Mobile Business Process Deployment in Device Independence & Context-aware Environment

Hassan Ali, Saqib Ali and Torab Torabi 1)

Abstract
Deployment of business process to a mobile environment in a chosen context and in a device independence environment is a difficult task. Business Process is desirable to be deployed to any location or any mobile environment the user tends to be in. The problem in business process deployment is in rapid changes in mobile technologies and user context. In this research we developed and implemented a mobile business process deployment framework that caters both for device independence and context aware environments. In this paper our focus is on how device independence and context awareness can be integrated into mobile business processes deployment. To present the viability of this framework, we have specified and then developed the business process for a logistic company where we adopt a simple scenario that uses users’ context and device specification for mobile business process deployment.

1. Introduction

“Mobile Business Process Deployment” is the new concern and desire of the companies that are evolving from electronic business to mobile business. The success of Mobile Business is dependent, not only to wireless technology; but it also requires substantial process deployment innovation. Organizations need to reshape their business processes and re-design their workflows to capture the new opportunities that are emerging as result of mobile technology.

According to [1], E-business is defined as “a business process that uses internet or other electronic medium as a channel to carry out business transactions”. In summary the mobile business can be defined as the exchange of goods, services, and information using mobile devices [5] [6]. The mobile phone market has exploded in recent years and the growth in mobile devices is set to continue. For this reason 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 these applications more available to the mobile users. Mobile Business is often described as the successor of electronic business and defined as the subset of it [2] [3]. Usually some researchers assume that any business operations performed via desktop computer can alternatively be carried out via wireless network. However mobile technology offers possibilities that are unique to the wireless world and it is not quite useful via fixed network. For example, providing the mobile users with the services that are location aware and context aware, which cannot be performed with a fixed internet connection [4].

1 Department of Computer Science and Computer Engineering, La Trobe University, Bundoora, VIC, 3083, Australia, ali.hassann@gmail.com, {saqib.ali,t.torabi}@latrobe.edu.au, Fax: (61 3) 9479 3060
The primary feature of mobile technology is the ability to reach people regardless of their location and for mobile users’ possibility to work from any location and at any time. Mobile Services are usually characterized by four features; namely drivers-mobility; reach-ability, localization, and identification [7] [8] [9]. Mobility is the central distinguishing feature of mobile technology. As mobile devices constantly accompany their users, people can receive and send data regardless of place and time. This feature is especially useful to logistic and supply chain companies that regularly need to reach their mobile workforce so to allocate various tasks to them.

An example of company introducing mobile business into its current applications is the Scottish company “Justfone” [10]. Another example of mobile business is an emergency plumbing service, receiving call from customers with a burst pipe. Using this system the call centre operator quickly determines the location of all plumbers within that geographic area and verifies their current status. The operator then simply selects the appropriate plumber’s online diary and assigns the job, automatically sending the details to the selected plumber’s handset. Once received, job acceptance and completion notification can then be signaled back in real time [11].

The success of the recent applications is not only based on the new technology, but also on its proper use [12]. Wireless applications would be advantageous only when they can be useful to their end users [7]. Mobile business value proposition originates from the fusion of the wireless technology with already available electronic business applications. On the other hand, the unique features of mobile business – mobility, and localization give rise to emergence of completely new applications and business models.

In this section we present a brief introduction on mobile business processes and emergence of technology. The next section would discuss the concept of mobile business processes. A company acquiring new technology and changing its applications from electronic business to mobile business does require its business processes to be mobile.

1.1 Mobile Business Processes

As the technology is changing very fast and the companies operate in complex environments that consist of thousand of processes, the business profit depends on efficient delivery of goods and services controlled by business process [13][14, 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]. Therefore according to Valiente and Can der Heijden [19], Mobile Business Processes can be based on these three assumptions:

- “Uncertainty of Location”.
- “Uncertainty of Location” is externally determined, and
- A corporation with external resources is needed in the execution of the process.
The first assumption is based on the concept of “uncertainty of location” by [19]. According to which the place of the execution of an activity can be different in different instances of the business process or the place can change during the execution of an activity. The second assumption deals with the uncertainty of the location which is caused by the external factors and the person executing the process has no freedom for choosing the place of the process execution. The third assumption restricts the “Mobile Business Process” to the necessity of cooperation with external resources, for instance being caused by the need for communication or coordination with other persons or interactions with other objects [17] [18][19].

1.2 Context-Aware Mobile Applications

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 [20]. An example of context-aware application is the scenario in which a consumer makes use mobile shopping. In the application a consumer registers a pre-shopping list (through web-based interface, or mobile phone) based on personal preferences or current position of a product in a supermarket. Whenever a consumer would enter a supermarket, the system would provide a suggested route (on his mobile device) that will guide a consumer directly to the products he wishes to buy. This kind of application can also benefit retailers, who by knowing the exact location of the consumer can allocate the workforce more efficiently, and can eliminate out-of-stock situations [21] [22] [44].

In the next section we are going to discuss the device independence and its role in mobile business process deployment. We use this research in development of our proposed framework.

1.3 Device Independence

In today’s world of rapidly changing technologies and the rise in the use of mobile environment has created a need for content adaptation. According to [23], 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. According to [24], “Content Adaptation has to be invisible to the end user”.

Device Independence technologies can be divided into three different categories namely: intermediate, client-side and server-side [25]. 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 colour images or changing it to the low resolution depending on the device display capabilities [26] [25]. In the client-side approach the content adaptation can occur in the device itself. The advantage for this approach is that the adaptation code has direct access to the device capabilities [25]. Client-side content adaptations can also occur within the content. An example of this can be the use of CSS which is used to style HTML elements [26] [25]. 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. An example of delivery context of a PDA’s HTTP request from a Pocket IE 2002 browser is shown below [25]:

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

Device manufactures and end users have different needs and expectations when it comes to Web Content. Device Independence is about trying to satisfy these needs and setting the path between the content and the end user by using the device capabilities [25]. As a result of device independence the content integrators do not need to develop content for every single device and the content can be integrated to different devices without accumulating more resources and effort.

1.4 Deployment

In terms of technology, the word “deployment” means “Installing, setting up, testing and running” [27, 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 even a process to other devices is a complex task which covers all the activities that have to be carried out from the completion of the development itself to the installation and maintenance of the application on the consumer devices. In V.Talwar et al and Rickman [28, 29], the authors have compared certain types of deployment techniques in terms of their scale, complexity, expressiveness, and barriers to first use. The deployment solutions handled by four different 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.

According to recent research, business processes are deployed through XML web services. A Web Service is defined as “a collection of protocols and standards used for exchanging data between applications or systems” [30]. 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 [31] [32] [33] [34]. 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 [35, 36] [37].

WSDL is used for describing the services available. It is a general framework based on XML for describing network services as collections of communication endpoints capable of exchanging
messages. 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 [35, 36]. 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 behaviour based on Web Services [38].

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 is becoming increasingly important to support. OTA delivery enables easy deployment and upgrades to the applications, thereby reducing the disrupting effect which installation of new applications and upgrades may have on mobile users. OTA delivery allows users to receive new applications and updates anytime and anywhere [39] [40].

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. The framework is designed to take into account the execution context such as the available resources or user’s terminal capabilities, in order to make the application accessible to the user and adapt to the execution context [39]. The framework is implemented with middleware technologies like CORBA [41] and SOAP [42] which facilitate the development of large scale distributed applications. These technologies allow an application to be described as an assembly of components which provides an opportunity to design new kinds of tools to deploy and connect the components of an application. SDI offers automatic deployment of multi-component applications and provides a deployment solution to customise installation and to adapt to device capabilities [39, 43].

In the next section we present our proposed framework for mobile business process deployment. We developed this framework to cater both for device independence and context aware environment.

2. A Framework for mobile business process deployment in device Independence and Context aware environment

In this section we present a novel framework proposed for business process deployment to a mobile environment taking in notice the context of the user and specification of the device it is using. To support this, architecture is being shown in Figure 1. Our proposed framework is consists of five different components, namely:

- Deployment component
- Process specification component
• Device Specification component
• Context-Aware component
• Mobile Interface component

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 Web Services and SOAP technologies.

The second component in our framework is “process specification component”. This is a traditional Business Process component consisting of activities, resources and user interfaces. Inside a Business Process there can be many activities, where each of this activity consists of different types of resources and a user interface is associated. It should be noted that not all activities would have user interfaces as shown in Figure 1. User Interfaces are stored as XML documents in the UI Database associated with Process specification component.

![Diagram](image-url)  
**Figure 1: Device Independence & Context aware mobile process deployment framework**
“Device Specification component” 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.

“Context-Aware component” consists of many parameters but for the purpose of simplicity, we only consider two parameters “Location and Environment”. The location of the Mobile User is determined using different technologies like GPS, GSM, or MPS as shown in location server defined in our 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 M number of mobile devices. The Mobile devices being used can be pocket pc’s, smart phone’s or PDA’s etc.

In the following section we are going to present a case study, where we adopt a scenario to deploy a business process on pocket pc.

3. Case study to implement deployment framework

In this case study we present a scenario for a company (Independent Logistics Company) that uses context-aware technologies and information intensive business processes to enrich its existing enterprise applications such as server dispatch and fleet management. The purpose of this case study is to show, how a business process can be deployed to a mobile environment (PDA, Smartphone or poked PC) from the standalone server taking into account the hardware specification of the device the user is using. The process can also be initiated by a mobile user requesting for the next available job.

The company (Independent Logistics Company) has different activities like order pickup, order delivery and invoicing. All these activities can be processed at different locations. Each location may have different set of processes and activities. The company has fleets that are being used to pickup 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 a package to be dispatched by entering its pickup 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 pickup 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 reaches to the server and sever determines the device user is using and deploys the business processes attached with user interfaces to the fleet’s device. If the user rejects the job then the server looks for the other fleet to pickup the package.

If the distance of the idle fleet and the customer require a long delay before pickup, then the server would automatically allocate the pickup to the busy fleet. Based on conditions and parameters the best suitable fleet is chosen in regard to the proximity of the location. Taking one scenario of order
pickup from the application, the server dispatcher looks for the fleet in the two suburbs of Kingsbury and Bundoora through GPS tracking technologies as shown in Figure 2. The client logs into the website, and enters its details for the order to be picked up.

The server shows the parcel pickup location on the map as shown in Figure 3. The parcel pickup location is shown as the green dot. The server now calculates the route from the available fleets to the order pickup location. For this scenario we will show route of number 1 and 2 fleets.

As we can see the result, fleet 2 is preferred over fleet 1. So the request should be sent to fleet 2. Before any request is sent to the fleet, the server has to determine the kind of mobile environment the fleet is using. Checking it from the server repository about the kind of device being assigned to the fleet, the server determines that fleet 2 is using Pocket PC with the following properties:

**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
The user Interface for job acceptance being sent to the fleet is customized according to the device capabilities. The user interface received by the fleet is shown in Figure 4.

On getting the job acknowledgement from the fleet, server rechecks the device capabilities using the **Apache Cocoon Server**. We are assuming that the fleet accepts the job; a Business Process for the order pickup is retrieved from the repository. All the user interfaces attached to the Business Process are customized according to the device capabilities. On accepting the job offer the fleet gets the second screen that is showing the route between fleet1 and the pickup location. In the second screen if the fleet wishes, he/she would also able to see the text instructions from the fleet location to the pickup location. At the end, when the fleet receives the parcel it sends the acknowledgement to the server.

![Figure 4: User interfaces received by the fleet](image)

4. **Conclusion**

In this paper we have discussed deployment of a business process to a mobile environment in a chosen context and in a device independence environment. Using the proposed framework we demonstrated that a Business Process can be deployed to any mobile device in any location or environment the user tends to be in. This framework helps companies and developers to deploy same business processes to different mobile devices without any software configuration or user interface re-designing complications. In comparison with earlier research in the context-aware and device independence, in this paper we the framework that integrates the deployment of business process with context-aware and device independence by customizing the process on the fly to user context and dynamically generate the user’s view based on the device specification.

In this paper, we have only tested the application using location as a context-aware parameter, but in our future work we plan to include other parameters of context-awareness. We also intend to take
our concept and idea about context-aware and device independence in business process deployment into the next level.

5. References


