Sudan University of Science and Technology
College of Postgraduate Studies and Scientific Research
College of Computer Science and Information Technology

Master of Computer Science Programme

Towards a Development of Automated-Adaptable Business-Process using Context-Aware and Event-Processing Concepts

نحو تطوير عمليات أعمال متكيفة ذاتياً بإستخدام مفاهيم البرمجيات الواعية للصياغ ومعالجة الأحداث

Submitted in partial fulfilment of the requirements for a degree of Master of Science in Computer Science

By:
Aimen Abdel Aziz Hussein Ali

Supervisor:
Ammar Thabit Zahary, PhD

May 2015

Dedication

To my family:

Mother Wafaa and father Abdel Aziz for supporting and teaching me what is important in life;

My sisters Rana and Aisha for making my life more beautiful!

I love you all …Thank you all.

Acknowledgement

I would like to acknowledge first to my supervisor Dr. Ammar Thabit Zahary and then to my family, friends, colleagues and those support me in my study and in completion of the complementary project, especially Mr. Ahmed Ibrahim Muhammed and Mr. Ashraf Khalaf Allah Abdella Hady.

Also, I would like to acknowledge Dr. Muhammed Othman Kambal, Prof. Eltahir Muhammed Haroun, and Prof. Shamboul Adlan Muhammed.

Abstract

In the capital markets, things happen fast. The markets move fast, positions change fast. Reacting quickly is the key to increase profit and/or managing risk. Enterprises need to adapt their processes to any observed changes or events in order to maintain their competitive levels and to provide better services.

Business processes are static by nature and, therefore, they cannot be modified at run time, and leaving them less useful in the new context. To change the behaviour of a business process, they need to be stopped first, modify it and redeploy it. By doing so, this tends to lose all data associated with the current executions of a running process.

To address these issues, this thesis presents a new reference model called Automated-Adapted Business-Process Using Context-Aware and Event-Processing (ABCE) that involves a new approach which allows system developers to design context-aware business processes in which context information can be considered as events monitored in run time. ABCE model developed in this thesis is based on a technique called Complex Event Processing (CEP) which is used to monitor the external working environment and capture relevant events. The captured events are then passed into the system to extract information from them. This extracted information is grouped with other information in order to create a business case with a high-level representation. Depending on the created business case, the adaptation process may starts. Business units are represented as software components which are communicated with each other via key performance indicators (KPIs) interfaces. Each business component (software component) has a set of KPIs with accepted values. The set of KPIs with their accepted values are used to invoke a business component. Business component interfaces are unique. Adaptation process starts on KPI violation. On KPI violation, business process execution changes and a request to appropriate business component will be sent. KPI values which are violated for a business process are accepted for another one. Each business component has one or more business services. When a business component is invoked, one business service is selected according to Service Level Agreement (SLA). After adaptation is

completed, the modified version of the business process will be sent for execution to an appropriate execution engine.

Finally, we validated our approach by implementing a public transport system using simulation software. The reason behind selecting a public transport system is to increase passengers' satisfaction by reducing the waiting time for buses and to increase buses owners' profit. We chose Radio Frequency Identifier (RFID) and Infrared technologies to monitor buses and passengers respectively. By using such technologies the system will be aware about the number of buses and the number of passengers in each passenger track in the station. And by employing supply/demand theory, the system will direct buses to passenger tracks with high demand. Doing so improves the system throughput by up to 40% and reduces passenger waiting time by up to 98%.

الخلاصة

في الأسواق الرأسمالية تحدث الأشياء وتتغير بسرعة. هذه التغيرات تشمل تغييرات في الأسواق بدخول سلع جديدة وظهور لاعبين جدد كما تشمل تغييرات في الحصص السوقية للشركات المتنافسة. الإستجابة السريعة لهذه التغييرات يؤدي إلى زيادة الأرباح وتقليل المخاطر. على ذلك فإنه يجب على شركات الأعمال جعل عملياتها قابلة للموائمة للتغيرات التي تطرأ على أسواق الأعمال للمحافظة على ميزتها التنافسية ولتقديد خدمات أفضل.

عمليات الأعمال هي ساكنة بطبيعتها، وبالتالي لا يمكن تعديلها في وقت التنفيذ أو التشغيل إستجابتة لمتطلبات عمل جديدة. لتغيير هذه الخاصية، يجب إيقاف عملية الأعمال أولا، تعديلها، ثم إعادة تشغيلها مرة أخرى. إلا أن ذلك يؤدي إلى فقدان كل البيانات المتعلقة بالتنفيذ الحالي لعملية الأعمال قيد التنفيذ.

لحل هذه المشكلة، يقدم هذا البحث نموذج مرجعي يسمى Automated-Adaptable Business-Process using Context-Aware and Event-Processing (ABCE). هذا النموذج المرجعي يقدم طريقة جديدة تسمح لمطوري النظم بتصميم وتطوير عمليات وحلول أعمال ذات دراية بالصياغ مبنية على مفهوم Complex Event Processing (CEP)؛ حيث تمثل معلومات الصياغ بأحداث في بيئة العمل الخارجية. النموذج المرجعي ABCE المقترح في هذا البحث يبنى على تقنية Complex Event Processing (CEP) والتي تستخدم لمراقبة بيئة العمل الخارجية وإلتقاط الأحداث ذات الصلة. الأحداث الملتقطة – بعد ذلك – تمرر إلى داخل النظام لإستخلاص المعلومات منها. هذه المعلومات المستخلصة تجمع مع معلومات أخرى مستخلصة من بيئة العمل الداخلية لإنشاء حالة عمل وتمثيلها بصورة مهنية. إعتمادا على حالة العمل المنشئة فإن عملية موائمة عملية الأعمال قد تستدعى. وحدات الأعمال – والتي يتم تمثيلها بمكونات برمجية بإستخدام نهج Component-Based Software Engineering (CBSE) – تتواصل فيما بينها بإستخدام ما يعرف ب Key Performance Indicators (KPIs). كل مكون أعمال يحتوي على مجموعة من KPIs مع قيم مقبولة. تستخدم PKIs مع قيمها في عملية إستدعاء مكونات الأعمال. مجموعة KPIs مع قيمها متفردة؛ بحيث لا يوجد مكونا أعمال لهما نفس المجموعة من KPIs مع قيمها. عملية موائمة عملية الأعمال تستدعى عندما إختلال في إحدى قيم KPIs لمكون أعمال معين في عملية الأعمال القياسية. عند حدوث الإختلال يتغير سير تنفيذ عملية الأعمال، ويتم البحث عن مكون أعمال لمواصلة عملية التنفيذ. قيم KPIs المختلة بالنسبة لمكون أعمال معين

تمثل قيم مقبولة بالنسبة لمكون أعمال آخر. كل مكون أعمال يحتوي على مجموعة من الخدمات. عندما يتم إستدعاء مكون أعمال معين فإنه يتم إختيار وتنفيذ إحدى خدماته بإستخدام ما يعرف بـ Service Level Agreement (SLA). عند إنتهاء عملية الموائمة فإن عملية الأعمال المعدلة يتم إرسالها إلى محرك تنفيذ مناسب للتنفيذ. قد يقدم مكون الأعمال أكثر من خدمة في نفس الوقت بترتيب تفضيلي ليتم إختيار الخدمة التالية في حالة حدوث فشل في تنفيذ الخدمة السابقة.

أخيرا قمنا بتجربة النموذج المرجعي المقترح عن طريق نظام مواصلات عامة بإستخدام برنامج محاكاة. السبب وراء إختيارنا لهذا النظام هو لزيادة الرضا لركاب موقف مواصلات الخرطوم بتقليل زمن الإنتظار للحافلات ولزيادة ربحية ملاك الحافلات ومشغل موقف المواصلات. قمنا بإستخدام تقنيات Radio Frequency Identification (RFID) والموجات تحت الحمراء Infrared لمراقبة حركة الحافلات والركاب على الترتيب. بإستخدام هذه التقنيات يكون بمقور النظام معرفة عدد الحافلات والركاب في كل مسار في الموقف وفي أي لحظة. بإستخدام نظرية العرض/الطلب الإقتصادية يقوم النظام بتوجيه الحافلات للمسارات التي تكتظ بالركاب. بتطبيق هذا النهج لمدة سبعة أيام متتالية لاحظنا تحسن في كفاءة النظام حتى 40% وإنخفاض في زمن الإنتظار بالنسبة للركاب حتى 98%.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Introduction

In the capital markets, things happen fast, markets move fast, and positions change fast. Reacting quickly is the key to increasing profit and managing risk. Enterprises need to adapt their processes to any observed changes or events in order to maintain their competitive levels and to provide better services. And as business processes evolve and become more complex, the data around them increases exponentially and there are more and more factors that can affect the correct performance or even the successful execution of the business processes.

The information of factors around the business process execution is known as *context information*. In order to capture context information, enterprises need to add monitoring activities throughout the business process definition to identify when special situations are developing, which could affect the execution of the business process (Hermosillo, 2012).

1.2 Problem Statement

Business process definitions are static by nature, which means that they cannot be modified at runtime. At the same time, there is a strong need to be able to adapt those processes dynamically, in order to respond to the changing conditions. Changing the processes manually takes considerably more time, and is more error-prone. Moreover, redeploying a modified business process would lead to downtime of the service and loss of information of all the currently executed instances of the process (Hermosillo, 2012).

There are many problems that prevent the successful execution of business processes, especially with high dynamically environments in which they are executed. The following issues have been identified:

- **Context Information.** Business processes lack the capabilities to constantly monitor the information around them. They have a limited scope on the information they can get, and the moment when they have access to it, beside the restraints of the specification.

- **Business Processes Nature.** Business processes are static by nature, which means that they cannot be modified at runtime. They were not meant to be dynamically changed, as they were thought simply as a predefined sequence of activities to achieve a goal.

- **Unnecessary Adaptations.** To reduce adaptation overhead, it is also necessary to prevent or reduce some adaptations to be made. Preventing or reducing adaptations optimise the total performance of the system by reducing the time and effort required for adaptation before executing the business process itself.

1.3 Research Question

"How to develop an automated adaptable business processes using context-aware and event processing concepts?"

The quest to answer this question has led to various sub-questions which needed to be addressed. These are:

1. How can enterprises get benefits of using *context-aware* and event processing concepts to improve their business processes?
2. How could business processes be automatically adaptable to the context?
3. What are the other approaches/technologies needed to be involved in the proposed solution?

1.4 Research Objective

Given the problems presented in Section 1.2, the objective of this research is to bring a solution to them by answering the questions in previous section. The plan is to *improve the execution of business processes* by providing them with *context-awareness* and *runtime self-adaptation*.

- **Context-awareness.** This objective is to make business processes able to constantly monitor the changes in the environment and detect when an adaptation is needed to continue an optimal execution. It will not affect the performance or the maintainability of the original process.

- **Run Time Self-Adaptation.** This objective means the ability to automatically adapt the business processes at runtime, without any downtime or loss of information. It is necessary to be able to rapidly respond to the changes in the context to guarantee that the business process is executed under the best possible conditions to achieve its goal.

- **Proactiveness.** This objective means the ability of system to prevent or reduce conditions of adaptation from occurring to reduce the needs for adaptation.

1.5 Research Scope

In combine with event processing, the main area of this research will be the context-aware, one of the *ubiquitous computing* fields, and how to get advantages of it in designing an automated adaptable business process. The research is about proposing a reference model and it will be validated using a public transportation system.

1.6 Document Organization

This document is divided into six chapters. The second chapter covers the domains in which the work takes place and reviews some related studies. The rest chapters present the proposed solution; including the methodology of the research, evaluation criteria, the reference model, a simulated results of implementation, and some recommendations and plans for future. These chapters are described below:

- **Chapter 2: Literature Review.** A brief introduction to the main domains of this research is presented, to allow a better understanding of the domain and context in which the work takes place, as well as the terminology and concepts presented in the subsequent chapters. It presents some studies that try to solve the lack of flexibility of business processes with different solutions.

- **Chapter 3: The Research Methodology.** This chapter presents the methodology of the research that aims to give the reader an idea about how the research can solve the lack of flexibility of business processes. It includes also criteria that can be used to evaluate the proposed reference model versus other solutions.

- **Chapter 4: ABCE Mode Implementation.** In this chapter the research objective is to present the implementation mechanism of the reference model ABCE. It explains how ABCE reference model works, discusses ABCE modules and processes including their inputs, outputs, and approaches and tools.

- **Chapter 5: Results Study and Evaluation.** Taking a real society problem, a novel contribution to the domain of public transportation will be presented in this chapter using the proposed reference model and simulation software. Evaluation results will be presented and discussed.
- **Chapter 6: Conclusions and Future Work.** This chapter presents the conclusions of the work, some recommendations, and plans for future work.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This research aims to add adaptability to business processes in an automatic manner at runtime. In other words, allowing business processes to be adapted to context information around them automatically and doing that at runtime. To achieve this goal, several technologies and approaches will be used to facilitate this task. This chapter serves to set a base for the reader in terms of research domains, which is followed by briefs of some related studies to the area of research.

The research objective in this chapter is to allow a better understanding to the domains being studied and the context in which the work takes place. This chapter presents context-aware and business process management domains in more details, as well as the terminology and concepts presented in the later chapters.

Another objective is to present some studies and contributions which are proposed to solve the problem of adapting business processes to their executing environment. Reviewing and studying those approaches allow both researcher and reader know about the trend of researching and where the researching has been reached to discover the potential areas of research.

2.2 Context Awareness

Humans are quite successful conveying ideas to each other and reacting appropriately. This is due to many factors, including the richness of the language they share, the common understanding of how the world works, and an implicit understanding of everyday situations. When humans speak with humans, they are able to use information apparent from the current situation, or *context*, to increase the conversational bandwidth. Unfortunately, this ability to convey ideas does not transfer well when humans interact with computers. Computers do not understand our language, do not understand how the world works, and cannot sense information about the current situation, at least not as easily as most humans can. In traditional interactive or desktop computing, users have an impoverished mechanism for providing information to computers, typically using a keyboard and mouse. Users translate what they want to accomplish into specific minutiae on how to accomplish the task, and then use the keyboard and mouse to articulate these details to the computer so that it can execute their commands. This is nothing like our interaction with other humans. Consequently, computers are not currently enabled to take full advantage of the context of the human-computer dialogue. By improving the computer's access to context, users can increase the richness of communication in human-computer interaction and make it possible to produce more useful computational services (Bardram, et al., 2010).

Many research areas are attempting to address this input deficiency, but they can mainly be seen in terms of two basic approaches (Bardram, et al., 2010):

- Improving the language that humans can use to interact with computers.
- Increasing the amount of situational information, or context, that is made available to computers.

The first approach tries to improve interaction by allowing the human user to communicate in a much more natural manner. This type of communication is still very explicit, in that the computer only knows what the user tells it. With natural input techniques such as speech and gestures, no other information besides the explicit input is available to the computer (Bardram, et al., 2010).

In human–human interactions, situational information such as facial expressions, emotions, past and future events, the existence of other people in the room, and relationships to these other people are crucial to understanding what is occurring. Since both human participants in such an interaction share this situational information, there is no need to make it explicit. This helps to explain why a driver finds it easier to talk to a passenger than to someone on a cell phone (with the passenger there is *grounding* without explicit communication). However, this need for explicitness does exist in human–computer interactions, because the computer does not share this implicit situational information or context. The goal of context-aware computing is to use context as an *implicit* cue to enrich the impoverished interaction from humans to computers, making it easier to interact with computers (Bardram, et al., 2010).

The process of building shared understanding between people is called *grounding* (Clark, et al., 1991).

Weiser coined the term *calm technology* to describe an approach to ubiquitous computing, where computing moves back and forth between the centre and periphery of the user's attention. To this end, the approach to context-aware application development is to collect *implicit* contextual information through automated means, make it easily available to a computer's runtime environment, and let the application designer decide what information is relevant and how to deal with it (Weiser, et al., 1997). This is a good approach because it removes the need for users to make all information *explicit* and it puts the decisions about what is relevant into the designer's hands. Furthermore, it is likely that most users will not know which information is potentially relevant and, therefore, will not know what information to provide. The application designer should have spent considerably more time analysing the situations under which their application will be executed and can more appropriately determine what information could be relevant and how to react to it (Bardram, et al., 2010).

The need for context is even greater when we move into ubicomp environments. Mobile computing and ubiquitous computing have given users the expectation that they can access *whatever* information and services they want, *whenever* they want, and *wherever* they are. With computers being used in such a wide variety of situations, interesting new problems arise, and the need for context is clear: *users are trying to obtain different information from the same services or systems in different situations*. Context can be used to help determine what information or services to make available or to bring to the forefront for users (Bardram, et al., 2010).

2.2.1 Context Concept

Most researchers have a general idea about what context is and use that general idea to guide their use of it. However, a vague notion of context is not sufficient; in order to use context effectively, we must attain a better understanding of what context is. A better understanding of context will enable application designers to choose what context to use in their applications and provide insights into the types of data that need to be supported and the abstractions and mechanisms required to support *context-aware computing* (Bardram, et al., 2010).

Schilit and Theimer refer to context as location, identities of nearby people and objects, and changes to those objects (Schilit, et al., 1994). Brown defines context as location, identities of the people around the user, the time of day, season, temperature, etc (Brown, et al., 1997). Ryan defines context as the user's location, environment, identity, and time (Ryan, et al., 1998). Dey enumerated context as the user's emotional state, focus of attention, location and orientation, date and time, and objects and people in the user's environment (Dey, 1998). Merriam-Webster defines context as "the interrelated conditions in which something exists or occurs." Brown defined context to be the elements of the user's environment that the user's computer knows about (Brown, 1996). Franklin and Flaschbart see it as the situation of the user (Franklin, et al., 1998). Ward views context as the state of the application's surroundings (Ward, et al., 1997). Rodden defines it to be the application's setting (Rodden, et al., 1998). Hull included the entire environment by defining context as aspects of the current situation (Hull, et al., 1997). Dey and Abowd (Dey, et al., 2000) define context as 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 application themselves. These definitions are either based on examples which are difficult to apply or on aspects with the impossibility to enumerate which aspects are important.

The researcher agrees with Schilit (Schilit, et al., 1994) about the important aspects of context they claim, which are: *where you are*, *whom you are with*, and *what resources are nearby*. They define context to be *the constantly changing execution environment*. They include the following elements of the environment:

- Computing environment — available processors, devices accessible for user input and display, network capacity, connectivity, and costs of computing.
- User environment — location, collection of nearby people, and social situation.
- Physical environment — lighting and noise level.

There are certain types of context that are, in practice, more important than others. These are location (where), identity (who), time (when), and activity (what). Location, identity, time, and activity are important context types for characterizing the situation of a particular entity. These context types not only answer the questions of who, what, when, and where, but also act as indices into other sources of contextual information (Bardram, et al., 2010).

2.2.2 Context-Aware Computing

Context-aware computing was first discussed in 1994 by Schilit and Theimer as software that *adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time*. But the first research investigation of context-aware computing was the Olivetti Active work in 1992 (Want, et al., 1992). Since then, there have been numerous attempts to describe context-aware computing.

The first definition of context-aware applications given by Schilit and Theimer expanded the idea of context awareness from applications that are simply *informed* about context to applications that *adapt* themselves to context (Schilit, et al., 1994). Context-aware also has become somewhat identical with other terms: *responsive* (Elrod, et al., 1993), *reactive* (Cooperstock, et al., 1995), *adaptive* (Brown, 1996), *situated* (Hull, et al., 1997), *environment directed* (Fickas, et al., 1997), *context sensitive* (Rekimoto, et al., 1998). Previous definitions of context-aware computing fall into two categories: *using context* and *adapting to context*.

Hull (Hull, et al., 1997) and Pascoe (Pascoe, 1998) (Pascoe, et al., 1998) (Ryan, et al., 1998) define context-aware computing as the ability of computing devices to *detect* and *sense*, *interpret* and *respond* to aspects of a user's local environment and the computing devices themselves. Dey (Dey, 1998) (Dey, et al., 1998) (Salber, et al., 1999) define context awareness as the use of context to *automate* a software system, to modify an interface, and to provide maximum *flexibility* of a computational service.

Many researchers define context-aware applications to be applications that dynamically *change* or *adapt* their behaviour according to the context of the application and the user (Schilit, et al., 1994) (Brown, et al., 1997) (Dey, et al., 1997) (Ward, et al., 1997) (Abowd, et al., 1998) (Davies, et al., 1998) (Kortuem, et al., 1998). Ryan defines them as applications that *monitor* input from environmental sensors and allow users to *select* from a range of physical and logical contexts according to their current interests or activities (Ryan, 1997). Ryan's definition has a little restriction by identifying the method in which applications acts upon context. Brown defines context-aware applications as applications that automatically provide *information* and/or take *actions* according to the user's present context as detected by *sensors* (Brown, 1998). He also takes a narrow view of context-aware computing by stating that these applications can act as information providers or to take some actions according to context using sensors. Fickas define environment-directed applications as applications that *monitor* changes in the environment and *adapt* their *operation* according to predefined or user-defined *guidelines* (Fickas, et al., 1997). Their definition focuses on operation adaptation according to according to some predefined or user-defined rules. Dey and Abowd define context awareness more generally with the following statement: a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task (Dey, et al., 2000).

2.3 Business Process Management

Business Process Management (BPM) is a collection of methods, policies, metrics, management practices, and tools used to design, run, and manage systems that support a company’s business processes (Chandy, et al., 2010). Business Process Management has received considerable attention recently by both business administration and computer science communities, to investigate business process properties and detect any deficiencies to propose necessary improvements. Business process management is based on the observation that each product that a company provides to the market is the outcome of a number of activities performed. Business processes, which can be performed manually or automatecally, are the key instrument to organizing these activities and to improving the understanding of their interrelationships (Weske, 2007).

2.4.1 Workflow Management Coalition (WfMC)

WfMC is a global standardisation organization founded in 1993. It includes adopters, developers, consultants, analysts, as well as university and research groups engaged in workflow and BPM. The WfMC creates and contributes to process related standards, educates the market on related issues, and is the only standards organization that concentrates purely on process. The WfMC created Wf-XML and XPDL, the leading process definition language used today in over 80 known solutions to store and exchange process models. XPDL is a process design format for storing the visual diagram and all design time attributes and product attributes (WfMC, 1995).

2.4.2 Workflow Terminology

In Table 2.1 and Figure 2.1, the workflow terminology will be presented as well as the relationships among them according to the Workflow Management Coalition.

Table 2.1: Workflow Terminology

Term	Definition
Workflow	is the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.
Workflow Management System	is a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.
Business Process	is a set of one or more linked procedures or activities, which collectively realise a business objective or policy goal. Normally it is made up within the context of an organization structure defining functional roles and relationships.
Process Definition	is a representation of a business process in a computerised form. The representation supports automated manipulation, such as modelling, or enactment by a workflow management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc..
Activity	is a description of a piece of work that forms one logical step within a process. An activity may be manual, which is not supported by computer automation, or a workflow (automated) activity. A workflow activity requires human and/or machine resources to support process execution: where a human resource is required, an activity is allocated to a workflow participant.
Automated Activity	is an activity which is capable of computer automation using a workflow management system to manage the activity during

	execution of the business process, which it forms a part of.
Manual Activity	is an activity within a business process, which is not capable of automation and hence lies outside the scope of a workflow management system. Such activities may be included within a process definition, for example to support the modelling of the process, but do not form part of a resulting workflow.
Process Instance	is the representation of a single enactment of a process including its associated data. It, therefore, represents an instance of a process definition that includes manual and automated aspects.
Activity Instance	is the representation of an activity within a single enactment of a process, for instance within a process instance.
Work Item	is a representation of the work to be processed by a workflow participant in the context of an activity within a process instance.
Invoked Application	is a workflow application that is invoked by the workflow management system to automate an activity, fully or in part, or to support a workflow participant in processing of a work item.

2.4.3 Business Process Management and Context-Aware

Business processes involve business events in a general sense. Conventional BPM engines control the flow of the process by evaluating conditions referring to the events that are generated by the application software in which the business process is running. In this way, BPM engines have a restricted knowledge of their environment, and they can take decisions based only on the information provided by the process, in contrast to real context-awareness, where the systems can benefit from a wide variety of information sources that provide it with the information to better understand its environment and execute accordingly (Hermosillo, 2012). To extend this awareness, the BPM engine can be complemented with an event processing engine. The EP engine can get information about the events that happen outside the business process from sources such as sensors, the Web, or other application systems. At the same time, the EP engine can get information about the events that are happening inside the business process from the BPM software, giving the system a complete context-awareness. From these base events, it can create derived events, that are then forwarded to the BPM software to enable sophisticated, context-dependent, situation-aware decisions (Hermosillo, 2012). Collaboration between BPM and EP was illustrated by (Chandy, et al., 2010) (Figure 2.2).

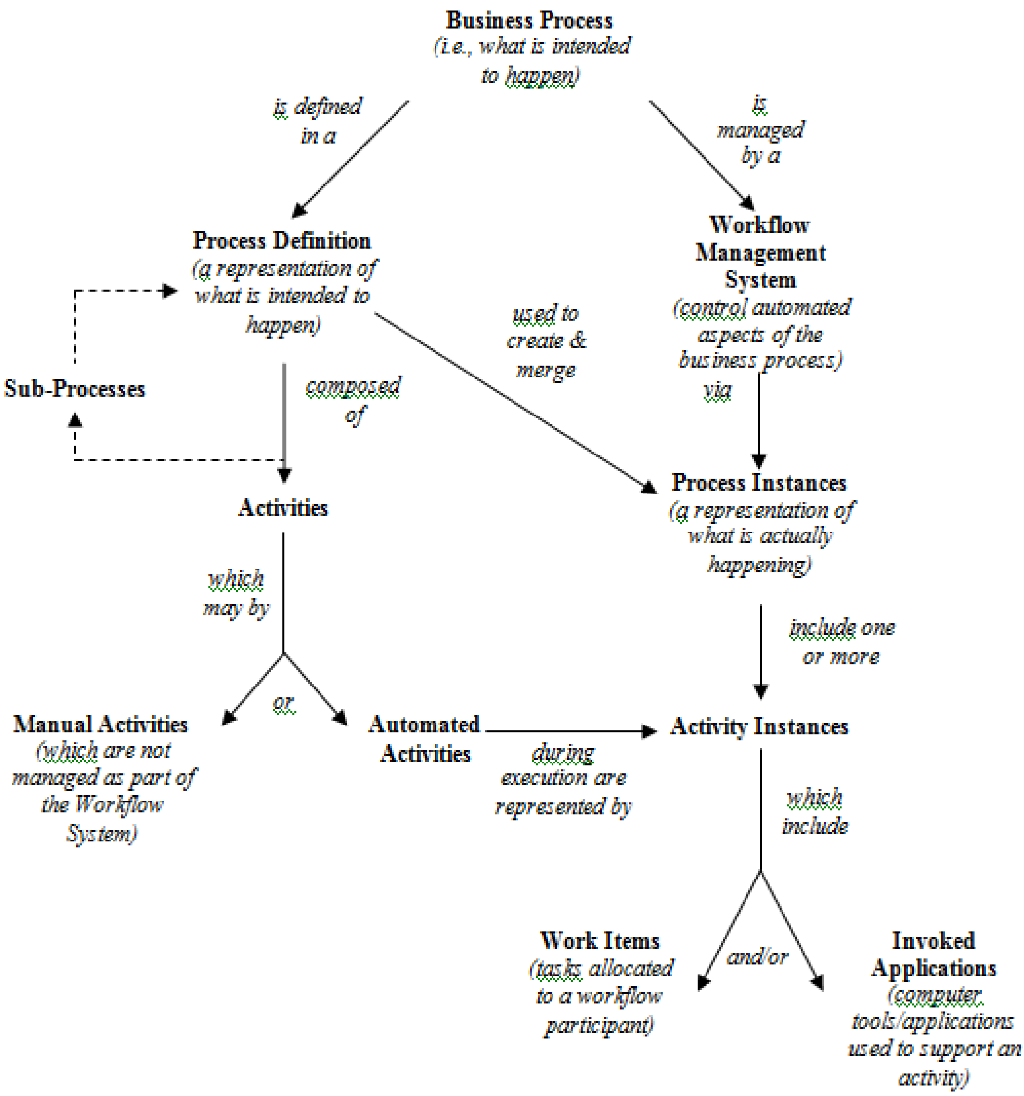


Figure 2.1: Business Process

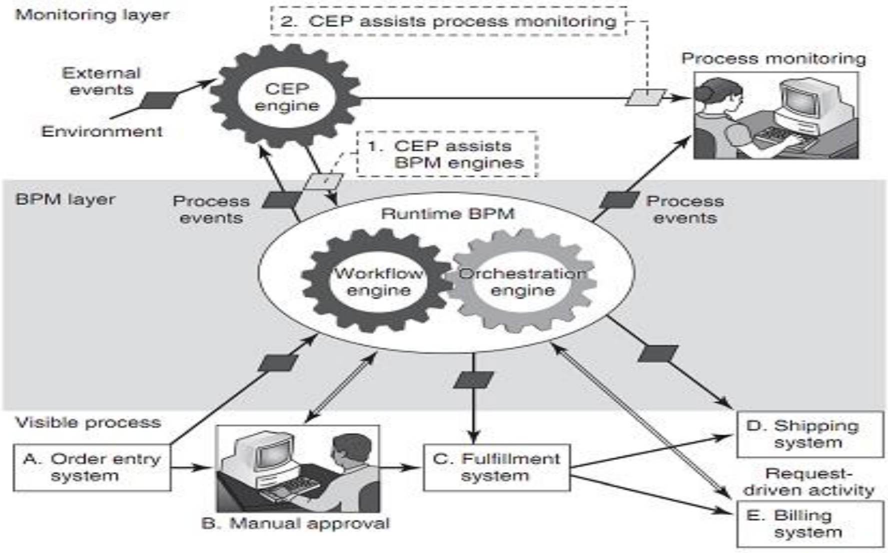


Figure 2.2: Using Event Processing with BPM

2.4 Related Work

In this section, some previous studies related to business process adaptations will be reviewed.

Authors in (Bucchiarone, et al., 2011) proposed a framework supporting context-aware evolution of business processes based on process instance execution and adaptation history. Instead of looking for recurring adaptations, it proposes to look for recurring adaptation needs. Based on the analysis of adapted instances, it automatically constructs and ranks corrective evolution variants which can handle the problematic context. At the same time, it tries to identify preventive evolution variants by constructing process variants which can prevent the adaptation need. It demonstrates the benefits of the approach using a car logistics scenario.

Authors in (Han, et al., 2014) present a building automation system adopting SOA paradigm with devices implemented by device profile for web service (DPWS) in which context information is collected, processed, and sent to a composition engine to coordinate appropriate devices/services based on the context, composition plan, and predefined policy rules. A six-phased composition process is proposed to carry out the task. In addition, two other components are designed to support the composition process: building ontology as a schema for representing semantic data and composition plan description language to describe context-based composite services in form of composition plans. A prototype consisting of a DPWSim simulator and SamBAS is developed to illustrate and test the proposed idea.

Authors in (Moon, et al., 2013) present an integrated architecture of context-aware business process management system based on ubiquitous computing technologies. By detecting the current health status of a patient using various ubiquitous devices such as RFID and smart sensors, the proposed system helps healthcare professionals provide personalized healthcare services.

Authors in (Bucchiarone, et al., 2011) provide an adaptation approach that can automatically adapt business processes to run-time context changes that impede achievement of a business goal. It defines a formal framework that adopts planning techniques to automatically derive necessary adaptation activities on demand. The adaptation consists in identifying recovery activities that guarantee that the execution of a business process can be successfully resumed and, as a consequence, the business goals are achieved. The solution proposed is evaluated on a real-world scenario from the logistics domain.

Authors in (Costa, et al., 2005) present a generic context architecture that is adaptable for all kinds of applications has been presented. The main contribution of this proposal consists on the support for the introduction of ubiquitous computing paradigm into applications. The objective of this paper is to remove the burden of obtaining information from sensors, locating and registering sensorial devices and managing stored contextual information from the applications themselves, so that application developers can focus only on application specific problems and on what to do with the information provided by the context system.

Authors in (Ardagna, et al., 2007) define a flexible process as one that can change its behaviour dynamically according to variable execution contexts. Authors developed a framework for flexible and adaptive execution of managed service-based processes - PAWS. The framework coherently supports both process design and execution. It selects the best available services for executing the process and defines the most appropriate quality-of-service levels for delivering them, and guarantees service provisioning, even in case of failures, through recovery actions and self-adaptation if the context changes.

Author in (Hermosillo, 2012) presents the CEVICHE Framework. He brings forward an approach which allows representing context-aware business processes where context information is considered as events which are monitored in real-time. He bases his work on a technique called Complex Event Processing. By using an external tool to monitor the context in real-time, he is able to surpass the limit of only accessing the information on specific places of the process. With CEVICHE he integrates the information obtained from the context with the capability of adapting business process at runtime. Also, one of the original contributions of the CEVICHE Framework is the definition of a correct adaptation undoing mechanism and its implementation. CEVICHE Framework, by using a component-based approach, offers flexibility and dynamicity to the business processes, allowing the modification of their bindings at runtime. He defines a simple language, the Adaptive Business Process Language (ABPL), as a pivot language, and uses a plug-in approach that allows the events defined in ABPL to be used in virtually any CEP engine. With ABPL, CEVICHE Framework facilitates the use of CEP without the drawbacks of early adoption.

Authors in (Hinz, et al., 2007) present an approach to dynamically adapt pipeline-based Web architectures according to the server and client context. A cluster-based distribution mechanism was proposed that distributes client requests according to varying load rates that are monitored permanently. The mechanisms facilitate optimizing the efficiency of the system architecture and reducing server load as well as fulfilling specific QoS requirements of the pipeline and its components. They were implemented and have proved their feasibility in different usage scenarios.

Authors in (Kostakos, et al., 2010) have a motivation of improving public transport's energy efficiency. A key factor contributing to public transport's current inefficiency is low seat occupation. This problem can be addressed by having accurate and frequently-updated OD matrix data to support an efficient and timely service, something which is currently too expensive for bus companies. They chose to improve this data collection process by developing a system that cheaply and automatically collects more data about passengers' travel behaviour than previously possible. The major output of this work has been the passenger OD matrix, and the subsequent graphs and analyses that can be derived from it.

Authors in (Antony, et al., 2013) attempted to develop a more comprehensive clinical data entry and patient monitoring system which in turn can relieve the clinical staff from lot of redundant activities. The Apps which are being developed focussing on ICU work flows will make life easier for them so that they will be able to concentrate on their primary objective that is to provide the required care. The conventional patient monitoring systems are not ubiquitous and thus will add to the delay in providing required care to the patient at the right time. Interfacing of wearable sensor devices to the central server will make the data available to the entire clinical team in a faster manner so that critical decisions are made faster. The decision support system under development, consisting of a classifier, context aware medical rule set, and smart alarm generator, can analyse the cross functional effects of different parameters and can generate alerts and possible advices based on a context aware rule set. These alerts will be generated well in advance so that care takers can provide medications earlier thereby saving the life of the critical patient.

2.5 Summary

In this chapter, some of the concepts that will be used throughout the rest of the thesis were introduced. These concepts are divided into two main groups: first group was about ubiquitous computing and context-aware; the second group was about business process and the benefit of using context-aware with it. This chapter, also, reviewed some recent contributions and approaches to business process adaptations.

In the next chapter, the objective is to present the research methodology to business process adaptation and the working environment. Some comparison criteria that can be used to evaluate adaptation approaches will be listed.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

One of the most powerful competitive tools business companies must have is the ability to adapt their business processes to any conditions in the marketplace. The adaptation enables the business process to be executed and achieved specified goals even in case of changing or unpredictable environment. Researchers have start proposing their approaches to make business processes adaptable to their environment for a few years ago. In this chapter the objective is to briefly introduce a new approach as well as the approaches and technologies involved.

The research objective of this chapter is to present a methodology that will be followed during the rest of the thesis based on the reviewed studies and technologies.

3.2 Adaptation Approaches

Approaches to business process adaptability, including those reviewed in Chapter 2, could be classified into vertical approaches and horizontal approaches. Each of these approaches has its own advantages and disadvantages (e.g. according to the type of applications in which they are used). But there are little approaches that trying to add adaptability to business process both vertically and horizontally. This thesis proposes a reference model that is able to adapt business processes vertically and horizontally.

3.2.1 Vertical Adaptation Approach

This approach refers to the changes done at a service-level that do not affect the structure or sequence of execution of the process. Works of this type of adaptation have a goal to improve the Quality of Service (QoS) and prevent Service Level Agreements (SLA) violations (Hermosillo, 2012).

The objective of adaptation approach is not to modify the behaviour of an existing process, but to give some assurance that the process will be successfully executed, and that the service provided by the partners of the process will comply with the predefined criteria (Hermosillo, 2012). Most contributions to business process adaptation classified as vertical adaptation approaches.

Benefits of the Approach

- Less invasive approach. It doesn’t need to modify the business process execution engine.

- Usually faster.
- Many choices with a dynamic pool of service providers.

Limitations of the Approach

- Limitation of flexibility. It is focused to maintain the QoS of the process and not really to respond to new needs.

Evaluation Criteria of the Approach

Table 3.1 lists some evaluation criteria for vertical approach solutions:

Table 3.1: Vertical Adaptation Approach Evaluation Criteria

Criterion	Description
Change service provider	It is the run time capability to dynamically change the binding of the service from one provider to another one.
Change activity behaviour	It is the run time capability to change, not only the service provider, but also the type of service that would be called.
Context monitoring	It is the capability to monitor the conditions under which it would adapt the process.
Context scope	It is the limits of the information sources to be considered for the context monitoring.
Automatic adaptation	It is the capability to react to the adaptation conditions and automatically make the necessary changes to the business process without any human interaction.
Proactive adaptation	Adaptations can either be proactive or reactive. If the adaptation is triggered in response to a specific event, then it is a reactive adaptation. Otherwise it is a proactive.
Adaptation mechanism	It is the specific approach used by the solution to achieve the adaptation of the business process.

3.2.2 Horizontal Adaptation Approach

This approach refers to the modifications made to the structure of the business process, which include the addition and removal of activities from the process to change its behaviour. Horizontal adaptations allow a broader manipulation of the business process, whereas vertical approaches are limited at service-level modifications (Hermosillo, 2012).

The goal of this adaptation approach is to allow business processes the ability to respond to new needs by modifying its behaviour without the need to redeploy them (Hermosillo, 2012). There are few contributions to this type of adaptation.

Benefits of the Approach

- Redeployment elimination. It has the ability to modify the process structure completely without the need to redeploy the process.
- Separation of concerns. It deploys only the main business process and then adapting the process to cover different concerns. This helps in the maintainability of the main process and makes it less complicated.
- Supporting vertical adaptation. Since the capability of changing the activity behaviour can also be used to modify the service providers.

Limitations of the Approach

- The need for customised engines. It needs a customized engine that can support dynamic reconfiguration. This hinders their usability when trying to work with standard solutions, since the new engine should be able to deal with an extended syntax that considers the adaptation points.
- Time consumption. It may consume more time than doing vertical adaptation depending on the kind of process to be adapted.

Evaluation Criteria of the Approach

Besides the criteria used to evaluate the vertical approaches, some criteria have been added (Table 3.2) to evaluate horizontal approach solutions:

Table 3.2: Horizontal Adaptation Approach Evaluation Criteria

Criterion	Description
Add Activity	It is to the capability to dynamically add new activities to the business process without the need to redeploy it.
Delete Activity	It is to the capability to dynamically delete existing activities from the business process without the need to redeploy it.
Adaptation Undoing	It is the capability to undo the changes done to the business process when the condition that caused the adaptation is no longer valid.

Following sections will introduce a reference model to business process adaptation – with the employed technologies and approaches. But before that the concept of reference model will be explained in the next section.

3.3 Reference Model

A reference model is an abstract framework for understanding significant relationships among the entities of some environment. It enables to develop a specific reference or concrete architectures using consistent standards or specifications that support that environment. A reference model consists of a minimal set of unifying concepts, axioms and relationships within a particular problem domain. It is independent of specific standards, technologies, implementations, or any other concrete details (MacKenzie, et al., 2006).

3.4 The Reference Model ABCE

The proposed approach called Automated-Adaptable Business-Process using Context-Aware and Event-Processing (ABCE) which has been developed as a component-based and service-oriented and uses context-aware and event processing approaches as adaptation mechanisms. This reference model provides developers a way to create context self-adaptable applications using event processing approach.

Because of the static nature of business processes and disability of cop with rapidly changing in business requirements at run time, the proposed reference model aims to provide business processes the ability to be context aware by modelling the environments in which business processes operates in and then monitoring these environments at run time by using event processing approach. Modelling and monitoring working environments make business processes aware with current business requirements. Business process must be also able to modify its behaviour as well as provide better services according to the changing working environment and doing so at run time and that is the second aim of this reference model. This ability improve the way of doing business by eliminate the time needed to stop, modify, and then redeploy business processes. The reference model aims also to add the ability to analyse the performance of execution of business processes. This ability optimises business process execution by reducing the time needed for business process adaptation and to handle any failure that can be occurred during execution. The reference model consists of the following modules:

- **Context Specification.** It responsible for determining what context-aware behaviours the application will have and in which situations each behaviour should be executed and how.

- **Context Acquisition.** It responsible for capturing real situations from the surrounding environment and creating of business case representations, which is then provided to system components for further use.

- **Adaptation Definition.** It helps the business process to define the adaptation needs, by responding to adaptation questions: What to adapt?, When to adapt it?, Where to adapt?, and How to adapt it?.

- **Process Adaptation.** It handles the actual manipulation of the business process. It manages the selection and binding of components and services to the business process to develop adapted business process alternatives.

- **Performance Analysis.** It handles the actual execution of the business process. Also, it analyse the performance of the executed business process weather it have been executed successfully or unsuccessfully. The analysis aims to upgrade the business process by calculate the probability of adaptation need occurrence.

3.4.1 ABCE Features

In addition to the advantages of using reference models, the main advantage of this proposed approach is *to add the ability of being self-adapted to business processes at runtime*. This ability is achieved by provide a system with one or more standard business processes at the *Design Time* – where the actual operation is not started yet. The system uses one of those standard business processes to derive a customised one at Run Time. A standard process is created by a Process Designer at Design Mode, whereas derived business processes are created automatically by the system itself at *Run Time* – where the organisation is in Operation Mode.

The second advantage is *to give a business process the ability not only to be reactive* to the environmental events *but also to be proactive*. Being proactive helps organisations to predict future business requirements depending on mining and analysing related data sources. Proactiveness increases system throughput by reducing the time needed for predictable business process adaptations. Proactiveness is useful also in systems where creating new business processes takes considerable amount of time.

Another advantage of this approach *is to support both vertical and horizontal adaptations*. Adaptation process starts by defining adaptation problem. After defining the adaptation problem, the system looks up vertically for the most suitable business process variation to deal with the problem. Vertical adaptation is useful with adaptation problems that occur *frequently*. Not surprising to find that the retrieved business process needs some adaptations. In such case, the retrieved business process variation is passed to the next step with the necessary modifications. The modification could be vertically at the service level or horizontally at the component level. Horizontal adaptation used with problems that occur *not frequently*.

One another advantage of this approach is to *be self-healing* by detecting failure during a business process execution and apply appropriate actions to let the business process terminate successfully.

3.4.2 ABCE Modules

In this approach, business process passes five stages, each of which is accomplished by a module. Each module consists of a group of related processes. Each process produces some outputs by processing some inputs using some tools and approaches. These outputs are passed to the next process for further processing. By executing all processes successfully, a business process can adapt to a specific context.

3.4.3 Component-Based Software Engineering (CBSE)

Component-Based Software Engineering (CBSE) is a new trend in software development. The main idea is to reuse already completed components instead of developing everything from scratch each time. Use of component-based development brings many advantages: faster development, lower costs of the development, better usability, etc. (Crnkovic, et al., 2001).

A component-based system is typically defined as n-tier structure, where n is a number of tiers or layers (mostly three). A tier or a layer is a part of the application which provides a specific functionality (also called business logic) and has a well-defined interface to other layers. Figure 3.1 shows an example of three-tier architecture. The lowest level represents data repository (i.e. relational database). The middle level presents the business logic (i.e. functional and computational part of the application), where data accessed from database is manipulated. The top tier presents a user interface (i.e. input forms and reports). Dividing applications in these levels makes them independent of each other as much as possible. This in turn enables more flexibility for reusing standard components, or updating parts of the application (Crnkovic, et al., 2001).

A Component Definition:

There are many definitions to component (Crnkovic, et al., 2001). Some of them are:

- A run-time software component is a dynamically bindable package of one or more programs managed as a uni t and accessed through documented interfaces that can be discovered at run time.
- A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third party.
- A Business component represents the software implementation of an autonomous business concept or business process. It consists of all the software artefacts necessary to express, implement and deploy the concept as a reusable element of a larger business system.

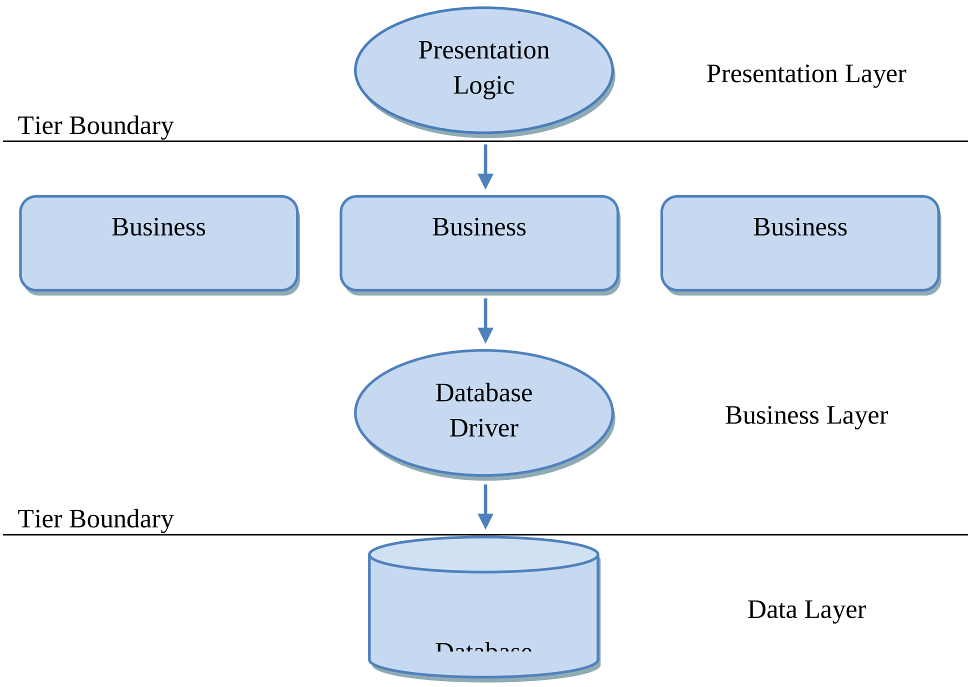


Figure 3.1: Three-tier Architecture

3.4.4 Service-Oriented Architecture (SOA)

A Service-Oriented Architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. And hence some means of connecting services to each other is required (Deori, 2010). SOA is defined by the Organisation for the Advancement of Structured Information Standards (OASIS as: *A paradigm for organizing and utilising distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations* (MacKenzie, et al., 2006).

SOA Collaborations

Figure 3.2 illustrates the entities in a service-oriented architecture that collaborates to support the 'find, bind and invoke' paradigm (Deori, 2010).

- **Service Consumer.** The service consumer is an application, a software module or another service that requires a service. It enquiries a service from the registry, binds to it over a transport, and then executes it according to an interface contract.
- **Service Provider.** It is a network-addressable entity. It publishes its services and interface contract to the service registry so that the service consumer can discover and access the service.
- **Service Registry.** It contains a repository of available discoverable services and allows for the lookup of service provider interfaces to service consumers.

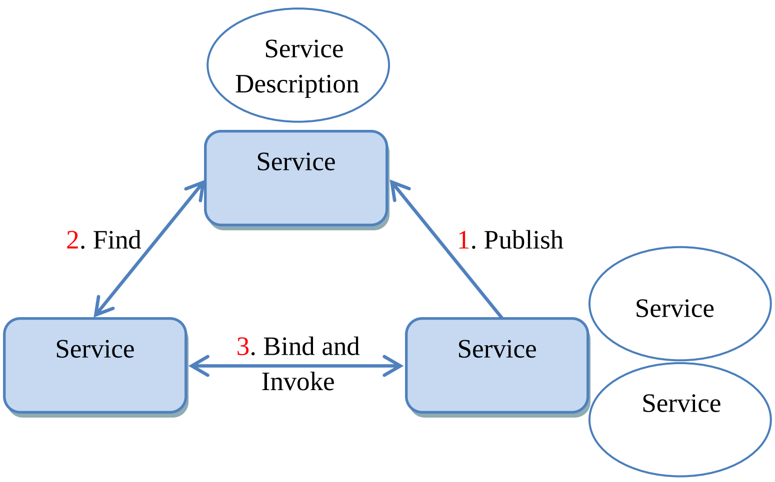


Figure 3.2: SOA Collaborations

SOA is based on three major technical concepts (Josuttis, 2007):

- **Service**. It is a piece of self-contained business functionality, which might be simple or complex.
- **Interoperability**. It is the ability of several systems to connect with each other and communicate successfully. Using enterprise service bus (ESB) makes it easier to distribute business processes over multiple systems using different platforms and technologies.
- **Loose Coupling**. It is the concept of reducing system dependencies. It refers to the amount of knowledge that one module has over another one in a system.

According to (Erl, 2007), Service-Oriented Architecture provides some features. Some of them are:

- **Standardized Service Contract.** Services adhere to a communications agreement, as defined collectively by one or more service-description documents.
- **Service Loose Coupling.** Services maintain a relationship that minimizes dependencies and only requires that they maintain an awareness of each other.
- **Service Abstraction.** Excepting its metadata and descriptions in the service contract, service hides its logic from the outside world.
- **Service Reusability.** Logic is divided into services with the intention of promoting reuse.
- **Service Autonomy.** Service has control over the logic it encapsulates.
- **Service Granularity.** It is a design consideration to provide optimal scope of business functionality in a service operation.
- **Service Statelessness.** Services minimize resource consumption by deferring the management of state information when necessary.
- **Service Discoverability.** It is the ability of services to be discoverable and interpreted by their associated metadata.
- **Service Composability.** It is the ability of services to be composited effectively regardless the size and complexity of the composition.

The process of developing applications as a collection of components and services, that exchange information via request/reply or events, is referred to as Service Component Architecture (SCA). SCA applications consist of a coalition of Service Components that communicate either via events (EDA) or via request/reply calls (SOA). SCA offers several key advantages over the traditional approach of monolithic application design:

- **Flexibility of Development.** Service Components are easier to develop because the semantics of each independent Service Component are significantly less complex than the overall of a single, (relatively large) monolithic application.
- **Reuse.** Since each Service Component has well-defined interfaces, each component can be developed, tested and debugged independent of the other components.
- **Dynamic Deployment and Runtime Modification/Replacement.** Service Components can be dynamically deployed to remote nodes at runtime, and components within a process can be easily replaced by new or updated components, further reducing the time taken to modify or change an existing process in response to business requirements.
- **Configuration Management and Version Control.** Service Components facilitate version control and dynamic configuration management, allowing fine-grained control over deployments across the enterprise.

3.4.5 Event Processing Approach

The concept of Complex Event Processing (abbreviated as CEP) evolved in the 90s (1989 – 95), and the term was coined by Prof. David Luckham. It was defined as a technology for extracting information from message-based systems. It is a technology for low-latency filtering, correlating, aggregating, and computing on real-world event data. It is an emerging network technology that creates actionable, situational knowledge from distributed message-based systems, databases and applications in real-time or near real-time (Deori, 2010).

A complex event processing system enables organisations to process distributed business events and identify opportunities or threats. Business events may be tracked individually, such as a stream of stock trades, or correlated with other events, producing derived or *complex* events often called *situations* (Deori, 2010).

Complex event processing software allows to process and analyse multiple streams of high-volume, high-speed business and system events, and to uncover opportunities and threats as they happen. It can be applied to extracting and analysing information from any kind of distributed message-based system (Deori, 2010).

CEP correlates multiple messages within given time frames. It is a technique to process message streams. These messages do not need to represent business events. A business event is something that happens when your business has planned to react in a predefined manner. A business event is represented by a message, but not all messages are representations of business events. (Deori, 2010).

There are three main roles in event processing (Hermosillo, 2012):

- *An event producer* is the entity that introduces the event to the system, and can be represented as physical or virtual entities.
- *Event consumers* are the ones that receive the events coming from the system and can be physical or virtual entities. An entity can be at the same time an event producer and an event consumer.
- *The event processing agent* is a software module that stands in the middle of the event producers and the event consumers. It analyses and processes the event, and can use its information for filtering, pattern detection, or for transforming to create new events.

Therefore, events can be of two types: raw events, and derived events. A raw event is the one that is introduced into the system by an event producer, and it is related only to its source and not to its structure. A derived event is the one that is generated as a result of event processing that takes a place inside an event processing system. The type of event is relative to the system where it is being considered, thus a derived event that is sent to a second event processing system will be perceived by it as a raw event (Hermosillo, 2012).

Event processing considers every change in the environment as a simple event and helps the user to specify a set of rules that will be used to create higher level events, called complex events that relate more to the business logic. Using event processing in combination with context-aware, allow organisations to find the information that is important for a specific application in run time and in an automated manner. This information can then be used to adapt business processes in order to respond to the new conditions in the environment.

3.4.6 ABCE Implementation and Evaluation

ABCE reference model consists of fourteen abstracted processes. To make things simple, we implement each process by a function with the same name. Some functions are implemented in sequence and some are implemented in parallel toward forming context-adaptable business process.

For simulation purpose, we built simulation software using Visual Basic.NET 2008 and MS-Excel within MS-Office version 2007. The development tools used are not a part of the proposed reference model. They are used for simulation purpose only.

The software is to simulate a public transport system. It monitors the amounts of buses and passengers in a bus station. Depending on the monitoring information, the system will react according to predefined conditions. Monitoring information can be used also to make the system proactive.

We employ the law of supply and demand in our system. In the law of supply and demand, producers must *supply* the goods that consumers *want* (Figure 3.3). *Supply* refers to the amount of goods a market can produce, while *demand* refers to the amount of goods consumers are willing to buy. In our system, supply and demand are represented by buses and passengers respectively.

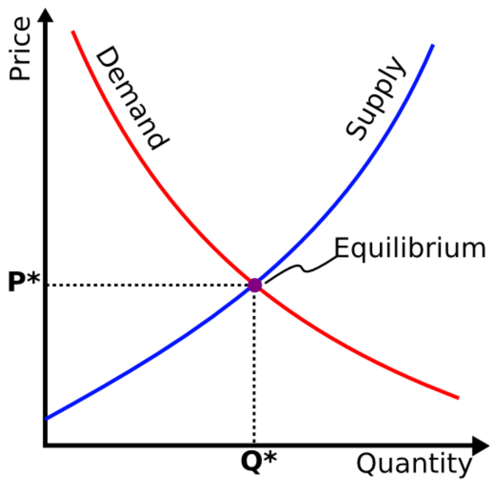


Figure 3.3: Supply and Demand Relationship

We use two technologies for monitoring. We use RFID technology to monitor buses and infrared technology to monitor passengers. Monitoring information will be saved in a log file. The system uses just-monitored information for reactive adaptation. In addition to just-monitored information and by applying data mining techniques, information stored in the system log file will be used for proactive adaptation.

Supply (buses) is determined by using the RFID technology. Each bus is identified by a unique radio frequency identifier. Each bus has a chip fixed in its body. When a bus enters the bus station, it passes under an RFID detector (the buses count is incremented by one). Also, when a bus exits the station, it passes under another an RFID detector (the buses count is decremented by one). In both cases a passing event is created in the log file. Using RFID detectors on entrance and exit gates, the system can determine how many buses in the bus station at each moment.

The Infrared technology is used as a demand (passengers) determinant mechanism. The system can determine the number of passengers by setting up two infrared devices in each track. When a passenger enters a track, the infrared ray of the first device will be cut (the passengers count is incremented by one). Also, when a passenger getting into a bus, infrared ray of the second device will be cut (the passengers count is decremented by one). In both cases a passing event is created in the log file. Using Infrared technology, the system can determine how many passengers in each track at each moment.

When a number of passengers in any track reach a specific level that means there is a demand in that track. As a consequence, buses to that track must be supplied. Doing so, buses and passengers do not need to wait. As a result, buses owners will increase their revenues and passengers will be served as they arrived. Also, the bus station will be in an equilibrium status.

There are two main approaches to trigger an adaptation: rule-based approach and machine learning approach. Rule-based approach systems are designed from a set of if-then rules: if the application senses a particular, then it should perform a particular action. Machine learning approach requires developer to collect data on the types of situations that a user will experience and the types of adaptation desired. It is about learning the probabilistic relationships between the situation and the adaptations, rather than have these relationships hard-coded and deterministic. Each one of them has its own advantages and disadvantages.

In this thesis we use goal-based approach to trigger adaptations. A business process is executed in order to bring about some state of affairs (goals) in the domain, which when achieved the business process can terminate. A goal is a set of stable states leads to business process termination. It is specified by a predicate over values of the state variables of the domain in which the process operates.

In this thesis, we represent business process goals by performance indicators with accepted values. In case of goals volatility, the system will react immediately. Based on historic information (i.e., violations information), the system can prevent goals volatility from being occurred. In our system, there are two main stakeholders, passengers and the operating company. We will use two metrics to reflect what they gain from our system. The metrics are *waiting time* and *system throughput*.

For evaluation purpose, we select to run our system from 2pm to 6pm for 7 days, which represents the rush hours in the Khartoum Bus Station. During this period the system will monitor the working environment and make necessitate actions. Monitored information and taken actions will be saved in the system log file. By the end of the period we will evaluate our system versus the existing one using the information in the system log file. We expect our system will improve the current one by more than 50%.

3.5 Summary

In this chapter, a brief introduction to business process adaptation types as well as some evaluation criteria in each was presented. It gave, also, an introduction to the new approach to business process adaptation. The new approach, which is domain-independent, is based on context-aware and event processing concepts. It is based also on the concepts of component-based software engineering and service-oriented architecture. Finally, the chapter explain the methodology used for evaluation.

In the next chapter the research objective is to present the mechanism of implementation of the proposed reference model. The main aim is to provide developers a methodology to build applications in any domain. A case study that has been applied based on our novel solution will be presented in Chapter 5.

CHAPTER 4: THE REFERENCE MODEL (ABCE) IMPLEMENTATION

4.1 Introduction

The design process for building the vast majority of context-aware applications can be boiled down to a pretty simple idea: *figure out what context your application needs* and when you receive that context, *figure out what you want to do with it*. Unfortunately, it takes a little more work than this to build a context-aware application (Bardram, et al., 2010).

The objective of this chapter is to present the implementation process of the new reference model Automated-adaptable Business-process using Context-aware and Event-processing (ABCE). The goal is to provide developers a mechanism that, possibly, can to be used with all types of applications regardless their domains.

4.2 The Reference Model ABCE

In section 4.3, we are going to discuss the reference model ABCE. But before that, we will revise previous approaches advantages and disadvantages. Approaches to business process adaptation are classified into vertical approaches like (Moon, et al., 2013) or horizontal approaches like (Bucchiarone, et al., 2011). Little approaches are classified as vertical and horizontal adaptation approaches like (Hermosillo, 2012).

Most of business process adaptation approaches are classified as vertical adaptation approaches. Their main concern is to improve the Quality of Service (QoS). Unfortunately, they do not take new business requirements into account and therefore they need a little amount of time for adaptation. So, the main limitation of this approach is the inability to respond to need requirements.

Horizontal adaptation approaches concern mainly about responding to new business requirements. Whereas they are able to deal with new needs, they consume more time than doing vertical adaptation. They also require customised execution engines that support dynamic reconfiguration. So, the main limitation of this approach is the time needed for adaptation which is also negotiable.

Our approach, which is domain-independent, is both vertical and horizontal adapter. We aim to provide better quality of service and respond to new requirements. And to solve the limitation of vertical approaches, we choose to use Service Component

Architecture (SCA) which makes business process structure modification more flexible and easy. (Hermosillo, 2012) also uses Service Component Architecture.

Some approaches have not taken into consideration the execution of business process in its proper manner, whereas other approaches like (Ardagna, et al., 2007) have taken that into account with different strategies. In our approach, we add the ability of being self-healing into business process.

Beside other approaches features, we add the feature of business process with many variants, according to probability of occurrence, to the approach to speed the adaptation process. Also, we design our approach to be proactive to context by applying data mining techniques which, also, reduce the time needed for adaptation.

We, also, add a new feature of self-optimisation to business process. We use Key Performance Indicators (KPIs) (Bucchiarone, et al., 2011) that can be used to evaluate the execution performance of business process against itself or against other business processes in the industry.

One of our contribution is the generic processes set that allow developers to develop context-aware applications. We provide them with generic graphical user interfaces.

4.3 The Reference Model ABCE Implementation

Using ABCE reference model, we have two modes: *design mode* and *run time mode*. In the design mode, the system does not started yet. In this mode, developers have to do some modelling and configurations.

Developers have to provide a standard business process (Figure 4.2) with business components and binding interfaces between them (Figure 4.3). Other business components that may be required, in case of business process modification, must also be provided to the system. Each business component has one or more business services that achieve a business goal.

Developers have to model also the working environment as well as determining system performance indicators with accepted values. Performance indicators will be used to measure the effectiveness of the business process which in case of violation, adaptation is going to be required. Models and metric values will be provided to the system using appropriate tools.

Detection components, that used to monitor the working environment, must also be configured. Detection components are a composition of services. They may differ from application to another according to the domain of the application. They share common configurations including component address, and services list (Figure 4.4).

In the run mode, the system starts operating by monitoring the working environment. *Context Acquisition* will capture the appropriate events using already configured detection components. It extracts business event parameters and passes them to the next component. *Adaptation Definition* uses the received parameters to create a business case. It uses these parameters to trigger an adaptation. Adaptation is done by *Process Adaptation*. Adapted business process will be send to *Performance Analysis* for execution and evaluation (Figures 4.5, 4.6).

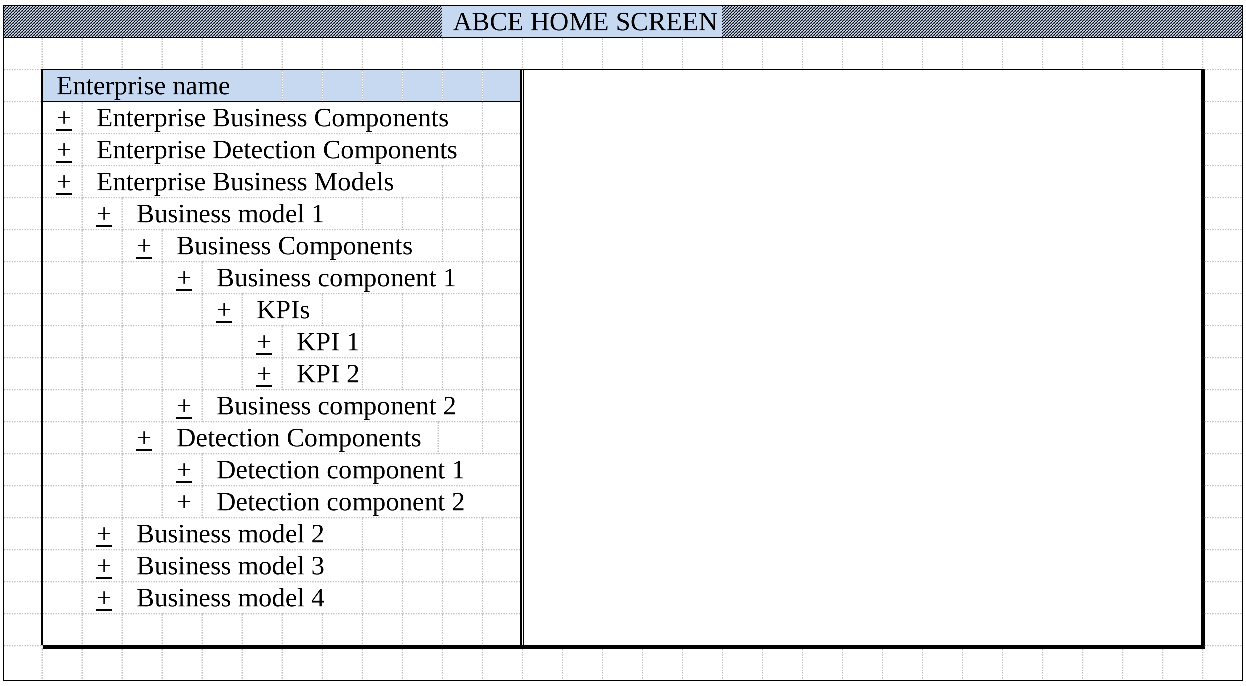


Figure 4.1: ABCE Home Screen

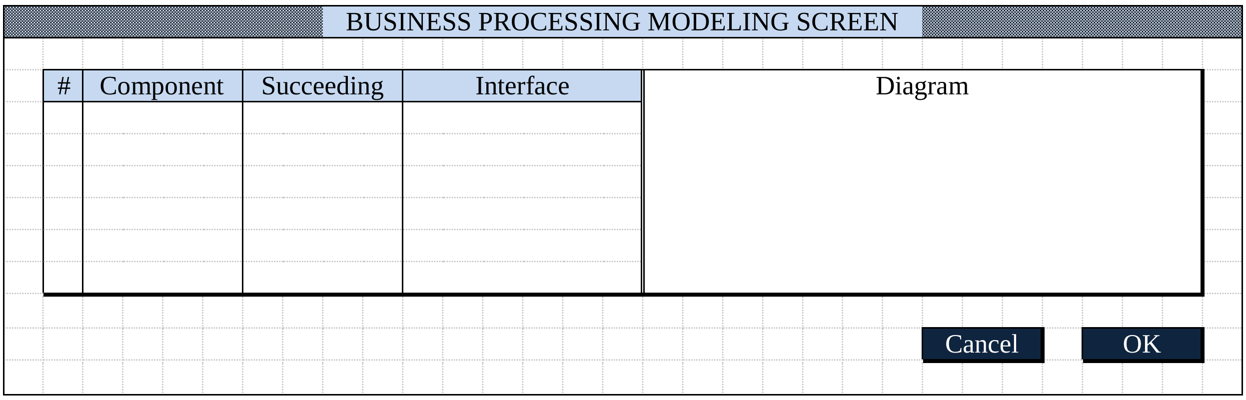


Figure 4.2: Business Process Modelling Screen

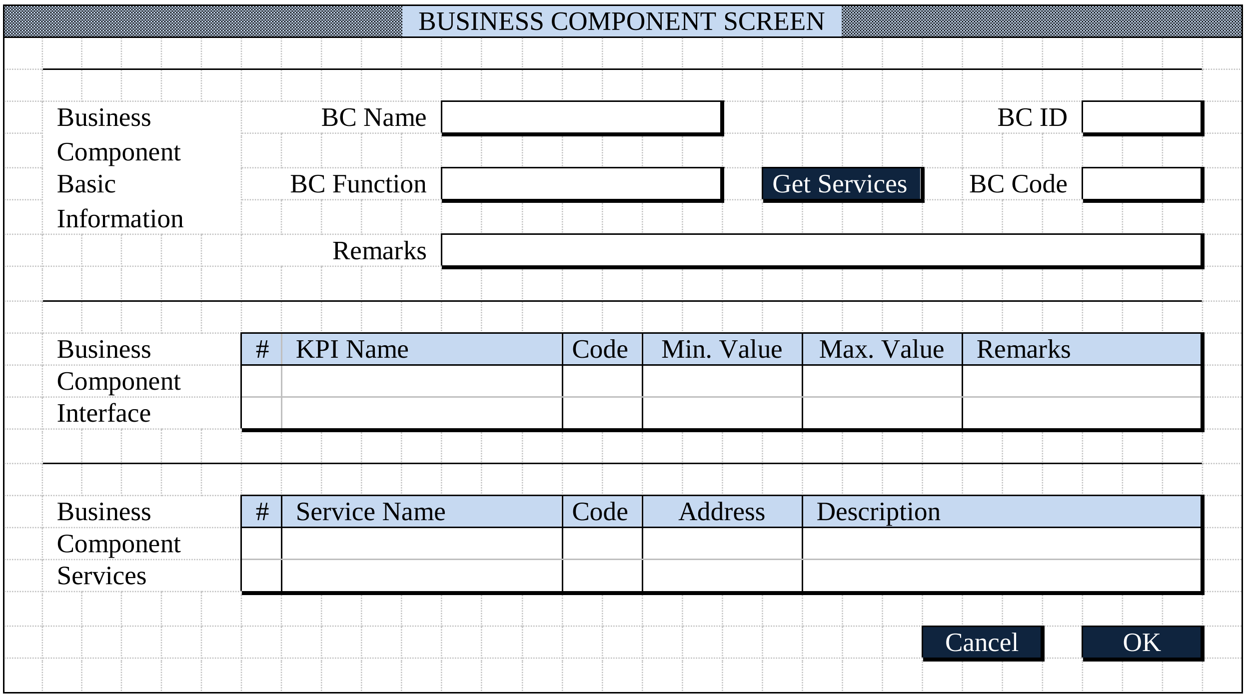


Figure 4.3: Business Component Screen

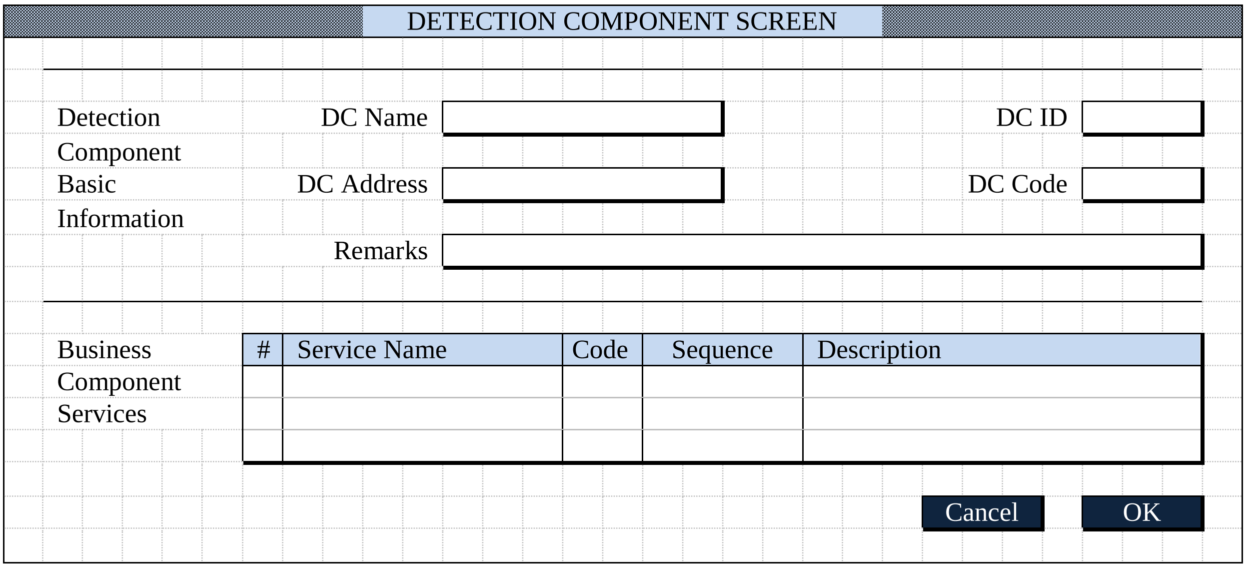


Figure 4.4: Detection Component Screen

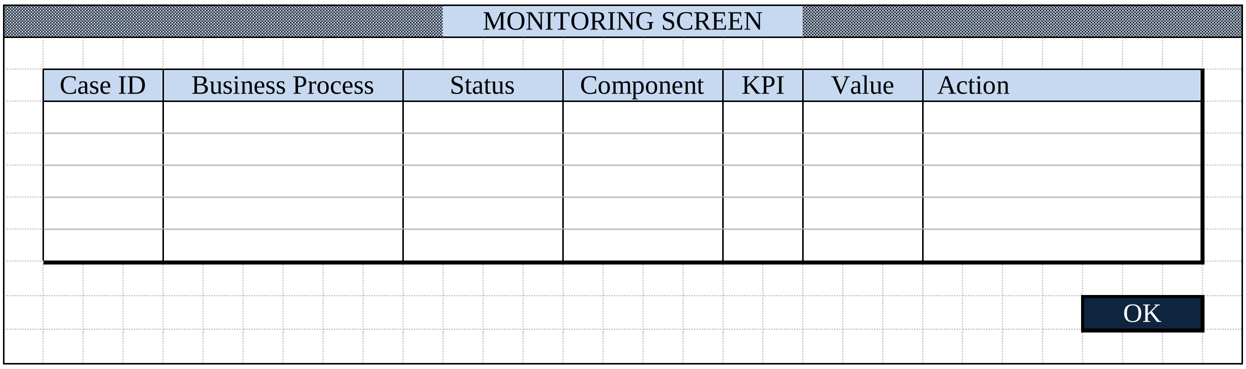


Figure 4.5: Monitoring Screen

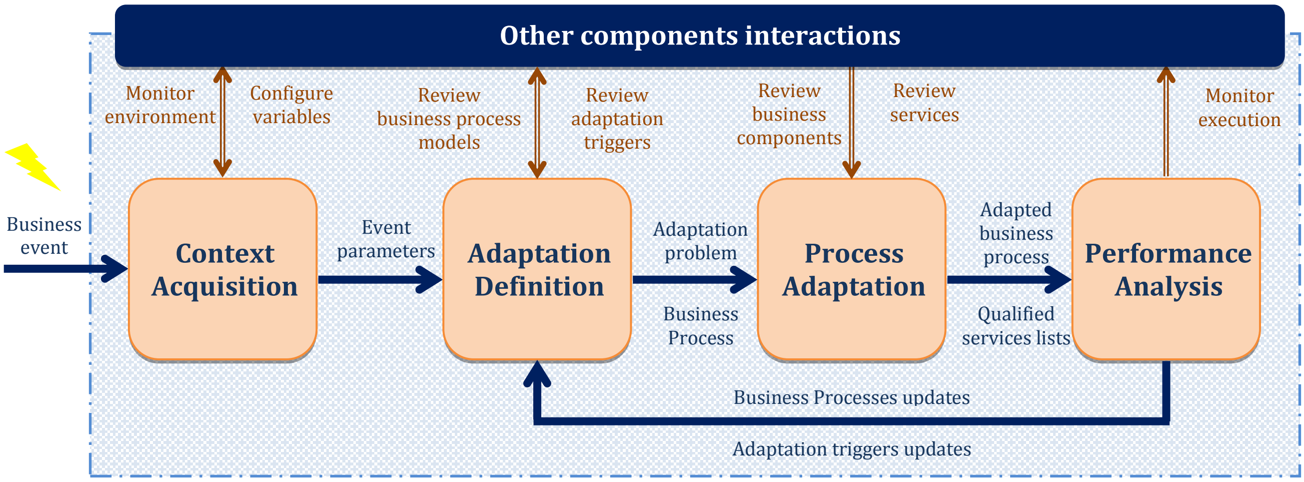


Figure 4.6: The Reference Model ABCE

4.4 ABCE Modules and Functions

In building context-aware applications, the first step is determining *what context-aware behaviour* the application should have and *in which situation*. The second step is determining *what hardware and/or software sensors* are required by the application to acquire the context determined in the first step. The third step is specifying *how context should be delivered* from the sensors to the application. The fourth step is having the application specify *what context it is in* and *receiving* that context. The last step is *analysing the received context* to determine what context-aware behaviour to execute, and then *execute* it (Bardram, et al., 2010).

In ABCE, a business process passes four stages: Context Acquisition, Adaptation Definition, Process Adaptation, and Performance Analysis. Each stage is accomplished by a module (Figure 4.6). But before these four stages, some specifications must be made. Each module consists of a group of processes. Each process needs some inputs to be processed by applying some tools/approaches to produce outputs that are passed to the next one. By executing all processes successfully, a business process can be adapted to a specific context (Figure 4.7).

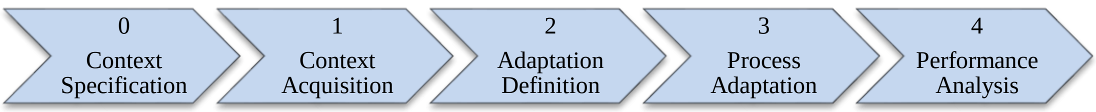


Figure 4.7: Framework's Modules

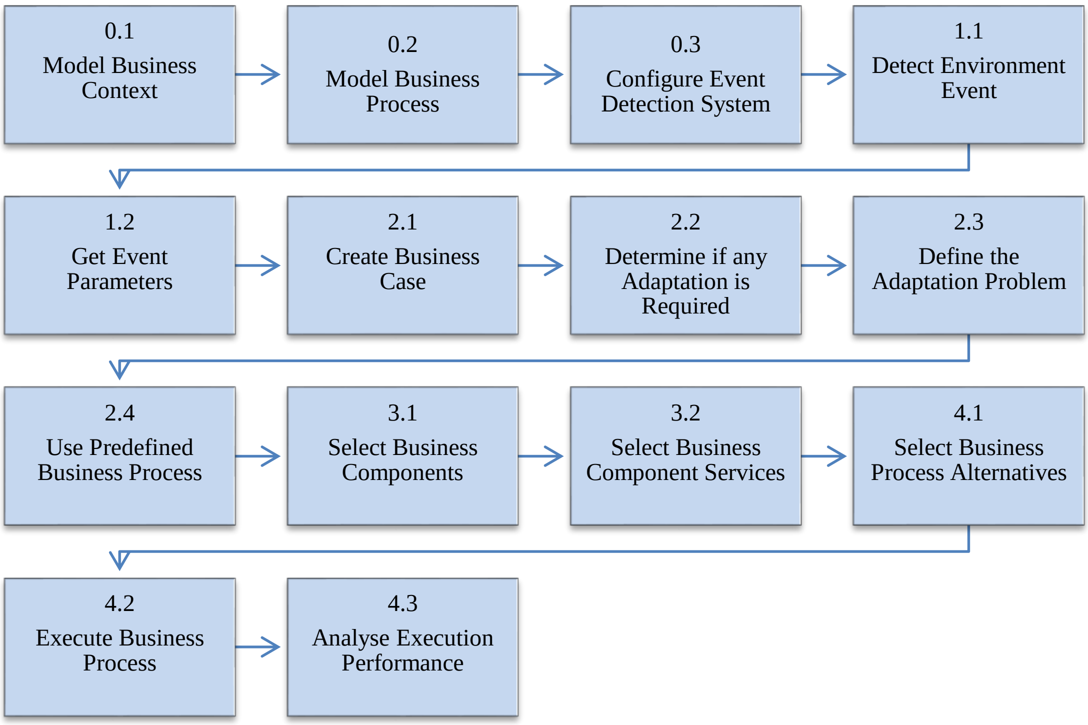


Figure 4.8: ABCE Processes

4.3.1 Context Specification

The processes in this module responsible for determine what context-aware behaviours the application will have and in which situations each behaviour should be executed and how. In this stage, the business context is modelled by the *Domain Expert* as a set of *context properties* (Bucchiarone, et al., 2011). The model contains information about possible configuration of properties with possible evolution respecting to events. The *Process Designer* uses the business context to create a business process model (Bucchiarone, et al., 2011). This module is in the Design Mode. Processes in this module are listed in Table 4.1.

Table 4.1: Context Specification Processes

Processes
Model Business Context
Model Business Process
Configure Domain Variables

1. Model Business Context

The purpose of this process is to model business context. This process is executed by the Domain Expert. It receives business context as inputs to create a business context model using modelling tools (Table 4.2).

Table 4.2: Model Business Context

Outputs	Inputs	Tools/Approaches
- Business context model	- Business context	- Business context modelling tools

2. Model Business Process

The purpose of this process is to model a business process. This process is executed by the Process Designer after *Model Business Context*. It receives context properties as inputs to create a business process model using modelling tools (Table 4.3).

Table 4.3: Model Business Process

Outputs	Inputs	Tools/Approaches
- Business process model	- Business context model	- Business process modelling tools

3. Configure Domain Variables

The purpose of this process is to define and configure domain variables. This process is executed after *Model Business Context*. It uses business context model as inputs to define the domain variables using analysis and then and configure them with sensors using configuration tools (Table 4.4).

Table 4.4: Define Key Performance Indicators

Outputs	Inputs	Tools/Approaches
- Key Performance Indicators	- Business context model	- Analysis tools - Configuration tools

4.3.2 Context Acquisition

The processes in this module are responsible for acquire context information which is then provided to system components for further use. This stage starts *when* the organisation *observes a change* in the marketplace or *receives a request* from a customer explicitly or implicitly. This module is in the Operation Mode. Processes in this module are listed in Table 4.5 (Bardram, et al., 2010).

Table 4.5: Context Acquisition Processes

Processes
Capture Environment Event
Get Event Parameters

1. Capture Business Event

The purpose of this process is to capture an event from the external environment to the organisation. This process is invoked when a change in the marketplace is observed or when receiving a request from a customer. It receives environment events as inputs and filters them to capture the event that it concerns with using event monitoring tools (Table 4.6).

Table 4.6: Capture Environment Event

Outputs	Inputs	Tools/Approaches
- Business event	- Environment events	- Monitoring tools

2. Get Event Parameters

The purpose of this process is to get the parameters of the captured event. This process is invoked after *Capture Business Event*. It receives a business event as input to extracts its parameters using converting/interpretation tools Table 4.7.

Table 4.7: Get Event Parameters

Outputs	Inputs	Tools/Approaches
- Event parameters	- Business event	- Event converting/ interpretation tools

4.3.3 Adaptation Definition

The processes in this module help the business process to define the adaptation needs, by responding to four adaptation questions: what to adapt?, when to adapt it?, where to adapt?, and how to adapt?. This module follows Context Acquisition module by receiving the information required for defining the adaptation needs. This module is in the Operation Mode. Processes in this module are listed in Table 4.8.

Table 4.8: Adaptation Definition Processes

Processes
Create Business Case
Determine if any Adaptation is Required
Define the Adaptation Problem
Use Predefined Business Process

1. Create Business Case

The purpose of this process is to create a business case which is represented by performance indicators. This process is invoked after *Get Event Parameters*. It receives event parameters as inputs and uses them to calculate the performance indicators of a specific business case using PI calculation tools (Table 4.9).

Table 4.9: Create Business Case

Outputs	Inputs	Tools/Approaches
- Business case PIs	- Event parameters	- PI calculation tools

2. Determine if any Adaptation is Required

The purpose of this process is to determine if any adaptation is required to the business process. This process is invoked after *Create Business Case*. It receives business case performance indicators as inputs and uses them to fire the adaptation process using adaptation trigger approach (Table 4.10).

Table 4.10: Determine if any Adaptation is Required

Outputs	Inputs	Tools/Approaches
- Adaptation request	- Business case KPIs	- Adaptation trigger approach

3. Define the Adaptation Problem

The purpose of this process is to define the adaptation problem if exists. This process is invoked after *Determine if any Adaptation is Required* by receiving a request for adaption from it. It receives, beside a request for adaptation, business case performance indicators and event parameters as inputs and uses them to define the adaptation problem using problem analysis approach (Table 4.11).

Table 4.11: Define the Adaptation Problem

Outputs	Inputs	Tools/Approaches
- Adaptation problem	- Adaptation request - Business case KPIs - Event parameters	- Problem analysis approach

4. Use a Predefined Business Process

The purpose of this process is to use a predefined business process that is most capable to handle the adaptation problem. This process is invoked after *Define the Adaptation Problem*. It receives the adaptation problem as inputs and uses it to select a business process with the same or almost same adaptation needs from the list of business processes alternatives (Table 4.12).

Table 4.12: Use Predefined Business Process

Outputs	Inputs	Tools/Approaches
- Business process	- Adaptation problem	- Business Processes List

4.3.4 Process Adaptation

The processes in this module handle the actual manipulation of the business process and consider the information coming from the *Adaptation Definition* to start the adaptation process. It manages the selection and binding of components and services to the business process to develop adapted business process alternatives and ranking them to select the best possible one. This module is in the Adaptation Mode. Processes in this module are listed in Table 4.13.

Table 4.13: Process Adaptation Processes

Processes
Select Business Components
Select Business Components Services

1. Select Business Components

The purpose of this process is to select necessary components to be involved in the adaptation process, if horizontal adaptation is required. This process is invoked after *Use Predefined Business Process*. It receives an adaptation problem and a business process as inputs to select components necessary to be bind in a business process from Business Components database using binding tools (Table 4.14).

Table 4.14: Select Business Components

Outputs	Inputs	Tools/Approaches
- Business process	- Business process - Adaptation problem	- Business Components - Binding tools

2. Select Business Component Services

The purpose of this process is to select the qualified services for each business components in the business process. This process is invoked after *Select Business Components*. It receives a business process as inputs and select alternative services to achieve each business components from Internal Services and External Services database using negotiation models (Table 4.15).

Table 4.15: Select Business Component Services

Outputs	Inputs	Tools/Approaches
- Business process - Services short lists	- Business process	- Internal Services - External Services - Negotiation models

4.3.5 Performance Analysis

The processes in this module handle the actual execution of the business process. Also, it analyse the performance of the executed business process weather it have been executed successfully or unsuccessfully. The analysis aims to upgrade the business process by calculate the probability of adaptation need occurrence. This module is in the Adaptation Mode. Processes in this module are listed in Table 4.16.

Table 4.16: Performance Analysis Processes

Processes
Select Business Process Alternative
Execute Business Process
Analysis Execution Performance

1. Select Business Process Alternative

The purpose of this process is to select best possible services for each component in the business process. This process is invoked after *Select Business Component Services*. It receives a business process and a list of services as inputs to select the best possible services using services evaluation tools (Table 4.17).

Table 4.17: Select Business Process Alternative

Outputs	Inputs	Tools/Approaches
- Business process alternative	- Business process - Services short lists	- Services evaluation tools

2. Execute Business Process

The purpose of this process is to execute a business process. This process is invoked after *Select Business Process Alternative*. It receives a business process as input to executes it. If a failure occurs during the execution, execution context is saved and the control returns to the previous process to select the next best service and resume the execution using process execution engine (Table 4.18) (Ardagna, et al., 2007).

Table 4.18: Execute Business Process

Outputs	Inputs	Tools/Approaches
- Execution information	- Business process - Adaptation problem	- Process execution engine

3. Analyse Execution Performance

The purpose of this process is to analyse the performance of a process under execution. This process is invoked with *Execute Business Process*. It receives execution information as inputs to analyse execution results, probability of occurrence of adaptation needs (Bucchiarone, et al., 2011), and performance indicators using Execution Log and performance analysis tools (Table 4.19).

Table 4.19: Analyse Execution Performance

Outputs	Inputs	Tools/Approaches
- Business Processes updates - Adap triggers updates	- Execution information	- Execution Log - Analysis tools

4.5 Summary

In this chapter, the implementation mechanism of the proposed reference model was presented and discussed. Implementation mechanism includes modules of the reference model as well as their functions and the relationships between them. It includes also the main user interfaces to be used by application developers. These user interfaces are generic.

In the next chapter the research objective is to validate the proposed reference model using a real society problem. By using simulation software, the chapter presents a novel cost-effective solution to the domain of public transportation.

CHAPTER 5: RESULTS STUDY AND EVALUATION

5.1 Introduction

The design process for building the vast majority of context-aware applications can be boiled down to a pretty simple idea: *figure out what context your application needs* and when you receive that context, *figure out what you want to do with it*. Unfortunately, it takes a little more work than this to build a context-aware application.

The objective of this chapter is to validate the proposed reference model using a real problem from the society. Using new designed simulation software; the chapter presents a novel cost-effective solution to the domain of public transportation.

5.2 Evaluation Metrics

For evaluation purpose, we have chosen two metrics. First metric, *waiting time*, is the amount of time that a passenger spends waiting for a bus in the ready queue. It is the time from first passenger detection to the time of first response for each round. This metric can reflect the degree of which customers (stakeholder) are satisfied. Second metric, *system throughput*, is the number of passengers that are served per a unit of time. It is the count of passengers served within a particular period of time. It reflects what bus station operating company (stakeholder) will gain from using this system.

5.3 The Transportation System

A transportation system provides the necessary connectivity that enhances the interaction between people. It is historical fact that, by facilitating the movement of people and the spreading of ideas, advances in transportation technology have been closely related to the evolution of civilisation (Papacostas, et al., 1990).

5.3.1 The Role of Government

Because of the profound role that transportation plays in society, governments have always become involved in the provision, operation, and regulation of transportation systems through the enactment of laws and through the establishment of public planning processes. Actions that a government takes at any given time as well as the method by which it chooses to implement those actions reflect the contemporary value system of the society it represents (Papacostas, et al., 1990).

The government intervenes in the marketplace by typical ways to accomplish objectives that it finds to be in the *public interest* include soft promotion, regulation, and investment (Papacostas, et al., 1990).

5.4 Case Study - Khartoum Bus Station

Khartoum Bus Station is located in Khartoum Locality one of seven localities in Khartoum State the capital of Republic of Sudan. Khartoum Bus Station was established in 2008 to provide and facilitate transportation service to Khartoum Locality residents and its visitors (the public). Khartoum Bus Station connects areas in Khartoum Locality to each other as well as to connect Khartoum Locality to the other localities through a number of destination lines. In Khartoum Bus Station, each locality has a *fixed* number of destination lines differs from other localities according to many factors like the area of the locality. Also, number of buses in a specific destination line may differ from the others according to the factor of people density in the destination area of that line and other factors.

5.4.1 Readiness Area

Because of the limitation in area of Khartoum Bus Station (passenger service track capacity is 9 buses maximum), each destination line has one or more tracks in Readiness Area beside its track in Passenger Service Area. Once a bus has arrived at Khartoum Bus Station, it will be directed to the Readiness Area. In Readiness Area, a bus driver can check his bus before entering Passenger Service Area. Readiness Area is outside the Khartoum Bus Station.

5.4.2 Current Operation Mechanism

There are two main departments in Khartoum Bus Station. The first department, Electronic Control Department, is responsible, beside other tasks, for monitoring and controlling the flow of buses inside Khartoum Bus Station using RFID technology. Each bus has a chip in the in front mirror. When a bus enters Passenger Service Area, an RFID reader fixed in the entrance of Khartoum Bus Station detects that bus by its chip. There are three bus entrances to Khartoum Bus Station; two for buses coming from points in Khartoum and Jebel Awleya Localities and one for buses coming from points in Omdurman, Karary, and Ombadda Localities. Destination lines of Bahry and Sharq Elneil Localities have not been activated yet. Each bus entrance has a group of RFID readers to monitor and detect the flow of buses to Passenger Service Area. But the RFID technology is used mainly for controlling the process of collecting entrance fees paid by buses. Each bus has an account in the system. The core information included in the account is: account ID, bus plate number, destination point, RFID tag, and balance. But it can be used also for other purposes.

As was mention above, each bus has a balance. At creation of an account for a new bus, the balance for this new bus will be initiated by a specific amount of money. Each time a bus arrives at the bus station, an amount of fee will be deducted from its balance. When the balance reaches zero, the driver must recharge bus account.

Second department has responsibilities of directing buses and on-ground supervision. Staff in this department use walky-talky devices to communicate. They exist in Passenger Service Area and Readiness Area. Each track in Passenger Service Area has a supervisor. A supervisor is responsible for monitoring his track using his eyes. If he observes a shortage of buses in his track, he communicate, using a walky-talky device, with his colleague in Readiness Area asking for directing some buses from Readiness Area to Passenger Service Area. This process continues all the day. There is some sort of collaboration between the two departments in issues like notifying bus drivers to recharge their accounts and maintaining the order.

5.4.3 Invested Technologies and Manpower

According to the description mentioned in the paragraph above, there are many technologies invested and used in Khartoum Bus Station operations beside manpower. Technologies include RFID system, web-based system, database system, and walky-talky communication system (Table 5.1).

Table 5.1: Invested Technologies

Technology	Remark
RFID system	Vendor
Server operating system	MS Windows
Client operating systems	MS Windows
Web-based system	ASP.NET
Database management system	MS SQL Server
Walky-talky communication system	Telecomm company

Besides those technologies, manpower is required in both the first and second departments. Manpower in the first department has responsibilities of managing accounts, updating balances, and monitoring bus flows. The number of operating staff in the first department is about 20. In the second department, manpower tasks are directing buses, monitoring flow of buses versus passengers, and maintaining the order in Passenger Service Area and Readiness Area. The number of operating staff in the second department is about 60.

5.4.4 Current Problems

There are many problems that reduce the effectiveness and efficiency of operation in Khartoum Bus Station. One of the main problems is the limited liability of bus owners toward public and ambiguity in defining the term of public interest. Each bus is dedicated to only one destination line regardless of whether there is a demand on that destination line or not. Other problems are related to the technologies used (for example, walky-talky devices). These technologies are rather ineffective in cost and operation.

The main problems to be solved, in this chapter, are the problem of *high waiting time* for passengers and the *low systems throughput*. With the aid of legislation, next section presents a novel cost-effective context-based solution that modifies the current one to achieve a public interest goal efficiently in Khartoum Bus Station.

5.5 The Case Study and the Reference Model ABCE

In our case study, there are two main categories of business components: the readiness area and passenger tracks. We represent the readiness area by one business component and a business component for each passenger track. No performance indicators are associated with the readiness area. We use *Number of Passengers Waiting (NPW)* as key performance indicator in each passenger track with accepted value of 10. This performance metric represents the demand for buses in a passenger track. High NPW values mean high demand and vice versa. Buses are the services in the two categories of business components. We assume that no bus is dedicated to any passenger track. We use another key performance indicator which is *System Throughput (ST)*. System Throughput represents the utility of the proposed solution to bus owners (another affected stakeholder). Current statistical studies show that each bus makes five trips per day as average. High ST values mean high profit for bus owners and consequence to bus station operating company (another affected stakeholder). Employing supply-demand theory as adaptation mechanism, the system must serve (supply) passenger tracks with high NPW values in descending order.

5.5.1. The Simulation Software

The simulation software has one form. This form consists of a five-button toolbox and a two-page tab. The five buttons (Figure 5.1) are used as follows:

- **New**. This button is for starting a new one.
- **Monitor Supply**. This button is for starting buses monitoring.
- **Monitor Demand**. This button is for starting passengers monitoring.
- **Export**. This button is for exporting monitoring information to an Excel file.
- **Exit**. This button is used for finalising the application.

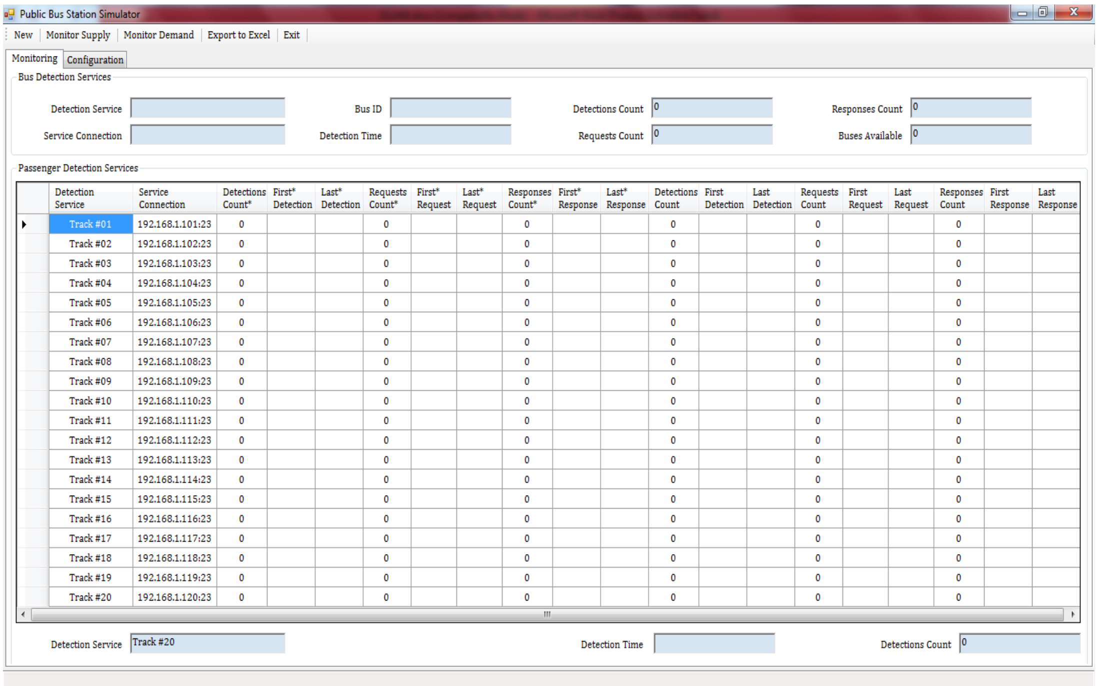


Figure 5.1: Main Interface Showing Buttons and Monitoring Page

The two tabs are used as follows:

- Monitoring. This tab is used for showing run time information about buses and passengers. It has the sections: Bus Detection Services and Passenger Detection Services. The first section shows information about detected bus like id, detection service, detection time, detections count, requested buses count, responses count, and number of buses available. The second section shows information of passenger detection time, detection service, and detections count. It consists of a grid table that shows monitoring information for each track. Information includes detections count, first detection time, last detection time, requests count, first request time, last request time, responses count, first response time, and last response time. Columns with asterisk are for current monitoring where others are for overall monitoring information.

- Configuration. It is used for managing services information and setting adaptation triggers.

Operation Mode

The system is turned to the operation mode by clicking on New, Monitor Supply, and Monitor Demand buttons. Monitoring information will be displayed as in Figure 5.2.

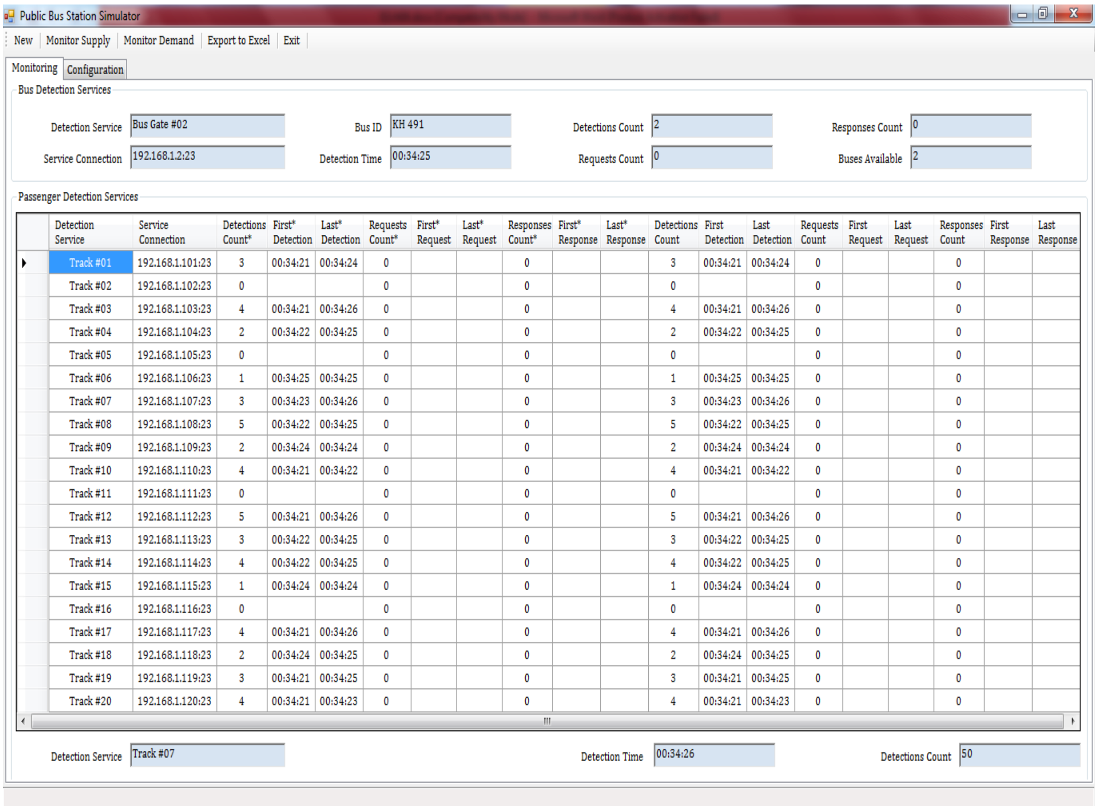


Figure 5.2: Main Interface Showing Run Time Information

In Figure 5.3, Track #01 requests a bus. Using FIFO, the request will be sent to the first bus in the queue. The track that needs a bus will be highlighted in red and font colour will be yellow. It will be showed in the status bar in red font colour.

In Figure 5.4, Track #09 request have been responded. The track that has been served will be highlighted in green and font colour will be white. The status bar shows also tracks that have been served in green font colour.

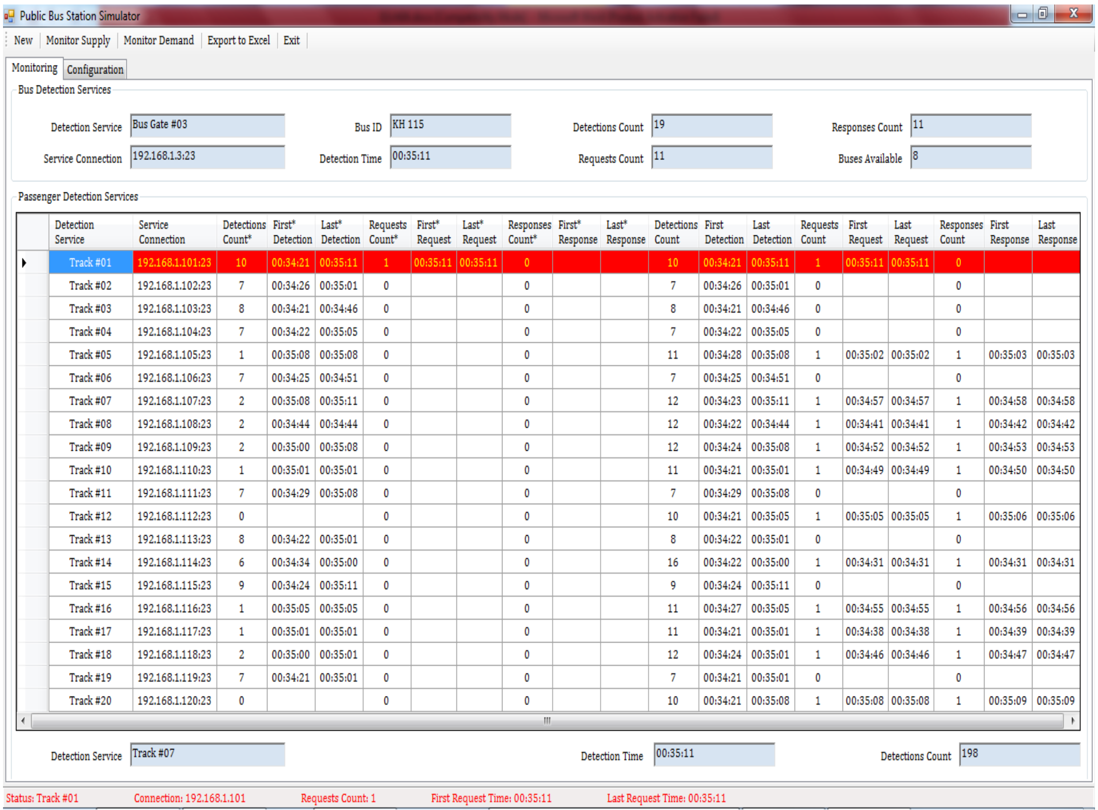


Figure 5.3: Main Interface Showing a Request from Track #01

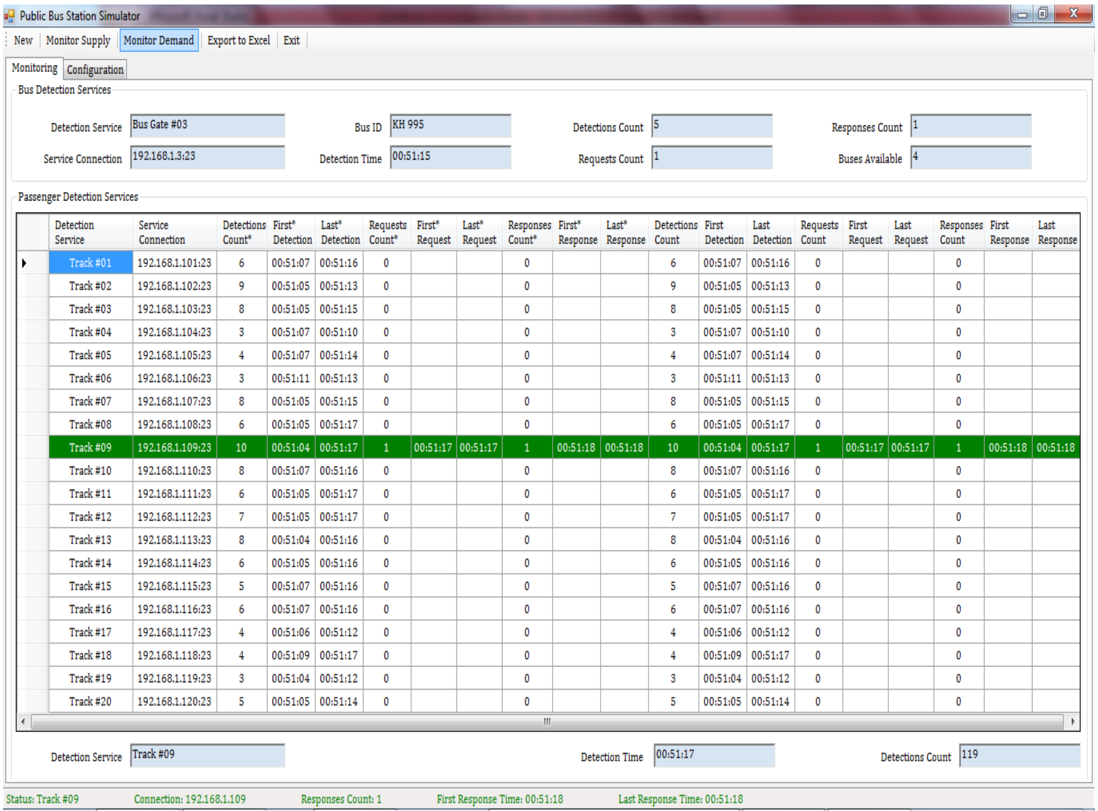
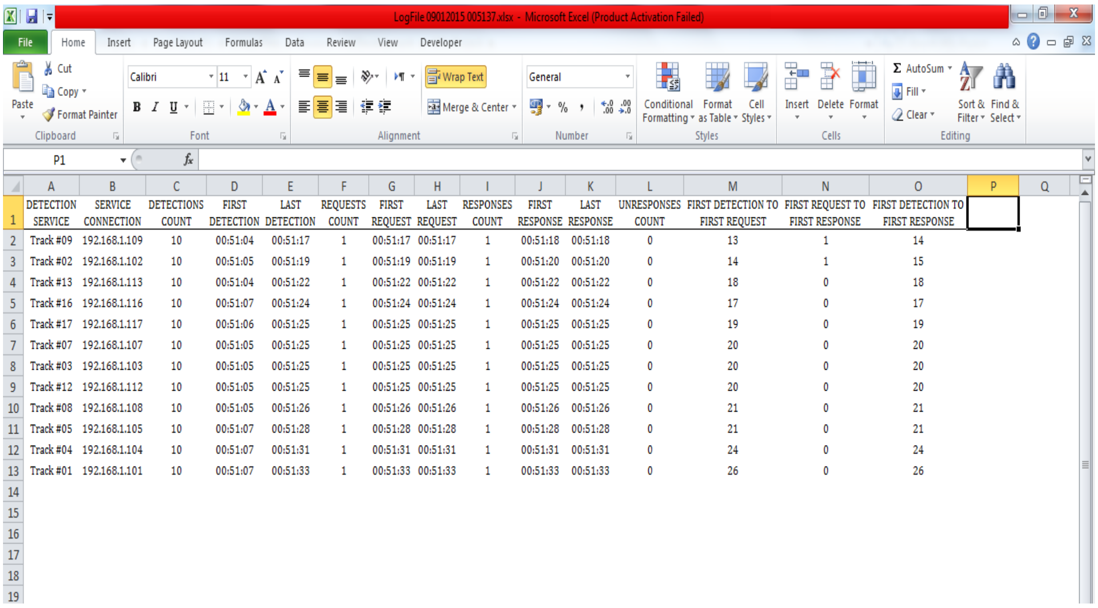


Figure 5.4: Main Interface Showing a Response to Track #09

Figure 5.5 shows a 15-column statistical report using MS Excel application. These columns are: Detection Service, Service Connection, Detections Count, First Detection, Last Detection, Requests Count, First Request, Last Request, Responses Count, First Response, Last Response, Unresponses Count, First Detection to First Request, First Request to First Response, and First Detection to First Response.



DETECTION SERVICE	SERVICE CONNECTION	DETECTIONS COUNT	FIRST DETECTION	LAST DETECTION	REQUESTS COUNT	FIRST REQUEST	LAST REQUEST	RESPONSES COUNT	FIRST RESPONSE	LAST RESPONSE	UNRESPONSES COUNT	FIRST DETECTION TO FIRST REQUEST	FIRST REQUEST TO FIRST RESPONSE	FIRST DETECTION TO FIRST RESPONSE
Track #09	192.168.1.109	10	00:51:04	00:51:17	1	00:51:17	00:51:17	1	00:51:18	00:51:18	0	13	1	14
Track #02	192.168.1.102	10	00:51:05	00:51:19	1	00:51:19	00:51:19	1	00:51:20	00:51:20	0	14	1	15
Track #13	192.168.1.113	10	00:51:04	00:51:22	1	00:51:22	00:51:22	1	00:51:22	00:51:22	0	18	0	18
Track #16	192.168.1.116	10	00:51:07	00:51:24	1	00:51:24	00:51:24	1	00:51:24	00:51:24	0	17	0	17
Track #17	192.168.1.117	10	00:51:06	00:51:25	1	00:51:25	00:51:25	1	00:51:25	00:51:25	0	19	0	19
Track #07	192.168.1.107	10	00:51:05	00:51:25	1	00:51:25	00:51:25	1	00:51:25	00:51:25	0	20	0	20
Track #03	192.168.1.103	10	00:51:05	00:51:25	1	00:51:25	00:51:25	1	00:51:25	00:51:25	0	20	0	20
Track #12	192.168.1.112	10	00:51:05	00:51:25	1	00:51:25	00:51:25	1	00:51:25	00:51:25	0	20	0	20
Track #08	192.168.1.108	10	00:51:05	00:51:26	1	00:51:26	00:51:26	1	00:51:26	00:51:26	0	21	0	21
Track #05	192.168.1.105	10	00:51:07	00:51:28	1	00:51:28	00:51:28	1	00:51:28	00:51:28	0	21	0	21
Track #04	192.168.1.104	10	00:51:07	00:51:31	1	00:51:31	00:51:31	1	00:51:31	00:51:31	0	24	0	24
Track #01	192.168.1.101	10	00:51:07	00:51:33	1	00:51:33	00:51:33	1	00:51:33	00:51:33	0	26	0	26

Figure 5.5: An MS Excel Log File Showing Statistical Information

5.6 Results and Discussion

We ran our system from Saturday to Friday. The results obtained from our system are compared to the results from the existing system using the two mentioned metrics waiting time, and system throughput.

Criteria #1: Waiting Time

The results of the current situation were collected, by Khartoum Bus Station operator, using Stratified Sampling method and Simple Probability Sampling method. Stratified sampling is a probability sampling technique wherein the entire population is divided into different subgroups or strata, then randomly selects the final subjects proportionally from the different strata.

After that, the Simple Probability Sampling method within each different stratum was applied. Probability sampling is a sampling technique wherein the samples are gathered in a process that gives all the individuals in the population equal chances of being selected.

Khartoum Bus Station passengers are divided into different strata. Each stratum represents a destination line. Within each stratum, some passengers were selected randomly as samples and their waiting times were recorded (Table 5.2).

Table 5.2: Passengers Waiting Time in Seconds

Day	Current Situation			Proposed Solution		
	Optimistic	Pessimistic	Common	Optimistic	Pessimistic	Common
Saturday	0300	0900	0600	0020	0239	0093
Sunday	0900	5400	3600	0015	0256	0092
Monday	0600	3600	1800	0018	0234	0100
Tuesday	0300	3600	0900	0019	0248	0082
Wednesday	0000	1800	0900	0017	0265	0091
Thursday	1800	7200	3600	0013	0236	0080
Friday	0000	0300	0120	0010	0224	0088

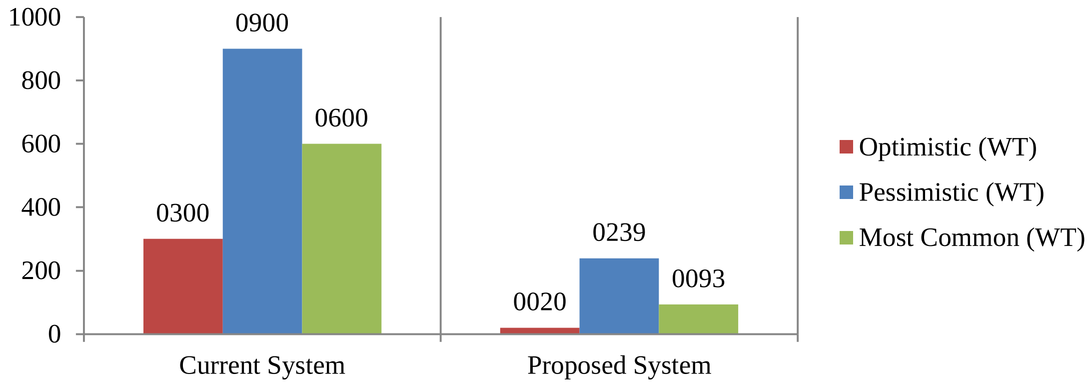


Figure 5.6: Waiting Time in Saturday

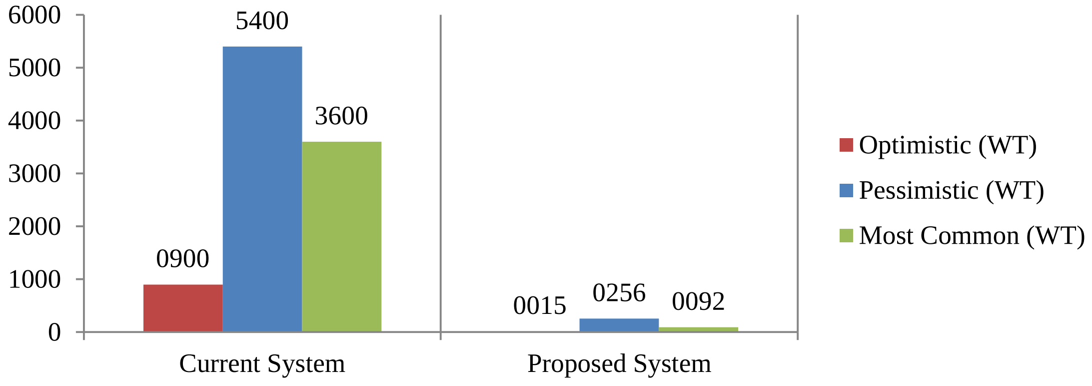


Figure 5.7: Waiting Time in Sunday

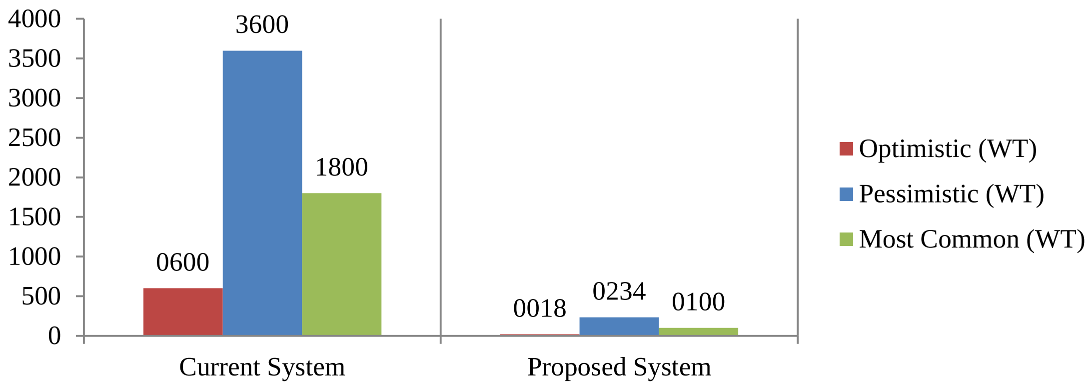


Figure 5.8: Waiting Time in Monday

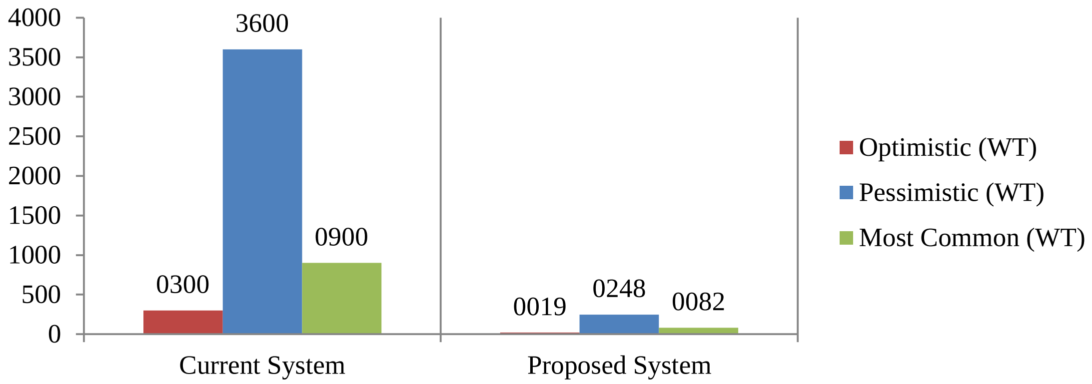


Figure 5.9: Waiting Time in Tuesday

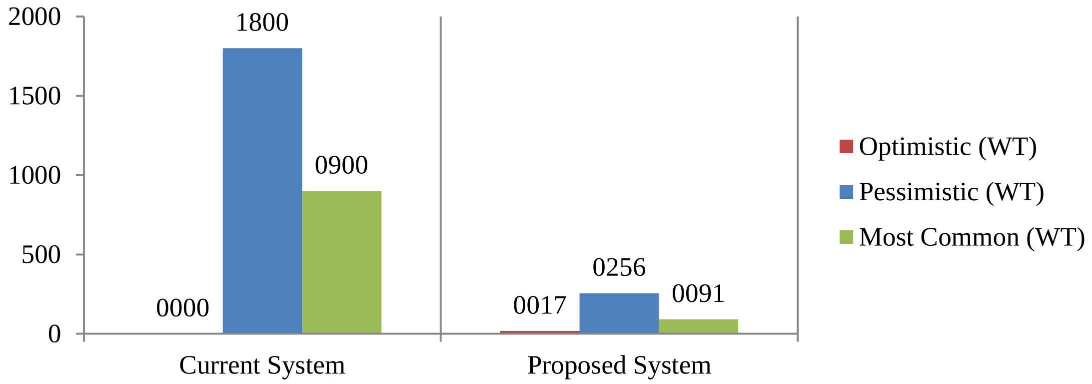


Figure 5.10: Waiting Time for Wednesday

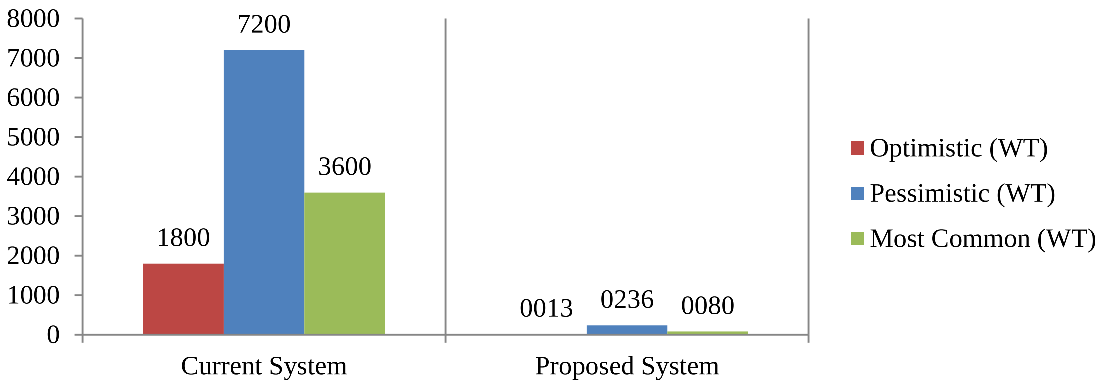


Figure 5.11: Waiting Time for Thursday

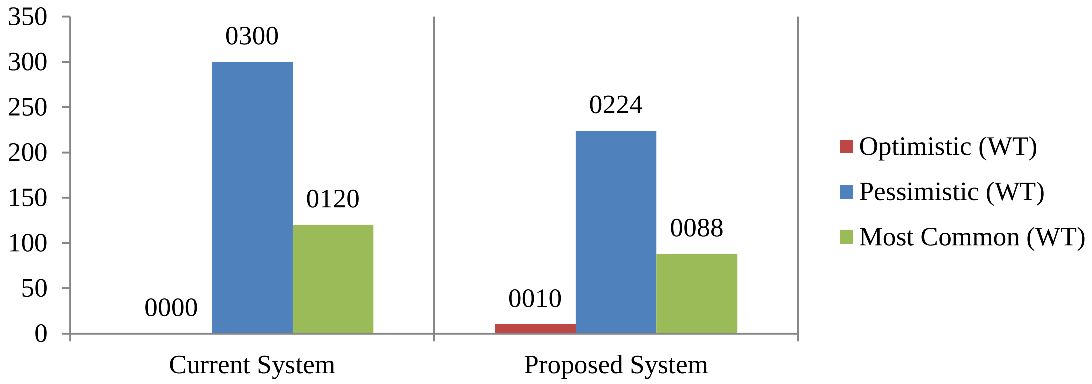


Figure 5.12: Waiting Time for Friday

Criteria #2: System Throughput

The results of the current situation were collected from Khartoum Bus Station operator database. Buses Log file contains the number of buses entered the bus station for each day and their capacities are known (Table 5.3).

Table 5.3: System Throughput (Served Passengers)

Day	Current Situation	Proposed Solution
Saturday	121629	161767
Sunday	136299	181278
Monday	143786	179733
Tuesday	146298	182873
Wednesday	124634	155793
Thursday	138420	184099
Friday	079752	111653

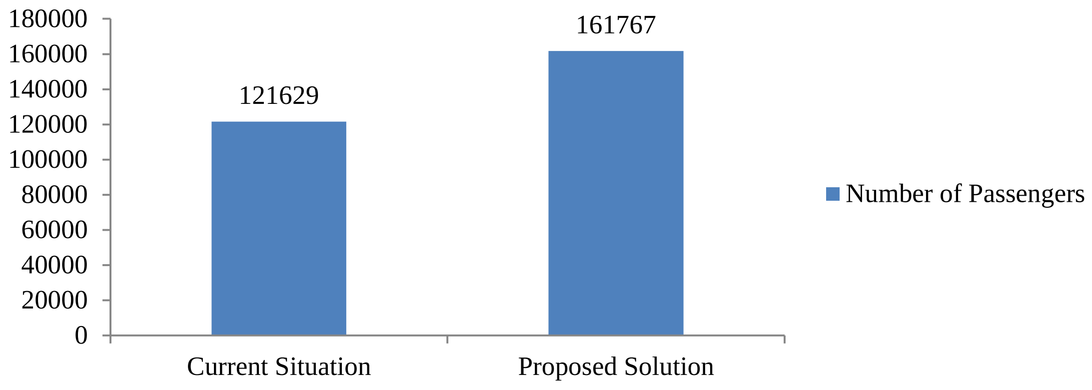


Figure 5.13: System Throughput in Saturday

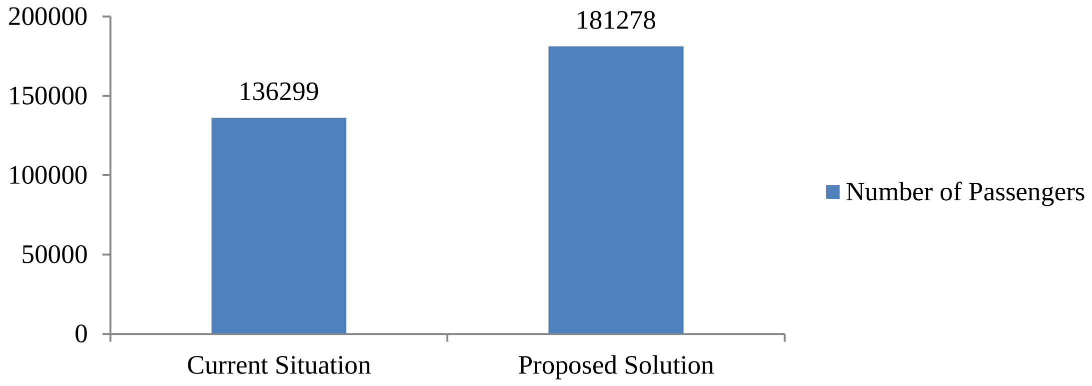


Figure 5.14: System Throughput in Sunday

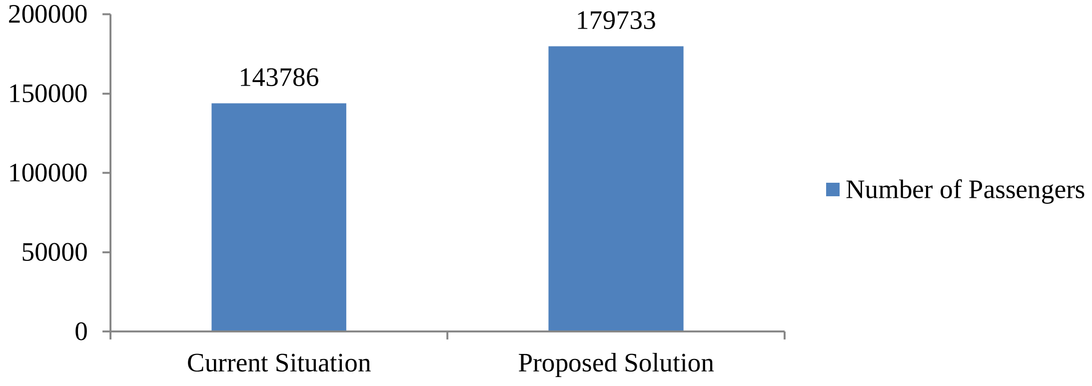


Figure 5.15: System Throughput in Monday

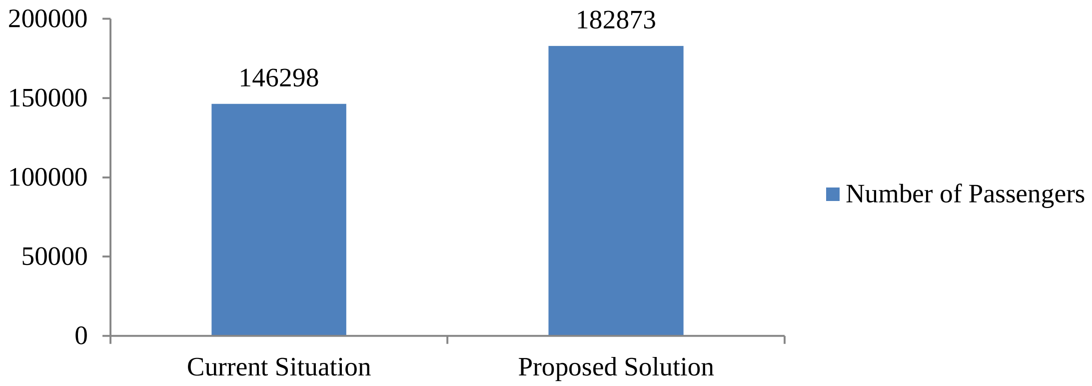


Figure 5.16: System Throughput in Tuesday

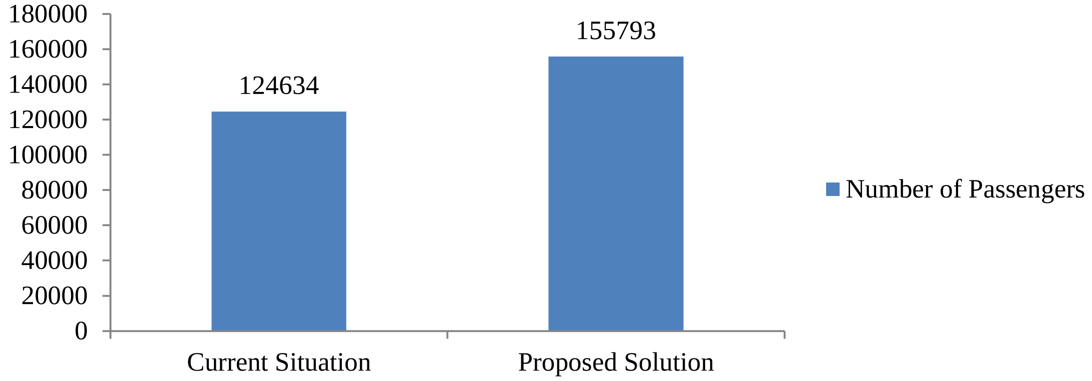


Figure 5.17: System Throughput in Wednesday

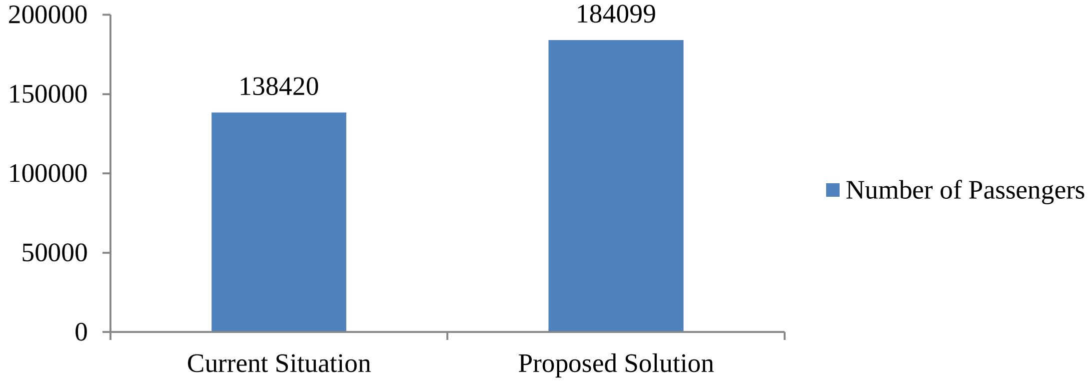


Figure 5.18: System Throughput in Thursday

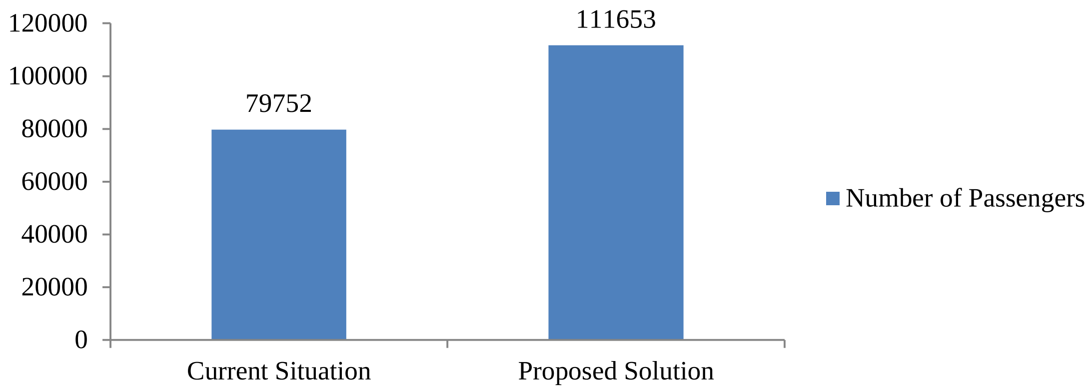


Figure 5.19: System Throughput in Friday

The question of this research, which has been mentioned in Chapter 1, is: ***"How to develop an automated adaptable business processes using context-aware and event processing concepts?"***

By using event processing techniques like RFID and Infrared technologies, the proposed solution is fully *automated* and *aware* to its working environment. The system use these technologies to monitor the bus station environment, detect buses and passengers, create a business case for each track, and *adapt* buses to tracks according to their *context* in terms of passengers demand.

From Chapter 1, this research aims to accomplish three objectives:

- **Context-awareness.** This objective is to make a business process aware to changes in the working environment. This objective has been achieved by using RFID and Infrared technologies to monitor the working environment and detect related events and create business cases.

- **Run Time Self-Adaptation.** This objective is to give a business process the ability to automatically adapt itself to its working environment at runtime. This objective has been achieved by using the captured context to route detected buses to tracks according to their demand.

- **Proactiveness.** This objective is to predict, prevent, or reduce conditions of adaptation from occurring to reduce the needs for adaptation. This objective has been achieved by applying data mining techniques on the log files to predict when the demand will be increased and in which tracks to reduce the time needed for requesting.

5.7 Summary

In this chapter the researcher proposed a framework for an automated adaptable business process. The framework is generic and can be used with all types of applications regardless their domains.

In the next chapter the researcher summarises the main contributions of this research, present the conclusions of the research, and define a set of perspectives for future work.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1 Introduction

The main contribution of this thesis is the reference model Automated-Adaptable Business-Process using Context-Aware and Event-Processing concepts (ABCE). This reference model is component-based and service-oriented and uses context-aware and event processing concepts to handle business process adaptation. ABCE consists of five modules, each of which groups logically related processes.

- **Context Specification.** It responsible for determining what context-aware behaviours the application will have and in which situations each behaviour should be executed and how.

- **Context Acquisition.** It responsible for capturing real situations from the surrounding environment and creating of business case representations.

- **Adaptation Definition.** It helps the business process to define the adaptation needs, by responding to the adaptation questions.

- **Process Adaptation.** It handles the actual manipulation of the business process. It manages the selection and binding of components and services to the business process to develop adapted business process alternatives.

- **Performance Analysis.** It handles the actual execution of the business process. It analyse the performance of the executed business process. The analysis aims to upgrade the business process by calculate the probability of adaptation need occurrence.

We found that making business process dynamically adaptable brings an additional cost to its execution. However, this cost can be justified by the ability of a business process to meet new business requirements without any redeployment or losing of any information or any downtime. Moreover, if we compare the time it takes to execute the process against the time it takes to adapt it, the adaptation time becomes negligible.

6.2 Dissertation Summary

This text was organised into six chapters. These chapters are:

Chapter 1: Introduction. This chapter gave an introductory to the dissertation, including the problem statement, research question, research objective, research scope, research contribution, and thesis organisation.

Chapter 2: Literature Review. This chapter presented a brief introduction to the main domains of this research as well as the terminology and concepts presented in the subsequent chapters. It presented, also, some studies that try to solve the lack of flexibility of business processes with different solutions.

Chapter 3: The Research Methodology. This chapter presented the methodology of the research to solve the lack of flexibility of business processes including implementation and evaluation. It covered also criteria that can be used to evaluate the proposed reference model versus other solutions.

Chapter 4: ABCE Model Implementation. In this chapter the reference model ABCE have been proposed. It explained how business processes can be automatically adapted to working environment changes, and how the reference model can prevent or reduce unnecessary adaptations according to their conditions and condition occurring probability. It also explained how the reference model will react in case of failure.

Chapter 5: Results Study and Evaluation. This chapter presented a novel contribution to the domain of public transportation using the proposed reference model and simulation software. It validated, using two evaluation metrics, the proposed solution against a current running one. Using the results obtained from the validation, the research question has been answered and the research objectives have been achieved.

Chapter 6: Conclusions and Future Work. This chapter summarises the whole thesis and presents what to be done in the future.

6.3 Future Work

In this thesis we proposed a reference model for business process adaptability. This reference model has satisfied the goal of the research, but we are looking forward to enhance it in areas of context specification, performance optimisation, and graphical user interface. We plan to develop an adaptive language to express adaptation needs to be used with business processes. Another plan is to extend the reference model self-optimisation-ability by taking account for fault probability. Also, we aim to enhance model user interfaces that facilitate purposes of modelling of context and business processes, configuring parameters, and monitoring of context and execution. Finally, we want to extend our case study to cover public transport system in Khartoum State and implement it in real.

REFERENCES

A Context-Aware Framework for Business Processes Evolution. Bucchiarone, Antonio, et al. 2011. s.l. : IEEE, 2011. 15th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW). pp. 146-154.

Abowd, G. D., et al. 1998. *Context-awareness in Wearable and Ubiquitous Computing.* 1998.

Adaptation and Distribution of Pipeline-Based Context-Aware Web Architectures. Hinz, M., et al. 2007. s.l. : IEEE, 2007. The Working IEEE/IFIP Conference on Software Architecture. p. 15.

Adaptation of Service-based Business Processes by Context-Aware Replanning. Bucchiarone, Antonio, et al. 2011. s.l. : IEEE, 2011. IEEE International Conference on Service-Oriented Computing and Applications (SOCA). pp. 1-8.

Antony, J., et al. 2013. Ubiquitous Patient Monitoring and Smart Alert Generation in an intensive care unit supported by Low Cost Tablet PC Based Automation System Powered Through Open Source Software and Hardware Platforms. 2013, pp. 334 - 339.

Ardagna, Danilo, et al. 2007. PAWS: A Framework for Executing Adaptive Web-Service Processes. *Software.* November - December 2007, Vol. 24, 6, pp. 39-46.

Bardram, Jakob E., et al. 2010. *Ubiquitous Computing Fundamentals.* [ed.] John Krumm. Boca Raton : CRC Press, 2010. 978-1-4200-9360-5.

Brown, M. G. 1996. *Supporting User Mobility.* 1996.

Brown, P. J. 1998. *Triggering Information by Context.* 1998.

Brown, P. J., Bovey, J. D. and Chen, X. 1997. *Context-aware Applications: From the Laboratory to the Marketplace.* 1997.

Bucchiarone, Antonio, et al. 2011. A Context-Aware Framework for Business Processes Evolution. 2011, pp. 146-154.

Bucchiarone, Antonio, et al. 2011. Adaptation of Service-based Business Processes by Context-Aware Replanning. 2011, pp. 1-8.

Chandy, K. M. and Schulte, W. R. 2010. *Event Processing - Designing IT Systems for Agile Companies.* s.l. : McGraw-Hill, 2010.

Clark, H. and Brennan, S. 1991. *Grounding in Communication.* 1991.

—. 1991. *Grounding in Communication - Perspectives on Socially Shared Cognition.* [ed.] L. Resnick, L. Levine and S. Teasley. Washinton, DC : APA Books, 1991.

Component-based Software Engineering - New Paradigm of Software Development. Crnkovic, Ivica and Larsson, Magnus. 2001. 2001.

Context-aware Applications: From the Laboratory to the Marketplace. Brown, P. J., Bovey, J. D. and Chen, X. 1997. s.l. : IEEE Personal Communications, 1997.

Context-Aware Business Process Management for Personalized Healthcare Service. Moon, Junho and Kim, Dongsoo. 2013. s.l. : IEEE, 2013. IEEE 10th International Conference on Services Computing (SCC). pp. 757-758.

Cooperstock, J. R., et al. 1995. *Evolution of a Reactive Environment.* 1995.

Costa, P. and and Botelho, L. 2005. Generic Context Acquisition and Management Framework. 2005.

Costa, Paulo and Botelho, Luís. ???. *Generic Context Acquisition and Management Framework.* ??? : ???, ??? ???

Crnkovic, Ivica and Larsson, Magnus. 2001. Component-based Software Engineering - New Paradigm of Software Development. 2001.

Davies, N., et al. 1998. *Developing a Context-sensitive Tour Guide.* 1998.

Deori, Lokhi Prasad. 2010. *Complex Event Processing.* s.l. : MphasiS Company, 2010.

Dey, A. K. and Abowd, G. D. 1997. *CyberDesk: The Use of Perception in Context-aware Computing.* 1997.

—. 2000. *Towards a Better Understanding of Context and Context-awareness.* 2000.

Dey, A. K. 1998. *Context-aware Computing: The CyberDesk Project.* 1998.

Dey, A. K., Abowd, G. D. and Wood, A. 1998. *CyberDesk: A Framework for Providing Self-integrating Context-aware Services.* 1998.

Disseminating Active Map Information to Mobile Hosts. Schilit, B. and Theimer, M. 1994. s.l. : IEEE Network, 1994.

Elrod, S., et al. 1993. *Responsive Office Environments.* 1993.

Erl, Thomas. 2007. *SOA Principles of Service Design.* Upper Saddle River : Prentice Hall, 2007.

Fickas, S., Kortuem, G. and Segall, Z. 1997. *Software Organization for Dynamic and Adaptable Wearable Systems.* 1997.

Franklin, D. and Flaschbart, J. 1998. *All Gadget and no Representation Makes Jack a Dull Environment.* 1998.

Generic Context Acquisition and Management Framework. Costa, P. and and Botelho, L. 2005. 2005. First European Young Researchers Workshop on Service Oriented Computing.

Han, Son N., Lee, Gyu Myyoung and Crespi, Noel. 2014. Semantic Context-Aware Service Composition for Building Automation System. *IEEE Transactions on Industrial Informatics.* February 2014, Vol. 10, 1, pp. 752-761.

Hermosillo, Gabriel. 2012. *Towards Creating Context-Aware Dynamically-Adaptable Business Processes Using Complex Event Processing.* Département de Formation, Universite Lille 1 - Science and Technology. Lille : s.n., 2012. PhD Thesis.

—. 2012. *Towards Creating Context-Aware Dynamically-Adaptable Business Processes Using Complex Event Processing.* Universite Lille 1, France. Lille : Universite Lille 1, 2012. PhD.

—. 2012. *Towards Creating Context-Aware Dynamically-Adaptable Business Processes Using Complex Event Processing.* Lille : Universite Lille 1, 2012.

Hinz, M., et al. 2007. Adaptation and Distribution of Pipeline-Based Context-Aware Web Architectures. 2007, p. 15.

Hull, R., Neaves, P. and Bedford-Roberts, J. 1997. *Towards Situated Computing.* 1997.

Josuttis, N. 2007. *SOA in Practice: The Art of Distributed System Design.* s.l. : O'Reilly Media, Inc., 2007.

Kortuem, G., Segall, Z. and Bauer, M. 1998. *Context-aware, Adaptive Wearable Computers as Remote Interfaces to 'Intelligent' Environments.* 1998.

Kostakos, V., Camacho, T. and Mantero, C. 2010. Wireless Detection of End-to-End Passenger Trips on Public Transport Buses. 2010, pp. 1795 - 1800.

Luckham, D. C. 2002. *The Power of Events: An Introduction to.* s.l. : Addison-Wesley Long-man Publishing Co., Inc., 2002.

MacKenzie, C. Matthew, et al. 2006. Reference Model for Service Oriented Architecture 1.0. *OASIS | Advancing open standards for the information society.* [Online] 1.0, 2006. http://www.oasis-open.org.

Mistry, Ross and Misner, Stacia. 2014. *Introducing Microsoft SQL Server 2014 Technical Overview.* Redmond : Microsoft Press, 2014.

Moon, Junho and Kim, Dongsoo. 2013. Context-Aware Business Process Management for Personalized Healthcare Service. 2013, pp. 757-758.

Nødtvedt, Jon Ole and Nguyen, Man Hoang. 2004. *Mobility and Context-awareness in Workflow Systems.* Institute for Computer and Information Science (IDI), Norwegian University of Science and Technology (NTNU). Trondheim : Norwegian University of Science and Technology (NTNU), 2004. MSc Thesis.

Papacostas, C. S. and Prevedouros, P. D. 1990. *Transportation Engineering and Planning.* Englewood : Prentice Hall, 1990.

PARC. 1991. 1991.

Pascoe, J. 1998. *Adding Generic Contextual Capabilities to Wearable Computers.* 1998.

Pascoe, J., Ryan, N. S. and Morse, D. R. 1998. *Human-computer-giraffe Interaction - HCI in the Field.* 1998.

Rekimoto, J., Ayatsuka, Y. and Hayashi, K. 1998. *Augmentable Reality: Situated Communication through Physical and Digital Spaces.* 1998.

Rodden, T., et al. 1998. *Exploiting Context in HCI Design for Mobile Systems.* 1998.

Ryan, N. 1997. *MCFE Metadata Elements, Version 0.2.* 1997.

Ryan, N., Pascoe, J. and Morse, D. 1998. *Enhanced Reality Fieldwork: The Context-aware Archaeological Assistant.* 1998.

Salber, D., Dey, A. K. and Abowd, G. D. 1999. *The Context Toolkit: Aiding the Development of Context-enabled Applications.* 1999.

Schilit, B. and Theimer, M. 1994. *Disseminating Active Map Information to Mobile Hosts.* 1994.

Schilit, B. N., Adams, N. I. and Want, R. 1994. *Context-aware Computing Applications.* 1994.

Sybase, Inc. 2012. *Analyze and Act on Fast Moving Data: An Overview of Complex Event Processing.* Dublin : Sybase, Inc., 2012.

The Workflow Management Coalition. 1999. *The Workflow Management Coalition - Terminology & Glossary.* s.l. : The Workflow Management Coalition (WfMC), 1999. Technical Report No. WFMC-TC-1011.

—. 1995. *The Workflow Reference Model.* s.l. : The Workflow Management Coalition (WfMC), 1995. Technical Report No. WFMC-TC-1003.

—. 1997. *Workflow Client Application (Interface 2), Application Programming Interface (WAPI) Specification.* s.l. : The Workflow Management Coalition (WfMC), 1997. Technical Report No. WFMC-TC-1009.

—. 2002. *Workflow Process Definition Interface – XML Process Definition Language.* s.l. : The Workflow Management Coalition (WfMC), 2002. Technical Report WFMC-TC-1025.

Ubiquitous Patient Monitoring and Smart Alert Generation in an intensive care unit supported by Low Cost Tablet PC Based Automation System Powered Through Open Source Software and Hardware Platforms. Antony, J., et al. 2013. South Asia Satellite : IEEE, 2013. Global Humanitarian Technology Conference. pp. 334 - 339.

2014. Visual Basic. *Microsoft Corporation.* [Online] 2014. www.microsoft.com.

Want, R., et al. 1992. *The Active Badge Location System.* 1992.

Ward, A., Jones, A. and Hopper, A. 1997. *A New Location Technique for the Active Office.* 1997.

Weiser, M. and Brown, J. S. 1997. *The Coming Age of Calm Technology.* 1997.

—. 1997. *The Coming Age of Calm Technology - Beyond Calculation: The Next Fifty Years of Computing.* [ed.] P. J. Denning and R. M. Metcalfe. New York, NY : Springer-Verlag, 1997.

Weiser, M. 1993. Some Computer Science Issues in Ubiquitous Computing. *CACM.* July 1993.

—. 1993. Ubiquitous Computing. *IEEE Computer Hot Topics.* October 1993.

Weske, Mathias. 2007. *Business Process Management: Concepts, Languages, Architectures.* New York : Springer, 2007. 978-3-540-73521-2.

WfMC. 1999. *The Workflow Management Coalition - Terminology & Glossary.* s.l. : The Workflow Management Coalition (WfMC), 1999. Technical Report No. WfMC-TC-1011.

—. 1995. *The Workflow Reference Model.* s.l. : The Workflow Management Coalition (WfMC), 1995. Technical Report No. WFMC-TC-1003.

Wireless Detection of End-to-End Passenger Trips on Public Transport Buses. Kostakos, V., Camacho, T. and Mantero, C. 2010. s.l. : IEEE, 2010. 13th International IEEE Conference on Intelligent Transportation Systems (ITSC). pp. 1795 - 1800.