Process Aware Mobile Systems

Applied to mobile-phone based data collection

Peter Khisa Wakholi

Dissertation for the degree philosophiae doctor (PhD)
at the University of Bergen

2013

RESEARCH ENVIRONMENT

REACHING THE LAST MILE

Travelling from Bergen to a village in Uganda I wondered how my journey was going to be. The last mile in Bergen is well served with infrastructure. One can choose to call a taxi, use the bus or use the tram. The means of transport is a choice between convenience and cost. At the airport, there are large aircrafts that carry passengers to their destination. So the journey from Bergen to Entebbe, Uganda is on a high-speed international airway route and takes about 12 hours to reach Entebbe Airport. On arrival at Entebbe, the available means of transport are private cars or taxis. A taxi takes me to the bus terminal where I board for the upcountry district of Busia – the kind of rural environment where 80% of Ugandans live.

As I move through the country side, I begin to contrast Norway and Uganda. The people in Norway stay in cities and small towns that are well served with infrastructure like roads, railway, electricity, water, telephone networks among others. In Uganda, these services are available depending on where you live. The major towns tend to have these services, although the quality of service may be poor due to congestion. As you move through the main roads connecting the towns, infrastructure and services available. For example, a road of tarmac or electricity network would typically provide services to communities near the main roads. So if you stay near a main road, you will find it pretty easy to access these services.

My village, however, is far off from the main road and the nearest major town (Busia) is 25 kilometres away. So on reaching Busia town, I hire a minibus. It will take me to a small trading centre approximately eight kilometres to my final destination. At that point the available means can be described as those that can navigate the narrow village paths. These include bicycles, motorcycles or simply walking. I find my cousin waiting for me on a motorcycle and as we ride to the village, I begin asking about our relatives and changes in the community. It is five years since I visited my mother's people and I am keen to know what is going on. My cousin tells me about the relatives who have died, those that got married and others that have started new families. As we move through the villages, I notice a mobile network mast and electricity power line in a small trading centre near home. The last time I was here, there was no such infrastructure. He tells me excitedly that things have changed, as he pulls out his Techno phone, made in China. Through this mobile phone he is able to keep in touch with relatives and friends. He mainly calls at night when the rates are low but also uses text messaging. He rides a motorcycle for private hire business locally known as *bodaboda* and his customers can reach him on phone, making it easier to do his business.

Being an ICT professional, this news sounds interesting. I wonder how much this communication infrastructure can bring him. Beyond his community, there is a world out there, where information flows across continents using the Internet. The infrastructure in his village, basic as it is can link him to the Internet. I ask him if he has heard about the Internet or email among others and he says yes. Having dropped out of school after completing his Ordinary Level, he is quite literate and aware of several things. However, he does not use Internet because it is of little significance and the only device that he has is the phone which can only provide little. I have a look at the phone and it is Internet-enabled, something he did not know. I try to access websites like Yahoo and Google and it is possible to search for information and send email messages. A pleasant surprise!

I realise that even though this infrastructure has reached my village, many web-based services are not available, largely because they were developed for an infrastructure that reflects the super road network in Norway. The situation above, at the edge of global connectivity, reflects what one could expect to find in many parts of the developing world. A complex infrastructure service like a tram cannot be expected to work in this village over the next 50 years, while less complex services like *bodabodas* are able to assist the population.

The OpenXdata project sought to address this type of challenges in order to bring services to such ‘edge’ communities. The project seeks to create *bodaboda*-like services by developing a generic tool for data collection that can work in constrained environments of rural parts in developing countries. Leveraging the growing mobile network infrastructure, increasing number of users on the mobile network and mobile devices getting smarter, the projects seeks to enable quick and easy development of data collection forms that can run on the simplest Java-enabled mobile devices, and therefore receive and provide data from the Internet. This research focuses on the processes related to data collection by integration of workflow systems to support mobile data collection.

The work undertaken in this thesis is part of the Open Mobile Electronic Vaccine Trials (OMEVAC) project. The project seeks to develop a platform for conducting clinical trials using mobile devices, based on OpenClinica. It is funded by the Norwegian Research Council. The OpenXdata system has been developed under this project as a generic tool for definition, collection and management of data.

ACKNOWLEDGEMENTS

The author wishes to thank several people. I would like to thank my wife, Barbara, for her love, support and patience during the past four or so years it has taken me to work on this research. She has worked hard to take care of the children when I was away. I would like to thank my parents, Mr and Mrs Wakholi for their unending love, support and encouragement for me to excel. I also wish to register my gratitude to my one brother and seven sisters for their moral support that gave me great morale to keep going.

I would also like to thank Professor Weiqin Chen who has not only been an advisor but a friend as she helped me cope with my numerous escapades in Bergen. Special thanks also go to Mr Jørn Ivar Klungsøyr and Professor Thorkild Tylleskär who provided sponsorship and mentorship under the OMEVAC project. I would like to thank the OpenXdata development team, especially Ronald Kayondo who provided the platform that has been used to develop and test the work in this thesis. Special thanks also go to the team at Makerere, SNV Uganda, Water Aid and the various districts in Uganda who provided avenues to validate this work. I also register my gratitude to Maria Muzaaki and Harriet Mutonyi for reading and editing this thesis.

Finally and most importantly, I thank the almighty God who I have known as my Lord and saviour from my teenage years. Without his guidance and favour, I would never have attained this. Indeed the wisdom he has given me gives me confidence of greater things ahead.

ABSTRACT

The quest to provide computing services to resource-constrained environments in developing countries is becoming a reality due to the wide use of mobile phones and penetration of mobile networks. Nowadays, many organisations use Mobile Data Collection (MDC) tools to enable the collection and digitalisation of data at source, hence improving quality and increasing efficiencies. Mobile devices and environments present challenges to computing and application design that need to be overcome.

Beyond mere digitalisation of data, MDC tools need to consider the process-related aspects of data collection used in paper-based routines expressed through paper trails. This lack of process-related support hinders the adoption of MDC routines in cases where great attention is paid to the data collection process. In conventional information systems, process-related features are implemented using workflows which may be embedded in an application or separately defined using Workflow Management Systems. This has led to the development of Process-Aware Information Systems (PAISs), which are software systems for managing and executing operational processes involving people, applications, and/or information sources on the basis of process models. PAISs facilitate the inclusion of process-related activities which include the ordering of various tasks undertaken to achieve a business goal (control flow), the collaboration among various entities, and the allocation and provision as well the exchange of relevant information necessary for decision making.

The use of mobile devices to carry out tasks is not the most preferable choice due to hardware limitations. Mobile-based systems should integrate with existing desktop-based solutions to provide a multiple access platform for work execution. This calls for integrating workflow systems with generic mobile data collection tools, which would require modifications in approach, methods and architecture to cater for device

and environmental constraints in order to enable the mobile devices to be used appropriately.

This thesis proposes a range of techniques that can be used to enable workflow support for mobile data collection. The overall goal is to minimise changes in workflow systems architecture, since these are based on widely agreed standards. Therefore, we propose an approach for online execution of work, for scenarios where network connection is readily available, and offline execution of work controlled by a workflow engine, when the connection is not available. A workflow adapter is proposed to enable matching of forms for data collection and workflow specifications. A distributed architecture for offline data collection based on partitioning a process model into fragments for distributed execution is also proposed.

The methods proposed have been implemented with the OpenXdata MDC suite used for data collection and YAWL workflow management system. The OpenXdata and YAWL platforms adhere to commonly agreed standards for mobile data collection and workflow management and thus provide generalizable concepts within the domain of process-aware mobile data collection. Experiments were carried out on foundational concepts in order to determine that all relevant workflow-related constraints are observed. In addition, artefacts developed from the application of these methods were implemented in real life projects. The findings and results of these applications were used to validate the methods and frameworks suggested.

LIST OF PUBLICATIONS

The main results of the work presented in this thesis are documented in the papers in Part II. In the following we give an overview of these research papers, describing the topics of each paper and indicating how much of the results are credited to the author of this thesis.

PAPER 1: WORKFLOW SUPPORT FOR MOBILE DATA COLLECTION

Authors: Peter K. Wakholi, Weiqin Chen and Jorn Klungsoyr

Publication status: Published in Enterprise, Business-Process and Information Systems Modeling, Lecture Notes in Business Information Processing, 2011, Volume 81, Part 7, 299-313, DOI: 10.1007/978-3-642-21759-3_22

Contribution: Peter K. Wakholi is the main author, responsible for about 90% of the work.

Main topics: This paper presents a framework for integration of generic Data Collection tools with Workflow Management Systems (WFMS) to enable MDC in such resource constrained environments. Furthermore the tool was implemented based on this framework using a vaccination registry project that uses mobile phones to record and track child immunisations as a case study.

PAPER 2: WORKFLOW PARTITIONING FOR OFFLINE DISTRIBUTED EXECUTION ON MOBILE DEVICES

Authors: Peter K. Wakholi, and Weiqin Chen

Publication status: Published in Proceedings of the CAiSE'12 Forum at the 24th International Conference on Advanced Information Systems Engineering (CAiSE), Gdańsk, Poland.

Contribution: Peter K. Wakholi is the main author, responsible for about 80% of the work.

Main topics: This paper proposes an approach for workflow partitioning and an algorithm that enables automatic discovery of such partitions from a process model in order to enable assigning grouped tasks for offline execution. The algorithm was implemented, evaluated and validated using a simulation tool.

PAPER 3: OPENXDATA WORKFLOWS – AN APPROACH TO PROCESS-AWARE MOBILE DATA COLLECTION

Authors: Peter K. Wakholi, Weiqin Chen and Jorn Klungsoyr

Publication status: Under review by Parallel and Distributed databases Journal

Contribution: Peter K. Wakholi is the main author, responsible for about 80% of the work.

Main topics: This paper proposes architecture for workflow execution in offline environments, where work items are uploaded with a partition of containing two or more tasks. These tasks are executed offline in a distributed and disconnected environment and synchronised with the main workflow model once connection is established. The paper also presents the results of an implementation of this proposed architecture and evaluation based on three real-life scenarios.

PAPER 4: A FRAMEWORK FOR MOBILE-BASED ELECTRONIC DATA COLLECTION IN CLINICAL TRIALS

Authors: Peter Wakholi, Weiqin Chen, and Jørn Ivar Klungsøyr

Publication status: Presented at 7th International Conference on E-Governance (ICEG-2010) and accepted for publication by Taylor and Francis Publishers

Contribution: Peter K. Wakholi is the main author, responsible for about 75% of the work.

Main topics: This paper proposes a framework for deployment of process-aware information systems for clinical trials utilising mobile-based routines as source documents. This framework is based on the process oriented view of electronic data collection and relevant industry standards.

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

API	Application Interface
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMDS	Business Process Modeling, Development, and Support
CAISE	Conference on Advanced Information Systems Engineering
CRF	Case Report Form
DWO	District water officer
EU	European Union
FDA	Food and Drug Administration - an agency of the United States
GPRS	General packet radio service
GPS	Global Positioning System
GUI	Graphical User Interface
HPM	Hand Pump Mechanic
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines
ICEG	International Conference on E-governance
ICT	Information and Communications Technology
IS	Information Systems

ITU	International Telecommunications Union
M4W	Mobile for Water
MDC	Mobile Data Collection
MWE	Ministry of Water and Environment
OMEVAC	Open Mobile Electronic Vaccine
OpenXdata	A generic mobile data collection tool developed under the OMEVAC project .
PAIS	Process Aware Information System
PDA	Personal Digital Assistant
SMS	Short Message Service
SMS	Short Message Service
TAR	Technical Action Research
WAP	Wireless Application Protocol
WFM	Workflow Management
WFMC	Workflow Management Coalition
WML	Wireless Markup Language
XML	Extensible Mark-up Language
YAWL	Yet Another Workflow Language

TABLE OF CONTENTS

Part 1: Context and Overview

1. INTRODUCTION

This thesis presents work carried out to enable workflow support in mobile environments. We present a number of related artefacts that contribute to the domain of mobile based workflow systems, with particular focus on enabling workflows using low-end mobile devices in challenging mobile environments. In this chapter we give some background and motivation for our work, and briefly present the artefacts, and give an overview of the thesis.

1.1 BACKGROUND

In organisations, the daily operation is governed by a set of cooperative business processes, in which interactions with humans and information systems are involved. These processes aim to produce a product or service for a user or a customer. The processes in an enterprise are not static but evolve over time to meet customer needs and provide greater efficiencies in managing the business. Conventionally, processes have been supported by the exchange of information recorded on paper. This paper-work enables the sharing and archival of information as work is transferred from one desk to another until the process is fully executed. Take for example an order from a customer. Once the order is received, it is stamped and forwarded to the relevant department. There may be additional business functions to be performed before the goods are finally delivered, e.g. checking inventory information, giving discounts, packaging, shipping, among others. These tasks are executed when the relevant office receives a request containing the relevant information about the order in a paper trail.

It can be seen that the paper-based process is inefficient due to the time it takes to move information from one desk to the other. This even becomes more prominent in modern organisations which are not located in one area but span across buildings and cities. Some work also involves teams working collaboratively in remote and hard-to-reach areas, making the paper-based transfer of information a major hindrance. With the advent of computers, organisations have developed software applications to manage the information. These are the so-called computer-based IS which use ICTs to support operations, management and decision making. For example all information about the customer order in the previous example would be stored in an Information System.

The mere storage of such information is not adequate as the underlying business process as seen in a paper trail, also need to be supported. Process Aware Information Systems (PAISs) are used to control the performance of business processes, through logic that deals with the control and flow of activities - referred to as workflows. Business process management (BPM) includes concepts, methods and techniques to support the design, administration, configuration, enactment, modelling and analysis of business processes. Sharp and McDermott (2009) define BPM as a "collection of interrelated work tasks, initiated in response to an event, achieving a specific result for the customer and other stakeholders of the process".

Whereas conventionally, computing has been conducted on stationary computers on desktops in an organisation, the advent of mobile computing has changed the landscape to include more platforms. Inflexible desktop computers have been complemented with mobile devices such as laptops, pads and mobile phones. Primarily, some mobile devices like phones have been developed to be used for communication unlike desktop computers and laptops, whose aim is to enable computing. However, their roles have gradually evolved to include advanced computing functions and are capable of running applications. Of interest are smart phones, which are a new class of cell phones that can facilitate data access and processing with significant computing power. In addition to conventional voice communication and messaging functionality, a smart phone usually provides ability to run applications. Smart phones are therefore like small, networked computers.

Mobile phones provide an opportunity for computing in developing countries, not because they are the preferred mode of operation but because they are cheap and available in rural areas. These devices were however, not manufactured as primary computing devices and therefore offer limitations related to screen size, processor speed, memory, keypad size, and many more. Any increase in these capabilities leads to unattractive prices for the devices which could make their use infeasible. Despite these limitations, mobile phones still provide great interest for computing and our focus in this thesis is on the cheapest devices that can run applications because they are feasible for use in rural parts of developing countries.

Conceptually, if process-aware mobile data collection is to be adopted in the place of paper-based routines, some challenges need to be overcome. Paper-based routines provide the flexibility to work in remote and disconnected environments. Conversely, conventional workflow systems are designed for client-server architecture or web-based-distributed architecture. In the client-server architecture mode, the workflow engine resides on the server and is accessed by other devices (clients) via a network link. The distributed architecture has several workflow engines residing on different servers but are coordinated to realise a business process. Such systems are usually implemented across multiple organisations. In all these cases, a constant communication link is needed between the servers and clients.

Therefore, workflow technologies need to be modified to support data collection in remote and server-disconnected locations. Moreover, the adoption of such environments should not impose undue constraints on conventional workflow systems. A typical deployment environment would require the ability to use the most appropriate method (web or mobile), hence requiring a multi-platform workflow capability. For example, a worker in the field conducting one or more tasks of an activity may need to complete them on a mobile phone without requiring connection to a central server. Once these tasks are completed, they may be transferred to the workflow server and accessed by another person via a desktop PC. This therefore requires the conventional client-server model used by many systems to be extended rather than replaced, in order to enable work execution in mobile-environments.

1.2 CONTRIBUTION

The main contribution of this thesis is architecture, methods and frameworks for extending conventional workflow systems to the mobile environment. These contributions are based on standard practice within the workflow paradigm and uses tools and reference models that cut across platforms. The first contribution is a framework to enable generic MDC tools to adopt the use of WFMS, thereby enabling workflow support for mobile data collection. The second and third contribution addresses issues to do with enabling offline execution of more than one task in a process in order to allow more work to be done before a connection is required. Lastly, we address the problem of how to make it all work by proposing a framework that allows for offline execution. This is complemented by an architectural design of a simple workflow engine to control the partitioned process and methods for synchronising the distributed systems, in order to maintain central control. All this work is evaluated using case studies, experiments and field deployments.

1.2.1 MOBILE DATA COLLECTION WITH WORKFLOW SUPPORT

As explained in section 1.1, the introduction of mobile phones as a possible platform for execution of workflows requires an extension of the conventional architecture. We considered a scenario where mobile devices were to replace paper for field data collection by using generic data collection tools. Architecture for extending conventional workflow systems to work with mobile phones was proposed. In addition, methods for mapping WFMS with MDC tools in order to provide process definition capabilities have been provided.

Using this approach, it is possible to design a form for data collection and define a workflow specification that will be used to control the process. Forms to be used for data collection are mapped to tasks in the workflow definition and questions in the form that are vital for decision making are mapped to task variables. The process is executed using a workflow adapter that implements client-agent-server architecture. Details of this work can be obtained in section 6.4.

1.2.2 MOBILE DATA COLLECTION IN DISCONNECTED ENVIRONMENTS

When mobile phones are used in environments where connection with the server is not always guaranteed, there is a need to provide for offline execution of work. Currently, MDC tools are designed for the offline mode of operation and the introduction of WFMS need not impose undue constraints.

We provide a method to partition the workflow process for distributed execution and an algorithm for automatically identifying such partitions from a process model. The partitions provide a group of tasks that can be executed offline without violating the workflow definition and process constraints. An algorithm that automatically discovers such partitions in order to enable the system to group tasks was developed. The offline work items are uploaded with the workflow logic which is decoded by a simple and light workflow engine on the mobile to determine the next activity. Finally, the activities need to be synchronised in order to maintain control by the main workflow engine. Details of this work can be obtained in section 6.5.

1.2.3 FIELD BASED ELECTRONIC DATA COLLECTION

This thesis proposes a framework that utilises workflow technologies for deployment in a mobile environment in order to control and manage the entire process. Key issues that the framework addresses are flexibility of work in disconnected environments; tracking work progress; keeping audit trails and generating automated reports for monitoring and conformance checking. A number of real-life application scenarios utilising the artefacts developed are discussed in section 4.3.

1.3 THESIS OVERVIEW

This thesis is based on a collection of the four research papers provided. The first paper explains work on workflow support for data collection published by the CAISE BPMDS conference of 2011. The second paper refers to workflow partitioning for offline distributed execution, published at the CAISE forum 2012. The third paper presents the tool that was developed for offline workflow mobile data collection which is under review by the Parallel and Distributed Databases journal. Finally, we present the framework that explains how this knowledge can be utilised for clinical trials presented at ICEG workshop 2010.

In chapter 2 we give a characterisation of the problem. We provide an explanation of paper-based data collection routines as used by teams working in the field. This is then contrasted with mobile-based data collection routines. The challenges of collecting data in the mobile computing environment are then discussed. This is followed by a discussion on conventional workflow systems. Based on the problem identified, the research questions are defined as how to enable ubiquitous and flexible collaborative mobile data collection in the place of paper-based routines.

In chapter 3 we review literature and discuss the current state of the art. The topics discussed include the paper-based data collection and the move to electronic data collection in clinical trials. The process related issues are highlighted by explaining workflow architecture and standards used on developing Information systems. The approaches of using distributed systems to manage workflows in disconnected environments are also explored. Finally, we provide the gaps addressed in this research.

In chapter 4 we provide the requirements for enabling workflow support in mobile computing environments. Based on the questions identified in section 2.6 and the related literature, we discuss the requirements for enabling workflow support for

MDC tools. Furthermore we discuss the requirements for workflow support to enable flexible and ubiquitous data collection. The goal is to reduce on the prohibitions for the use of mobile device for data collection in the place of paper. Finally, we discuss the required architecture to enable distributed workflow systems. The distributed environment is created when partitions are created to enable offline data collection.

In chapter 5 we discuss the design science research methodology that provided the general approach used. The action research method was used to develop artefacts and evaluate them. This was complemented with case studies and real-world based applications to understand and contextualise the problem and to validate the artefacts. Experiments were used to test out the algorithms developed using computer applications. The section discusses in detail how these approaches were applied at each stage of the research.

In chapter 6 we provide an overview of the artefacts developed in this research. These include the enabling generic data collection with support of workflow systems and architecture for mobile workflow execution. Furthermore, we present methods for partitioning workflow process for offline execution. A tool that utilises these concepts is then presented. It contains architecture for implementing a service for offline data collection working with conventional workflow systems and design of a simple workflow engine that guides the execution of work on mobile devices. Using a case study of a clinical trial, guidelines are provided of how mobile phones can be used to support data collection following guidelines and regulations.

In chapter 7 we discuss the fulfilment of the research objectives related to generic workflow support for data collection, providing for flexibility in data collection as provided by paper-based routines, validation of developed artefacts and the significance of the research. In addition, we provide and compare related scientific approaches to the problems identified in mobile-based workflow systems, distributed workflow systems and workflows in disconnected environments.

The thesis ends with a summary of the work done and issues that still need to be addressed for future work presented in Chapter 8.

2. PROBLEM CHARACTERISATION

This chapter discusses the purpose and motivation of using mobile devices for data collection. In addition we examine the paper-based workflow routines represented by what is commonly referred to as paper trails. Thereafter we discuss the mobile computing environment and highlight the limitations that it posits to achieving ubiquitous workflow implementations. These challenges are best understood by examining current workflow system implementations and architectures. The last part of the chapter then outlines the issues that this thesis aims to address.

2.1 DATA COLLECTION PROCESS

Data collection is a term used to describe a process of preparing and collecting data, for example, as part of a clinical trial or a similar project. The purpose of data collection is to obtain information to keep on record, conduct analysis and make decisions. Data collection usually takes place early on in a project, and is often formalised through a data collection plan which often contains the following activities (contributors 2012).

- Pre-collection activity — agreement on goals, target data, definitions, and methods.
- Collection — data collection.
- Present findings — usually involves some form of sorting analysis and/or presentation.

For instance in a clinical trial, the pre-collection process starts with identifying the instruments that will be used for data collection. These instruments are usually structured as forms that need to be completed for each case. The data collection process involves using some tools to collect the required data. The tools used may be paper-based or electronic (e.g. web, mobile phone, PDA, etc.). The paper documents act as source document. A source document is a document in which data collected for a clinical trial is first recorded, commonly the case report form (CRF). A CRF is a paper or electronic form used in clinical trial research to collect data from a particular event, for instance a participant attending a clinic visit. In multi-centre trials, the process of capturing data on paper and entering it into the system is labour intensive, inefficient and error prone. Despite advances in mobile computing, there still remain many impediments to replace paper as authentic source documents.

2.2 FROM PAPER-BASED TO MOBILE DATA COLLECTION

Conventionally data collection has been done using paper-based instruments that have inherent limitations and risks. The risks relate to the fact that data can be lost between the field and the research office and it requires manual coding and double checking for accuracy, which adds both time and the potential for human error (Weber, Yarandi et al. 2005). This has led to the need for computerised data collection systems to allow investigators to instantly capture data electronically, thus enabling them to maintain control over data and minimise loss and errors. Moreover, data collection and management processes can be added to include tasks such as skip logic, dynamically selecting from a list of possible values, validation on inputs, enabling forms when due, providing/auto filling background information, among others.

Using paper-based routines has many disadvantages which result in erroneous data in the database and longer duration of clinical trial. There is therefore an increasing reliance on electronic CRF, specifically web-based systems to address these challenges and improve electronic data collection (EDC). Mobile devices provide an opportunity for EDC in developing countries due to their portability and wide use – unlike web based systems.

Specialised and generic software allow for the creation of customised data collection forms that meet the requirements for a data collection effort have been developed. Initially, the systems run on personal computers, but due to the limitation of carrying them in the field, mobile phone based tools have been developed. Examples of MDC tools include Cell Life, Dimagi, EpiSurveyor, GATHER, Open Data Kit, OpenXdata and many more. These tools are based on agreed open standards under Openrosa Consortium (OpenROSA 2011). They provide generic, modular and extensible platforms for data collection and use common standards and tools for design of forms to collect data, rendering the form on a mobile device and storing collected data on a server to analyse. These standards are widely recognised in the MDC community (Anokwa, Hartung et al. 2009). In order to enable data collection in disconnected environments as it often is the case with mobile networks, data collection forms are downloaded to the mobile device. When forms have been appropriately filled, they are uploaded to a central server when connection is established.

MDC applications consist of three components; the mobile client, the application designer and the server storage. The mobile client provides the ability to record information, display information and wirelessly transmit information to a server. The server-side data storage system provides a mechanism to easily store and retrieve the digital records (Anokwa, Hartung et al. 2009). The form designer allows users to build forms and questionnaires with complex logic and interactions. It produces the logic used by all the other tools to render the application to the user as well as create databases from which data can be extracted for visualisation and reporting. The mobile client is a mobile phone application that allows users to download application logic, interact, using physical or on-screen keyboards, and sends information to servers wirelessly. The server storage is where data is stored in a format that saves the application logic built by the user.

2.3 PAPER-BASED WORKFLOW ROUTINES

There are three types of data; non-time dependent, time dependent and cumulative data (Moon 2006). Non-time dependent data is the data collected at a snapshot in time. Such data include subject demographics and medical history. Time dependent data is data collected repeatedly over time through multiple visits. Cumulative data is data collected over time but not linked to a specific visit. After clinical trial data are collected on the paper forms they have to be entered in the electronic database in order to perform computer data analysis. For this purpose, investigators usually send the copies of paper CRF to the data centre where data entry into a database is done.

Clinical trials use work plans like checklists, schedules, protocols, work programmes etc., to provide workflow related elements alongside the data collection process. The plans are defined based on study protocols, which specify the duration and structure of the study and the standard guidelines which must be followed by all participants. The data collection process is most difficult as it involves field-based studies which could be in single or multiple sites, often in remote rural environments.

Workflow systems provide ways of routing information objects among users, and to specify automatic actions to be taken in that routing typically according to certain process models. Such process models should be understood as a formal representation of work procedures that control the order in which a sequence of tasks are to be performed (Bardram 1997). They provide the plans for collaboration by presenting information on activities undertaken in order to aid decision making on next course of action.

2.4 MANAGEMENT OF FIELD DATA COLLECTION

Field data collection aims to capture data from areas that are located away from the main administration centre. Fields are usually located in many sites and can span across several countries. Teams may consist of enumerators working individually or in groups. In many cases, the data collection process is collaborative, in that teams are required to work together to complete a data collection process.

An information system should therefore support such processes and collaborations. We undertook a study about the management of field data collection in a clinical trial using the PROMISE-PEP study (ClinicalTrials.gov 2008). Figure 1 provides four scenarios for field data collection, using mobile phones. The requirements for workflow support for each of these scenarios are explained below:

1) **Collaborative team** (Figure 1(a)): This is composed of more than one member each having a mobile device, collaborating in a field data collection activity. One of the team members plays the role of supervisor and therefore manages and controls the work done by other team members. In a workflow scenario, one mobile device would work as a server to clients possessed by other team members. The mobile server delivers and controls the tasks to the clients, and reports progress to the main system.

2) **Lone field worker** (Figure 1 (b)): This is a situation where each team member works separately. At the end of the field activity, he/she updates the team with the work done by submitting the CRFs. In this case, the user has a mobile device with a workflow engine and a workflow definition guiding the execution of tasks. The workflow engine on the device should be able to report back to the main system when connection is established, on work undertaken in order to coordinate with other activities.

3) **Centralised Team** (Figure 1 (c)): The scenario is common when the team needs constant update of the process in order to complete work in the shortest time possible. Each member of the team can receive tasks on the mobile device, execute them and update the main workflow engine immediately. This

scenario would require the team to be constantly connected so as to receive the tasks and report on completion status.

4) **Mixed team:** Many organisations do not have clearly defined paradigms as provided above, but rather maximise the opportunities present. A typical scenario would therefore provide for all possibilities so that they can be used when necessary. This is the *mixed team* scenario, which would require all the users to either work in a connected manner or disconnected with distributed workflow engine clients, depending on the nature of work to be done and the environment in which it will be executed.

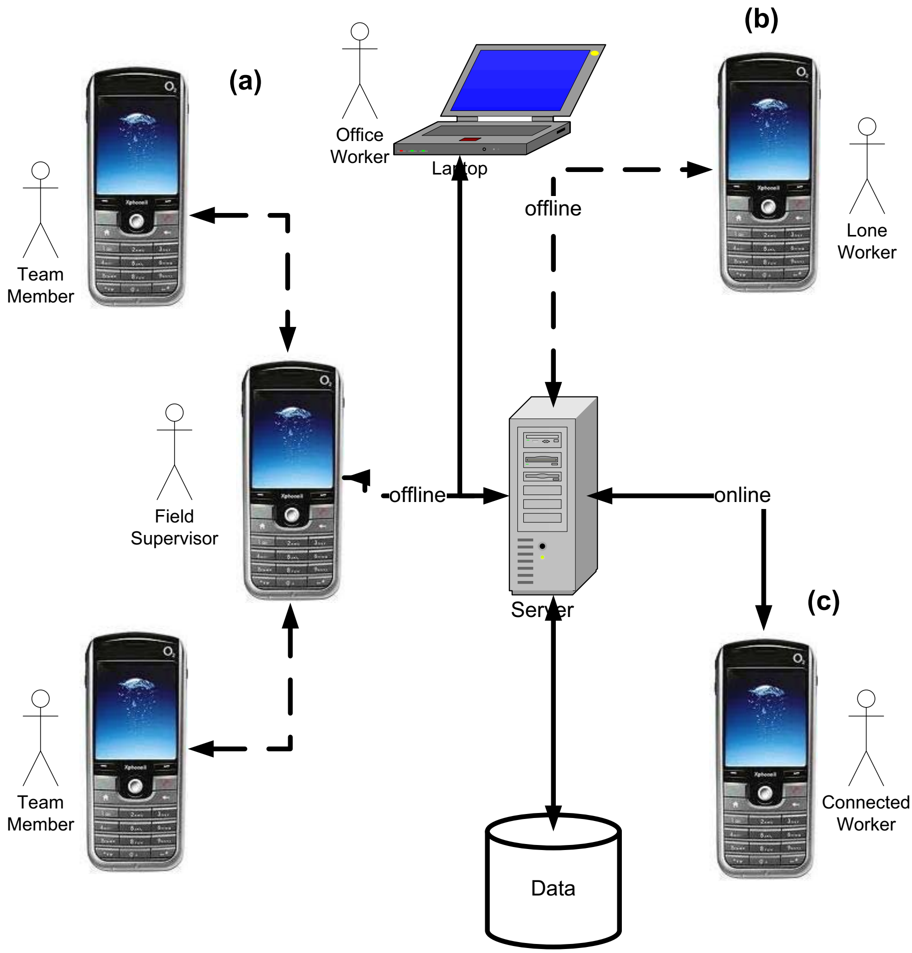


Figure 1: Topology diagram for typical application scenario

2.5 THE MOBILE COMPUTING ENVIRONMENT

2.5.1 SMART PHONES

Smart phones are cell phones that use an advanced operating system, on which customised mobile computer applications and services are based. The range of smart phones available on the market is diverse with varying capabilities. Zheng and Ni (2006) provide a general comparison of cell phones along with the three generation of cellular systems. From this analysis, it can be observed that while cell phones are becoming smaller and lighter, they are more powerful and sophisticated and can perform a wide variety of operations. Mallick (2003) classifies mobile devices as web-enabled phones, low-end smart phones, palm-sized PDAs, High-end smart phones, handheld PCs, Tablet PCs and Custom devices. These functional capabilities and costs rise as one moves across the spectrum.

Over the last ten years, Africa has experienced an incredible boom in mobile phone usage. An observer newspaper survey (Fox 2011) reports that in 1998, there were fewer than four million mobiles on the continent. Today, there are more than 500 million. In Uganda alone, 10 million people, or about 30% of the population, own a mobile phone, and that number is growing rapidly every year (Fox 2011). The mobile phone therefore provides an immense opportunity for extending computing where a billion people use only 4% of the world's electricity and many cannot afford to charge a computer, let alone buy one. According to an African survey report (News 2012), smart phone penetration rates in Africa are now at a 17 to 19 % of the handsets being used. The rest are split between either "feature" phones or basic "dumb" phones (albeit with SMS capability). A survey conducted by research among teachers in rural schools under the CU@school program that tracks pupil and teacher absenteeism (Wakholi 2010), showed that most people in Uganda use basic phones, web-enabled phones and low-end smart phones.

The constraints of using mobile devices for computing are a result of portability and mobility requirements. These include short battery life which is approximately a maximum of five hours if in constant use. The devices are also subject to theft and destruction and hence are unreliable. They are also often unavailable as they powered-off to conserve battery. They also have limited capability in display, memory, input devices, and disk space.

Zheng and Ni (2006) argue that smart phone computing requires rethinking of system design with regard to power consumption, user interface, network capability, storage, and software architecture. This is needed even more, in the OpenXdata project which seeks to extend computing to feature phones. Technologies that have been conventionally designed for PC based solutions and the Web need to be modified to enable smart devices to support mobility and adaptability. The design challenges are even greater when the target devices are at the bottom end of the classification spectrum.

2.5.2 WIRELESS NETWORKS AND INTERNET ACCESS

According to the ITU Africa report on telecommunication ICT indicators (ITU 2008), there were only 8.6 million telephone subscribers in Africa, most of them in Northern Africa and South Africa in 1990. Norway had more telephone subscribers than all of Sub-Saharan Africa and mobile communications was virtually non-existent with only six networks in operation. No African country was connected to the Internet in 1990 (ITU 2008). This situation has changed dramatically, with all countries having mobile networks and all connected to the Internet.

Numerous wireless technologies are being utilized by mobile computing to enhance a user's experience with regard to both user mobility and device mobility. The coverage and quality of the wireless network is influenced by many factors. These include the type of technology in use, number of users served by base stations, distance from base station, among others. According the ITU report on Africa's mobile markets, Africa's ICT future is definitely a wireless one. In the absence of fixed-line networks, in addition to a lack of PCs, Mobile phones have become increasingly used as means to access the Internet. The common technologies used include GPRS and 3G, thus providing real opportunities for Internet.

Wireless networks pose some constraints in their use for mobile data collection. Frequent disconnections occur due to handoff blank out (>1*ms* for most cellular phones) when a mobile device moves from one cell to another, drained battery disconnection, battery recharge down time, voluntary disconnection (automatically turned off to preserve battery power, also off overnight), theft and damage (hostile environment). In addition there is limited communication bandwidth making the magnitude and capacity of mobile networks slower than fixed network. This is caused by higher transmission bit error rates, uncontrolled cell population making it difficult to ensure quality of service thus reducing on network availability. The limited communication bandwidth exacerbates the limitation of battery lifetime as devices make several attempts to establish communication. Enabling the use of mobile phone for computing in areas where there is no network and allowing for synchronisation once network has been restored is the common practice.

2.6 ISSUES ADDRESSED IN THIS THESIS

A workflow solution for MDC should take into consideration the challenges observed by employing new computing paradigms. Paper-based routines provide the flexibility to work in remote and disconnected environments. Work conducted in the field is often done on paper and filed at the end of a specified time. The workflow system therefore needs to allow for this flexibility by enabling execution of multiple tasks before connection to the server is required. This is a shift from the conventional workflow systems which provides for client-server architecture and requires synchronisation before the next task is executed and in some cases web-based-distributed architecture which require constant connection links.

This research focused on the development of mobile systems for EDC by addressing workflow-related problems in their use for field-based activities in rural parts of Sub-Saharan Africa. The main research question was to investigate ”*how workflow systems can be used to support mobile data collection, management and dissemination in place of paper-based routines used for field activities in resource constrained environments*” In order to answer the main question and achieve new knowledge in this area, the following sub-questions were investigated.

1. How can generic workflow support be implemented for mobile data collection tools?

2. How can mobile-based workflows be implemented to mirror the flexibility and ubiquity provided by paper-based routines in field activities?

3. How can workflows be implemented in environments where work is conducted offline?

3. LITERATURE REVIEW

This chapter presents literature on current practice and approaches to workflow management in mobile environments. In section 3.1 we discuss the fundamental concepts necessary to contextualise and develop a scientific understanding of the problem. Section 3.3 discusses the environment by highlighting key issues in mobile devices and networks. Finally section 3.3 contrasts these approaches and advances knowledge by proposing new approaches to mobile process-aware information systems.

3.1 FUNDAMENTAL CONCEPTS

In this section, we provide a clear understanding of workflow systems, by defining various concepts related to workflow systems. We explore patterns used to understand and classify workflow processes, describe the construction of conventional workflow systems, explain the workflow management reference model and explain the use of petri nets as a conceptual approach for understanding workflows.

3.1.1 WORKFLOW DEFINITIONS

Formal approaches to defining and enacting workflow process provide a theoretical basis for developing cross-cutting artefacts that can be used in any workflow scenario. The basic view of a workflow process is that it is case-driven, i.e., tasks are executed for specific cases. Three perspectives are used for describing workflows: (1) the control-flow dimension, (2) the resource dimension, and (3) the case dimension (van der Aalst 2004).

In the control flow definition, a description of the process itself is defined by routing of work as tasks are undertaken to meet a business process. For example in clinical trials, the control flow definition could be the ordering of tasks like registration of a new patient, conducting blood test, etc. There are several patterns that may be followed in defining the ordering of tasks. Common among these are conditional, sequential, parallel and iterative routing (van der Aalst, ter Hofstede et al. 2003).

The resource perspective is concerned with the classification of the resources to be used and how to map work to resources. They aim to capture the various ways in which resources are represented and utilized in workflows. Resources are commonly human beings (e.g. doctors, nurses, and other field staff in a clinical trial) or non-human (e.g. X-ray devices, software, hardware, etc). Resource classifications refer to grouping based on organisational roles (e.g. doctors) or organisational units (e.g. Statistics Department). Russell, Ter Hofstede et al. (2005) present rich set of resource patterns which define the allocation of resources in automated business processes.

The case perspective gives a concrete piece of work to be undertaken. When a case is available, a process model is created by the workflow system, such that each task that can be undertaken is enabled. The case perspective also defines the data that is used in making decisions regarding the routing of the case.

The work presented in this thesis explores these three perspectives in relation to mobile data collection.

3.1.2 WORKFLOW PATTERNS

Workflow patterns provide a conceptual basis for workflow based systems in defining process models (van der Aalst, ter Hofstede et al. 2003). The patterns are classified in four perspectives; control flow, data, resource, and exception handling. Tools and languages that are developed to support workflows can be evaluated based on the number of patterns that they can support (Recker, Rosemanny et al. 2007). In this thesis, we use these patterns to draw conclusions on the generalisation of the ideas presented and define the bound in which the methods can be implemented.

Control flow patterns are defined when developing a process model. Their execution semantics apply to process instances and therefore guide the execution of a process. The use of control flow patterns enabled us to examine the suitability of the artefacts developed and to assess the relative strengths and weaknesses of various approaches to process specification and implementation in a mobile computing environment. There are some commonly occurring control flow patterns in business processes. These include *sequence, and split, and join*, *exclusive or split and exclusive or join (Weske 2005)*. The "*sequence"* pattern is when a task in a process in enabled after completion of a preceding task in the same process. The "*and split"* pattern is divergence of a branch into two or more parallel branches each of which execute concurrently. The "*and join*" pattern is a point in process where multiple concurrent threads converge into a single thread of control. The "*exclusive or split*" is point in a process model where one of several branches is chosen. The "*exclusive join*" is a point in a process model where two or more alternatives come together without synchronisation. These control flow patterns are vital for the definition of process models and should be supported by any meta-model or workflow system (van der Aalst, ter Hofstede et al. 2003).

Common resource allocation patterns include direct, role-based, deferred, authorisation, separation of duties, case-handling, history-based allocation and organisational (Weske 2005). *Direct allocation* is where an individual, rather than a position in an organization, is allocated to all activity instances of a particular activity model. *Role-based* allocation where work is allocated to the members of organizations based on their roles. *Deferred allocation* is where the decision about who performs an activity instance is only made when the business process is executing (run time) of the business process. *Authorization allocation* is where persons are allocated to activity instances based on their positions. *Separation of duties* refers to a requirement to allocate to different resources within one business process. In *case handling* certain activities in a business process require an understanding of the overall case and therefore allocation based on the resource that was earlier engaged. *History-based allocation* is where a resource is allocated to an activity instance based on what this resource worked on previously, i.e., on the history. Finally, *organizational allocation* is when positions and not roles within the overall organization are used to allocate activity instances.

There are several data patterns which describe how data resources are managed during business process execution (van der Aalst, Adams et al. 2010). These features describe the data visibility, data interaction, data transfer and data based routing. Data visibility patterns relate to the scope and visibility of data elements in a process. Data interaction, patterns focus on the way in which data is communicated between active elements within a process. Data transfer patterns present the means by which the actual transfer of data elements takes place between process elements. Data-based routing characterises the manner in which data elements can influence the operation of other aspects of a process, particularly the control-flow perspective.

3.1.3 CONVENTIONAL WORKFLOW SYSTEMS

A WFMS is defined as: "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." (Fischer 2002). Workflow systems can trace their origins from concepts of office automation which begun in the 1980s. Its aim was to develop IS that support groups and whole organisations in routine activities with the goal of creating a "paperless" office (Mahling, Craven et al. 1995). The basic idea behind the term "workflow" is that work flows through an organisation. A workflow can therefore be regarded as the automation of these processes in whole or in part as documents, information, or tasks are passed from one participant to another for action, based on some procedural rules (Weske 2005).

In organisations, work is undertaken to meet a business objective. Workflows therefore foster a mainly process-oriented perspective on organisations. The process-oriented view comprises activities and their relationships within and to an organisation's context. Each work item is considered to be a case where tasks are executed. In order to provide a definition of a workflow, the following needs to be specified: control flow definition, resource classification and resource management rules (van der Aalst 2004).

Conventional PAISs are not always based on pure WFMS; rather have workflow related elements. (van der Aalst 2004) identifies four categories of workflow support to information systems as (1) pure WFMS; (2) WFM components embedded in other systems; (3) custom-made WFM solutions which are built for organisation or industry segments and (4) hard-coded WFM solutions where the processes are hard-coded in the applications. The concepts and insights provided by the first two categories can be applied to systems that are developed in the third and fourth category. The work presented in this thesis lies squarely in the second category as we seek to embed WFM with MDC to provide generic workflow support to the process of data collection.

3.1.4 WORKFLOW REFERENCE MODEL

Many of the foundational concepts in BPM are based on common standards propagated by the Workflow Management Coalition (WFMC) which is a global organisation of adopters, developers, consultants, analysts, as well as university and research groups engaged in workflow and BPM (Hollingsworth 1995). The WFMC developed a workflow reference model (Hollingsworth 1995), which specifies a framework for workflow systems, identifying their characteristics, functions and interfaces. It has acted as a development standard for workflow systems. Figure 2 illustrates the reference model, which consists of five interfaces; process definition tools, workflow client applications, invoked applications, workflow interoperability and administration and monitoring.

The process definition tools interface provides a definition of a standard interface between process definition and modelling tools and the work flow engine(s). The workflow client applications interface provides for definition of interfaces for applications that request services from the workflow engine to control the progression of processes, activities and work-items. The invoked applications interface provides for definition of APIs to allow the workflow engine to invoke a variety of applications, through common agent software. The workflow interoperability interface is used for definition of workflow interoperability models and the corresponding standards to support interworking. Finally, the administration and monitoring tools interface provides for the definition of monitoring and control functions.

Many workflow systems have been built based on this model. The YAWL system used in this research is inspired by this model (van der Aalst, Adams et al. 2010) and therefore provides a good basis for this thesis. The artefacts proposed in this thesis utilise this architecture by providing and defining client applications that are capable of being used in the mobile environment. Additionally, interface 4 (other workflow engines) is used to define the distributed system for offline workflow execution.

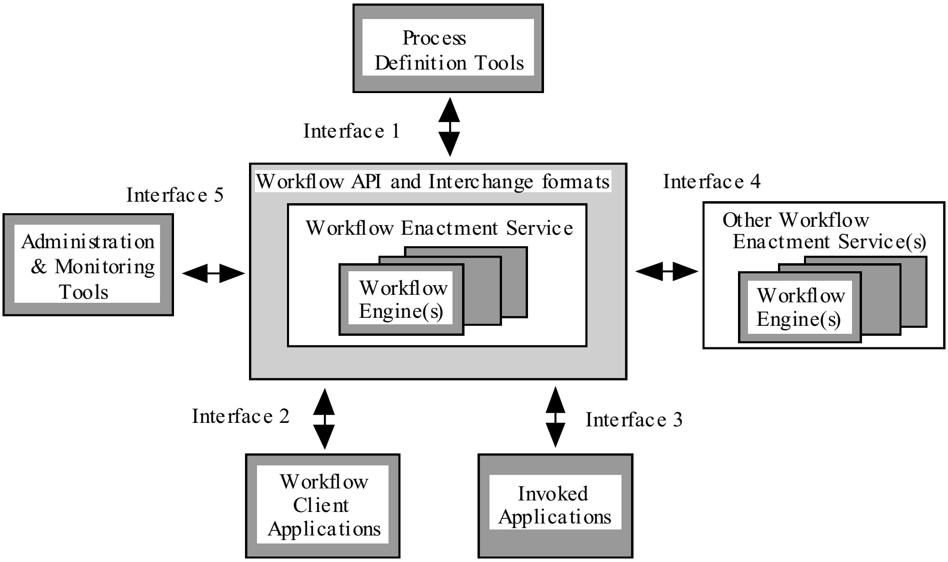


Figure 2: Workflow reference model (Hollingsworth 1995)

3.1.5 PETRI NETS

Petri nets are a graphical and mathematical modelling tool that can be used for describing and studying process related systems (Murata 1988). Petri nets contain three elements: places, transitions and arcs. Places correspond to the state of a discrete transition system; transition represents the events that transfer to another state while arcs provide the flow of the system. Using Petri nets it is possible to provide a graphical and mathematical description of workflows. The rationale of using Petri nets is to provide a way of representing a workflow process that may be defined in any notation / execution language. This provides a good basis for formal semantics for workflows and allows for mathematical analysis.

To clearly understand the application of Petri net theory in representing processes, we look at the formal definitions of Petri nets and workflow nets. Definitions 1,2 and 3 are obtained from (van der Aalst 2004).

Definition 1: *A classical Petri net is a four-tuple (P,T,I,O) where:*

P is a finite set of places,

T is a finite set of transitions such that $P \cap T = \emptyset$*,*

I: $P \times T \rightarrow \mathbf{N}$ *is the input function, and*

O: $T \times P \rightarrow \mathbf{N}$ *is the output function.*

Definition 2: *The state (marking) of a Petri net (P,T,I,O) is defined as follows:*

s: P-> ***N****, i.e., a function mapping the set of places onto {0,1,2, … }.*

In order to define the control-flow dimension of a process, building blocks such as the AND-split, AND-join, OR-split, and OR-join are used to model sequential, conditional, parallel and iterative routing (Fischer 2002). Petri nets are used to specify the routing of cases, whereby tasks are modelled by transitions and causal dependencies are modelled by places and arcs. A Petri net which models the control-flow dimension of a workflow is called a workflow net.

Definition 3: *A workflow net is a Petri net PN = (P, T, F) if and only if:*

a. *There is one source place* $i \in P$ *such that* $\bullet i = \emptyset$*.*
b. *There is one sink place* $o \in P$ *such that* $o \bullet = \emptyset$*.*
c. *Every node* $x \in P \cup T$ *is on a path from i to o.*

The control flow perspective of BPM is represented as a workflow net. In designing and analysing workflow nets, it is important to assume the soundness properties. These properties are vital for the accurate representation of a business process and ensure that for each case, it is always possible to terminate correctly, there is no deadlock and it is always possible to have a finite number of occurrences. The soundness properties of a workflow net are therefore relevant for properly defining a workflow process. Such properties are:

- For each token put in the place *start, one (and only one) token eventually* appears in the place *end;*
- When the token appears in the place *end, all the other places are* empty; and
- For each transition (task), it is possible to move from the initial state to a state in which that transition is enabled.

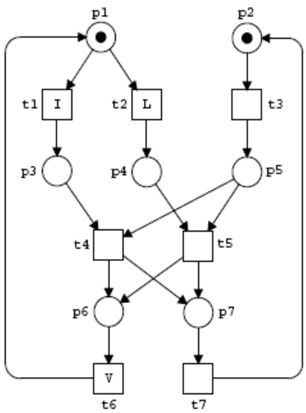
3.2 DISTRIBUTED WORKFLOW SYSTEMS

3.2.1 DISTRIBUTED WORKFLOW APPROACHES

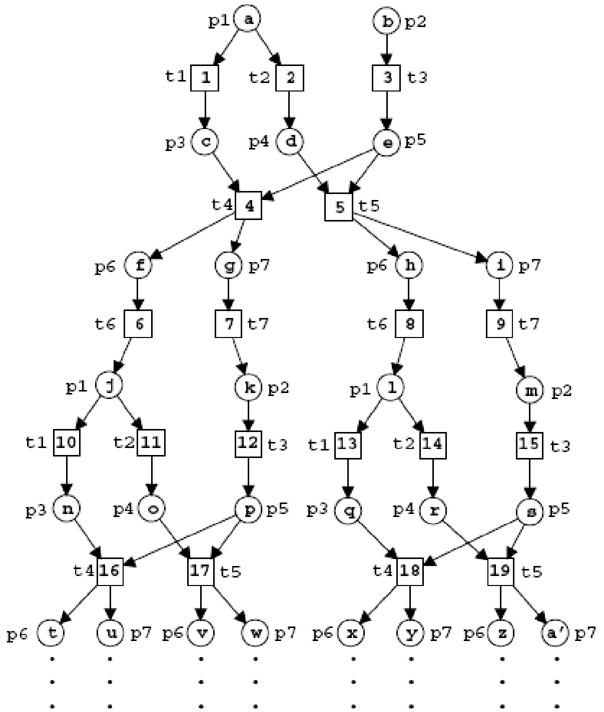
From literature, it can be observed that a common approach to distributing workflow execution is through partitioning (Muth, Wodtke et al. 1998; Baresi, Maurino et al. 2005). Crucial however is the purpose for distribution which determines the partitioning approach taken. For web-based systems, the main reason for partitioning is twofold; to enable parallel processing of complex workflow systems or to enable decentralised execution of geographical diverse systems. This contrasts with the reason for distribution in a mobile environment which has to do with coping with mobility of the devices used in disconnected environments and limited computing resources which calls for a different approach. Partitioned workflows form a distributed architecture that need synchronisation to ensure that the overall process executed as per the predefined procedures. In this research, we focus on approaches for partitioning in disconnected environments and synchronisation in a distributed environment.

3.2.2 PARTITIONING FOR DISTRIBUTION

A number of methods for partitioning workflows for distributed execution have been proposed (Guth, Lenz et al. 1998; Tan and Fan 2007). The main thrust for partitioning workflows is to enable distributed execution. In creating a partition, in particular, Petri nets and state chart specifications are amenable to model checking (Burch, Clarke et al. 1994; Esparza and Heljanko 2001), so that critical workflow properties that are expressible in temporal logic can be formally verified. Esparza and Heljanko (2001) propose model checking using a partial order technique for the verification of concurrent and distributed systems referred to as model unfolding, initially introduced by McMillan (Esparza, Römer et al. 1996). The unfolding technique is applied to systems modelled by Petri nets, communicating automata, or process algebras to enable formal reasoning about specifications in a distributed workflow environment. This unfolding technique is explained below:



(a)



(b)

Figure 3: Unfolding of a Petri net (adopted from (Esparza and Heljanko 2001))

Given a Petri net (P) shown in Figure 3(a), an unfolding shown in Figure 3(b) can be obtained. An unfolding (N) of a transition system is a tree where the transitions are events and the places (i) are conditions. Intuitively, an unfolding of a product can be seen as a synchronisation of trees. It can be observed that in an unfolding, the following properties hold:

1) N has no cycles, i.e., no (nonempty) path of arcs leads from a node to itself.
2) For every reachable marking of N puts a token in exactly one i-place.

3) The set of i-nodes of the unfolding N forms a tree with the i-root as root. Moreover, the tree only branches at places, i.e., if a node of the tree has more than one child, then it is a place.
4) A place of N can get marked at most once (i.e., if along an occurrence sequence it becomes marked and then unmarked, then it never becomes marked again), and an event of N can occur at most once in an occurrence sequence.

Based on these propositions, the following properties of unfolded nets hold and can be used to analyse the consistency. Esparza and Heljanko (2001) use model unfolding to analyse the properties of *causality, conflict,* and *concurrency* of a Petri net.

Two tasks of the unfolding of a Petri net are *causal* if they are connected by a path of net arcs. For instance, tasks T_2 and T_5 in figure 2 (b) are connected by a path, while tasks T_2 and T_4 are not. In the first case, the task at the end of the path can only be enabled after the one at the beginning of the path has completed; we say that the events are causally related. In the second case, no occurrence sequence of the unfolding contains both events, and so we say that the events are in *conflict*. The task T_4 in figure 2 (b) is certainly not a cause of T_5 and vice versa. Moreover, they are not in conflict, since they are in the same sequence of an occurrence sequence of the unfolding. We refer to such a situation as concurrency.

3.2.3 SYNCHRONISING DISTRIBUTED SYSTEMS

A distributed workflow execution requires synchronisation between the underlying workflow engines. Synchronisation is necessary to create a common view of the workflow process. In wide area networks, the latency for establishing a connection is a major performance factor whereas in mobile-based systems, frequent network disconnections and cost of data transmission are the major factor. Therefore, the goal is not only to reduce the overall message size, but also to reduce the number of synchronisation messages required for further execution of a workflow process.

Many approaches to synchronisation of distributed workflow systems have been proposed to address different challenges. The kind of approach proposed depends largely on the problem to be addressed. For instance, (Muth, Wodtke et al. 1998) propose an approach to enable workflow support for large-scale distributed, enterprise wide applications. Their approach is based on the distributed execution of state and activity charts. They used formal semantics of state and activity charts, to develop an algorithm for transforming a centralised state and activity chart into a provably equivalent partitioned one, suitable for distributed execution. Furthermore, they propose a synchronisation to ensure that the original execution sequence of a non-distributed system is maintained.

Schuster, Jablonski et al. (1994) focus on developing architecture for distributed systems that execute tasks using parallelism. The use of parallelism is due to performance requirements and involves data and applications that are spread across a heterogeneous, distributed computing environment. Their architecture for WFMS aims to provide scalability through transparent parallelism and heterogeneity of various systems involved.

3.3 Mobile-based Process Aware Information Systems

Advances in mobile computing capabilities have enabled the development of light weight mobile workflow process execution engines. Mobile devices can now provide a complete platform to run end user driven work processes (Hackmann, Haitjema et al. 2006). In order to adapt mobile devices to support workflows, it requires a general extension of existing workflow systems by adding mobile device as one of the environments where activities can be executed. Most mobile solutions consist of three parts; the database clients, servers – which are usually, connected to corporate Information Systems and the synchronisation software.

There are many vendors who have developed a wide range of mobile related solutions using some workflow technology. An example is SILVER which is based on business process execution language (BPEL) (Hackmann, Haitjema et al. 2006). It is a demonstration of how groupware applications are implemented on mobile devices. Another example is the MARPLE system (Pryss, Tiedeken et al. 2011) which integrates process management technology with mobile computing frameworks in order to enable mobile process support in mobile environments. The MARPLE architecture consists of a light-weight process engine that runs on the mobile device and that is able to interact with backend processes if required.

There have been attempts to develop a framework that delivers workflow definitions to mobile devices in disconnected environments (Bahrami, Wang et al. 2006). The IBM FlowMark (Alonso, Gunthor et al. 1996) is an example of a meta-model that seeks to address the constraints of deploying mobile workflows. Work is loaded to a mobile with the hope that a user is committed to do it. The activities are considered locked and can only be unlocked and reassigned once the user has synchronised with the workflow server.

Another approach proposed by IBM is to develop a distributed (across multiple devices) workflow system which spans between devices and backend infrastructure. Exotica is an example of a distributed workflow platform where processes are transferred to sites thereby eliminating the need of a centralised server (Alonso, Gunthor et al. 1996). To achieve this, a copy of the process definition is maintained at each of the sites or to pre-compile the solution and determine which site executes a particular process. Muth, Wodtke *et al*. (1998) also propose a general approach to synchronise distributed workflows based on state and activity charts to give a status of a workflow process.

3.4 CONTRAST WITH THIS RESEARCH

Whereas many solutions have been provided for mobile data collection, key issues addressed in this research are not adequately tackled in related literature. The primary goal of this thesis is to provide an approach to enable mobile data collection with process support within resource constrained environments. This requires a platform which extends the conventional workflow systems to the mobile computing environment, with special emphasis on MDC. Such solutions should not require high-end mobile computing devices and readily available mobile networks to work. This research therefore seeks to provide solutions for low end mobile devices commonly referred to as *feature phones*.

Indeed mobile workflows and distributed systems have been an area of great interest to researchers. What needs to be done is to provide a rich application domain, in order to refine theory and make a strong case for such workflow based systems for mobile task automation and field work executions. Specifically a deployment of mobile workflows for data collection in developing countries would require research to develop a light weight workflow engine for mobile phones which can enable workflow execution in the constrained environment discussed in section 1.1. This thesis proposes a framework for mobile workflows comprising a complete workflow engine executing within the workflow client and a backend system to deploy and manage the lifecycle of the workflows.

4. RESEARCH METHODS

In this chapter we provide an explanation of the methods used in this research and illustrate how these techniques were applied. The design science technique which focuses on the development and performance of (designed) artefacts with the explicit intention of improving the functional performance of the artefacts provides the umbrella for this research. Within this approach, technical action research was used at the implementation to validate the artefacts by implementing them in real world projects. Finally, a case study based on a clinical trial was used to obtain requirements for process related implementation of mobile data collection in the place of paper based routines.

4.1 RESEARCH TECHNIQUES USED

4.1.1 DESIGN SCIENCE RESEARCH METHOD

Design science research is technical and engineering science that validates proposed artefacts through rigorous study. The design science methodology involves identifying a practical problem and proposing a technical artefact. A validation of this artefact involves answering the question, "does this work?" In order to answer this, various validation research options are explored. The methods used in this thesis include field experiments and action research. Using the method, researchers then reflect on the problem and the proposed artefact to see if there is a different understanding of the problem or improvements in the artefact.

The design science research process is conducted in three cycles i.e. the *relevance cycle, design cycle* and *rigor cycle* as illustrated in Figure 4 (Hevner, March et al. 2004). The *relevance cycle* is the first stage and its main purpose is to bridge the contextual environment of the research project with the design science activities. This is followed by the *design cycle* which iterates between the core activities of building and evaluating the design artefacts and processes of the research. The final cycle is the *rigor cycle*, which connects the design science activities with the knowledge base of scientific foundations, experience, and expertise that informs the research project. They contend that the three cycles must be present and clearly identifiable in a design science research project.

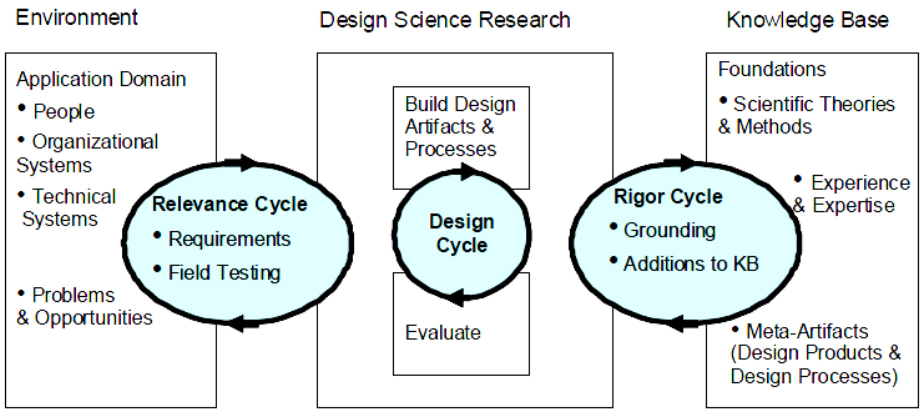


Figure 4: Design science research process (Hevner, March et al. 2004)

4.1.2 Technical Action Research

Technical Action Research (TAR) aims to make research results more relevant, by adding a problem-solving cycle to the design cycle. TAR starts with artifact design and then looks for organisational problems that could be solved by this artifact (Wieringa and Moralı 2012). The goal of the researcher is to develop this artifact for use in a class of situations imagined by the researcher. The artifact is usually first tested on toy problems under idealized circumstances in a laboratory and then it is scaled up to conditions of practice by solving more realistic problems with it, until it can be tested by using it in one or more concrete client organizations to solve real-world problems.

TAR was used in this research at the artifact validation stage. Using these techniques, an artifact is designed for a client in appropriate conditions of practice that fit the problem domain. By using a case study, TAR was used in an engineering cycle that aimed at solving a problem and transferring the lessons learned to a class of similar problems.

4.1.3 Case Studies

Case studies are used to organise a wide range of information about a case and then analyse the contents by seeking patterns and themes in the data and by further analysis through cross comparison with other cases (Tellis 1997). Sources of evidence in case-based research are documentation, archival records, interviews, direct observation, and physical artefacts (Yin 1984).

Cavaye (1996) recommends this approach for Information Systems research because "*case study research can be used for testing or building theory, with a single or multiple case study design, using qualitative or mixed methods*". Benbasat, Goldstein *et al.* (1987), provide some suggestions about how to conduct and evaluate case study research. They argue that it is particularly well-suited to IS research, because when technology is relatively new, the interest is in demonstrating its applicability within a particular domain rather than technical issues.

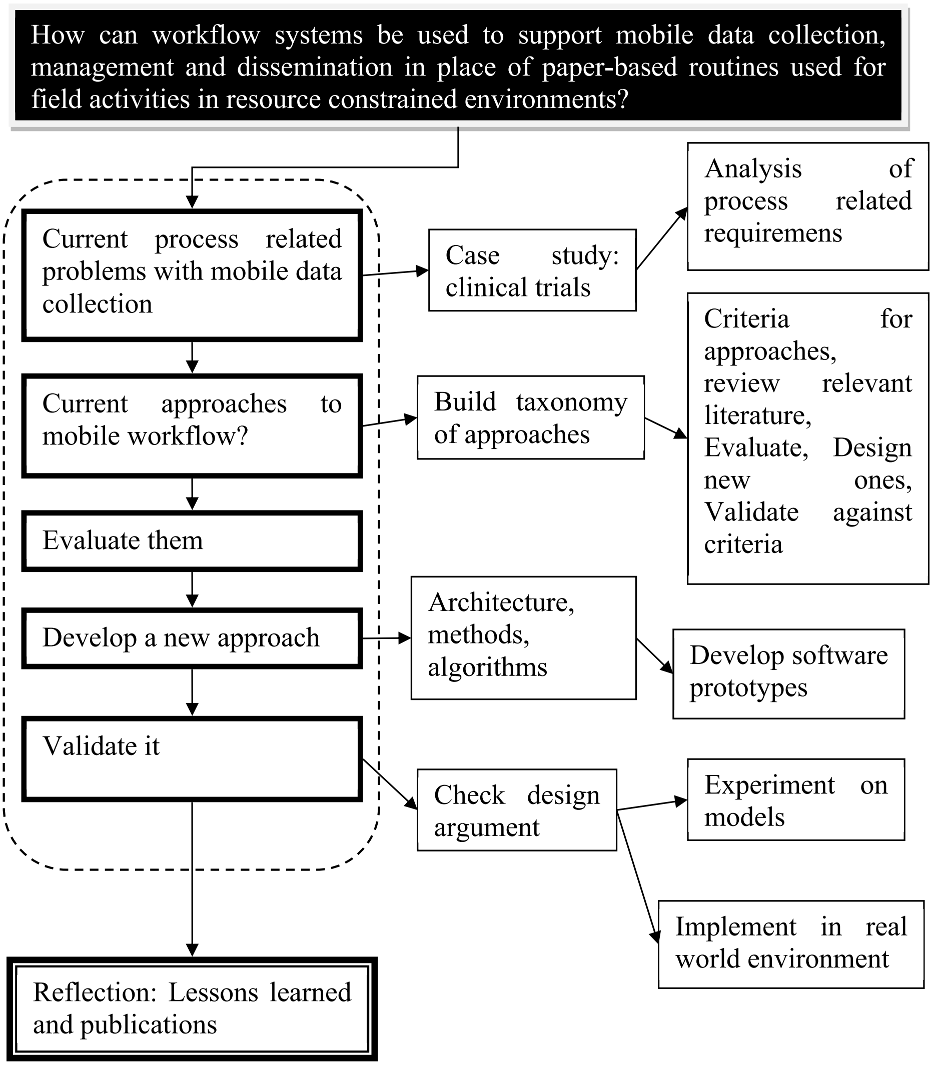


Figure 5: Research process used in this thesis

4.2 HOW THESE METHODS WERE APPLIED

Figure 5 above illustrates the process that was used in conducting the research. The problem to be addressed is clearly defined in section 2. In order to contextualise the problem, a case study was undertaken to understand the process-related requirements for using mobile workflows. Literature about the current approaches was reviewed to build taxonomy of current approaches. These were evaluated and their inadequacy in solving the problem was identified. New approaches were then designed and evaluated against the requirements. These new approaches included architecture, methods, algorithms that provide generalisable and scientific methods to solving the problem at hand. These were validated using experiments and real world problems, the results of which have been published in various papers.

4.2.1 THE RELEVANCE CYCLE

In order to clearly understand the problem and define the factors and constraints, we used a case study based on clinical trials. We analysed the PROMISE-PEP (ClinicalTrials.gov 2008) study protocol in order to determine the process related requirements that should be supported by electronic data collection tools. In addition, we studied FDA and EU guidelines on data collection with specific reference to guidelines to industry in the use of computer systems for data collection (FDA 2007). Furthermore a study of the paper-based data collection routines for field activities was conducted to analyse the process related features that need to be supported by mobile phones if they are to replace paper.

4.2.2 THE DESIGN CYCLE

From a scientific point of view, the challenge of having workflows running on mobile devices and disconnected environments have been addressed. The research sought to find out existing scientific approaches and how they could be used to solve the problem. A taxonomy of these approaches was developed and is discussed in later chapters of this thesis. In order to address the gap in current approaches, we devised an evaluation criteria based on requirements for ubiquitous mobile data collection. A review of relevant literature was conducted to determine the extent to which the requirements have been addressed. Based on these evaluations, new approaches were developed. The first approach addressed the issue of integrating WFMS into generic data mobile collection tools. The second approach addressed the issue of enabling offline execution of two or more tasks through a distributed WFMS approach. Each of these were validated by implementing them within the OpenXdata platform (openXdata).

4.2.3 THE RIGOR CYCLE

The system was then deployed in typical field environments where we conducted evaluations and reflected on the outcomes. Through the first iteration, we developed a tool that enables generic data collection with workflow support. We implemented the architecture to enable the OpenXdata collection tool to work with the YAWL WFMS. The system that was developed enabled users to flexibly define workflows and forms for data collection, match them and deploy them on a mobile device. This approach was validated by the deployment in the M4W project (M4W 2012).

4.3 CASE STUDIES

More than any other field research, studies like clinical trials, demographic surveillance studies, and many more, provide the best platform to test the artefact produced. Case studies were used to evaluate artefacts and draw lessons for requirements for mobile based workflow systems.

4.3.1 REAL WORLD PROJECT - M4W

The M4W project aims to improve water and sanitation in rural parts of Uganda by enabling access to information by various stakeholders and players in the water and sanitation sector. Using mobile phones, baseline and inspection information is collected about each water point based on the Uganda MWE guidelines and standards. The project is a PAIS that uses the OpenXdata and YAWL to provide generic workflow management for MDC.

The M4W project uses SMS messages and OpenXdata application on mobile phones (approx. $50 each) and web interfaces to collect information. Once collected, the data is uploaded to a database that is hosted on the Internet. The data collection processes mirrors existing local government procedures where data is collected by the Hand Pump Mechanics (HPM) and approved by the extension worker at the sub-county before it is stored as a record. This functionality is implemented through a workflow that delivers work assigned to a HPM and upon completion, collected data is sent to the extension worker who verifies the information before it is archived in the database. The workflow system enables additional services such as SMS, to support the data collection processes like sending an SMS message to notify users of new work items.

The system also has an automated process for reporting faults by community members and management of the response process, involving data collection on assessment, repairs and messages to appropriate stakeholders. These were accomplished by the use of PAISs through MDC tool using OpenXdata and YAWL. A member of the community or any member of the public can report a fault by sending a coded SMS with the unique identifier of the water source and problem to a short code (8888). The system creates a ticket and notifies the HPM and District Water Officer (DWO) for action. HPMs are able to report about the assessments and repairs undertaken by downloading work items to their phones.

Through the use of this generic system, two workflow process are implemented; for collection of baseline data and managing the fault handling process. The use of a generic approach enabled easier and faster definition of a complete application to accomplish the identified tasks. The artefacts used on this project are related to the workflow architecture for MDC described in section 6.4. One limitation identified was the need to make a connection before the next work item was enabled. This formed the basis for the next iteration of the research process.

4.3.2 Experiments - Petri-net Based Simulation

Experiments were used to test ideas for partitioning workflow systems for offline execution. As explained in section 3, models are used to define workflow processes. Therefore partitioning a workflow process requires studying the static and dynamic behaviour of the model and identifying rules for partitioning. Experiments were used to determine the adherence to the pre-defined rules using process models that represent a wide range of scenarios.

This thesis proposes an approach to enabling offline execution by partitioning a workflow process for distributed execution. The method for partitioning needs to meet a set of requirements as provided in section 5.2. An algorithm to enable automatic partitioning based on the method was devised as explained in section 5.3. Experiments using a tool based on petri nets was developed using the PIPE 2 platform (Bonet, Lladó et al. 2007).

These experiments were conducted on a number of process models, to assess the correctness of the proposed approach in meeting the requirements for partitioning. In order to ensure that the experiments were appropriate, several workflow patterns (van der Aalst, ter Hofstede et al. 2003) were considered. Each of these partitions was evaluated for correctness and completeness by determining the number of partitions omitted.

4.3.3 Field Evaluations – Iganga DSS

In order to evaluate the offline data collection tool, a demographic surveillance scenario was used. The Iganga-Mayuge Demographic Surveillance Site (DSS) was established in 2005. It collects longitudinal data about households in Iganga and Mayuge districts in Uganda using paper-based questionnaires related to pregnancy and birth outcomes. These questionnaires are administered to all household members in the catchment area every four months. A team of enumerators conduct a survey in all the villages and record information about all females in the household. During the subsequent visits, females between the age of 12 and 49 are asked for pregnancy information which is completed on data entry forms. If the woman is pregnant, a pregnancy questionnaire is completed. During the subsequent follow up (after about 6-8 months), the process is repeated. In addition, women who answered that they are pregnant are asked about the status of pregnancy and child. If she is no longer pregnant or has no child, she is asked additional information about what happened to the pregnancy. Other information collected on the households includes death registration, in and out migration and change of residence.

The use of workflows in this project was meant to guide enumerators (whose literacy and competence levels are low) on what is required to be completed depending on the responses given. Mobile network within the villages is patchy and therefore enumerators operate offline and upload work upon reaching areas with network connection. The offline workflow tool was appropriate in this project in that it improved the quality of data by ensuring adherence to the guidelines and enabled linking of forms, thus making it easier to administer. The project provided a good evaluation of this research based on the fact that it represents a typical example of cumulative data collection as explained in section 2.3. The field evaluation was used to test the artefacts explained in sections 6.5, 6.6, and 6.7.

4.3.4 FIELD EVALUATIONS - PROMISE-PEP PROJECT

In this case study, we use the example from the PROMISE-PEP study (ClinicalTrials.gov 2008). It is a clinical trial that compares the efficacy and safety of prolonged infant peri-exposure prophylaxis (PEP) with Lopinavir/Ritonavir (LPV/r) versus Lamivudine to prevent HIV-1. It aims at making the comparison of transmission through breast milk in children born to HIV-1-infected mothers not eligible for Highly Active Antiretroviral Therapy (HAART) and having benefited from prenatal antiretroviral (ART) regimens. Being a clinical trial, there is a pre-defined process and data variables that need to be collected about mothers and infants. The aim of using workflows is to automate the process of data collection and ensure adherence to the pre-defined procedures. The study has about 1,500 mother-infant pairs operating through field clinics in four African countries. We base our example on a field clinic set up in a hospital in Uganda.

The PROMISE-PEP study uses the Openclinica information system (Collins 2007) which has a workflow embedded for easy customisation of the data collection process. Our solution therefore sought to extend this process to enable data collection on mobile, which required workflows to be used. The workflow MDC tool developed for this case enables users to download work items and form definitions before embarking on a data collection process. For each case, the relevant information about mother and child are saved on the device. Once the work has been loaded on the mobile phone, it can be executed and saved on the device. On completion of the data collection effort, users can upload the data to the server.

5. REQUIREMENTS

In order to answer the questions identified in section 2.6 we outline the requirements related to workflows and mobile computing environment. In section 5.1 we discuss the requirements for enabling workflow support for MDC tools. Section 5.2 examines the requirements for flexible workflow support to enable the ubiquity that paper-based routines give for field-based activities. Finally in section5.3, we discuss the requirements for developing distributed workflow systems to support offline mobile data collection.

5.1 ENABLING WORKFLOW SUPPORT FOR MDC TOOLS

In order to enable generic workflow support for MDC tools, two issues need to be addressed. First, WFMS should be modified to support MDC, especially since the mobile client may not always be connected to the server. Second, a mechanism to map the data elements is necessary to enable data flow between the workflow engine and MDC tool.

A workflow solution for MDC should take into consideration the characteristics and limitations outlined in section 2.5 by employing new computing paradigms. MDC is not always the preferred choice for electronic data collections due it limitations already identified. Therefore any solution that uses mobile phones should not seek to replace existing systems but rather provide additional platform through which information can be obtained. In addition, the Workflow-based systems should support the collaboration among various parties involved in order to ensure completion of set tasks.

MDC tools and WFMS follow generic software architectural standards that have bearing on their integration. They however have in common data elements related to data types and visibility that need to be matched. Russell, ter Hofstede et al. (2005) conducted a study a detailed study of a number of workflow tools and business process modelling paradigms discovered that the way in which data is structured and utilized within these tools have a number of common characteristics. These were used to developed workflow data patterns, which aim to capture the various ways in which data is represented and utilised in workflows as explained in section 3.1.2. Therefore, one of the key requirements for using generic WFMS in data collection is providing for how to map the data elements of a workflow specification with MDC forms. Such a mapping should provide appropriate methods for handling data variables shared between WFMS and MDC tools.

The artefacts developed to meet these requirements are described in section 6.4 and 6.7. Details can also be obtained from paper 1 in part II of this thesis.

5.2 DISTRIBUTED WORKFLOW SYSTEMS IN DISCONNECTED ENVIRONMENTS

There are several motivations for developing distributed workflow systems in literature as explained below (Muth, Wodtke et al. 1998):

1. **Scale:** Large scale workflow applications which involve several business units, create a very large number of concurrent workflow instances, and thus impose a high load on the workflow engine. Such applications may involve a very large number of concurrent workflow instances which impose a high load on the workflow engine. Consequently, scalability and availability considerations dictate that the overall workflow processing must be distributed across multiple workflow engines that run on different servers, and this workload partitioning may itself require the partitioning of individual workflows.

2. **Geographical distribution:** Whenever a workflow spans multiple business units that operate in a largely autonomous manner, it may be required that those parts of a workflow that are under the responsibility of a certain unit are managed on a

server of that unit. Thus, the partitioning and distribution of a workflow may fall out naturally from the organisational decentralisation.

3. **Disconnected systems:** Workflow systems that operate in disconnected environments often require work to be undertaken on the client device even when the server of offline. Partitioning may be necessary to enable more than one task to be assigned to the client. Thus, it is necessary that such clients have workflow engines capable to controlling processes based on workflow partitions that are provided with work items.

A common approach to distributing workflow execution is through partitioning. Crucial however, is the purpose for distribution which determines the partitioning approach taken. For web-based systems, the main reason for partitioning is twofold; to enable parallel processing of complex workflow systems or to enable decentralised execution of geographical diverse systems. This contrasts with the reason for distribution in a mobile environment, therefore requiring a new approach. The aim for partitioning in such cases is to cope with mobility of the devices used in disconnected environments and limited computing resources.

The provision of having mobile clients to execute portions of a workflow specification, independent of the main server creates a distributed workflow scenario. This raises a further requirement for synchronisation of the various fragments of the workflow process in order to create a uniform process coordinated by the main workflow engine. Moreover, to the main workflow engine, it should always be as if only one process is running so as to maintain the required control.

5.3 FLEXIBLE WORKFLOW APPROACH FOR DATA COLLECTION

Partitioned workflows form a distributed architecture that provides the flexibility required to use mobile devices in disconnected environments. If these partitions are to be executed by a mobile client, it requires the workflow client to have an engine that is capable of reading the process model and guiding the execution of subsequent tasks. Moreover, taking into consideration the limitations to mobile devices provided in section 2.5, such an engine should use the least possible resources (memory and processing speed), to provide the required solution. By a large extent, this is determined by the complexity of the partition and therefore partitions should not be overly complex. Keeping the objective for partitioning in mind, the solution proposed seeks to enable simple and commonly occuring control flow patterns provided in section 3.1.2 to be executed on the mobile client.

A number of methods for partitioning workflows for distributed execution have been proposed (Guth, Lenz et al. 1998; Tan and Fan 2007). Many of the approaches while addressing the need for distributed offline execution of work do not cater for the need to provide an option that enables the server to maintain control. Centralised control requires that the underlying logic of the original model maintained on the server be adhered to, by all client devices executing the partitions. This enables distributed execution to be an option and not a requirement.

The following requirements are necessary for partitioning process models in order to enable execution on mobile devices in offline mode:

1. A partition should not alter the underlying logic of a process model. Any combination of tasks should preserve the order of execution as defined by the process modeller.

2. All tasks from the partition can be assigned to one resource and can be executed by the service(s) on the client device.

3. All data dependencies for the tasks to be executed are contained in a partition and hence there is no need to connect to the server during the lifetime of its execution.

4. The user can define the optimum number of tasks to be assigned or select from a set of possible partitions.

Whereas a key motivator for partitioning in this thesis is to enable offline and geographical distribution for teams located in the field, consideration needs to be taken, such that the resulting partitions are simple enough to be executed by a light workflow engine residing on the mobile device.

6. OVERVIEW OF DEVELOPED ARTEFACTS

In this chapter, we give an overview of the artefacts that are proposed and developed in this thesis. We begin by presenting the overall picture where we explain how the artefacts are related and how they combine into a larger framework for mobile data collection. Subsequently, each of the artefacts is presented separately in more detail. We also give references to the chapters in Part II for further details.

6.1 CONTEXT OF ARTEFACTS

In our approach, the workflow engine and MDC tools are treated as external entities that are generic and based on commonly agreed standards. The work that was done aimed to create artefacts that would enable these tools to be integrated into the mobile computing environment. Two modes of mobile clients were considered; online and offline. In the case of online mode, it is assumed that the client is in a networked environment and is able to access work when required. In the offline mode, the client operates in an area without network and is unable to connect during execution of tasks. The former mode requires an adapter that enables forms defined by MDC tools to be matched with tasks in workflow specifications in order to enable data collection. The latter mode requires additional functionality for workflow partitions to be assigned along with the work items to enable execution of more than one task before connection is required.

Figure 6 provides the general description of how work items are handled in a mobile data collection environment. The workflow engine generates a work item that is matched with the relevant form, based on pre-designed form/workflow specification maps stored in the system. The work item and its form reference are then assigned to a mobile "agent" that receives work on behalf of a mobile user, whose profile is pre-configured. When the user requests for download of work items, a work list is provided together with the form references for forms already configured on the mobile device. On completion of work, the data is uploaded to the server where the relevant data is extracted and sent back to the workflow engine for the next task to execute. In cases where the mobile client is unable to make a connection before advancing to the next task, a workflow partition may be provided, with the work item that is enabled as the first task. After the first item is executed, another work item may be generated depending on the workflow definition in the partition. This provides a distributed workflow which requires synchronisation upon upload of data.

The artefacts implemented relate to the processes mentioned above. These include a method for mapping of forms used in MDC to workflow specification from WFMS. In addition, a workflow adapter to enable matching of work items to forms and assigning to mobile agents. We then present a method for partitioning processes based on prescribed process models for offline execution. This is followed by description of architecture to enable offline execution of work items. Finally we present a platform that implements and tests all these approaches.

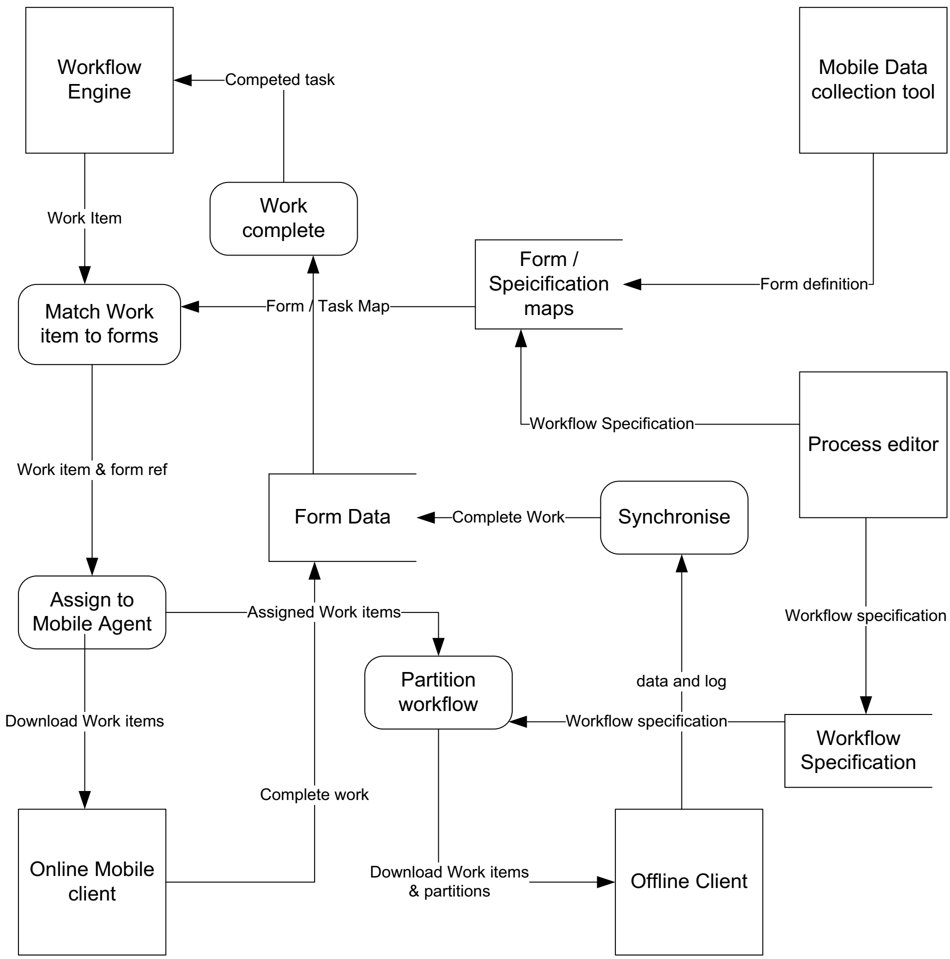


Figure 6: Data flow between implemented artefacts

6.2 PARTITIONING WORKFLOWS FOR OFFLINE EXECUTION

The work presented in this section is published in paper 2 of part II of this thesis. The requirements for partitioning were provided in section 5.2. Figure 7 provides an illustration of the partitioning approach proposed in this thesis. Figure 7(a) shows a process model containing choice, parallelism and sequential constructs common in many process models. This model is unfolded using techniques proposed by (Esparza and Heljanko 2001) in order to determine structural and behavioural aspects to enable the discovery of partitions that meet the requirements. Esparza and Heljanko (2001) use model unfolding to analyse the properties of *causality, conflict,* and *concurrency* of a Petri net.

Two tasks of the unfolding of a Petri net are *causal* if they are connected by a path of net arcs. For instance, tasks T_2 and T_4 in Figure 7 (b) are connected by a path, while tasks T_2 and T_3 are not. In the first case, the task at the end of the path can only be enabled after the one at the beginning of the path has completed; we say that the events are causally related. In the second case, no occurrence sequence of the unfolding contains both events, and so we say that the events are in *conflict*. The task T4 in Figure 7(b) is certainly not a cause of T_6 and vice versa. Moreover, they are not in conflict, since they are in the same sequence of an occurrence sequence of the unfolding. We refer to such a situation as concurrency.

Model unfolding represents the full reachability graph using partial orders that preserve the relations between transition occurrences (Esparza and Heljanko 2001). All reachable markings are represented in a petri net unfolding thus enabling us to determine the dependences and conflict between states. Figure 7 (c,d,e & f) provides the four possible partitions of this model. It can be observed that Figure 7(f) is an extension of Figure 7(e) by adding task T_8. This sort of extension could be increased for any subsequent tasks in the process model. Therefore it is important to specify the maximum number of tasks that could be included to a partition.

Definition 4: Given an unfolding of a workflow net $W = \langle \mathrm{S};\ \mathrm{T};\ \alpha; \beta, \mathrm{is} \rangle$, such that $\mathrm{S} = \{\ \mathrm{s}_0, \mathrm{s}_1, \ldots\ldots, \mathrm{s}_\mathrm{m}\}$ is a set of states and $\mathrm{T} = \{\mathrm{t}_0, \mathrm{t}_1, \ldots\ldots, \mathrm{t}_\mathrm{n}\}$ is a set of transitions; the relation α: T $\rightarrow$ S associates to each transition its source state; β: T $\rightarrow$ S associates to each transition its target state. A partition $\mathrm{W}' = \langle \mathrm{S}', \mathrm{T}'\alpha, \beta, \mathrm{is} \rangle$ where $T' = \{t_i, t_{i+1}, \ldots, t_{i+m}\} \subset T$ and $S' = \{s_i, s_{i+1}, \ldots, s_{i+n}\} \subset S$ is possible if:

Rule 1: $\forall W' \exists \{\alpha(t_i) = is, \beta(t_i) = s_i, \ldots \beta(t_n) = s_{i+n}\}$ (causal and connected)

Rule 2: $\forall W' \neq \{W \notin W'\}$ (No contradiction)

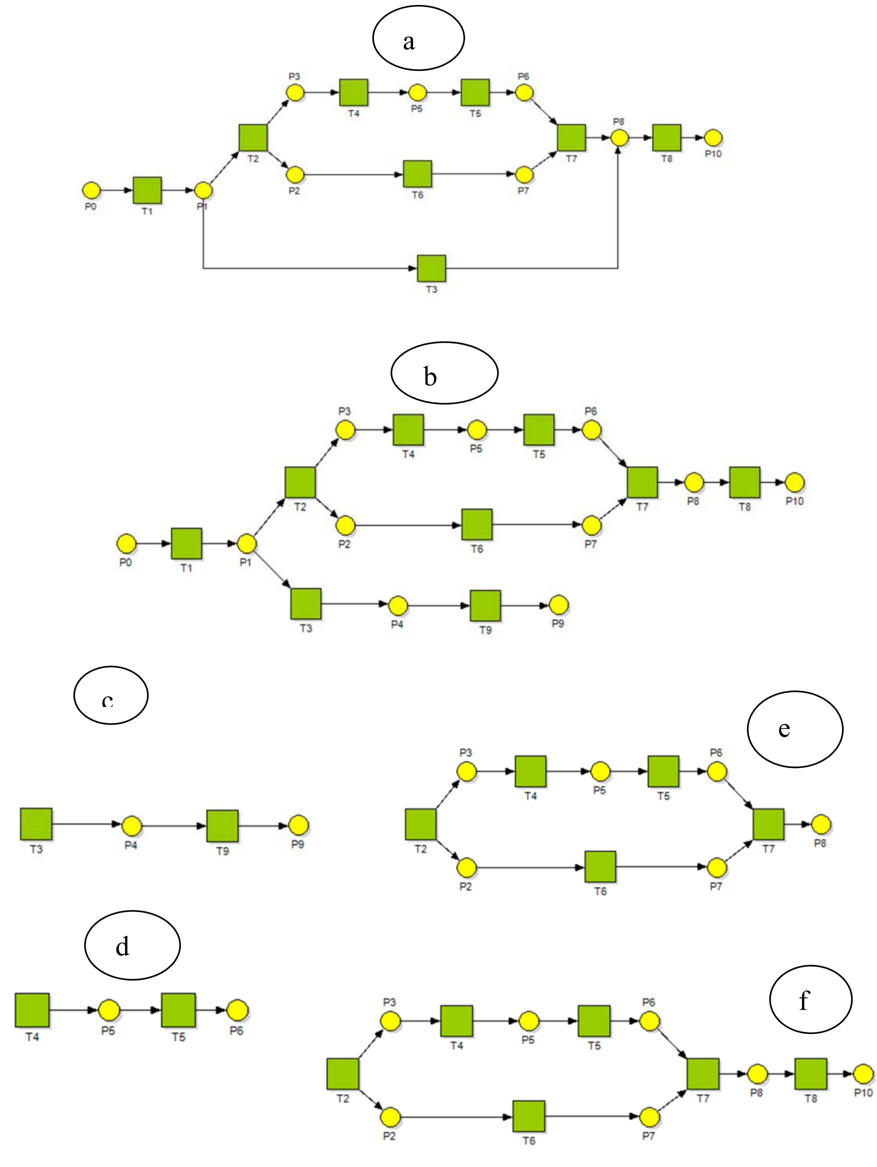


Figure 7: Partitioning a workflow process

6.3 ALGORITHM FOR PARTITION DISCOVERY

Discovering partitions in a complex workflow process can be a daunting task. The complexities of unfolding the model to discover dependencies increase with models implementing advanced control flow patterns. In addition, a typical deployment environment would require for partitions to be dynamically discovered when an offline assignment problem occurs. The work presented in this section is published in paper 2 of part II of this thesis.

We present a mathematical approach to discovering possible partitions using properties of an incidence matrix. The basic idea is that; given a workflow process, dynamically identify grouping of tasks that may be done together (partition). Table 1 presents an incidence matrix of the unfolded net presented in Figure 7(b). In the incidence matrix, each row corresponds to a place while each column corresponds to a transition. For a transition column a value of 1 represents an arc entering from a place in the corresponding row, while -1 is for an outgoing arc. The incidence matrix of a directed graph $M = (P, T)$ is a $P \times T$ matrix $B = (b_{ij})$ such that

$$b_{ij} = f(x) = \begin{cases} -1, & \text{if edge } j \text{ leaves transition } j \ (the\ relation\ \beta(t_j)) \\ 1, & \text{if edge } j \text{ enters transition } j \ (the\ relation\ \alpha(t_j)) \\ 0, & \text{otherwise} \end{cases}$$

It can be observed from Figure 7 (c, d, e and f), that each of the partitions has an initial task (T2, T3, and T4) which are enabled and subsequent tasks that need to be completed. Each of the partition has one thread (series of arcs forming a branch) leaving and no other thread entering. This means that all threads created within the partitions must eventually close. Since a row in the matrix corresponds to a place and each column a transition, the total sum for the rows and columns must be equal to zero. Table 2 gives the sum of the columns and rows in the partition in Figure 7(e). A valid partition is obtained using the equation in proposition.

	T1	*T2*	*T3*	*T4*	*T5*	*T6*	*T7*	*T8*	*T9*
P0	-1	0	0	0	0	0	0	0	0
P1	1	-1	-1	0	0	0	0	0	0
P2	0	1	0	0	0	-1	0	0	0
P3	0	1	0	-1	0	0	0	0	0
P4	0	0	1	0	0	0	0	0	-1
P5	0	0	0	1	-1	0	0	0	0
P6	0	0	0	0	1	0	-1	0	0
P7	0	0	0	0	0	1	-1	0	0
P8	0	0	0	0	0	0	1	-1	0
P9	0	0	0	0	0	0	0	0	1
P10	0	0	0	0	0	0	0	1	0

Table 1: Incidence matrix of unfolded net

Proposition 1: Given an incident matrix of an unfolded workflow net W, a partition W' whose place elements correspond to rows and columns to transitions and can be automatically discovered if:

$$\sum_{j=1}^{n} b_{ij} + \sum_{i=1}^{m} b_{ij} = 0$$

The method for partitioning enables work to be dynamically assigned at different stages of the execution process as long as they can be carried out independent of the original model. These ideas were implemented and a simulation carried out to discover partitions based on several process models. In addition, they were used as a basis for a prototype that was developed and tested in a real life scenario.

	T1	*T2*	*T3*	*T4*	*T5*	*T6*	*T7*	*T8*	*T9*	*total*
P0	-1	0	0	0	0	0	0	0	0	-
P1	1	-1	-1	0	0	0	0	0	0	-
P2	0	1	0	0	0	-1	0	0	0	0
P3	0	1	0	-1	0	0	0	0	0	-
P4	0	0	1	0	0	0	0	0	-1	0
P5	0	0	0	1	-1	0	0	0	0	0
P6	0	0	0	0	1	0	-1	0	0	0
P7	0	0	0	0	0	1	-1	0	0	0
P8	0	0	0	0	0	0	1	-1	0	-
P9	0	0	0	0	0	0	0	0	1	-
P10	0	0	0	0	0	0	0	1	0	-
total	-	1	-	0	0	0	-1	-	-	0

Table 2: Matrix calculation of partitions

The following algorithm gives the method for automatically discovering partitions from an incidence matrix of an unfolded workflow net.

```
Require: A = ⟨S;T;α;β;is⟩              <Unfolded Workflow net>
    M = (P,T)                                <Incidence Matrix>
    T_j = (b_ij)                    <Initial transition column>
    n = 1
    While (n < Number of transitions)do
      If T_j ⇒ T_{j+n} then                          <Rule 1>
        add T_{j+n}                     <Adjacent transition>
        SumTransitions Σ_j^{j+n} T
        sumplaces Σ_i^{i+n} P
        If (sumTransitions+SumPlaces=0) then      <Rule 2>
          Partition = (T_j, T_{j+1},.....T_{j+n}
        End If
      End If
    End While
```

This algorithm was implemented by creating a module in the PIPE framework (Bonet, Lladó et al. 2007). Based on the unfolded model, partitions for the first transition were searched, and then the next, until all transitions in the sequence had been visited. Figure 8 gives the results of the partitions discovered for the model in

Figure 7(b). It can be observed that these results match the partitions that had been observed in Figure 7.

Combined incidence matrix *I*

	T1	T2	T3	T4	T5	T6	T7	T8	T9
P0	-1	0	0	0	0	0	0	0	0
P1	1	-1	-1	0	0	0	0	0	0
P2	0	1	0	0	0	-1	0	0	0
P3	0	1	0	-1	0	0	0	0	0
P4	0	0	1	0	0	0	0	0	-1
P5	0	0	0	1	-1	0	0	0	0
P6	0	0	0	0	1	0	-1	0	0
P7	0	0	0	0	0	1	-1	0	0
P8	0	0	0	0	0	0	1	-1	0
P9	0	0	0	0	0	0	0	0	1
P10	0	0	0	0	0	0	0	1	0

Partition 2

	T2	T6	T4	T5	T7	T8	sum
P0	0	0	0	0	0	0	0*
P1	-1	0	0	0	0	0	-1*
P2	1	-1	0	0	0	0	0
P3	1	0	-1	0	0	0	0
P4	0	0	0	0	0	0	0*
P5	0	0	1	-1	0	0	0
P6	0	0	0	1	-1	0	0
P7	0	1	0	0	-1	0	0
P8	0	0	0	0	1	-1	0
P9	0	0	0	0	0	0	0*
P10	0	0	0	0	0	1	1*
sum j	1**	0**	0**	0**	-1**	0**	0

Partition 1

	T2	T6	T4	T5	T7	sum
P0	0	0	0	0	0	0*
P1	-1	0	0	0	0	-1*
P2	1	-1	0	0	0	0
P3	1	0	-1	0	0	0
P4	0	0	0	0	0	0*
P5	0	0	1	-1	0	0
P6	0	0	0	1	-1	0
P7	0	1	0	0	-1	0
P8	0	0	0	0	1	1*
P9	0	0	0	0	0	0*
P10	0	0	0	0	0	0*
sum j	1**	0**	0**	0**	-1**	0

Partition 3

	T3	T9	sum
P0	0	0	0*
P1	-1	0	-1*
P2	0	0	0*
P3	0	0	0*
P4	1	-1	0
P5	0	0	0*
P6	0	0	0*
P7	0	0	0*
P8	0	0	0*
P9	0	1	1*
P10	0	0	0*
sum j	0**	0**	0

Partition 4

	T4	T5	sum
P0	0	0	0*
P1	0	0	0*
P2	0	0	0*
P3	-1	0	-1*
P4	0	0	0*
P5	1	-1	0
P6	0	1	1*
P7	0	0	0*
P8	0	0	0*
P9	0	0	0*
P10	0	0	0*
sum j	0**	0**	0

Figure 8: Discovered partitions using a computer program

6.4 WORKFLOW ARCHITECTURE FOR MOBILE DATA COLLECTION

The work in this section has been published in paper 1 in part II of this thesis. We proposed integrating generic workflow system with current MDC tools as a way of addressing data flow related issues. A server-agent-client architecture, which enables a mobile device to act as the client that communicates with the server through an agent, is proposed. The client resides on the mobile phone while the agent on the same computer as the server. The agent receives all work items on behalf of the client and ensures that the authorised user is assigned the work item. In addition, we provide a mechanism for matching workflow specification definitions with MDC form definitions. A tool was implemented based on the YAWL (van der Aalst, Aldred et al. 2004) workflow system and OpenXdata tool for MDC. The tool was used in several projects described in case studies presented in section 4.3.

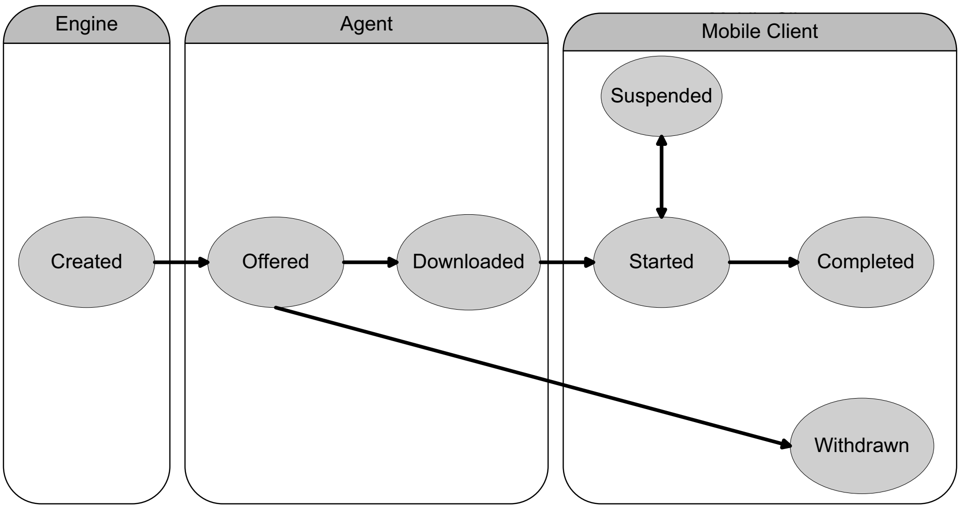


Figure 9: Work item lifecycle in a mobile environment

Figure 9 shows the lifecycle of a work item from the time it is created to completion. A work item comes into existence in the created state. This is done by the workflow engine when all the preconditions required for its enablement have been satisfied and it is capable of being executed. This work item is allocated to an agent for execution which represents the mobile user as a resource. Once connection with the client device is established, the work item is downloaded to a mobile phone. The work item may then be started and any further state transitions are triggered by the resource responsible for the work item up until the time that it is completed or withdrawn. Each state has a specific meaning in terms of how the work item is handled:

- Created - means that the work item has come into existence but no resources are yet aware of it.

- Offered - means that a resource has been identified and the work item has been given to the resource. The resource has the responsibility of getting it completed.

- Downloaded - means that a mobile client has received the work item on the phone work list and is able to execute it.

- Started - is when the responsible resource has commenced executing the work item. It may be suspended if the resource decides to cease execution for a time and be resumed later.

- Withdrawn - happens when the work item is cancelled. In this case the work item will be removed from the client device when a connection is established and any data discarded.

- Completed - identifies that a work item that has been successfully executed to completion and returned to the workflow engine for the next task to be enabled.

6.5 Enabling offline execution in a distributed environment

The work in this section has been published in paper 3 in part II of this thesis. A workflow system that enables offline execution of a partition and synchronises with a server was implemented based on YAWL workflow system and OpenXdata. YAWL provides a service oriented architecture and therefore we describe a service for offline workflow execution.

Figure 10 shows the interaction between the workflow engine and the mobile agent to manage the execution of the offline workflow tasks. This interaction takes place in a server-agent-client architecture. The server controls the execution sequence by enabling tasks that are due for execution. The enabled task (work item) is received by the mobile agent application, which resides on the same computer as the server and added on the list of work items to be done (work list). The agent prepares the work item on behalf of the client by adding functions like partitioning and form bindings for the tasks. It also receives the completed work from the mobile device and channels it back to the server. The client (mobile device) provides the platform through which users execute the work items. Upon completion of work, a log of each activity is kept and sent to the agent when connection is established. This log is used for synchronising the work status as explained below.

The workflow process is instantiated each time a new case (work that needs to be done) is available. When the case advances to the point of the mobile workflow service, the work item is checked to ascertain if its task is part of a partition. If it is not, it is sent to the mobile phone (client) for execution. If it is part of a partition, then its execution status is checked by the mobile agent. If it has not been executed, the work item is sent to the mobile phone for execution. If, however, execution of the task is already completed, the agent waits for a connection to obtain a log of the executed work items. The relevant parameters are inserted into the work item and execution is complete. The work item is then returned to the workflow engine. This process enables synchronisation between the tasks executed on the mobile client and the server. During synchronisation, the client transfers all work to the agent and obtain a new work list which will have been update by the engine when it receives the work.

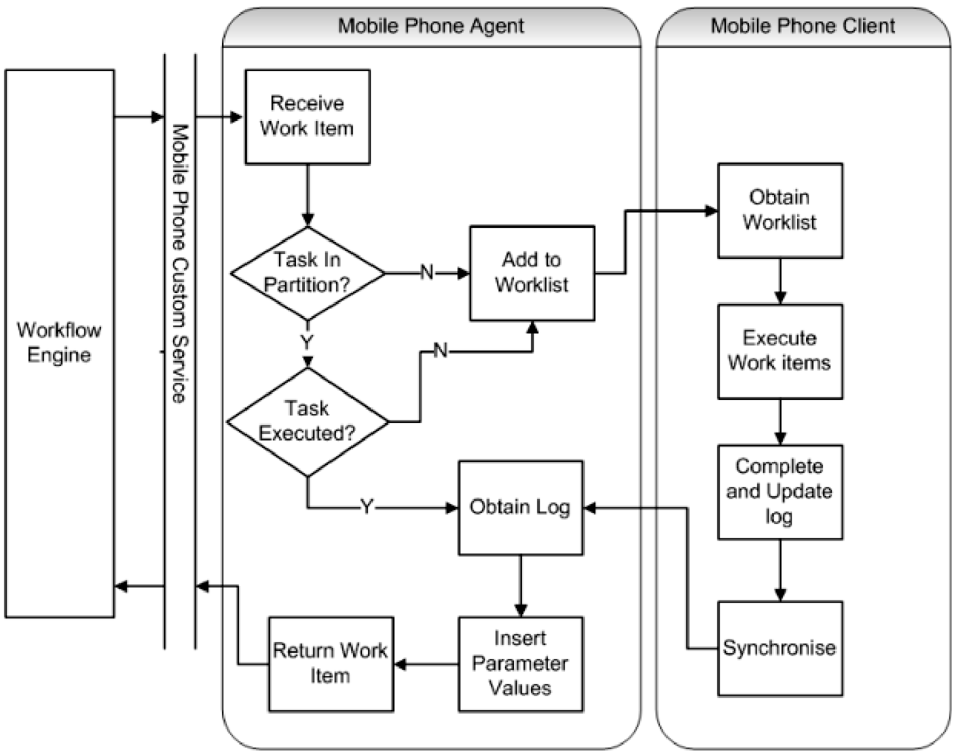


Figure 10: Implementation of an offline workflow service

6.6 DESIGN OF MOBILE CLIENT WITH PARTITION EXECUTION CAPABILITIES

The work in this section has been published in paper 3 in part II of this thesis. In order to enable execution of a fragment of a workflow process, there is need to design a workflow client that is capable of controlling the execution of the workflow based on the partition. Figure 11 illustrates the design of a mobile client to execute workflow segments partitioned from the main workflow process. The mobile agent which resides on the server contains the components *Partition assembler*, *Worklist handler*, *Communication service* and *Server synchroniser*. When a work item is assigned to the mobile agent by the server, it is added to the worklist. If the work item has a partition attached to it, the work list handler notifies the Partition assembler to prepare the partition for the mobile client. Preparation of the work partition involves extracted the relevant process elements from the process model and serialising them in a way that reduces the required processing of the mobile workflow engine.

The mobile phone is designed to obtain a work list from the *Agent* using the *Communication service* which handles the handshaking communication between the mobile client and the agent located on the server. The work items are executed using forms that are pre-loaded on the mobile phone. The mobile process engine provides a mechanism of reading the workflow partition from the repository and determining if subsequent tasks are defined and need to be executed. When the mobile device establishes connection, the synchroniser passes the log and current execution status to the mobie phone agent. Synchonisation can only be done if there is no work item executing.

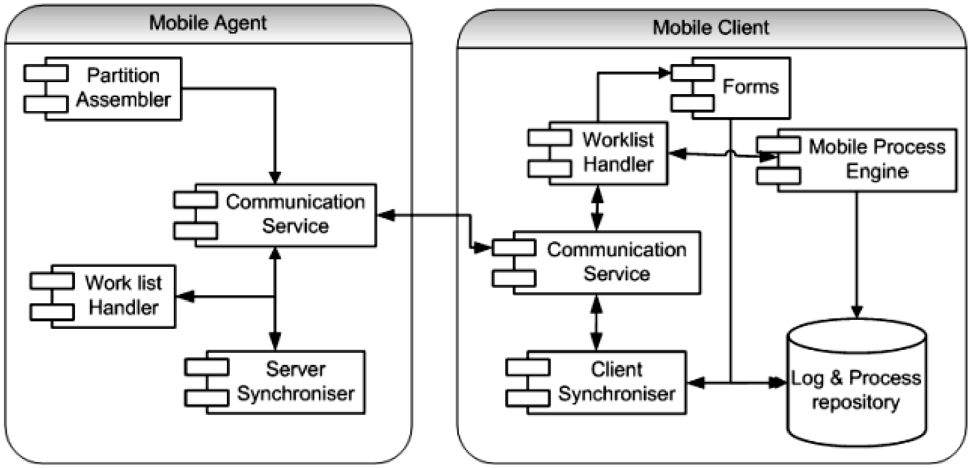


Figure 11: Architecture of a mobile client for offline workflow execution

Figure 12 below, shows a flow chart of the engine process for controlling the workflow process. The engine instantiates a net (partition) to determine the execution sequence of the tasks. If a task is enabled after completion of the first task, then a work item is created and added to the work list. After complete execution of the work item, a log of the event and data is kept. This cycle is repeated until all the tasks in the case are complete.

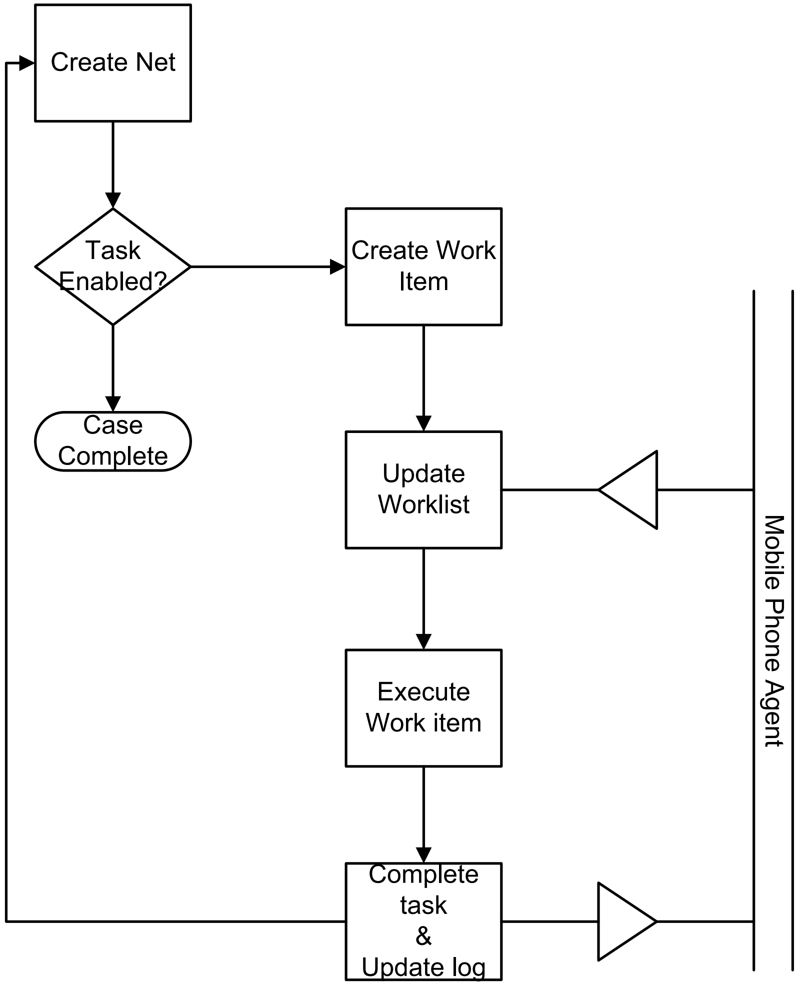


Figure 12: Process for controlling workflow partition

6.7 MAPPING WORKFLOW SPECIFICATIONS TO STUDY DEFINITIONS

The work in this section has been published in paper 1 in part II of this thesis. We provided an entity relationship model that enables mapping of data elements between WFMS and MDC in order to enable data exchange. Figure 13 shows an entity relationship diagram that illustrates the mappings. There exists a hierarchical relationship between the entities of a study and a workflow. In the mapping, each study is matched to one workflow. A task may be matched with one form whereas a form can be matched with many tasks. This is to enable the same form to be used at different stages of the process. The input and output variables of a task can be matched to questions in a form. Each variable can only be matched to one question and vice versa. A form service that utilises these mappings, based on the YAWL system and OpenXdata was implemented.

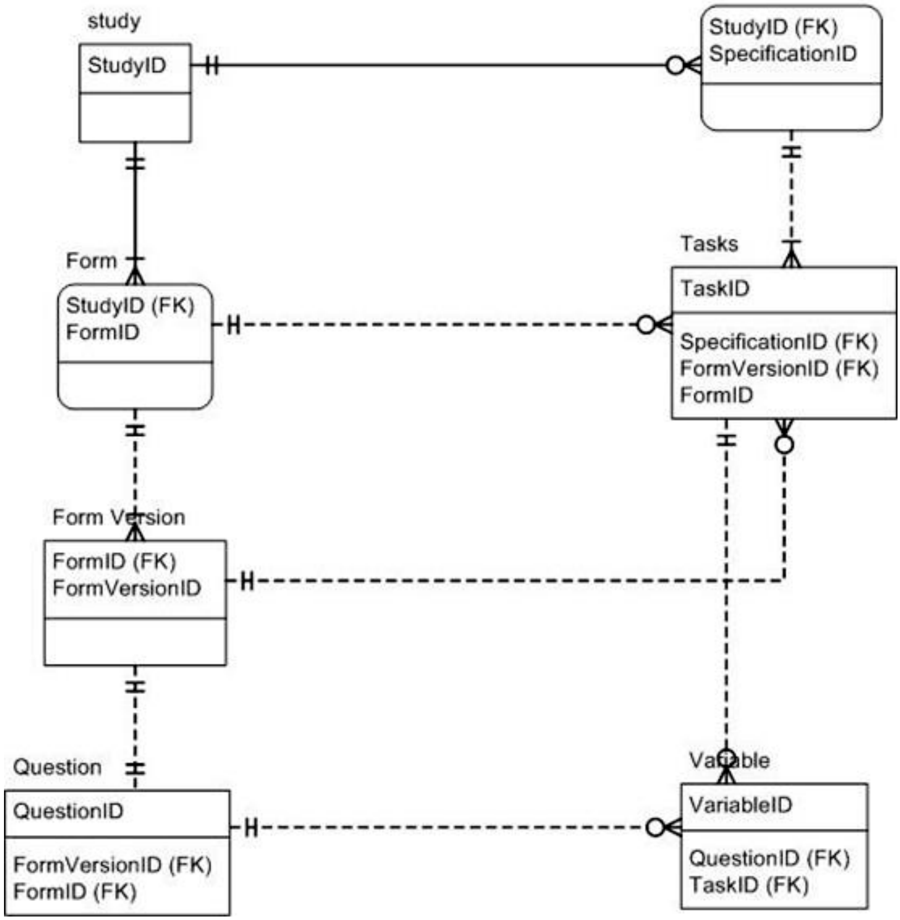


Figure 13: Mapping MDC forms to Workflow specifications

7. DISCUSSION

In this chapter we discuss and evaluate the contributions of this thesis. In Section 7.1 we evaluate each of the invented artefacts against the research questions formulated in Chapter 2, and in Section 7.2 we discuss related work.

7.1 FULFILMENT OF THE RESEARCH QUESTIONS

Question 1: How can generic workflow support be implemented for mobile data collection tools?

This question has been addressed by identifying the key issues that need to be addressed to enable generic workflow support for mobile data collection. In section 6.7 we provide a mechanism for mapping the workflow specification and form definition for data collection. We then provide a client-agent-server architecture implemented through a workflow adapter in section 6.4 to enable data collection when client is offline. These methods were implemented and tested in a number of projects hence answering the question.

Question 2: How can mobile-based workflows be implemented to mirror the flexibility and ubiquity provided by paper-based routines in field activities?

In section 2.4 we analysed the paper-based data collection process in a clinical trial and identified the requirements for mobile-based systems. It was stated that data could be collected by a lone field worker, or a team working in connected or disconnected environments. Regarding flexibility of paper-based routines, there is no need for collaboration/update of other teams when a task is completed. The workflow approach proposed allows such flexibility by proposing partitioning workflow process in a distributed architecture. Ubiquity is provided by enabling the mobile device to work in offline mode, whereby all tasks and data required to complete the tasks, are pre-loaded on the mobile phone.

In section 5.2 we identify the requirements for partitioning and propose a method for partitioning workflow system in order to enable offline execution. Teams are able to take the forms and partitions on their mobile devices and carry out prescribed work and only require synchronisation once work is compete. In section 6.2 we go further and provide an algorithm for workflow discovery to enable automated discovery of partitions based on a workflow model. These algorithms were tested by carrying out experiments based on Petri-net simulations of workflow processes. Each simulation considered the complexity of patterns and tasks. The findings showed that the approach works well for a variety of process models.

In this thesis, we describe a system that enables work to be executed in offline manner. Two modes of execution are discussed. In both modes, the client device is offline but in one case, it is always possible to establish connections to the server while in the other case, connections can be established intermittently. The availability of these two modes of operation enables the system to operate flexibly and therefore ably replace paper as a tool for data collection.

Question 3: How can workflows be implemented in environments where work is conducted offline?

In order to answer this question, we provide architecture for workflow execution in connected and offline environments and implement a system using YAWL and OpenXdata systems. The YAWL system architecture is based on the recommendations of the WFMC and therefore provides a generic case for workflow systems. Conversely, the OpenXdata system implements common standards for mobile data collection under the OpenROSA consortium and therefore provides a good example of MDC tools.

We provide architecture for distributed workflow system for MDC in section 6.6. The architecture enables work items and related partitions to be uploaded to a mobile phone which is equipped with a light-weight workflow engine. Additionally, we describe a method for synchronisation in order to maintain control of the entire process by the workflow server. These ideas were tested in a real world project that collects demographic data from rural parts of Uganda. The tool was found to be generic enough to support various data definitions appropriately for the case study.

7.2 RELATED WORK

This section discusses work that is related to the work presented in this thesis. In Section 7.2.1 we address existing approaches to workflow support in mobile-based systems. In Section 7.2.2 we discuss approaches to partitioning workflow systems and contrast with the work presented in thesis. In section7.2.3, the approaches to the design of distributed workflow are discussed. Finally section 7.2.4 presents current approaches to the implementation of mobile-based workflow systems. The reader is also referred to the related work section of the papers presented in Part II of the thesis.

7.2.1 CURRENT APPROACHES TO WORKLOWS IN MOBILE ENVIRONMENTS

MDC tools based on OpenROSA standards like Javarosa (Klungsöyr, Wakholi et al. 2008) do not explicitly implement workflows but are developing elements that are workflow related. This includes the ability to view pre-filled forms and link forms in a study. These have been used as a basis for requirements and motivation for workflow implementation for MDC. The work presented in this research provides a comprehensive, ontology and standards-based approach to data collection processes which are time dependent and cumulative.

Tan and Fan (2007) provide a Petri-net based approach for fragmenting workflows for distributed execution. The fragments created can migrate to servers where tasks are performed and new fragments are created. Through this approach a case can be executed on several servers in succession thus enabling the outsourcing of business functionalities. Guth, Lenz et al. (1998) present an approach for the distributed execution of workflows based on the fragmentation of high-level Petri-nets. The Petri-nets are fragmented horizontally, vertically and diagonally, and fulfil the necessary requirements for formal workflow behaviour like completeness, minimality and disjointedness.

Conceptually, formal methods presented in (Guth, Lenz et al. 1998; Tan and Fan 2007) for distributed workflow differ from our approach due to the problem addressed and deployment environment. Our approach seeks to move work to a client with a light-weight workflow engine for offline execution by combining two or more tasks while maintaining semantics of the original model.

7.2.2 WORKFLOW PARTITIONING

Baresi, Maurino et al. (2005) use workflow partitioning in BPEL to structurally provide rules, based on graph transformations. Transformations are based on rules that ensure that the system exposes the functional behaviour and the flow of the original workflow is observed. The partitioned BPEL processes are executed onto a network of mobile phones. Through this approach, they are able to produce an overall execution model that is equivalent to a centralised one, implemented using disconnected components and independent workflow engines. (Pryss, Tiedeken et al. 2011) present the MARPLE architecture that enables the execution of processes on mobile devices. Through their system, they realize generic process management. The architecture meets the performance requirements of mobile scenarios to cope with specific requirements like broken connections and limited GUIs. They provide a set of requirements for the architecture to work which form part of the basis for the partitioning algorithms proposed. In their work, conceptual issues regarding the partitioning of processes are not addressed.

7.2.3 Distributed Workflow Systems

Many distributed workflow systems have been proposed to address different challenges. The kind of approach proposed depends largely on the problem to be addressed. For instance, (Muth, Wodtke et al. 1998) propose an approach to enable workflow support for large-scale distributed, enterprise wide applications. Their approach is based on the distributed execution of state and activity charts. They used formal semantics of state and activity charts, to develop an algorithm for transforming a centralised state and activity chart into a provably equivalent partitioned one, suitable for distributed execution. Furthermore, they propose a synchronisation to ensure that the original execution sequence of a non-distributed system is maintained.

(Schuster, Jablonski et al. 1994) focus on developing architecture for distributed systems that execute tasks using parallelism. The use of parallelism is due to performance requirements and involves data and applications that are spread across a heterogeneous, distributed computing environment. Their architecture for WFMS aims to provide scalability through transparent parallelism and heterogeneity of various systems involved.

From literature, it can be observed that a common approach to distributing workflow execution is through partitioning. Crucial however is the purpose for distribution which determines the partitioning approach taken. For web-based systems, the main reason for partitioning is twofold; to enable parallel processing of complex workflow systems or to enable decentralised execution of geographical diverse systems. This contrasts with the reason for distribution in a mobile environment which has to do with coping with mobility of the devices used in disconnected environments and limited computing resources which calls for a different approach. Our approach uses the concept of partitioning workflows for distribution and applies it in the mobile computing environment.

7.2.4 MOBILE BASED WORKFLOW SYSTEMS

More related to the ideas proposed in this thesis, Bolcer (2000) proposes MAGI – an architecture for mobile and disconnected workflow. The problem addressed is that of tasks disconnected from the rest of a business process – similar to the problem tackled in this thesis. MAGI - an architectural framework that explicitly addresses the coordination of e-business messaging and deployment across a range of computing platforms. The focus is on technology by specifying open-source interoperability specification consisting of complementary protocol standards, formats, and implementations based on the HTTP protocol. This enables MAGI to support natural mapping of the Workflow Management Coalition's model to web primitives.

Dustdar and Gall (2003) also address the challenge of coordinating work using mobile devices in a distributed manner in order to share expertise across locations and different mobile devices. They propose a framework for distributed and mobile collaboration and define a set of requirements for virtual communities and discuss mobile teamwork support software. In their architecture, they aim to support mobility by addressing the difficulties of bandwidth restrictions, unreliable connections and disconnected operations as highlighted in this thesis. (Pryss, Tiedeken et al. 2011) introduce the MARPLE that integrated process management technology and mobile computing frameworks so as to workflow support for mobile systems. They provide architecture and its components consisting of process engine which enables light-weight as well as flexible process support on mobile devices.

It can be observed that there is increasing maturity in respect to the demands for workflow support for mobile computing environments that address the unique constraints. Various applications from light-weight frameworks and process engines for mobile computing have been proposed. One key variation with our approach is on the technologies used. Whereas, (Bolcer 2000; Dustdar and Gall 2003) propose to use WAP and WML protocols to access the server, MDC tools use applications residing on the device as a client to enable offline processing of information. The variation in approach impacts significantly on the architecture and methods required. (Pryss, Tiedeken et al. 2011) provide architecture for light weight workflow engine and an approach to assigning work in partitions. The work presented in this thesis builds on this and addresses gaps like approaches towards partitioning workflow processes and enabling such system to work in a distributed environment with control by a powerful web-based server.

7.3 REFLECTIONS

The work presented in this thesis has been based on rigorous design science research and development of artefacts to enable the use of mobile devices in rural parts of Africa. The need to conduct research and the desire to solve real world problems by providing workable solutions was a great challenge. Whereas, it would have been possible to have different approaches to achieving the desired result, the scientific approach presented in this thesis enables generalisation within the mobile computing and workflow domains. The experiments used to validate a key component of this research – partitioning workflow for mobile execution could have been more varied in order to provide clear limitations and boundaries of the proposed methods.

The key problem addressed related to using workflow with constraints of limitations of handheld devices and mobile computing environments. In cases where these are not a big issue, the methods proposed may not necessarily provide the best approach. It can be argued that these constraints make it rather difficult to use WFMS and therefore the workflow approach could be left out altogether.

Furthermore, using OpenXdata and YAWL to create generalisations for workflow systems in mobile environments may not always be prudent. Within the workflow area, there are many approaches to defining and enacting workflow process and the YAWL approach is just one of the many. In addition, OpenXdata tool is just one of the many tools for MDC, and the standards are yet to be widely accepted and adopted by the MDC community. Therefore it can be argued that this research is limited in its applicability in a variety of other workflow modelling approaches and tools.

Nevertheless, we have demonstrated that the ideas presented could apply across many mobile computing environments. Even in third and fourth generation networks which are fast and stable and cases where devices like iPads have all computing capabilities, disconnection can still occur due to battery down time and switching off by the user. It can therefore be argued that as long as ubiquitous computing remains the motivating factor, the mobile environment constraints observed will remain. Therefore, this thesis contributes to scientific knowledge by proposing generalisable approaches to the use of workflows for MDC.

8. CONCLUSION

This chapter concludes Part I of the thesis by summarising the results and pointing out directions for future work.

8.1 SUMMARY

The rapid growth in the use of mobile devices in developing countries coupled with the ever-increasing number of features and capability has led to increased use for computing functions. The demand for using mobile device for data collection provides a compelling case to overcome many limitations of the devices and mobile networks. Many generic mobile-based applications have been developed to enable data collection offline and upload/synchronisation with the server, once connection is established. This mode of operation provides flexibility as the system may be used in places where there is no network and can therefore replace paper-based routines for field data collection.

This thesis contributes to the domain of workflow systems and mobile systems by providing mechanisms for linking generic workflow systems and mobile data collection tools in order to allow for process related definition of the data collection process. It further provides methods for enabling offline execution of workflows in order to reduce the need for frequent connection with the server and thereby maintain the flexibility provided by paper-based routines.

The work presented in this thesis is implemented using the OpenXdata System for mobile data collection and YAWL workflow system. These systems are based on largely agreed standards for mobile data collection and workflow management respectively. Through case-studies, experiments and field evaluations, and case-based studies the ideas developed through the action research methodology are validated.

It can be argued that the work presented makes scientific contribution in partitioning workflow systems for distributed execution in mobile environments. Additional contribution is the architecture provided for designing mobile-based systems to work in offline and distributed mode thus enabling work when network connection is not available. The last contribution is providing methods for linking generic data collection systems to generic workflow systems. Furthermore, even though the focus for this research has been mobile data collection, the methods and architectures devised can be used for workflows in mobile based systems in general.

8.2 FUTURE WORK

It is important that the artefact developed in this thesis is tested in a wider context to determine design constraints that may need to be addressed. The method partition discovery presented is limited and would not be appropriate for complex workflow constructs. Future work will need to devise better approaches to partitioning and partition discovery.

Although the driving force in this thesis was to develop artefacts that could be used in a real-world environment under the OMEVAC project, there is still work that needs to be done to ensure that the methods proposed address the requirements and regulations for clinical trials. Future work could include large-scale evaluations based on real-life scenarios – most preferably clinical studies.

The application of the concepts developed in this thesis will depend largely on the extent to which workflow patterns can be supported, without violation. As explained in section 3.1, in PAISs various perspectives related to the control-flow, data, resource and exception handling can be distinguished. van der Aalst, ter Hofstede et al. (2003) conducted research and proposed a range of patterns in these perspectives. The work proposed in this thesis concentrated more on the control flow perspective. The patterns supported are those that are commonly occurring and therefore more advanced patterns which provide more challenges to the mobile computing environment need to be considered. Resource and data patterns, though supported in the artefacts developed were not fully evaluated. Future work needs to study the support for resource and data patterns with an aim of determining the extent to which they can be supported. Finally, exception handling poses the greatest challenge in offline environments. This is due to the fact that it impossible to access the mobile client when it is offline, in order to cancel a process or attend to any other exception on the server side. Future work needs to address the issue of exception handling in offline mobile computing environments so as to enable actions to be undertaken as a result of an exception.

9. BIBLIOGRAPHY

Alonso, G., R. Gunthor, et al. (1996). "Exotica/FMDC: A Workflow Management System for Mobile and Disconnected Clients." Distributed and Paralell databases **4**(3): 229-247.

Anokwa, Y., C. Hartung, et al. (2009). Open Source Data Collection in the Developing World, IEEE. **42:** 97-99.

Bahrami, A., C. Wang, et al. (2006). The Workflow Based Architecture for Mobile Information Access in Occasionally Connected Computing. IEEE International Conference on Services Computing, IEEE Computer Society.

Bardram, J. E. (1997). Plans as situated action: an activity theory approach to workflow systems, Springer.

Baresi, L., A. Maurino, et al. (2005). "Workflow partitioning in mobile information systems." Mobile information systems: 93-106.

Benbasat, I., D. Goldstein, et al. (1987). "The Case Research Strategy in Studies of Information Systems." MIS Quarterly **11**(3): 369-386.

Bolcer, G. A. (2000). "Magi: An Architecture for mobile and disconnected Workflow." Internet Computing, IEEE **4**(3): 46-54.

Bonet, P., C. M. Lladó, et al. (2007). PIPE v2. 5: A Petri net tool for performance modelling. Proc. 23rd Latin American Conference on Informatics (CLEI 2007).

Burch, J. R., E. M. Clarke, et al. (1994). "Symbolic model checking for sequential circuit verification." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on **13**(4): 401-424.

Cavaye, A. L. M. (1996). "Case study research: a multi-faceted research approach for IS." (3): 227-242.

ClinicalTrials.gov. (2008, 26/01/2012). "Comparison of Efficacy and Safety of Infant Peri-exposure Prophylaxis With Lopinavir/Ritonavir Versus Lamivudine to Prevent HIV-1 Transmission by Breastfeeding." Retrieved 13/05/2012, 2012, from http://clinicaltrials.gov/ct2/show/NCT00640263.

Collins, C. (2007). "OpenClinica - Open Source for Clinical Research." Retrieved 05/03/2010, 2010, from http://www.openclinica.org/entities/entity_details.php?eid=374.

contributors, W. (2012, 18 August 2012 21:03 UTC). "Data collection " Retrieved 5 September 2012 09:56 UTC from http://en.wikipedia.org/w/index.php?title=Data_collection&oldid=508035137.

Dustdar, S. and H. Gall (2003). "Architectural concerns in distributed and mobile collaborative systems." Journal of Systems Architecture **49**(10): 457-473.

Esparza, J. and K. Heljanko (2001). "Implementing LTL model checking with net unfoldings." Model Checking Software: 37-56.

Esparza, J., S. Römer, et al. (1996). "An improvement of McMillan's unfolding algorithm." Tools and Algorithms for the Construction and Analysis of Systems: 87-106.

FDA (2007). Guidance for Industry Computerized Systems Used in Clinical Investigations. O. o. t. C. U.S. Department of Health and Human Services Food and Drug Administration.

Fischer, L. (2002). Workflow handbook, 2002, Future Strategies Inc.

Fox, K. (2011). Africa's mobile economic revolution. The Observer. United Kingdom, Guardian News and Media Limited**:** 1.

Guth, V., K. Lenz, et al. (1998). Distributed workflow execution based on fragmentation of petri nets, Citeseer.
Hackmann, G., M. Haitjema, et al. (2006). Sliver: A BPEL Workflow Process Execution Engine for Mobile Devices. Service-Oriented Computing – ICSOC 2006. Berlin, Springer Berlin / Heidelberg**:** 503-508.
Hevner, A. R., S. T. March, et al. (2004). "Design science in information systems research." MIS Quarterly **28**(1): 75-105.
Hollingsworth, D. (1995). Workflow Management Coalition: The Workflow Reference Model.
ITU (2008). African Telecommunication/ICT indicators 2008: At a crossroads Place des Nations – CH-1211 Geneva 20 – Switzerland, International Telecommunications Union.
Klungsöyr, J., P. Wakholi, et al. (2008). OpenROSA, JavaROSA, GloballyMobile–Collaborations around Open Standards for Mobile Applications.
M4W. (2012). "Mobile for improved safe access to water." from http://m4water.org/.
Mahling, D. E., N. Craven, et al. (1995). "From office automation to intelligent workflow systems." IEEE Expert **10**(3): 41-47.
Mallick, M. (2003). Mobile & Wireless Design Essentials, Wiley-India.
Moon, K.-h. K. (2006). Techniques for Designing Case Report Forms in Clinical Trials. ScianNews. **9**.
Murata, T. (1988). Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, IEEE. **77**.
Muth, P., D. Wodtke, et al. (1998). "From Centralized Workflow Specification to a distributed workflow execution." Journal of Intelligent Information Systems **10**: 159 - 184.
News, A. T. (2012). African Mobile Factbook 2012, Africa & Middle East Telecom-Week.
OpenROSA, c. (2011). Retrieved 30/November/2011, 2011, from http://openrosa.org/.
openXdata, C. "OpenXdata - opensource tool for data collection." Retrieved 27/11/2012, 2010, from http://www.openxdata.org/.
Pryss, R., J. Tiedeken, et al. (2011). "Towards flexible process support on mobile devices." Information Systems Evolution: 150-165.
Recker, J., M. Rosemanny, et al. (2007). "Ontology- Versus Pattern-Based Evaluation of Process Modeling Languages: A Comparison." Communications of the Association for Information Systems **20**(48).
Russell, N., A. ter Hofstede, et al. (2005). "Workflow data patterns: Identification, representation and tool support." Conceptual Modeling–ER 2005: 353-368.
Russell, N., A. H. M. Ter Hofstede, et al. (2005). Workflow resource patterns, Citeseer.
Schuster, H., S. Jablonski, et al. (1994). A client/server architecture for distributed workflow management systems, IEEE.
Sharp, A. and P. McDermott (2009). Workflow modeling: tools for process improvement and applications development, Artech House Publishers.
Tan, W. and Y. Fan (2007). "Dynamic workflow model fragmentation for distributed execution." Computers in Industry **58**(5): 381-391.
Tellis, W. (1997). "Application of a Case Study Methodology." The Qualitative Report **3**(3).
van der Aalst, W., M. Adams, et al. (2010). Modern Business Automation, Springer.
van der Aalst, W., L. Aldred, et al. (2004). Design and implementation of the YAWL system, Springer**:** 281-305.

van der Aalst, W. M. P. (2004). Business Process Management Demystified: A tutorial on Models, systems and standards for workflow management. Eindhoven, Springer Berlin / Heidelberg.

van der Aalst, W. M. P., A. H. M. ter Hofstede, et al. (2003). "Workflow Patterns." Distributed and Parallel Databases **14**(1): 5-51.

Wakholi, p. K., H. (2010). "CU@SCHOOL: Tracking School attendance in Uganda." Retrieved 08/10.2012, 2012, from http://twaweza.org/index.php?i=221.

Weber, B. A., H. Yarandi, et al. (2005). "A comparison study: paper-based versus web-based data collection and management." Applied Nursing Research **18**(3): 182-185.

Weske, M. (2005). Business Process Management - Concepts, Languages, Architectures. Berlin, Springer.

Wieringa, R. and A. Moralı (2012). "Technical Action Research as a Validation Method in Information Systems Design Science." Design Science Research in Information Systems. Advances in Theory and Practice: 220-238.

Yin, R. K. (1984). Case Study Research, Design and Methods. Beverly Hills, California, Sage Publications.

Zheng, P. and L. M. Ni (2006). Smart phone and next generation mobile computing, Morgan Kaufmann.