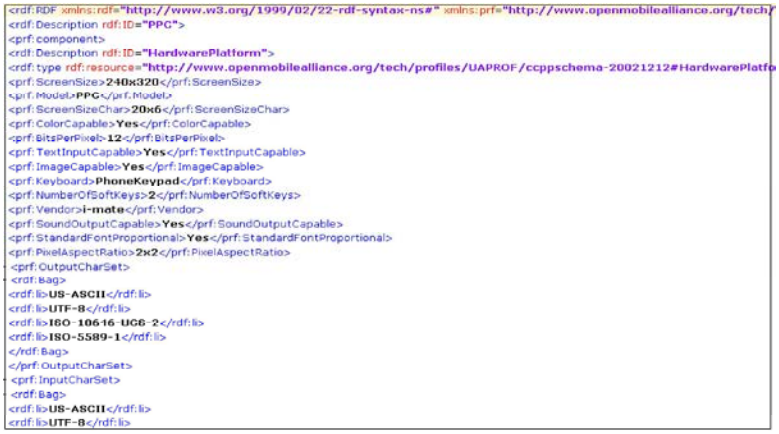
```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
atributeFormDefault="unqualified">
    <xs:element name="Resources" type="ResourceType"/>
    <xs:complexType name="ResourceType">
        <xs:sequence>
            <xs:element name="resource_id" type="xs:integer"/>
            <xs:element name="resource_name" type="xs:string"/>
            <xs:element name="resource_description" type="xs:string"/>
            <xs:element name="resource_created_by" type="xs:string"/>
            <xs:element name="resource_date_created" type="xs:string"/>
            <xs:element name="resource_start_date" type="xs:string"/>
            <xs:element name="resource_end_date" type="xs:string"/>
            <xs:element name="resource_condition" type="xs:string"/>
            <xs:element name="resource_consumable" type="xs:string"/>
            <xs:element name="resource_shareable" type="xs:string"/>
            <xs:element name="resource_usage_rate" type="xs:string"/>
            <xs:element name="resource_notes" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```


Fig. 18: XML Schema for resources



```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:prf="http://www.openmobilealliance.org/tech/
<rdf:Description rdf:ID="NokiaN-Gage">
<prf:component>
<rdf:Description rdf:ID="HardwarePlatform">
<rdf:type rdf:resource="http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschema-20021212#HardwarePlatfo
<prf:ScreenSize>176x208</prf:ScreenSize>
<prf:Model>NokiaN-Gage</prf:Model>
<prf:ScreenSizeChar>15x6</prf:ScreenSizeChar>
<prf:ColorCapable>Yes</prf:ColorCapable>
<prf:BitsPerPixel>12</prf:BitsPerPixel>
<prf:TextInputCapable>Yes</prf:TextInputCapable>
<prf:ImageCapable>Yes</prf:ImageCapable>
<prf:Keyboard>PhoneKeypad</prf:Keyboard>
<prf:NumberOfSoftKeys>2</prf:NumberOfSoftKeys>
<prf:Vendor>Nokia</prf:Vendor>
<prf:SoundOutputCapable>Yes</prf:SoundOutputCapable>
<prf:StandardFontProportional>Yes</prf:StandardFontProportional>
<prf:PixelAspectRatio>1x1</prf:PixelAspectRatio>
<prf:OutputCharSet>
<rdf:Bag>
<rdf:li>US-ASCII</rdf:li>
<rdf:li>UTF-8</rdf:li>
<rdf:li>ISO-10646-UCS-2</rdf:li>
<rdf:li>ISO-5589-1</rdf:li>
</rdf:Bag>
</prf:OutputCharSet>
<prf:InputCharSet>
<rdf:Bag>
<rdf:li>US-ASCII</rdf:li>
<rdf:li>UTF-8</rdf:li>
```


Fig. 19: Device specification for Nokia N-Gage



```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:prf="http://www.openmobilealliance.org/tech/
<rdf:Description rdf:ID="PPC">
<prf:component>
<rdf:Description rdf:ID="HardwarePlatform">
<rdf:type rdf:resource="http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschema-20021212#HardwarePlatfo
<prf:ScreenSize>240x320</prf:ScreenSize>
<prf:Model>PPC</prf:Model>
<prf:ScreenSizeChar>20x6</prf:ScreenSizeChar>
<prf:ColorCapable>Yes</prf:ColorCapable>
<prf:BitsPerPixel>12</prf:BitsPerPixel>
<prf:TextInputCapable>Yes</prf:TextInputCapable>
<prf:ImageCapable>Yes</prf:ImageCapable>
<prf:Keyboard>PhoneKeypad</prf:Keyboard>
<prf:NumberOfSoftKeys>2</prf:NumberOfSoftKeys>
<prf:Vendor>i-mate</prf:Vendor>
<prf:SoundOutputCapable>Yes</prf:SoundOutputCapable>
<prf:StandardFontProportional>Yes</prf:StandardFontProportional>
<prf:PixelAspectRatio>2x2</prf:PixelAspectRatio>
<prf:OutputCharSet>
<rdf:Bag>
<rdf:li>US-ASCII</rdf:li>
<rdf:li>UTF-8</rdf:li>
<rdf:li>ISO-10646-UCS-2</rdf:li>
<rdf:li>ISO-5589-1</rdf:li>
</rdf:Bag>
</prf:OutputCharSet>
<prf:InputCharSet>
<rdf:Bag>
<rdf:li>US-ASCII</rdf:li>
<rdf:li>UTF-8</rdf:li>
```


Fig. 20: Device specification for pocket PC (PPC)

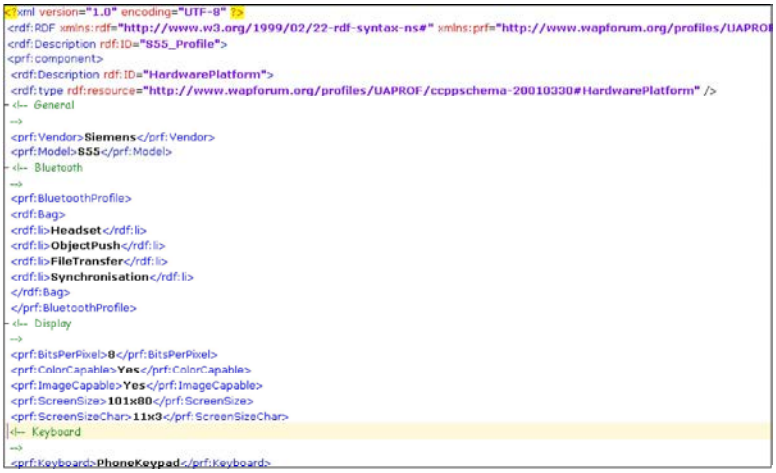


Fig. 21: Device specification for Siemens S5500

Context Information: Every business process is deployed depending upon its context as discussed in previous sections. In case study implementation, only *location* is taken as a context parameter. Also due to limited resources, it is assumed that in an implementation all the positions for the fleet are predefined and stored as a XML document as shown follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<markers>
<marker timestamp="2007-10-12 15:06:10.0" fleetID="1" lng="144.881072" lat="-37.827232" />
<marker timestamp="2007-10-12 15:06:46.0" fleetID="2" lng="144.818416" lat="-37.768196" />
<marker timestamp="2007-10-12 14:11:50.0" fleetID="3" lng="145.415968" lat="-37.921512" />
<marker timestamp="2007-10-12 14:03:30.0" fleetID="4" lng="145.014304" lat="-37.680296" />
<marker timestamp="2007-10-12 14:09:24.0" fleetID="5" lng="144.983188" lat="-37.796144" />
<marker timestamp="2007-10-12 14:49:13.0" fleetID="6" lng="144.927552" lat="-37.790728" />
<marker timestamp="2007-10-12 14:52:14.0" fleetID="7" lng="145.155232" lat="-37.962628" />
<marker timestamp="2007-10-12 15:14:23.0" fleetID="8" lng="145.019776" lat="-37.907824" />
<marker timestamp="2007-10-12 15:19:12.0" fleetID="9" lng="144.961344" lat="-37.77156" />
<marker timestamp="2007-10-12 15:21:18.0" fleetID="10" lng="145.285408" lat="-37.886532" />
</markers>
```

The same XML document would be used in case study scenarios in the next section. The information being recorded includes fleetID, its position in terms of Latitude and Longitude, timestamp meaning time the position was recorded for a specific fleet. The Latitude, Longitude values from this document would be used to calculate the distance and time from fleet location to order pickup location. In real time environment, the context information for the fleet would be determined using positioning technologies and this information would refresh after every specified time to keep the track of the fleet.

Deployment: This section will show the deployment of the "Order Pickup" Business Process taking in notice the context information and device specification. This would be presented with the help of a simple scenario that include order dispatch and order pickup through finding appropriate fleet.

As discussed in the previous section that "Order Pickup" Business Process has number of activities. This scenario will discuss each activity along with its resources and user interfaces in order. The first activity for "Order Pickup" Business Process is "Receive Order" as shown in the tree structure shown in Fig 22.

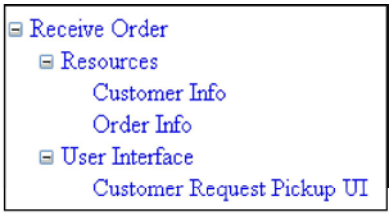


Fig. 22: Receive order activity

Resources being attached are "Customer Info" and "Order Info". User Interface associate with the activity is "Customer Request Pickup UI". As being discussed in the previous sections that user interfaces are not specified, they are only referenced. This activity would be executed when the client logs into the website and enters its details for the order to be picked up as shown in the Fig 23.

Fig. 23: Enter order pickup details

Fig. 24: Find available fleet

Fig. 25: Map for order pickup

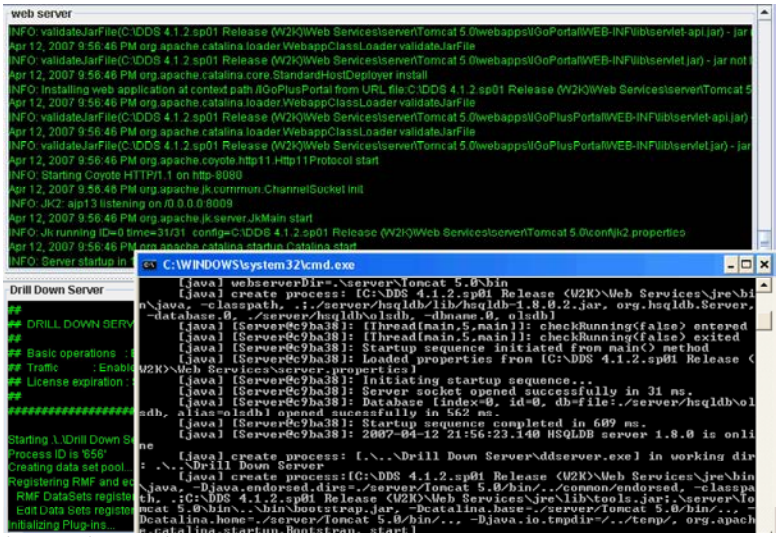


Fig. 26: Backend running routing engine

After the user enters order details, the server tries to find the available fleet that is the result of execution of the second activity "Find Available Fleet". The details for this activity are presented in Fig 24.

"Fleet Info" would be the context information in terms of Latitude and Longitude stored as the XML document in the local repository as discussed earlier. The routing engine then uses this context information to determine the fleet around the order pickup location. The fleet around pickup location is shown in the map presented in Fig 25.

The screen shot for the routing engine running on the server is shown in Fig 26.

After finding the available fleet, the server needs to calculate the route. The detail for this activity is shown in Fig 27.

The server needs the customer and fleet positions so that it can find the distance and time from the fleet location to the pickup order location. The parcel pickup location is shown as a green dot and route for two closest fleets are shown in Fig 28.

The time taken (Time A) for the first fleet to reach destination is 1 minute and 1 seconds and the distance is 0.4895 miles. For the second fleet, time taken (Time B) is 1 minute and 09 seconds and the distance to the destination is 0.5674 miles. The XML request and response from the routing engine for fleet 1 is shown in Figure 29 and Fig 30 below:

The activity "Choose Available fleet" chooses the fleet after calculating the route between the fleets and the destination. According to that fleet 1 is preferred over fleet 2. And now fleet 1 has to be sent with the request. For this activity "Send Job Request" the details are as presented in Fig 31.

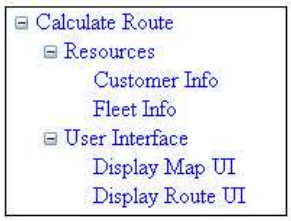


Fig. 27: Calculate the route activity

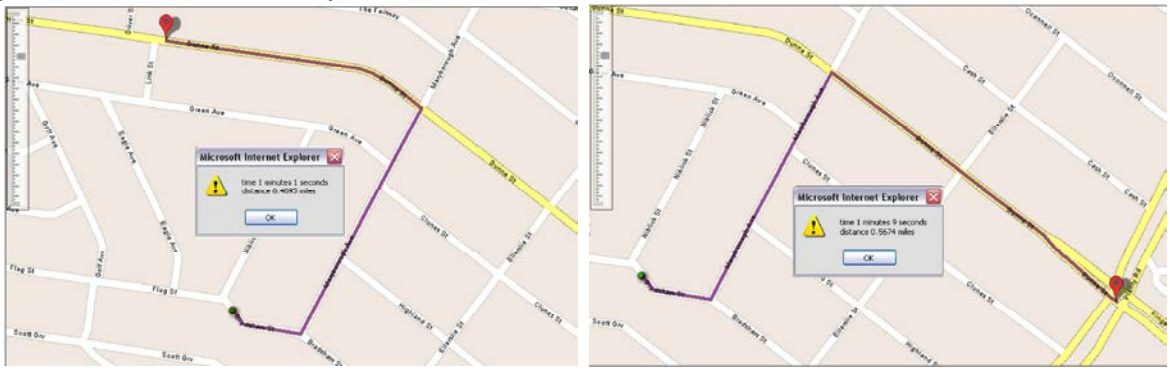


Fig. 28: Parcel pickup location map with time A and B

Fig. 29: XML request from the routing engine

Fig. 30: XML response from the routing engine

Fig. 31: Send job request activity

Fig. 32: User interface

Before something is sent to the fleet, the server has to determine the kind of mobile device the fleet is using. Checking it from the server repository about the kind of mobile device being assigned to the fleet, the server determines that fleet 1 is using Pocket PC. According to that the User Interfaces are customized as discussed in the Device Specification section of the case study. The User Interface received by the fleet is shown in the Fig 32.

On accepting the job, the server rechecks the device capabilities using the Apache Cocoon Server, the response from the server would be as follows:

```
User-Agent: Mozilla/2.0 (compatible; MSIE 3.02; Window CE; PPC; 240x320)
UA-OS: Windows CE(POCKET PC) - Version 3.0
Accept-Encoding: gzip, deflate
UA-CPU: ARM SA1110
UA-Pixels: 240x320
```

Then the "Forward Detail" activity is being executed with the details shown in Fig 33:

According to the activity, the fleet would be shown with the routing map showing the route between start and destination. The Text Instructions are shown in Fig 34.

At the end "Acknowledge Order Pickup" Activity would be executed where the fleet sends the acknowledgement to the server once the order is being picked up. The detail of the activity is shown in Fig 35.

The fleet confirmation screen is shown in Fig 36.

Coordination and Integration: The coordination and integration between the different components is shown in Fig 37 with the help of the sequence diagram. The order pickup scenario is adopted where user can specify the order pickup process and send for its deployment. In this sequence diagram it is showed that how deployment component interacts with other framework components

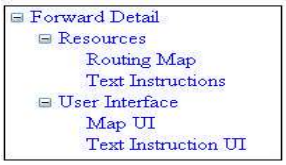


Fig. 33: Forward detail activity

Fig. 34: Fleet routing map with instructions

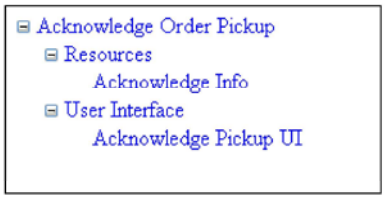


Fig. 35: Order pickup acknowledgement activity

Fig. 36: Fleet screen confirmation

and successfully able to deploy a pickup process. All the coordination and integration will took place in between two main engines, a) process specification engine and b) deployment engine. The deployment engine will only initiate its process once process specification is completed. All behind the scene technical processes are already explained and discussed in the proposed framework section.

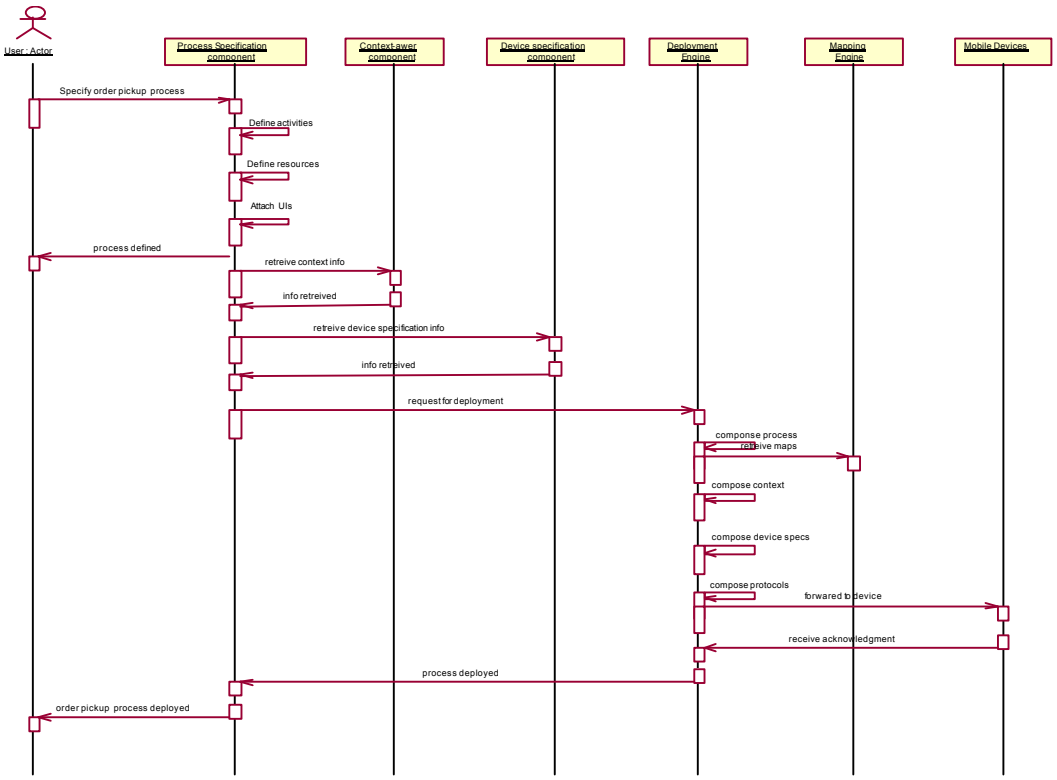


Fig. 37: Process deployment sequence diagram or order pickup service

It is also showed how the context information and device specification is retrieved from the database. It also portrayed how the business process is coordinated, integrated and customized with the context information and the device specification. It is also demonstrated the deployment of the business process in terms of activities, resources and user interfaces.

The case study implementation is managed to show deployment on only one device "Pocket Pc" due to limited resources. At the end, the coordination, automation and synchronization between the components is shown with the help a sequence diagram (Fig 37).

Evaluation: In this research, it is showed how a business process can be specified and deployed to a mobile device taking in notice the context and the specification of the device user is using. As being discussed in the previous sections screen resolution (User Interfaces) from the device specification will be adopted. It was assumed that the user interfaces with different screen resolutions would already be stored in the database. Making dynamic user interfaces is out of context for this research scope.

Evaluation on the proposed framework with the other existing models, architectures or applications is presented in this section. This research compare the framework with the "Web Services Deployment Framework" [38, 39]. The particular framework deploys a business process to a mobile environment through XML web services. BPEL is a language that is used to implement business processes in the framework [40].

Framework deploys the business process using the XML web services but proposed framework deployment approach also incorporates the software and hardware capabilities of the mobile device and context of the user. In proposed framework Business Process is specified along with its activities and resources. User interfaces are attached to each activity in a Business Process. Business Processes are stored as XML documents in a database. In proposed framework there is a component that stores the specification for mobile devices in the form of XML documents in the database. The context component stores the context information in this case are *location* and *environment* into the database. There is also a Deployment Engine that takes care of the process deployment. There is a composition layer that interacts with all the components so that a specified business process can be integrated and coordinated with a particular context and capabilities of the device where business process is going to be deployed. Depending on the context, the composition layer interacts with the

routing engine to retrieve the map. Protocol Layer then deploys the customized business process to a mobile device using XML web services and HTTP protocols.

There is another framework called "Content Adaptation for Heterogeneous Mobile Devices Using web-based Mobile Services". This framework talks about the concept for adapting web content to a very heterogeneous set of mobile devices by using Device Capability Database (DCDB) driven mechanism [44]. All the device specifications are stored into DCDB. This framework enables to deploy user interfaces to the mobile devices without the need to change user interfaces for each device. This kind of framework greatly reduces the maintenance complexity implied by the heterogeneity of the mobile devices. The setback for the framework does not take care of all the mobile devices. It only does the deployment for the devices on the GSM network. Extending to other networks is part of their future work. For Implementation this framework uses J2ME as the programming language. On the other hand proposed framework uses. Net platform as C#.Net and ASP.Net as the programming languages with XML web services which are tend to be far more efficient and faster as compared to Java platform. This research already presented the detailed about these implementation tools and techniques in section 2.2.

Proposed framework follows the use of scalable, flexible architecture. As being discussed in the earlier sections that proposed framework is divided into different components and system architecture is divided into three parts or tiers that are client tier, business logic tier and database tier. All interact with each other to achieve a specific goal. Proposed framework follows the use of reusable components. As can be seen from the design and the system architecture that have followed component based approach in which components can be reused. For example, a specification for a device stored in the database and the context information unless updated can be reused over and over again. Same goes for the Business Processes that would be stored in the database and can be deployed to different mobile devices.

Another advantage is customization, integration and synchronization of a business process in proposed framework. Integration, coordination or synchronization is done by the composition layer where it interacts with all the three components to get the required business process with the attached context and device capabilities. The composition layer also communicates with the routing engine so that map can also be deployed. The Business Process being deployed to the mobile device is customized according to the device and the location or context of the mobile device. The disadvantage for the proposed framework is that user interfaces are already stored in the local repository. The approach only gives reference to these user interfaces. The real thing would have been to create these user interfaces according to the specification stored in the repository or received from the device. Storing the user interfaces for each screen resolution can increase the size of the application and reduce the performance of the system.

The proposed framework achieved automation at two stages: first automation of mobile business process specification and second automation in process deployment. Automating a mobile business process is a very delicate task in terms of complex context-awareness and device independence. The framework approach incorporate automation at the point at which the business analyst finished defining process specification. In automated system, human intervention is required only when the business analyst specified at each activity of the process. The automated level varies at different processes – some processes are totally automated and some do not depend upon process specification. As already mentioned earlier, the word deployment means installing, executing and automating a process. This is a very critical and most difficult phase in terms of process execution. To deploy the process on mobile environment is most difficult in terms of mobile location, deployment environment, their culture, platform, communication channels etc. The proposed framework approach achieved this automation through composition layer in deployment engine. This framework approach is not automating deployment process of user interfaces as placed them on specific devices and simply refer them at process specification. Automation of user interfaces is part of future research work.

CONCLUSION

In this research, a novel framework has been presented that support mobile business process deployment in device independence and context-aware environments. In summary a user business process can be deployed to any mobile device in any location and platform. The proposed framework will specially help organization and IT developers to deploy similar business processes to diverse mobile devices without any changes to software configuration or graphical user interface. A considerable research has been done in the field of

context-aware and device independence, but contributions to this research is to compile characteristics together and present framework that can bring additional benefits to the supplier and user which is service subscriber.

In future work, this conceptual approach about context-aware and device independence will be developed to the next level. Additional enhancement can be Service Composition. A service composition refers to a functionality (services) drawn from several different processes within a service-oriented architecture. Composite services often incorporate process application logic to control how the composed services interact with each other to produce the new, derived functionality or service. In future research, this framework will incorporate the concept of service composition. Intelligent agents were also being incorporated to control different services operating within a complex business system.

REFERENCES

1. Swaminathan, J.M. and S.R.Tayur, 2003. Models for Supply Chains in E-Business. Management Science, 49: 1387-1406.
2. Adam, O., P. Chikova and A. Hofer, 2005. Managing Inter-orgnizational business processes using an architecture for M-Business scenarios. in International Conference on Mobile Business (ICMB'05).
3. Zhang, J.J., N.P. Archer and Y. Yuan, 2003. Driving Forces for M-Commerce Success. E-Business Management: Integration of Web Technologies with Business Models, 1(1): 51-76.
4. Paavilainen, J., 2002. Mobile Business Strategies: Understanding the Technologies and Opportunities International Journal of Electronic Commerce, Addison Wesley Professional, 1(1).
5. Kaasinen, E., 2003. User needs for location-aware mobile services. Personal and Ubiquitous Computing, 7(1): 70-79.
6. Siau, K. and Z. Shen, 2003. Mobile Communications and Mobile Services. International Journal of Mobile Communications, 1: 3-14.
7. Justfone. Justfone, 2006. 12/2/2007]; Available from: www.justfone.com.
8. Brown, S., T. Morton and A. Telecom, 2003. The Hub of Mobile Excellence.
9. Centerbase.com. Centerbase On-Premise Database Business Application, 2012. 07/11/2012]; Available from: http://www.centerbase.com/default.aspx?gclid=CLefm9_ikbQCFQx76wodJVwA5g.
10. Clancy, H., 2012. 14 DIY mobile app development resources for small businesses. Small Business Matters [cited 2012 08/11/2012]; Available from: http://www.zdnet.com/blog/small-business-matters/14-diy-mobile-app-development-resources-for-small-businesses/2288.
11. Liang, T.P. and C.P. Wei, 2004. Introduction to the special issue: Mobile commerce applications. International Journal of Electronic Commerce, 8: 7-17.
12. Kalakota, R. and M. Robinson, 2001. Trends Shaping the Mobile Economy. in M-Business: The Race to Mobility. Mc Graw- Hill.
13. Beaulieu, M., 2002. Building the Four Wireless Application Families of the Wireless Internet. [cited 2012 10/11/2012]; Available from: http://www.informit.com/articles/article.aspx?p=24905&seqNum=10.
14. Davenport, T.H., 1992. Process Innovation: Reengineering Work Through Information Technology, ed. H.B.S. Press.
15. Ali, S., T. Torabi and H. Ali, 2006. Location Aware Business Process Deployment. in The 2006 International Conference on Computational Science and Its Application, ICCSA. Glasgow, UK.: (LNCS 3983).
16. Ali, S., B. Soh and T. Torabi, 2005. A New Adaptive Business Model for E-Commerce. in Third International Conference on Information Technology and Applications (ICITA). Sydney, Australia.
17. Ritz, T. and M. Stender, 2003. B2B mobile business processes: scenarios and technologies. in 14th. International Workshop on Database and Expert Systems Applications (DEXA'03).
18. Berger, M., A.G. Siemens, Muenchen, Germany, M. Bouzid, M. Buckland, H. Lee, N. Lhuillier, D. Olpp, J. Picault and J. Shpherdson, 2003. An approach to agent-based service composition and its application to mobile business processes. IEEE Transactions on Mobile Computing, 2: 197-206.
19. Barkhuss, L. and A. Dey, 2003. Is context-aware computing taking control away from the user? Three levels of Interactivity Examined. in The fifth annual conference on Ubiquitous Computing (UBICOMP'2003).
20. Hakkila, J. and J. Mantyiarvi, 2005. Collaboratin in Context-Aware Mobile Phone Applications. in The 38th Annual Hawaii International Conference on System Sciences (HICSS'05).

21. Koolwaaij, J. and P. Strating, 2003. Service Frameworks for Mobile Context-Aware Applications. in Conference on e-Challenges. Bologna, Italy.
22. Chen, G. and D. Kotz, 2000. A Survey of Contxt-Aware Mobile Computing Research. Dartmouth College Hanover, NH, USA.
23. Fisher, P.J., 2012. Mobility Strategy – Remain Device Independent/Adopt HMTL5. [cited 2012 09/09/2012]; Available from: http://mobility-strategy.ideascale.com/a/dtd/Mobility-Strategy-%E2%80%93-Remain-Device-Independent-Adopt-HMTL5/89997-17523.
24. Mikhalenki, P., 2004. Introduction to Device Independence.
25. Dheap, V., 2012. A Device Independent Mobile Platform. [cited 2012 09/09/2012]; Available from: http://www-01.ibm.com/ software/ ebusiness/ jstart/asap/.
26. Butler Mark, Giannetti Fabio, Gimson Roger and Wiley Tony, 2002.Device Independence and the Web. IEEE Internet Computing, 6(5): 81-86.
27. Hwang, Y., J. Kim and E. Seo, 2003. Structure Aware Web Transcoding for Mobile Devices. IEEE Internet Computing, 7(5): 214-221.
28. Ali, S., T. Torabi and H. Ali, 2006. A Case for Business Process Deployment for Location Aware Applications. International Journal of Computer Science and Network Security, (IJCSNS), 6(8): 118-127.
29. Vanish Talwar, D.M., 2005. Qinyi Wu,Calton Pu,Wenchang Yan,Gueyoung Jung, Approaches for Service Deployment. IEEE Computer Society, IEEE Internet Computing(March-April 2005).
30. Richman, D.M., 2001. Model based process deployment. in The 20th. Conference on Digital Avionics Systems: DASC:IEEE.
31. IBM. Developing a webservices.xml deployment descriptor for JAX-WS applications, 2012. [cited 2012 02/11/2012]; Available from: http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=%2Fcom.ibm.websphere.express.doc%2Finfo%2Fexp%2Fae%2Ftwbs_jaxwsdeploydescriptor.html.
32. AG, S. Web services & soa standards--leading the industry in open standards adoption. Standards support 2012 [cited 2012 03/08/2012]; Available from: http://www.softwareag.com/corporate/products/wm/xml/default.asp.
33. Network, M.D., 2012. XML Web Services. [cited 2012 05/09/2012]; Available from: https://developer.mozilla.org/en/docs/XML_Web_Services.
34. Bazaar, D., 2010. Web Services: JSON vs. XML. [cited 2012 08/08/2012]; Available from: http://digitalbazaar.com/2010/11/22/json-vs-xml/.
35. Group, W.C., 2006. Web Services Architecture.
36. Hammadi, A.M., S. Ali and F. Liu, 2006. A new modelling approach to enhance reliability of transactional oriented web services. in The 8th International Conference on Information Integration and Web-based Application & Services (iiWAS 2006). Yogyakarta Indonesia.
37. Zhang, J., L.J. Zhang and J.Y. Chung, 2004. WS-trustworthy: a framework for Web services centered trustworthy computing. in IEEE International Conference on Services Computing (SCC 2004). 2004. Shanghai, China.
38. Austaller, G., J. Kangsharju and M. Muhlhauser, 2004. Using Web Services to Build Context-Aware Applications in Ubiquitous Computing. in Web Engineering 4th. International Conference, ICWE.
39. Gokhale, A., B. Kumar and A. Sahuguet, 2002.Reinventing the Wheel? COBRA vs. Web Services. World Wide Web (WWW).
40. Andrews, T., F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic and S. Weerawarana, 2003. Business process execution language for web services version 1.1 (Technical report)..
41. Taconet, C., E. Putrycz and G. Bernard, 2003.Context aware deployment for mobile users. in 27th. Annual International computer software and applications conference (COMPSAC).
42. Fjellheim, T., 2006. Over-the-air deployment of applications in multi-platform environments. in The Australian Software Engineering Conference (ASWEC).
43. Courtney, P.E., 2000. Sterling Commerce's e-Business Process Integration. EAI Journel, First Impression: pp: 75.
44. Schmohl, R., U. Baumgarten and L. Koethner, 2007. Content adaptation for hetrogeneous mobile devices using web-based mobile services, in In proceedings of the 5th International Conference in Computing and Multimedia (MoMM 2007). pp: 77-86.