9-18-2015

Modeling, Designing, and Implementing an Ad-hoc M-Learning Platform that Integrates Sensory Data to Support Ubiquitous Learning

Hien M. Nguyen
Florida International University, hnguy002@fiu.edu

DOI: 10.25148/etd.FIDC000194

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

MODELING, DESIGNING, AND IMPLEMENTING AN AD-HOC M-LEARNING PLATFORM THAT INTEGRATES SENSORY DATA TO SUPPORT UBIQUITOUS LEARNING

A dissertation submitted in partial fulfillment of

the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Hien Nguyen

2015

To: Interim Dean Ranu Jung
College of Engineering and Computing

This dissertation, written by Hien Nguyen, and entitled, Modeling, Designing, and Implementing an Ad-Hoc M-Learning Platform that Integrates Sensory Data to Support Ubiquitous Learning, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

S.S. Iyengar

Kia Makki

Deng Pan

Naphtali Rishe

Kang Yen

Niki Pissinou, Major Professor

Date of Defense: September 18, 2015

The dissertation of Hien Nguyen is approved.

Interim Dean Ranu Jung
College of Engineering and Computing

Dean Lakshmi N. Reddi
University Graduate School

Florida International University, 2015

DEDICATION

I would like to dedicate this dissertation to:

My wife, Dung Phan-Nguyen, and our parents, for their continued support of my education and career. Without their patience, understanding, support, and most of all love, the completion of this work would not have been possible.

Our closest friend, Tran Dang and her family, for their great support of me from start to finish.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my academic parents and advisors: Dr. Niki Pissinou and Dr. Kia Makki for their excellent guidance, unbelievable support to open my mind in the mobile learning research fields and ultimately I was able to complete this dissertation. Without their supervision, advice, and help, it would have not been possible. They mentored and encouraged me to stay focused to complete the research work. They inspired me as great professors who I look up to pursue and perform well in my academic career. Dr. Pissinou and Dr. Makki have spent many of their valuable hours during weekday and weekend to provide me numerous discussions, great lectures, feedback to my research work. They helped and guided me on related research topics to become a better thinker and a better researcher. I am also grateful to be a FIU graduate student because I have been able to use many great learning resources at the library and at the school of Computing and Information Sciences, just to name a few.

I would like to also thank my committee members—Dr. S. S. Iyengar, Dr. Deng Pan, Dr. Naphtali Rishe, and Dr. Kang Yen—for their support and motivation to finish my research. Thanks to Ebtissam Wahman, the Research Experience Undergrad (REU) student I mentored during the ten-week summer research experience in 2014 at FIU, and Nguyen Minh Vo Le, an IntelliMath associate, for the fruitful discussions and work resulting in a successful paper publication.

Most importantly, none of this would have been possible without the love and patience of my wife, Dung Phan-Nguyen, and our parents: Dung Van Nguyen, Nam Thi Van, Viet Phan, Kim Hue Pham. They have been a constant source of love, concern, support and

strength all these years. I thank them for their loving support and understanding throughout my academic journey. They helped me to stay focused on my graduate work and all academic matters as well as accomplishments. I would like to express my heart-felt gratitude to all members of my immediate family and my wife’s family for their incredible support. I would like to thank our best friend, Tran Dang, and her entire family for their support and care that helped me to overcome challenges and setbacks. I deeply appreciate the generosity, the way they believed in me and how they encouraged me to keep learning and moving forward throughout this endeavor.

I would like to give a special mention for the mental support given by the Buddhist monk/Zen master, Vien Thuc, aunt Kim Thuan Nguyen and uncle Thu Minh Nguyen who we all met and introduced master Vien to me. They all helped me to stay calm and focused. The Zen way helped me to write this dissertation and over one hundred fifty Zen poems. Their constant encouragement and incessant inspiration enabled me to stay focused to complete the research and writing of this dissertation.

Last but not least, I appreciate the grant and financial support of the National Science Foundation (Award Number: 1407067 and 1263124), Department of Homeland Security (Award #2008-ST-062-000012) that funded parts of the research discussed in my dissertation.

ABSTRACT OF DISSERTATION

MODELING, DESIGNING, AND IMPLEMENTING AN AD-HOC M-LEARNING PLATFORM THAT INTEGRATES SENSORY DATA TO SUPPORT UBIQUITOUS LEARNING

by

Hien Nguyen

Florida International University, 2015

Miami, Florida

Professor Niki Pissinou, Major Professor

Learning at any-time, at anywhere, using any mobile computing platform learning (which we refer to as "education in your palm") empowers informal and formal education. It supports the continued creation of knowledge outside a classroom, after-school programs, community-based organizations, museums, libraries, and shopping malls with under-resourced settings. In doing so, it fosters the continued creation of a cumulative body of knowledge in informal and formal education. Anytime, anywhere, using any device computing platform learning means that students are not required to attend traditional classroom settings in order to learn. Instead, students will be able to access and share learning resources from any mobile computing platform, such as smart phones, tablets using highly dynamic mobile and wireless ad-hoc networks. There has been little research on how to facilitate the integrated use of the service description, discovery and integration resources available in mobile and wireless ad-hoc networks including description schemas and mobile learning objects, and in particular as it relates to the consistency, availability, security and privacy of spatio-temporal and trajectory information. Another challenge is finding, combining and creating suitable learning modules to handle the inherent constraints of mobile learning, resource-poor mobile devices and ad-hoc networks.

The aim of this research is to design, develop and implement the cutting edge context-aware and ubiquitous self-directed learning methodologies using ad-hoc and sensor networks. The emphasis of our work is on defining an appropriate mobile learning object and the service adaptation descriptions as well as providing mechanisms for ad-hoc service discovery and developing concepts for the seamless integration of the learning objects and their contents with a particular focus on preserving data and privacy. The research involves a combination of modeling, designing, and developing a mobile learning system in the absence of a networking infrastructure that integrates sensory data to support ubiquitous learning. The system includes mechanisms to allow content exchange among the mobile ad-hoc nodes to ensure consistency and availability of information. It also provides an on-the-fly content service discovery, query request, and retrieving data from mobile nodes and sensors.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

ABBREVIATIONS AND ACRONYMS

M-Learning	Mobile Learning
E-Learning	Electronic Learning
U-Learning	Ubiquitous Learning
P2P	Peer-to-Peer
LO	Learning Object
CMS	Content Management System
LMS	Learning Management System
SCORM	Sharable Content Object Reference Model
LOM	Learning Object Metadata
SOAP	Simple Object Access Protocol
IEEE	Institute of Electrical and Electronic
ADL	Distributed Learning Initiative
API	Application Programming Interface
QR Code	Quick Response Code
CSS	Cascading Style Sheet
HTML	Hypertext Markup Language
GPS	Global Positioning System

PAN	Personal Area Network
LAN	Local Area Network
WLAN	Wireless Local Area Network
WAN	Wide Area Network
J2ME	Java Micro Edition
MLA	Mobile Learning Application
MLO	Mobile Learning Object
AMLO	Adaptive Mobile Learning Object
SLE	Learning and Sensing Environment
AE	Adaptation Engine
ISO	International Organization for Standardization
GPRS	General Packet Radio Service
GSM	Global System for Mobile communication
XML	Extensible Markup Language
JSON	Javascript Object Notion
SQL	Structure Query Language
MD	Mobile Device
GLOS	Global Learning Object Sharable

SRM	Searching and Ranking Module
DMM	Data Management Module
RRM	Request and Response Module
NP	Neighbor Profile
GUI	Graphical User Interface
IDE	Integrated Development Environment
JDK	Java Development Kit
SDK	Software Development Kit
ADT	Android Development Tool

CHAPTER 1: INTRODUCTION

The 21st century Information Technology (IT) users are digitally savvy users who depend extensively on mobile technologies for everyday life and learning. These users dependence on technology has fostered "learning mediated by networked computing and communications technologies where the forms of steering of learning can arise in a hybrid manner from a variety of personal, educational or collective sources and designs " referred to as "cyber-learning" [46]. Cyber-learning encompasses many forms of learning including e-learning [18] and M-Learning [51, 45, 31-35].

M-learning uses mobile devices, sensors and wireless transmissions [45] to provide anytime and anywhere access to content as well as just in-time training or review of content. It delivers any-time, anywhere device learning (which we fondly refer to as "education in your palm" [4, 51]) and empowers informal science education. M-learning supports mobility, collaboration, advanced personalized learning experiences and on-the-fly interaction with adaptive feedback among users and real-world situations [53, 63, 64]. It also provides access to learning modules, hence fostering the continued creation of a cumulative body of knowledge in formal or informal education. Anytime, anywhere, any device learning means that users are not required to attend classes like in a traditional school. Instead of coming to class or a library to gain knowledge, the users/learners should be able to access learning modules from their computers or mobile devices. These devices may be attached to a high performance fixed or wireless network, ad-hoc networks, wireless sensor networks. Under these situations, users would need to find suitable modules and combine them freely.

Furthermore, users will be more successful and more likely to stay on task if they have the opportunity to cooperate and communicate with their fellow users. While this can easily be achieved in a traditional classroom setting, it is far more challenging to provide adequate means of this support in a teaching environment as the one described here [41, 51, 56]. To realize the vision we must foster emerging technologies and develop new collaborative modes, media richness, and virtual worlds which can facilitate and support inter-operability between heterogeneous local and specialized systems [9, 10, 51]. M-learning is still in its infancy because much of the research is theoretical in nature [41, 42]. M-learning can be taken place at anytime, anywhere, on any-device, and platform with or without the predetermined locations. Existing cyber-learning systems are ineffective due to many technological challenges associated with ad-hoc or peer-to-peer networking. This includes limitations posed by the inherent characteristics of wireless systems such as transmission interferences, bandwidth, limited resources as well as content inconsistencies of sharable content objects because of rapid updates (here today, outdated tomorrow) and semantic knowledge representation among learning objects among others.

1.1 Problem Statements

The problem under investigation is in the area of mobile learning in ad-hoc networks. In other words, the core theme of our research emphasizes M-Learning solutions for ad-hoc and sensor networks. More precisely, the study focuses on modeling and designing adaptive M-Learning mechanisms for building and delivery content using privacy centric in an ad-hoc network where mobile devices in the M-Learning system have limited resources. Their connections are infrastructureless, meaning only ad-hoc on-the-fly connections among those mobile nodes. Currently, M-Learning systems are depended on the underlined infrastructures, wired and wireless networks that are used the inter-networking technologies as a multi-tier system dependent to allow mobile users/learners to access and retrieve the learning materials. Therefore, there is a lack of M-Learning solution that allow mobile learners to connect, collaborate and learn via the mobile ad-hoc network where no network infrastructure is available. There have been works on mobile learning systems where mobile users get access to retrieve the learning content on-the-go via reliable infrastructure wireless networks such as WiFi or their cellular network carriers. However, those solutions are not viable when there are only ad-hoc connections available among mobile nodes. A complete M-Learning solution is a dynamic solution where mobile users can use their mobile devices to access, request, or retrieve their on-the-fly content, anywhere and anytime. Thus, a service discovery, efficiency adaptive mechanisms to request and retrieve content with privacy centric are needed. The effort to solve the problem in this dissertation is divided into the flowing sub-problems:

1. Modeling, designing, and developing a mobile learning system in an ad-hoc network:
    a. Defining efficiently mechanisms to allow content exchange among the mobile ad-hoc nodes to ensure consistency and availability of information.
    b. Providing on-the-fly content service discovery, query request, and retrieving data from mobile nodes and sensors.
2. Proposing a hybrid solution of a mobile learning system.

1.2 Goals and Objectives

This section describe the goals of this research contained in this dissertation, including specific, measurable objectives that must be attained in order to satisfy those goals.

Research Goals:

The aim of this research is to model, design, develop, and prototype a M-Learning system solution to be used in an ad-hoc network environment utilizing adaptive learning object mechanism with content privacy centric.

Research Objectives:

The emphasis objectives of our work are on defining appropriate mobile learning objects and their adaptation descriptions as well as providing mechanisms for ad-hoc service discovery and developing concepts for the seamless integration of the learning objects and their contents with a particular focus on preserving data and privacy.

1.3 Summary of Contributions

M-Learning is in high demand and is among the top 14 challenging research topics in the 21st century [1, 9, 46, 65]. Our research will shed light on how ad-hoc networking and wearable sensing technology can enable new forms of informal educational practices that actively engage student-centered learning to support anywhere, anytime learning. Our research may lead to technological advancements that will allow more personalized learning experiences and allow on-the-fly access to learning resources anytime and anywhere [37, 38]. Since our research is one of the first M-Learning works that focuses on ad-hoc network, it is expected to shed light on how emerging technologies can enable new forms of educational practices. This dissertation establishes the following novel contributions in the area of mobile learning:

1. Creation of an adaptive and reusable approach to building mobile learning objects
2. Modeling and Designing a M-Learning system using learning objects in an ad-hoc network environment

Prototyping our M-Learning system solution that works in a mobile learning hybrid wireless and ad-hoc network environments.

1.4 Scope and Limitations

The scope of this dissertation is confined to the investigation of modeling and designing an ad-hoc mobile learning platform. Although, we try to adhere to the IEEE standards of learning object, we neither promote specific learning management systems, nor do we suggest that the underlying platform used in this dissertation for mobile learning in ad-hoc network produces the best results. Rather, the modeling, designing, and developing M-Learning system presented in this dissertation, is one of the first of its kind, and it is the primary focus of the work. The use of mobile ad-hoc, wireless networks, and technologies along with concepts regarding learning objects are used to build the presented approach.

Furthermore, findings identified in this dissertation hold across the investigated learning management system projects. It may not be possible to extend the findings of the study involving other open-source software and database management languages to proprietary learning management system due to the different development practices adopted by current commercial entities. Further validations with both open-source and propriety learning management software systems are necessary to help us extend the research work to the next level

1.5 Outline of the Dissertation

The rest of this dissertation is organized as follows: Chapter 2 provides the background and related work on the problem under investigation. Chapter 3 focuses on the modeling and designing of an ad-hoc mobile learning system. We present the development of the mobile learning object authoring tool in chapter 4. The proof of concept of sharing a mobile learning object in infrastructure-less platform, used to demonstrate its potential uses is presented in Chapter 5. Chapter 6 concludes the research investigation and discusses future work.

CHAPTER 2: RELATED WORK

Research on e-learning was instruction focused proceeding along several directions including proposing generic e-learning models, content management systems, standards and tools, learning objects and ontologies. Learning management systems [2] were designed to manage learning content by maintaining and organizing context that could be delivered in various formats such as WBT, CBT, etc. The system allowed instructors to organize, assemble, approve, publish and deliver content. In general, these types of management systems are considered to be instructional centered in that they facilitate the instructor in managing and delivering instructional material by managing content. A Content Management System [2] (CMS) is a computer program that allows publishing, editing, modifying and centrally maintaining content. Such systems also provide procedures to manage workflow in a collaborative management. However their functionalities are limited in that they are not efficiently developed or interoperated. They don't support the transfer of content along different learning management systems nor can they create searchable learning content libraries or media repositories across different learning management platforms. The development of these learning systems was followed by the emergence of numerous learning tools, such as authoring, screen capturing, content packaging and meta-data editing, synchronous communication, etc. [2].

To support the sharing of learning modules across different platforms two standards were introduced: The Sharable Content Object Reference Model (SCORM) [5] and Learning Object Meta Data Model (LOM). SCORM provides a platform for developing, packaging and delivering high quality training materials for online training courses. It is composed of three modules: the aggregation model, run time environment and sequencing and navigation. The SCORM framework specifies how to develop and deploy content objects that can be shared and contextualized to suit the needs of the learner. It provides a tag content for later discovery and access in a distributed environment. Identification of an object and knowing where it resides are different things. SCORM is silent about how discovery and access is to be implemented.

Any digital resource that can be freely adapted and reused to mediate learning is defined as a learning object [11,12]. Learning objects are like Legos in that they are digital or non-digital entities that can be grouped and reused or referenced during technology supported learning. The functionality of learning objects is limited however because learning objects are not really interchangeable, have inconsistent classifications and therefore can't be found, may become obsolete sooner than anticipated and are rigid. Also learning objects are e-learning system dependent. Thus there are many barriers to using learning objects including the structure of their repositories, quality of learning objects in the repositories, their granularity as well as privacy, openness and sharing concerns. Learning objects are still an evolving definition. Clearly learning objects are not designed for ad-hoc networks and wearable learning.

Semantic knowledge representations across learning objects are described through ontologies. Ontologies and semantic repositories enhance learning material distribution and retrieval within learning management systems. The construction of ontologies is governed by the Resource Description Framework which provides for describing identified things. Semantic repositories use ontologies as sematic schemata to automatically reason about the queried data. They incorporate numerous data sources and provide a simple and quick querying mechanism against diverse or rich data schemata. Semantic repositories are highly effective for data interoperability but are ineffective for M-Learning applications because the semantic webs are not yet fully developed.

The largest body of research works in cyber-learning centers around e-learning management systems [2, 7, 8]. Most recently some efforts focused on M-Learning but most of the research is on instructional designs and focuses on fixed, infrastructure based wireless networks. Research on M-Learning focuses on proposing information services for M-Learning or 3-tier based system architectures [1]. For example, introducing an AXML data and web service integration architecture focuses on the application layer of M-Learning services and in particular on how to effectively find suitable services and how to efficiently use the services found. In particular, it uses the notion of a service ring for service discovery by grouping devices offering similar services to rings, where the notion of “similarity” is derived from the ontology. One member of the ring acts as the leader. Still this work is limited and relies on identifying group leaders who are assumed to be permanently stationed in a wireless environment. They proposed a three tier context-ware for pervasive e-learning where the 1st tier includes the mobile devices and intelligent agents, the 2nd tier contains the information stations and the 3rd tier is the information station center. This solution acts more like a client/proxy/server solution rather than a truly ad-hoc network. The Simple Object Access Protocol (SOAP) is also a peer-to-peer learning system but does not scale well for ad-hoc networks.

In sum, much of the research on cyber-learning focuses on e-learning systems. More recently some papers addressed the issue of M-Learning but are not comprehensive enough for use in an ad-hoc network platform. Furthermore, very few papers focus on algorithms to detect learner profiles and learning object references or “on-the-fly” learning objects.

2.1 Mobile “On-the-fly” Learning

Multiple definitions of the term M-Learning have been performed covering a wide variety of aspects. Some definitions identify the M-Learning as the mere evolution of e-learning, while others define it as an independent trend that is originated in the ubiquitous nature of the actual communication systems and identify M-Learning as location-independent and situation-independent [4]. We will interpret M-Learning as the latter definition.

M-learning supports mobility, collaboration, advanced personalized learning experiences, and on- the-fly interaction with adaptive feedback among users and real-world situations. It also provides access to learning modules, hence fostering the continued creation of a cumulative body of knowledge in formal or informal education. Anytime, anywhere, any device learning means that a student is not required to attend classes like in a traditional school. Instead of going to class or a library to gain knowledge, the student would be able to access learning modules from her or his computer or any other device such as a smartphone or a sensor [3].

2.2 Learning Objects (LOs)

A LO can include anything that has educational value – digital or non-digital such as a case study, film, simulation, audio, video, animation, graphic image, map, or book so long as the object can be contextualized by individual learners. The learner must be able to make meaningful connections between the LO and his or her experiences of knowledge he or she previously mastered. For the purpose of e-learning and M-Learning, a LO is digital in nature [5]. Because of its digital nature, the notion of platform independence (reusability, interoperability, and accessibility) becomes a vital issue. Reusability is the ability of the LO to be used over and over in different instructional contexts. Reusability of a LO depends upon interoperability and accessibility. Interoperability is the ability of the LO to function in various environments regardless of the platform. Accessibility is the ability of the learning object to be accessed by learners in any location regardless of the learner experience, or the type of platform the learner uses. The notion of platform independence requires following a set of standards. These standards contribute to reusability, interoperability, and accessibility [5]. A few widely used examples of these standards are Institute of Electrical and Electronics Engineers (IEEE) Learning Object Metadata (LOM) and Advanced Distributed Learning Initiative (ADL)'s Sharable Content Object Reference Model (SCORM).

2.2.1 Learning Object in E-Learning

2.2.1.1 LOM

In 2002, the IEEE Learning Technology Standard Committee proposed a LOM standard (IEEE 1484.12.1-2002 LOM v1.0) to provide a unified description of LO resources. By using LOM, LOs can be retrieved and acquired easily and precisely among various LMS. The LOM is now serving as the principal standard internationally to specify LOs. LOM is mainly composed of nine categories: general, life cycle, meta-metadata, technical, education, rights, relation, annotation, and classification. These categories are used to annotate LOs in a comprehensive manner. Each category has its own subcategories and specific vocabularies to describe LOs in detail [6].

2.2.1.2 SCORM

In 1997, the United States Department of Defense and the White House Office of Science and Technology Policy launched the ADL. The ADL initiative aims to establish a new distributed learning environment that permits the interoperability of learning tools and course content on a global scale. As a foundation for accomplishing those goals, ADL’s SCORM aims to foster the creation of reusable learning content as ‘instructional objects’˙I within a common technical frame-work for computer-based and Web-based learning. SCORM was developed to solve these problems: move a Web-based course from one LMS to a different LMS, reuse Web-based learning content in different Web-based courses, and define the logical flow sequence of Web-based learning activities in an interoperable, repeatable way. SCORM is defined in four books: the Overview Book, the Run-Time Environment Book, the Content Aggregation Model Book, and the Sequencing and Navigation Book. The Overview Book describes the model as a whole, and the other three books describe specific capabilities in technical detail [2].

2.2.1.3 Project Tin Can

Project Tin Can, also created by ADL, was the first step toward supporting the next generation of SCORM. The Tin Can API is merely one of several components of the next generation of SCORM. The first major effort of the Tin Can API was to determine the different types of verbs and activities appropriate for learning as adopted from the activity stream specification actor-verb-activity or I-did-this model. The Tin Can API can be thought of as the communication component of the next generation of SCORM just as the Run-Time Environment was the communication component of legacy SCORM. In addition, communities of practice are expected to enforce certain behaviors and taxonomies by creating their own vocabulary profiles. A virtual world training community may define hundreds of verbs that relate to the large context of training in another world [7].

2.2.2 Learning Objects in M-Learning

A LO can include anything that has educational value – digital or non-digital such as a case study, film, simulation, audio, video, animation, graphic image, map, or book so long as the object can be contextualized by individual learners. The learner must be able to make meaningful connections between the LO and his or her experiences or knowledge he or she previously mastered. For the purpose of e-learning and m-learning, a LO is digital in nature [5]. Because of its digital nature, the notion of platform independence (reusability, interoperability, and accessibility) becomes a vital issue. Reusability is the ability of the LO to be used over and over in different instructional contexts. Reusability of a LO depends upon interoperability and accessibility. Interoperability is the ability of the LO to function in various environments regardless of the platform. Accessibility is the ability of the learning object to be accessed by learners in any location regardless of the learner experience, or the type of platform the learner uses. The notion of platform independence requires following a set of standards. These standards contribute to reusability, interoperability, and accessibility [5]. A few widely used examples of these standards are Institute of Electrical and Electronics Engineers (IEEE) Learning Object Metadata (LOM) and Advanced Distributed Learning Initiative (ADL)'s Sharable Content Object Reference Model (SCORM).

2.3 Authoring Tools and LMS

2.3.1 MASLO

The Mobile Access to Supplemental LOs system is an open-source tool for m-learning developed by the Academic Advanced Distributed Learning Co-Lab at the University of Wisconsin. Content authors use a desktop or laptop computer running Microsoft Windows (Microsoft, Redmond, WA, USA) and Mac OS X (Apple, Cupertino, CA, USA), or Linux to create or curate materials by forming content packs. These packs can contain text, image, audio, video, or simple quiz content. The system includes an automatic check on file size to help ensure that users are aware when they are creating a pack that might be too big to download over a 3G connection, or even just bulky enough that it will take up an inordinate amount of space on a user's device. After creating a content pack, the learning expert can preview the pack and then publish it to a cloud-based server solution. Once a content pack is posted to the cloud, content creators or content managers have options depending on their desired use of the system and how it's been configured. Finally, after content has been posted to the cloud and any subsequent review or pricing is complete, learners can download content packs to their iOS or Android device by simply selecting a pack they want to retrieve from the content store. Learners can then access their content packs on or offline in any location they choose to [8].

2.3.2 Mobilogue

Mobilogue is an M-Learning tool for authoring and deploying learning content with location awareness and guidance on mobile devices. The application covers informal learning settings like field trips and museum visits, as well as formal classroom settings.

Physical locations are associated with QR codes that are scanned by the application and used to obtain supporting learning content stored in a remote server [9].

2.3.3 GoMo Learning

GoMo Learning is a commercial content authoring tool for multi-device content delivery by utilizing Web technologies (hypertext markup language (HTML), JavaScript, and cascading style sheets (CSS). It supports a collaborative and Web-based creation of content that can be delivered as an HTML package or a native application for different devices. GoMo-created scenarios do not support location-based content delivery [9].

2.3.4 Treasure HIT

Treasure-HIT is a research project that works on a treasure hunt game mechanics. The system offers an authoring environment for teachers to create station-based activities for GPS-identified locations with associated content, tasks, and feedback for the learners. These games are shared through a repository and can be run on mobile phones in the field. Treasure-HIT also uses Web technologies for content visualization on the devices, thus making it platform-independent [9].

2.4 Peer-to-Peer and Ad-hoc Networking

Recent hardware advances in embedded mobile devices such as mobile phones have enabled us to envision new applications and services based on the context and the collaboration between the mobile devices and fixed infrastructures available around us. Technologies for peer-to-peer networking include fixed and wireless network technologies, for example, Bluetooth, wireless local area network, WiMax, general packet radio service, and Universal Mobile Telecommunications System, allowing devices both to be mobile and to interconnect with mobile or fixed peer-to-peer infrastructures [10]. The capability for ad hoc wireless device-to-device communication between ubiquitous mobile devices such as smartphones and laptop computers opens up the possibility of exciting new types of networking applications. From a practical technological point of view, the capabilities of short-range communication devices over unlicensed links are improving exponentially, in terms of performance measures such as cost, power consumption, throughput, and spectral efficiency. The throughput available on a short-range wireless communication channel operating on an unlicensed band is often far greater than what is available over a long-distance wired communication channel, and is, moreover, free to the user. It is thus a compelling engineering problem to design a self-contained wireless network whose communication resources consist exclusively of short-range unlicensed communication links. This has led to many proposals for wireless ad hoc network designs [11].

2.4.1 Bluetooth Based E-Learning

The Bluetooth based e-learning system proposed by Zhang et al. uses Bluetooth technology as the communication method. The system was developed using the Java API for Bluetooth Wireless Technology on Java platform, making it operating system-independent and device-independent. To be secure, the system uses both authentication and encryption to ensure safety during communications. In order to support the short range of Bluetooth (roughly 10 m), when a client is out of the serving area of the server, it needs to connect to the server via the help of other clients using ad hoc on-demand distance vector routing. In this case, the client will send connect-to-server request to its neighboring clients, which then forward the request to their neighbors. Upon receiving such a request, the server sends back a route reply to the client and the intermediate clients will forward the reply to the requesting client. This way, the client will be able to connect to the server. If multiple replies are received via different routes, the route with minimum hop count will be selected for further communication [12]. Using Bluetooth to do similar work in an M-Learning environment would be useful and takes advantage of the mobile hardware technologies.

2.4.2 Proem

Proem is an open computing platform that provides a complete solution for developing and deploying peer-to-peer applications for mobile ad hoc networks. Its objectives included versatility, interoperability, platform independence, built-in security, and high-level development support [13]. Proem is currently no longer developed, but its objectives remain similar to the features our work aims to include.

2.4.3 Mobile Chedar

Mobile Chedar (CHEap Distributed ARchitecture) is an extension to the Chedar peer-to-peer network allowing mobile devices to access the Chedar network and also to communicate with other Mobile Chedar peers. Chedar is a peer-to-peer middleware designed for peer-to-peer applications. Mobile Chedar is implemented using Java 2 Micro Edition (Oracle, Redwood Shores, CA, USA) and uses Bluetooth as a transmission technology for connecting to other peers. Current Bluetooth implementations have the restriction that nodes can be connected to only one piconet at a time. Therefore, the only topology available for constructing Bluetooth network is star-shaped. One device functions as a master and others as slaves (hybrid architecture) [10].

2.4.4 Peer2Me

Peer2Me is an open-source project developing a framework for mobile collaborative applications on mobile phones. Peer2Me enables developers to create collaborative applications for mobile phones using network technologies such as Bluetooth while also using a hybrid peer- to-peer model. The architecture and concepts of Peer2Me are independent of the kind of personal area network (PAN) technology supported in the mobile device. The project was initiated to enable rapid development of proximity-based peer-to-peer applications for mobile devices on the J2ME platform [10].

Mobile Chedar and Peer2Me use the J2ME platform. However, J2ME is being phased out as more and more mobile devices move on to different operating systems, therefore making these systems outdated.

While there have been plenty of e-learning and M-Learning tools and implementations developed, most solutions still fall into a niche situation and are not generic enough to be taken out of the contexts for the applications they were designed for. Instead, we choose to provide an alternative to these systems, one that can take and improve upon the strengths of the previous mentioned methods and combine them into one system, while addressing and hopefully eliminating the weaknesses in those systems previously discussed.

CHAPTER 3: MODELING, DESIGNING MOBILE LEARNING SYSTEM FOR AD-HOC NETWORKS

3.1 Introduction

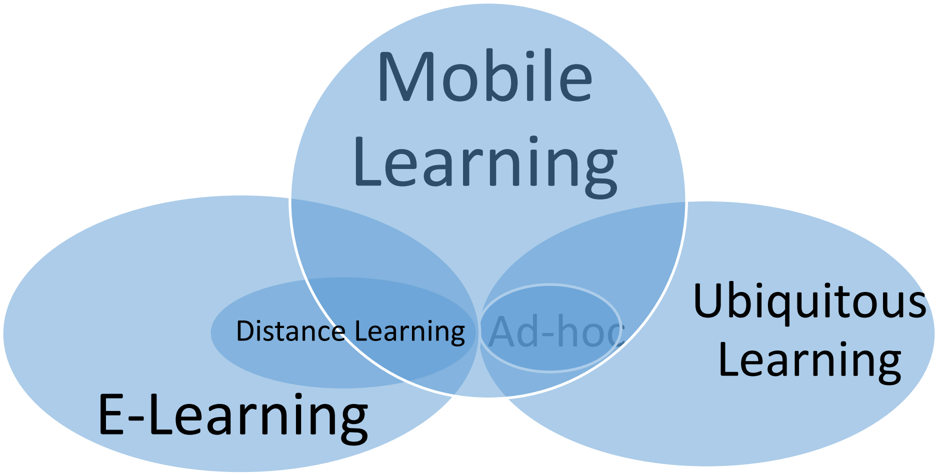


Figure 3.1: Relationship between different types of cyber learning

Figure 3.1 above depicts mobile learning as the harmonic convergence of e-learning and ubiquitous learning. It addresses the current elements of cyber education systems driven by current technological advances. E-learning has enabled many learners to access virtual classroom environments thus creating what is commonly referred to as distance learning. Ubiquitous learning is a separate but similar environment that is still being debated but nonetheless allows users to learn anytime and anywhere using any device that uses sensory technology or constantly collects data that is forwarded to its host. The mature technology of mobile learning allows the advancement of both e-learning and u-learning by combining elements from both environments and allowing learners to access and collect information anytime, anywhere via network infrastructures and mobile wireless ad-hoc networks. Mobile learning using ad-hoc networks is still a developmental field. This chapter further discusses modeling and designing mobile learning systems for ad-hoc networks.

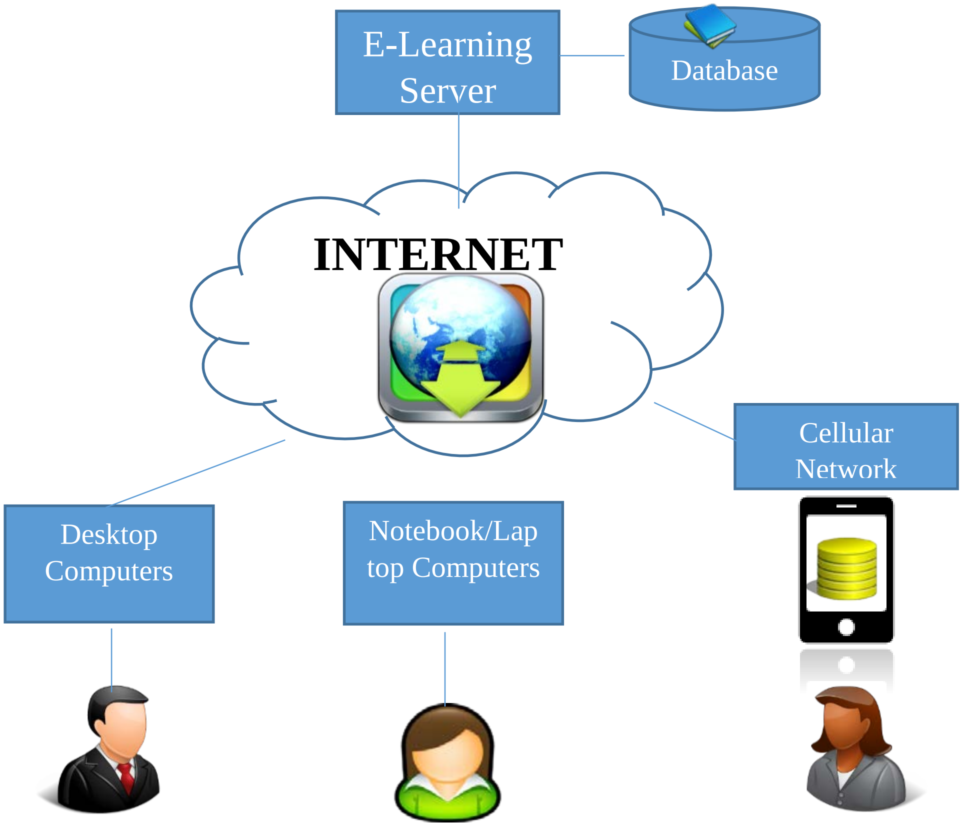


Figure 3.2: General Model of Cyber Learning

3.2 M-Learning System Model in Mobile Wireless and Ad-hoc Networks

Figure 3.3 is a general learning system model that depicts how users with access to the internet connect to traditional e-learning servers. Mobile learning is considered a non-formal extension of the traditional formal structured learning environment. Typical e-learners use desktop or laptops with internet connections to access a web based classroom environment or learning applications. Thanks to the advancement of smart-phones and touch technologies, users are now able to access the same e-learning environments on the fly. However, these applications still depend heavily upon e-learning system network infrastructures for functionality. For example, users of current smart-phone learning applications can only use said applications if they are connected to a wireless networks or a cellular network. If a user is in a remote area or out of range of these connections the application will render itself useless. The incorporation of mobile ad-hoc networks into the current learning systems would enable true on the fly learning independent of cellular or wireless network connectivity. Figure 3.2 is also a conceptual design model for the hybrid mobile learning framework which can be used to understand at a high level of how mobile users can use their mobile devices such as mobile phones or tablets to exchange mobile learning objects in the absent of the wireless structure network. In this framework, we also include the capability of acquiring sensor data to enhance the learning experience. The mobile users will communicate among others via mobile wireless ad-hoc networks, possible using Bluetooth connection. This peer-to-peer network connection allow mobile users to connect, collaborate and exchange information to those closed distance.

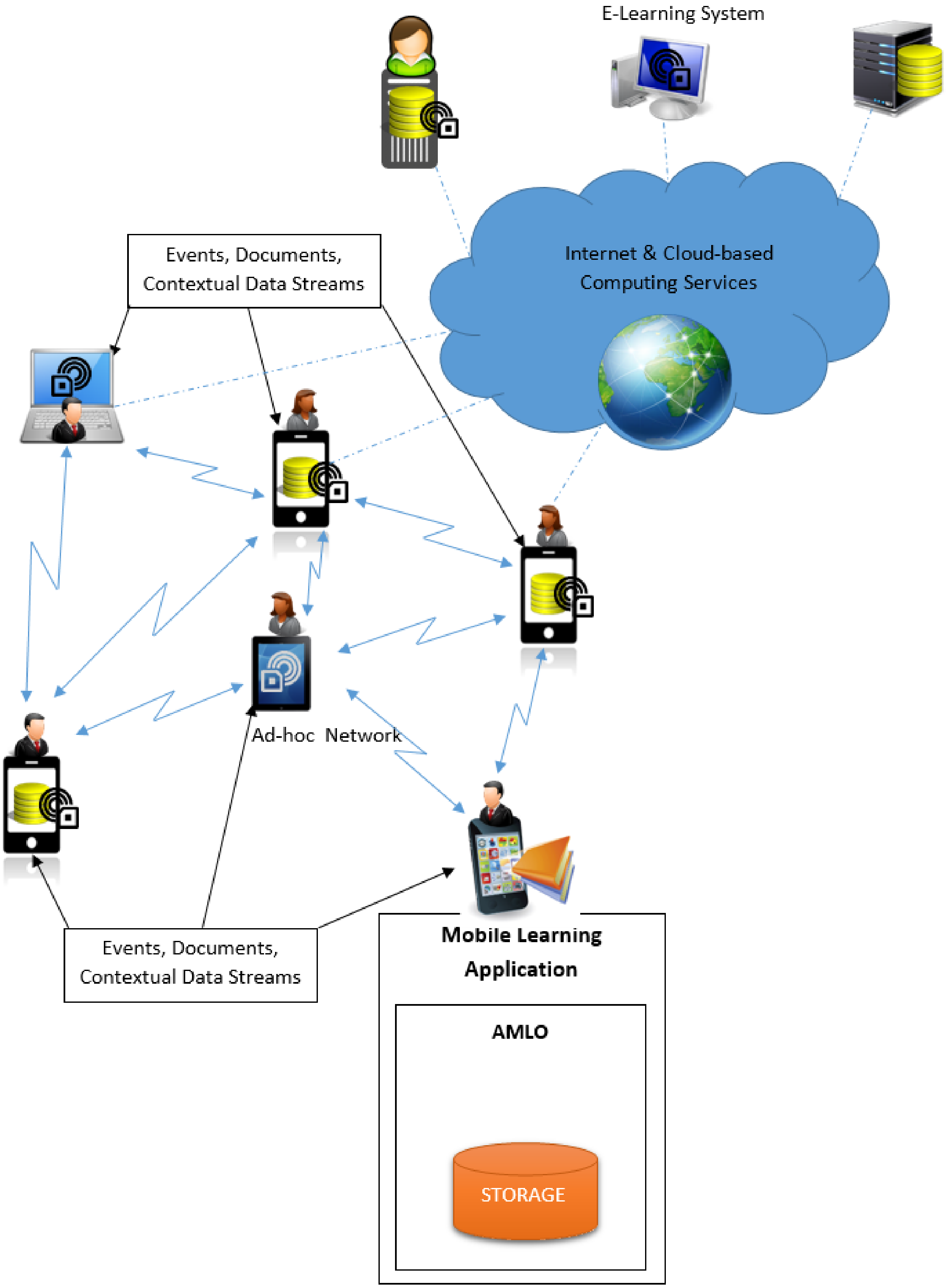

E-Learning System
Events, Documents, Contextual Data Streams
Internet & Cloud-based Computing Services
Ad-hoc Network
Events, Documents, Contextual Data Streams
Mobile Learning Application
AMLO
STORAGE

Figure 3.3: Abstraction of the mobile ad-hoc learning platform

3.3 Conceptual Design for the Adaptive Mobile Learning System Architecture

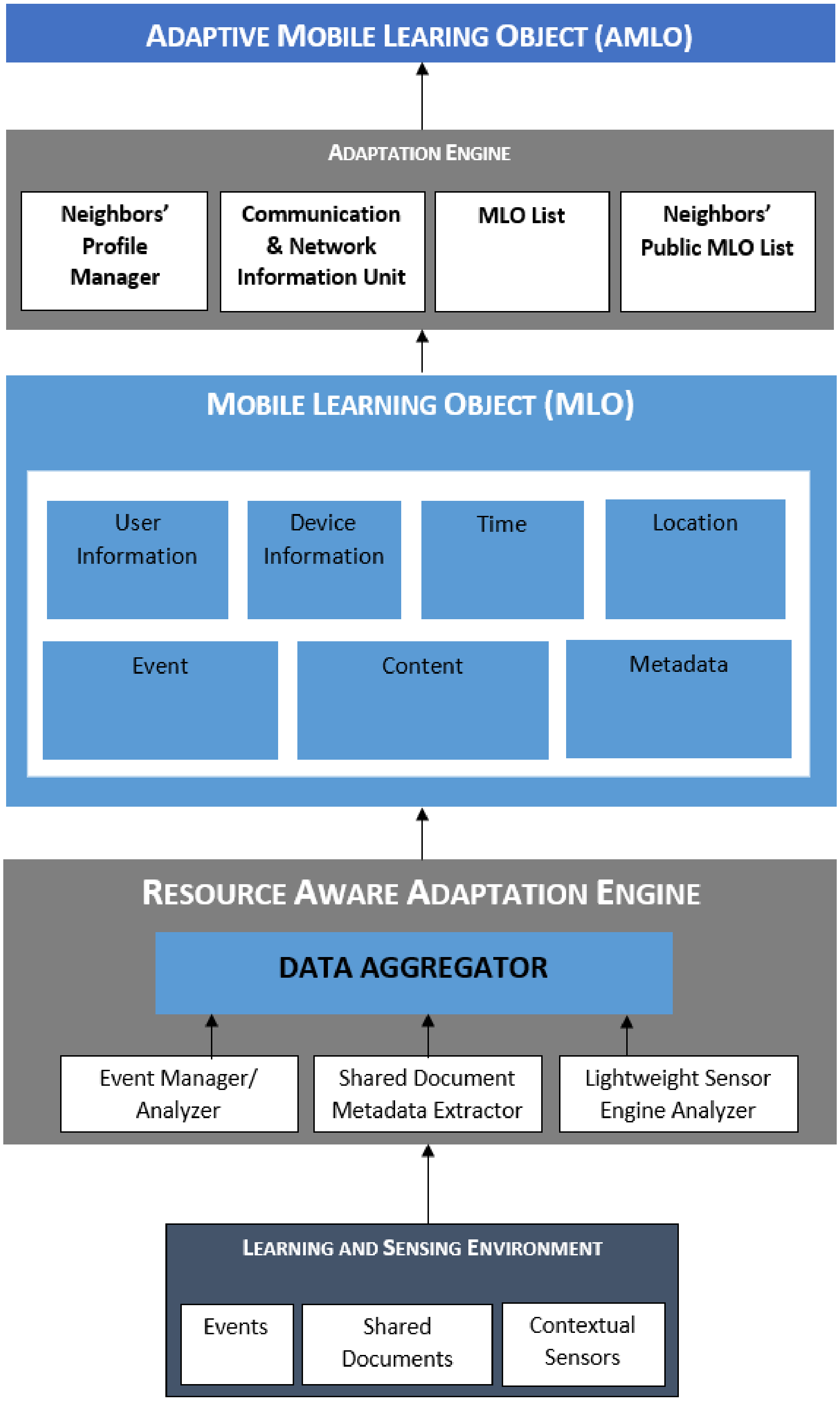

ADAPTIVE MOBILE LEARING OBJECT (AMLO)
ADAPTATION ENGINE
Neighbors' Profile Manager
Communication & Network Information Unit
MLO List
Neighbors' Public MLO List
MOBILE LEARNING OBJECT (MLO)
User Information
Device Information
Time
Location
Event
Content
Metadata
RESOURCE AWARE ADAPTATION ENGINE
DATA AGGREGATOR
Event Manager/ Analyzer
Shared Document Metadata Extractor
Lightweight Sensor Engine Analyzer
LEARNING AND SENSING ENVIRONMENT
Events
Shared Documents
Contextual Sensors

Figure 3.4: High Level Mobile Learning System Architecture

The proposed Mobile Learning System (MLS) Architecture given in figure 3.4 above is a high level abstraction of an adaptive context-aware mobile learning system architecture that supports our “on-the-fly” learning concept. The system is also to be used as a platform in a blended approached in an informal infrastructure and infrastructureless networks to allow mobile users to connect and share information via modules at anytime, anywhere. In a typical scenario, MLS is used to support just some basic learning and sharing when mobile users do not have access to the internet via WiFi or cellular networks but still require a temporary solution to connect and share learning material with their neighbor mobile devices. The MLS is not intended to replace or to support all mobile learning activities in the modern technological advancement of teaching and learning environment.

MLS is designed and organized as a mobile learning application that allow connectivity between mobile ad-hoc nodes as well as traditional client-server e-learning and M-Learning systems. Mobile users use their mobile devices to connect, communicate, and collaborate with their neighbor mobile devices via mobile learning objects. In this system, mobile learner role is considered dual role as a learning object consumer and producer dynamically. This concept is much different than the traditional mobile learning where users play a role of teach and students statically. Since we proposed the concept solution for mobile learning on the fly where learner use the ad-hoc network as communication architecture to allow them to consume and produce knowledge and information via mobile learning object anywhere, anytime.

Layers of the System

The MLS is comprised of several layers such as Learning and Sensing Environment (LSE), Resource Aware Adaptation Engine (RAAE), Mobile Learning Object (MLO), Adaptation Engine (AE), and the Adaptive Mobile Learning Object (AMLO). Each layer serves the layer above it and is served by the layer below it. The bottom layer is used to receive input from learning data, trigger events, and the contextual sensors. The learning data is any file or document that users would like to include to the learning object. An event is a command, or query request from the mobile users/owner to tell others what they want to do with the learning data. The event can be as simple as a command to request information about a particular learning object such as place, time, location or it can be as complex as the combination of several queries. The contextual sensors are those sensors that collect basic data and aggregate into a simple stream of input data to the MLS such as temperature, space, and time. Input data can be events, shared documents, or raw sensor data from the contextual sensors and it is treated as the feed-in data to the next layer called resource aware application engine. This layer consists of three components which are event manager/event analyzer, shared document metadata extractor, and lightweight sensor engine analyzer.

The event manager fetches the incoming events (where the event are either in the form of external source or internal source) and then places them into the corresponding event module at the above layer. In the typical ad-hoc network environment, mobile devices are faced with limited resources. Therefore, we propose to use the lightweight event processing method as describe in the algorithm below and in figure 3.5.

Lightweight Event Processing Algorithm:

Step 1: Lightweight Event Processing Unit (LEPU) receives a simple event from the event sources

Step 2: The LEPU checks against the predefined allowable event list.

Step 3: If it is in the list, then process this event in step 4.

If it is not in the list, then do not process this event and go back to wait for the next incoming event in step 1.

Step 4: Add the metadata of this event to the metadata unit.

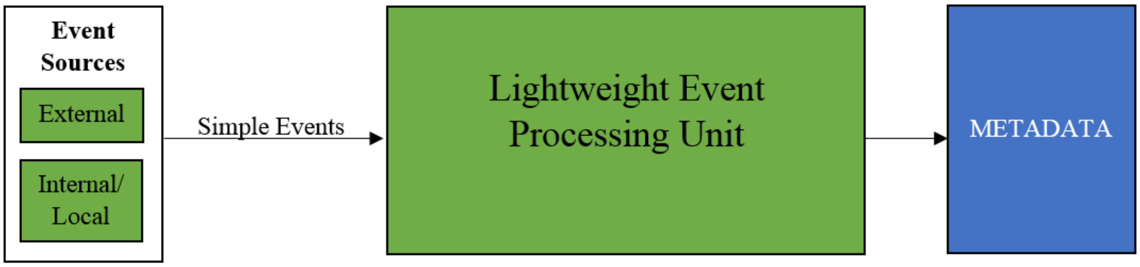


Figure 3.5: Simple Event Processing Architecture

If the mobile device has the connection to the cloud which gives it a much more capabilities for processing complex events, then we can use the complex event processing architecture (see figure 3.6). Since the complex event processing unit (CEPU) is being processed on the cloud-based servers, we are able to take advantage of a more powerful distributed system that can be used to process those complex events in order to output the metadata which can be consumed by the next layer of MLS. The following is the proposed conceptual complex events processing algorithm.

CEPU Algorithm:

Step 1: CEPU receives complex events from the event sources

Step 2: The CEPU break those incoming complex events into simple events

Step 3: Check each simple event against the feasible list.

If there is a match, then include that event into the selected event list to be to process.

If there is no match, then store it to the pending approval list to be checked later in order to be added to the feasible list.

Step 4: Select the feasible events to be processed.

Step 5: Add the metadata of these feasible events to the metadata unit.

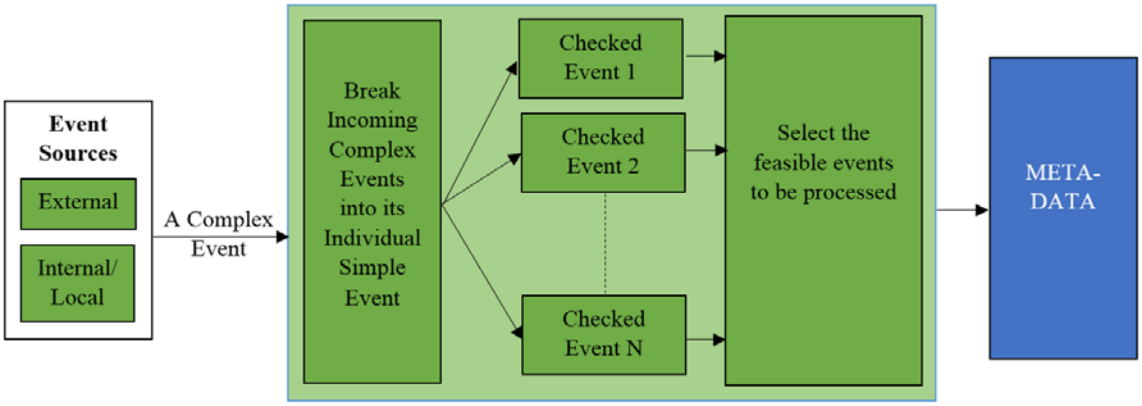


Figure 3.6: Complex Event Processing Architecture

The second component of this layer is called document metadata extractor which is responsible to organize and classify the document property in two clearly defined areas called document content and document metadata. Document metadata is the data about the document such as title, subject, author/creator, keywords, description, date created, etc. There are two common methods to extract metadata, these are support vector machine

(SVM) and Conditional Random Fields (CRF) which they normally use either the rule-based approach or the machine-learning approach to quickly read and extract the metadata needed [74]. Our proposed M-Learning solution is primary for the ad-hoc network environment, we suggest to use the lightweight metadata extraction using rule-based approach due to the limitation of the ad-hoc networks as shown in figure 3.7. The suggested general extraction algorithm used in this document is given below:

Step 1: User selects a file or a document from the local storage of the mobile device

Step 2: Checks the file format or the file extension

Step 3: Document/OCR processing is done via the rule-based technique

Step 4: Output the metadata in a text file or XML file.

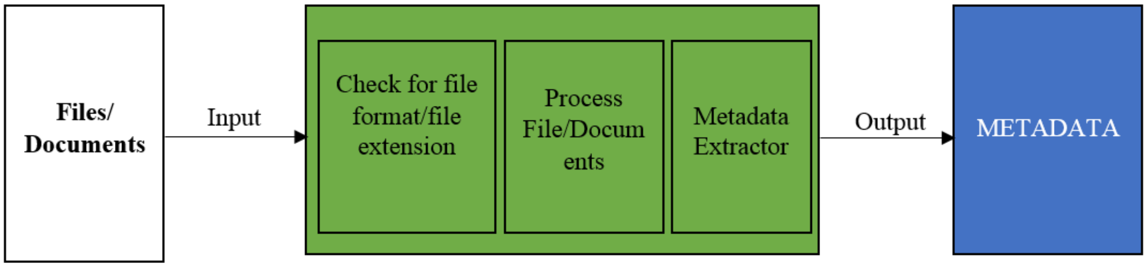


Figure 3.7: Ad-hoc System Metadata Extractor Architecture

In case if there is a possibility to connect to cloud, we suggest to use either the automated template-based or the rule-based for the hybrid solution in figure 3.8 since metadata extraction process can be quickly executed on the servers. This way, the document processing and metadata extractor process would be done externally at the server side to save processing power and battery life for the mobile devices which has limited resources compare to the more powerful and rich resources from the server. These solutions have already been used for the automatic document extraction using support vector machines or conditional random fields. There are existing metadata tools that are currently available online such as Wget, EXIFtool, Metagoofil, Meltego, Meta-extractor which can be utilized to extract the document's metadata.

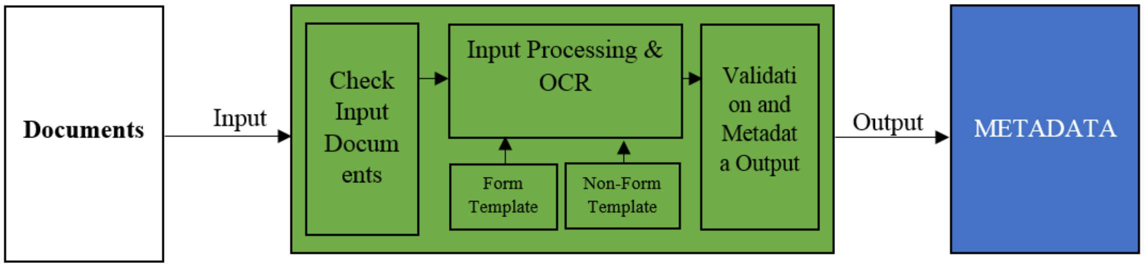


Figure 3.8: Metadata Extractor Architecture

The third sub-component is called lightweight sensor data stream analyzer that is supposed to receive basic sensor data as an input in figure 3.9. Then it goes through the necessary steps to be able to process and extract the input sensor data and output as the metadata to feed into the next layer/component above. Now the metadata aggregator analyzes metadata to aggregate and distribute the received metadata to the mobile learning object level to include necessary information in those dimension of the mobile learning object. Once it is analyzed, then it is placed into the corresponding module like content, metadata, users, device, time, location.

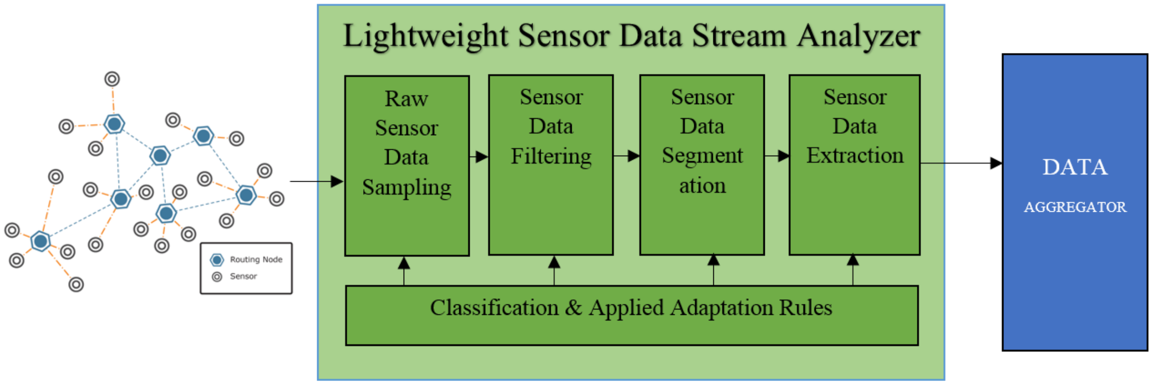

Routing Node
Sensor
Lightweight Sensor Data Stream Analyzer
Raw Sensor Data Sampling
Sensor Data Filtering
Sensor Data Segment ation
Sensor Data Extraction
Classification & Applied Adaptation Rules
DATA
AGGREGATOR

Figure 3.9: Lightweight Sensor Data Stream Analyzer Architecture

The data output from the analyzer/extractor will be fed into the metadata aggregator to create a well-structured metadata which includes those adaptable categories such as user information, device information, time, location, event, actual data content, and metadata of the mobile learning object as in figure 3.10. The category list can be updated to exclude or include more category names depending on the users need.

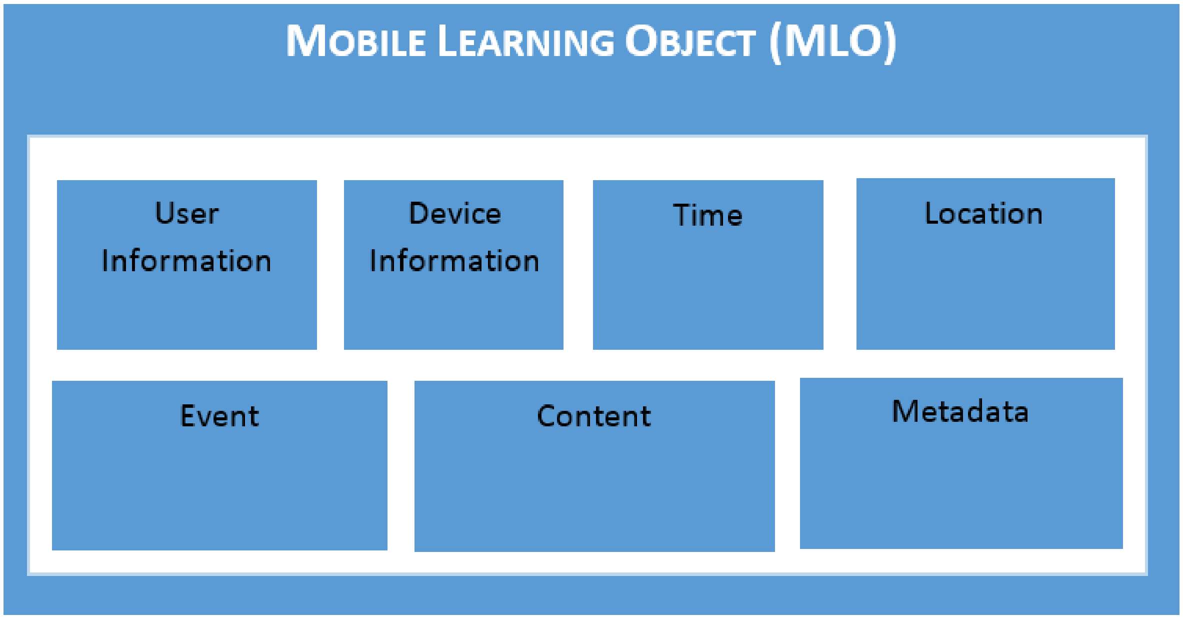


Figure 3.10: Mobile Learning Object Structure

The layer above the MLO layer is the adaptation engine layer which consists of the following four components:

1. Neighbors' Profile Manager (NPM),
2. Communication and Network Information (CNI) ,
3. MLO List
4. Neighbors' Public MLO List.

Neighbors' Profile Manager contains the information about the currently connected neighbors. Also this component keeps track of the previously searched keywords from its neighbors. This information will be helpful when a neighbor user can quickly obtain information about this user so that they can share the related MLO. Communication and Network Information component contains the network related information about the current mobile node (such as the number of nodes it is currently associated to, or the current level of its battery life, its computation capability, and its storage limitation.) The MLO list contains the list of one or more public MLOs which will be used to create a new object called Adaptive Mobile Learning Object (AMLO). Neighbors' Public MLO List contains the public MLO list of the current neighbors of the mobile node. This adaptation mechanism is very useful because it shares the public MLO list of its neighbors with the intended mobile nodes which will use those information for their benefit. The output of the adaptation engine layer is the adaptive mobile learning object which will be used for sending and receiving information between the connected mobile devices.

3.4 Technical Design Details of the Adaptive Mobile Learning Object and its Hybrid Mobile Learning System

At the core level this system is comprised of mobile devices with capabilities of communication via Bluetooth, WiFi or GPRS/GSM Cellular networks. Our focus solution in this M-Learning system is mobile devices interconnected via ad-hoc network. Therefore, we suggest to use the common mobile ad-hoc architecture using Bluetooth where ad-hoc network can be formed anywhere, anytime with data rate that can be up to 2 Mbps among the nodes within distance of 30 to 100 feet. Once the connection is established, the data will be transferred among the nodes in the binary object format called Adaptive Mobile Learning Objects.

The underlying concept of this mobile learning object in ad-hoc network environment is to efficiently share and reuse the MLOs among the mobile devices instead of transferring files/documents. Typically, file sharing technique may involve compressed files which can be transferred over the wire or wireless medium from sender to receiver. This method can be inefficient due to high overhead bandwidth if mobile users keep sending heavy load of documents. For example, mobile device A asks mobile device B for a particular file such as “Lecture notes of Java in 2015”. Mobile device B immediately packs all the related documents into the compressed file and then send back that huge file to A. This common method is working well in the case that both mobile devices are using traditional wireless network and have full processing power as well as strong battery power. However, in the ad-hoc network environment, mobile devices have limited resources. In our approach, both sender and receiver devices are able to communicate directly via the binary AMLO objects. The lightweight transmission of AMLOs reduces the overhead bandwidth requirements of traditional file transfer system which in turn enhances the connectivity between mobile devices and at the same time will increase the transmission efficiency.

In addition, our work has been extended to incorporate the hybrid solution which allows mobile devices to connect and share reusable AMLOs via the cloud based storage system. The storage system solution used in this research is a high-performance and scalable document-based database (NoSQL) which is designed to be well-adapted to the high demand in mobile wireless ad-hoc networks.

Figure 3.11 below depicts the adaptive mobile learning object architecture for our proposed system. This system consists of a communication module, an adaptive mobile learning object management module, an adaptive mobile learning object module, and a storage module.

The communication and networking module is the connection between mobile devices such as Bluetooth, WiFi, and cellular network. The adaptive mobile learning object management module processes and manages AMLO requests. This module consists of five individual components which are Sync Data, Query Request/Response Unit, Pairing Bluetooth Devices Unit, Paired Neighbor Devices Unit, and Searching and Ranking AMLO for adaptation purposes.

The sync data is a gateway which facilitates bidirectional, multi-master sync, mobile object replication protocol, and authorizes AMLO access. When an AMLO is shared through paired devices and then the connection is no longer available, the sync data from the sender can be automatically synced to the cloud-based server namely Couchbase.

However, as soon as the receiver mobile device gets connected back to the Couchbase server, the data will be synced to it automatically.

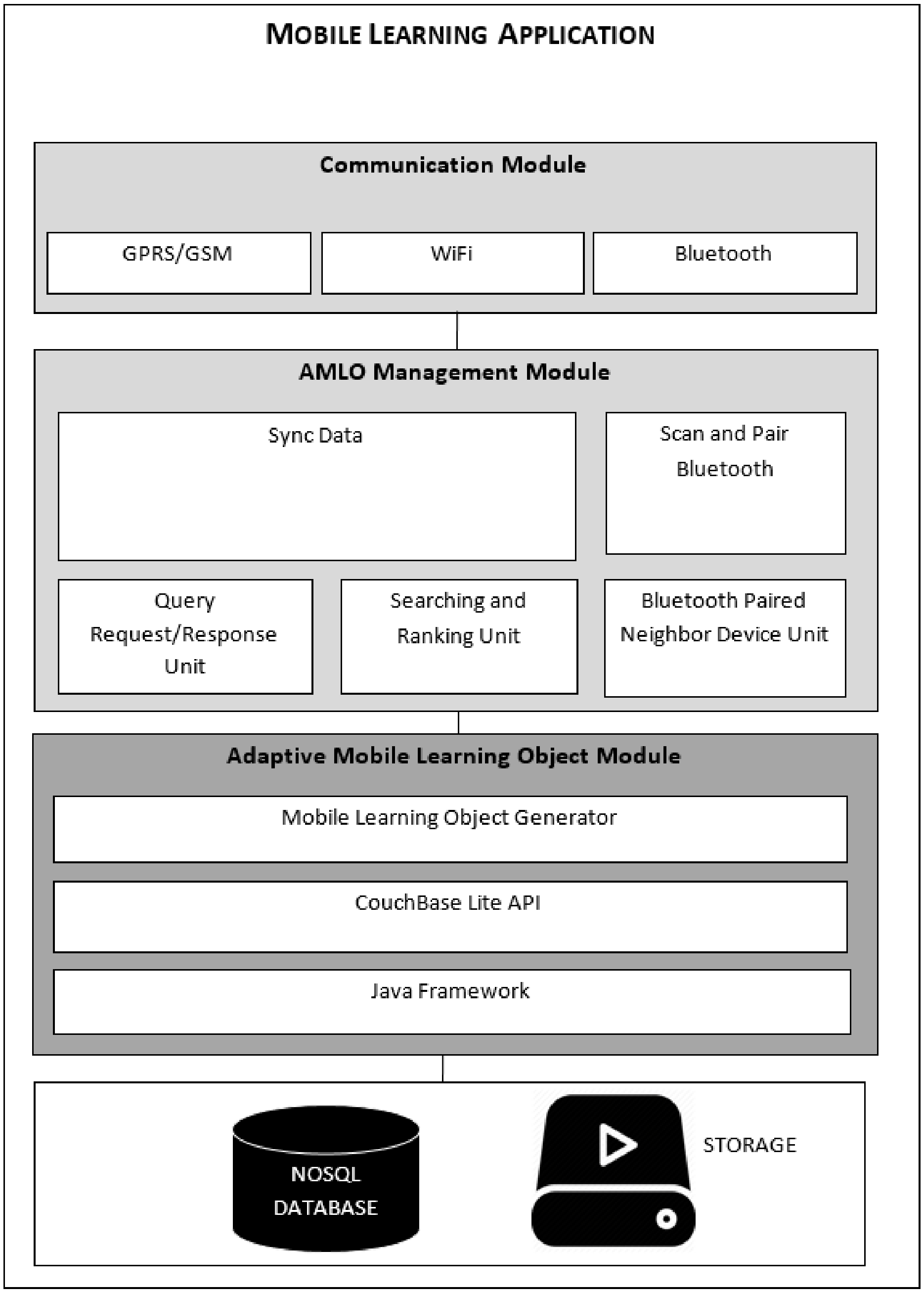

MOBILE LEARNING APPLICATION
Communication Module
GPRS/GSM
WiFi
Bluetooth
AMLO Management Module
Sync Data
Scan and Pair Bluetooth
Query Request/Response Unit
Searching and Ranking Unit
Bluetooth Paired Neighbor Device Unit
Adaptive Mobile Learning Object Module
Mobile Learning Object Generator
CouchBase Lite API
Java Framework
NOSQL DATABASE
STORAGE

Figure 3.11: Adaptive Mobile Learning Application Design Architecture

The query request and response unit receives request as an AMLO format and then processes the request and provides the response which is also in the AMLO format. The searching and ranking unit provides search history tracking services for both sender and receiver devices which also keeps track of searched keywords and rank them based on their usages. Bluetooth Paired Neighbor Devices Unit keeps the list of all the paired and connected devices.

Our proposed solution addresses the security and privacy issue regarding AMLO by creating three access levels which are classified as private, public, and protected. The private access rights does not allow any neighbor devices to search and query but the public access rights allows neighbors to access all public AMLOs. The protected access rights allows mobile hosts to create group level access among specified connected mobile devices.

The adaptive mobile learning object module consists of a AMLO generator, Couchbase Lite Application Programming Interface, and the Java programming framework. These components allow the application to create and manage AMLOs. The Couchbase Lite is an embedded JSON database that can work standalone, within a peer to peer network, or as a remote endpoint for Couchbase servers for synchronization purposes.

The storage module is a NoSQL database where AMLOs data file paths are stored.

Figure 3.12 below depicts our Mobile Learning Object design architecture consisting of three layers. The first layer is the data input layer which serves as resources for the next layer. The resources are devices, sensors, commands and files. The second layer is the Learning Object Generator in which the data input analyzer takes information from the resources in the data input layer and verifies the security access rights. After verifying the accessibility, it then creates a learning object that will be converted into a JSON data format via the Couchbase lite API. Once data is in the JSON format it can be sent to the storage in which a user can choose to store it locally on their mobile devices or save it onto the Couchbase server for offline synchronization to paired devices.

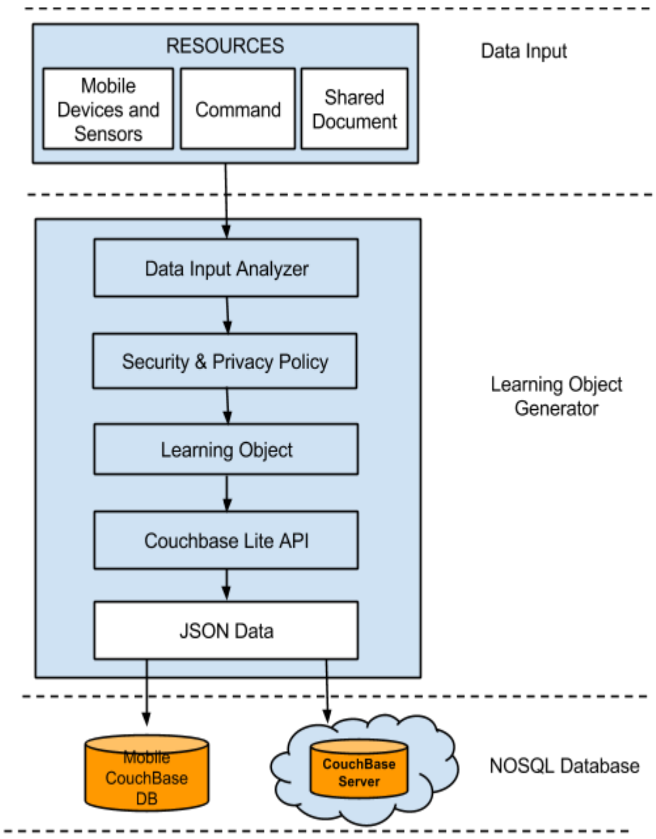


Figure 3.12 Mobile Learning Object Design Architecture

Figure 3.13 depicts the Adaptive Mobile Learning Object design architecture. The creation of MLO and AMLO upon first glance are similar due to the similar functions of the three layers. The AMLO design architecture is more efficient because AMLOs are created to adapt. In the data input layer the learning object data is the MLO that was described in figure 3.12. The learning object generator layer has 4 main components. Query analyzer which analyzes the learning object data input, then execute the query. Once the result is completed, the next step will be to build the adaptive mobile learning object that writes the data results to the JSON object and then output the AMLO onto the third layer.

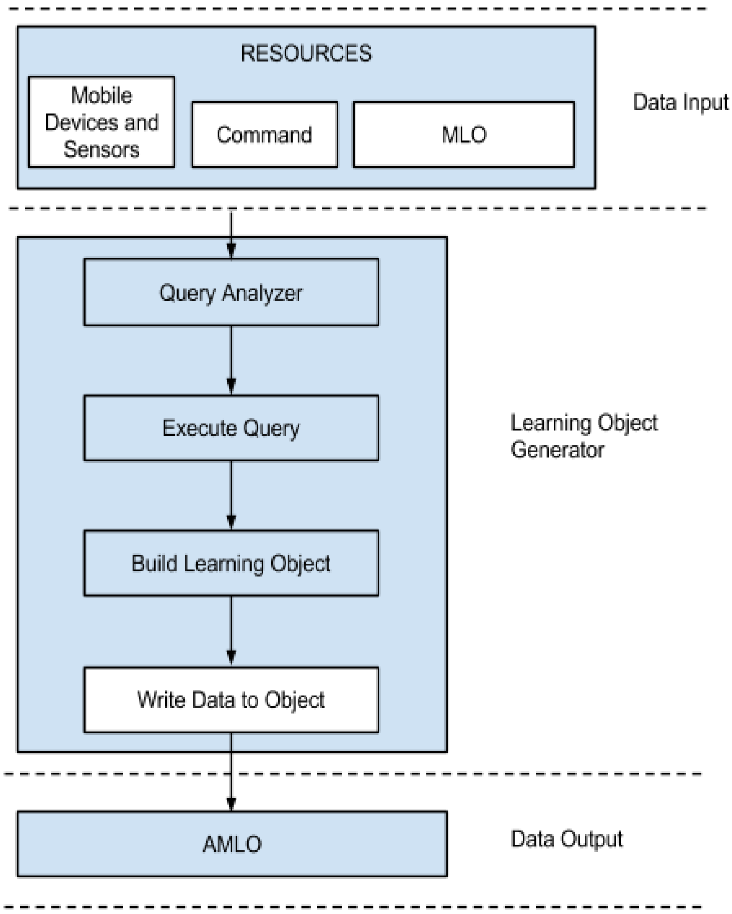


Figure 3.13 Adaptive Mobile Learning Object (AMLO)

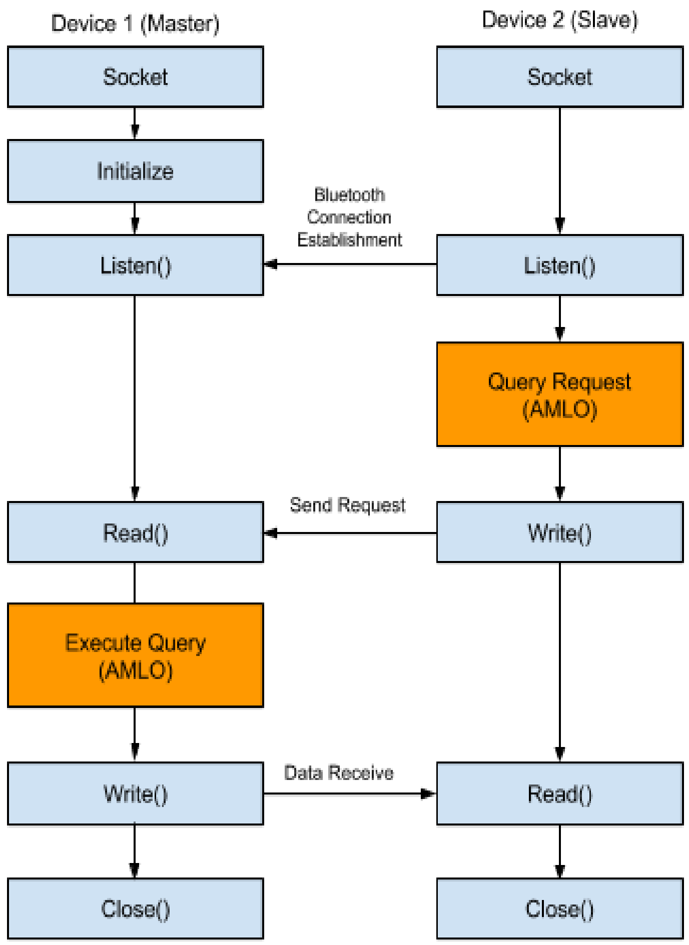


Figure 3.14 Learning Object Data Communication Architecture

Figure 3.14 is the learning object data communication object architecture which is based on the standard bluetooth architecture. One mobile device plays the role of “master” and the other mobile device functions as a “slave” to the master device. Both devices must activate their bluetooth capabilities to initiate communication and pairing. Once the connection is established the master device listens for requests from other slave mobile devices. The slave device generates a learning object request and then sends the query request in the format of an AMLO. The master device receives the AMLO request and executes a query AMLO and responds to the slave device with the results of the query in the AMLO format through which the slave device is granted access rights to the results. Once the AMLO is transferred the process starts over and the slave is able to request another AMLO from the master until the devices are disconnected or are out of range.

Figure 3.15 below depicts our proposed system design model for a hybrid mobile learning ad-hoc and wireless network architecture. This model addresses the ability to transfer mobile learning objects over ad-hoc network and additionally an option to transfer those objects over the Internet to share the reusable mobile learning object to improve the system communication efficiency.

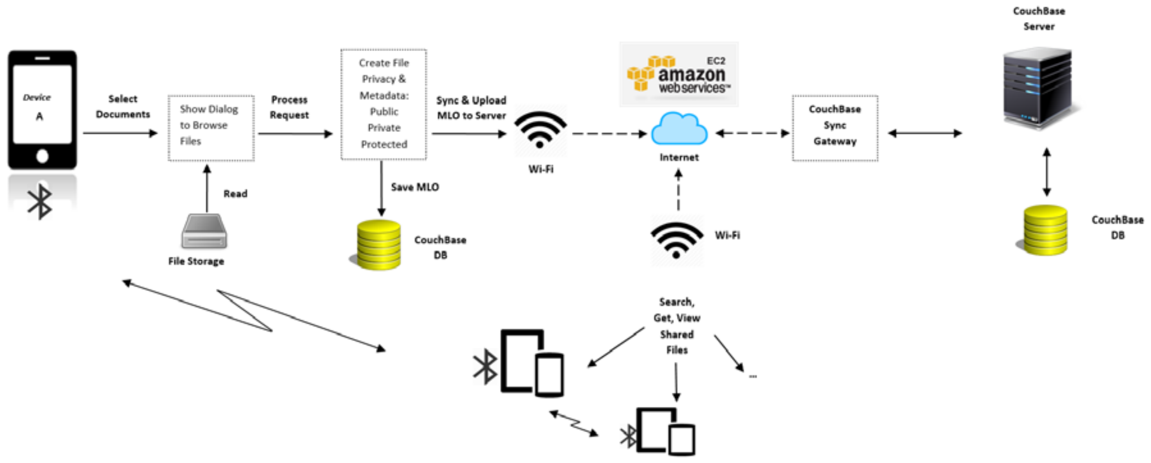


Figure 3.15: System Design Model for the Hybrid Mobile Learning Ad-hoc and Wireless Network Architecture

3.5 Conclusions

In conclusion, throughout this chapter we present a novel mobile learning design model for mobile ad-hoc and wireless networks. Mobile learning is the harmonic convergence of e-learning and ubiquotous learning. The unique concept of our solution compared to the standard mobile learning solution is not only the adaptive mobile learning object but also design to work in mobile ad-hoc environment. The incorporation of mobile ad-hoc networks into the current learning systems enables true on the fly learning independed of cellular or wireless network connectivity. The transmission of AMLO via bluetooth increases efficiency of the typical file transfer process. The lightweight transmission of AMLOs reduces the overhead bandwidth of traditional file trasfer mediums and enhances connectivity between mobile devices in order to make it faster for mobile users to transfer files over the mobile ad-hoc networks. The hybrid solution allows mobile devices to connect and share reusable AMLOs via the cloud-based storage system on a scalable document-based database. Our design incorporates an off line synchronization feature using the Couchbase sync gateway that allows mobile devices to synchronize to the NoSQL server. This high performance document-based database is well adapted to the demand of mobile ad-hoc and wirless networks environments where mobile users are constantly surrounded by the heavy traffic of information and constantly on the move. One of the unique features of our learning object designed solution is the ability to modify access rights on AMLOs into the subcategories of private, public or protected. Thus, it is improving the security and privacy of the file transfer process among mobile device users.

CHAPTER 4: ENABLE ON-THE-FLY LEARNING FOR MOBILE AD-HOC WIRELESS NETWORKS USING BAYESIAN THEORY AND MOBILE LEARNING OBJECT AUTHORING TOOL

The 21st century student is a digitally savvy student who depends extensively on mobile technologies for everyday life and learning. This student dependence on technology has fostered "learning mediated by networked computing and communications technologies where the forms of steering of learning can arise in a hybrid manner from a variety of personal, educational or collective sources and designs " referred to as "cyber-learning" [8]. Cyber-learning encompasses many forms of learning including e-learning [8] and M-Learning [7-9].

M-learning uses mobile devices, sensors and wireless transmissions [7] to provide anytime and anywhere access to content as well as just in-time training or review of content. It delivers any-time, anywhere device learning (which we fondly refer to as "education in your palm" [4, 9] and empowers informal science education. M-learning supports mobility, collaboration, advanced personalized learning experiences and on-the-fly interaction with adaptive feedback among users and real-world situations [11]. It also provides access to learning modules, hence fostering the continued creation of a cumulative body of knowledge in formal or informal education. Anytime, anywhere, any device learning means that a student is not required to attend classes like in a traditional school. Instead of coming to class or a library to gain knowledge, the student would be able to access learning modules from their computer or any other device such as a smart phone or a sensor. These devices may be attached to a high performance fixed or wireless network, ad-hoc networks, wireless sensor networks or even wearable sensors. Under these situations, students would need to find suitable modules and combine them freely.

Furthermore, students will be more successful and more likely to stay on task if they have the opportunity to cooperate and communicate with fellow students. While this can easily be achieved in a traditional classroom setting, it is far more challenging to provide adequate means of this support in a teaching environment as the one described here [9]. To realize the vision we must foster emerging technologies and develop new collaborative modes, media richness and virtual words which can facilitate and support inter-operability between heterogeneous local and specialized systems [9, 10].

M-learning is immature. Much of the research is theoretical in nature [6]. This research articulates what mobile and conceptual learning is all about, but not how to foster emerging technologies, methodologies and systems to design, develop and implement learning systems to advance mobile learning and establish interoperability between otherwise heterogeneous local and specialized e-learning systems and new collaborative modes, media richness, and virtual worlds. Since M-Learning can occur "anytime, anywhere, any-device, any-platform" with or without the predetermined locations, existing cyber-learning systems are ineffective due to many technological challenges associated with ad-hoc or peer-to-peer networking and moving object databases. This includes limitations posed by the inherent characteristics of wireless systems such as transmission interferences, bandwidth, limited resources as well as content inconsistencies of sharable content objects (a.k.a. learning objects) because of rapid updates (here today, outdated tomorrow) and semantic knowledge representation among learning objects and others. The follow section outlines a methodology for this type of learning.

4.1 On-the-fly Learning

Let us have a closer look at the requirements for a future anytime, anywhere learning environment: anytime learning means that the student is not required to attend classes like in a traditional school system. To gain the required knowledge at the student's convenience, it should be possible to access different kinds of learning information at any time and from any location. These teaching modules could be video-taped lectures from the student's university, different institutions, online reference materials, tutorials for specific subjects, virtual labs, environmental or health care sensor data streams, and so on. Thus, in order to support anytime learning, a wide variety of teaching modules need to be available on-line or discoverable on the spot. Students need to be able to find appropriate modules and combine them freely. To allow this, a description schema for modules needs to be developed, modules from different institutions need to be made compatible and combinable, and payment methods for different user groups need to be developed.

Anywhere learning means that the student is not required to come to a classroom or a library in order to obtain knowledge. It should instead be possible for the student to access learning modules from their computer at home or work, but also to use specialized equipment at a university lab (i.e., a computer that offers high quality virtual reality environments) when necessary or even to quickly look up some information using a smart phone. The computers used may be attached to a high performance fixed network or may have a wireless connection to a network. Obviously, the amount and kind of data that can be stored, processed, and displayed will depend on the computer and the connection used. However, from a user's point of view it is highly desirable to have access to similar interfaces independent of the machine being used.

Also, information about the students' progress, modules they downloaded, and so on, should be available regardless of which machine they use. In order to support anywhere learning [4, 9], the teaching environment needs to be based on a reliable network that provides a high quality of service for users in both wired and wireless networks, and mechanisms to ensure consistency and availability of information. Furthermore, as experience shows, students will be more successful and likely to stay on task if they have the opportunity to cooperate and communicate with other students. While this can easily be achieved in a traditional classroom setting, it is far more challenging to provide adequate means of this support in a teaching environment as the one described here. Both means to locate fellow students and technical support for cooperation among groups of students need to be provided.

Consider, for example, a group of students deciding to meet to prepare for their final exam. The students will bring their laptop computers to their meeting point, which may or may not provide network access to some or all of them. If no network access is available, it should be possible to download important information on their laptops beforehand. If the student's computers can communicate via wireless connections with one another at the meeting, it is not necessary to download each piece of information on each of the machines. Rather, it suffices for each piece of information to be available on just one of the inter-connected machines at certain locations [12].

Thus, what is needed here is a kind of semantic, cooperative, preemptive caching. It should be obvious that the resolution of these issues will not only enable the development of a wireless learning and teaching environment, but will have tremendous impact on a number of other areas as well. Wireless network management, for example, is needed wherever wireless networks are used. The same is true for sophisticated methods to support work through disconnections and for mechanisms to support groups of wireless users. Tele-Geo monitoring queries have important applications in areas as diverse as fleet management and environmental control. Even though wireless networks coupled with high performance hardware, such as optical fiber and high bandwidth gateways, are widely used in today's cyber-learning systems, these capabilities must be complemented with corresponding system advances for ad-hoc networks to obtain a real benefit. Without this capability we cannot gain access to more data or improve access to the type and quality of information needed for decision making that characterizes M-Learning. This includes mobility, context awareness [1, 2, 3, 5], collaboration, advanced personalization and access to learning modules possibly without the use of a solid networking infrastructure.

Thus, the core theme of our research emphasizes M-Learning solutions for ad-hoc and sensor networks. Currently, mobile learning systems are depended on the underlining infrastructures, wired and wireless networks that use the inter-networking technologies as a multi-tier system dependent to allow mobile learners to access and retrieve the learning materials. Therefore, they are lacking of M-Learning solutions that allow mobile learners to connect, collaborate and learning via the mobile ad-hoc network where no network infrastructures are available. Our research aims at designing a cutting-edge ad-hoc networking and wearable sensor learning platform to support cooperative information sharing and learning in informal settings. In particular, we will focus on developing novel protocols to describe, compare, and combine learning modules and mechanisms in infrastructure-less networks. Our work will center on creating and managing the content to be used by the learning application for an ad-hoc network. This includes on-the-fly content identification and creation as well as updating and removal of old or obsolete material. Support for cooperative information access in ad-hoc networks will include methods to describe, compare, combine and share learning modules and mechanisms.

4.2 Exploiting Bayesian Theory to Support On-the-fly M-Learning

This section delves into the application of the Bayesian theory to support high volume, event-based, in-network and on-deterministic pervasive learning object management solution. First, an argument for providing learning objects management functionality based on classical probability is presented. Then we introduce the conceptual system architecture that support the global learning object sharable that is discoverable and in ad-hoc networks

4.2.1 Motivation Scenario

Widespread adoption and use of Internet-connected, mobile, ‘smart’ devices has paved the way for a technologically enhanced culture in which users can consume content anytime and anywhere. This ‘Information at Your Fingertips’ culture is further propagated with mobile-centric websites, search engines, and applications that help users view, create, and share content with others around the world regardless of technical prowess and with little-to-no setup. Tech start-ups, major companies, and educational institutions have taken notice, and many have created apps and user experiences for the mobile-connected user. For years, the Internet and computing technology has been used as an ideal tool for educators and learners to exchange knowledge and build skills. The use of electronic media and/or technology for educational purposes can be categorized under the term electronic learning, or ‘e-learning’ [1]. A learning management system (LMS) is often used by educators to facilitate the delivery, management, and tracking of e-learning resources with their students [2]. Typically, an e-learning system sets out to mimic or enhance a traditional classroom environment. Depending on the course design, e-learning can be carried out asynchronously or synchronously and makes distance learning possible as long as the user has access to the e-learning resources, commonly via Internet. E-learning can also enable and enrich collaborative learning. With the rapid growth and spread of Internet-enabled mobile devices came the emergence of mobile learning (M-Learning). M-learning is mobile-centric; it focuses on the user's ability to be mobile in time and space and can take advantage of common mobile device hardware features such as camera, location tracking, and gyroscope data. M-learning enables users to participate in less-traditional, informal learning, where learners do not necessarily follow a classroom model or module-based learning pattern.

Mobile learning can add an educational context to any physical environment. With the use of ad hoc networks and mobile authoring tools, we can create an 'on-the-fly' [3] learning scenario, where educators and learners can create, share, and view content from their mobile devices without the need for server–client or infrastructure-based liaisons. The goal is to present a new perspective to M-Learning, learning objects (LOs), and content sharing and management in an m- learning system. We will discuss a lightweight system and the creation of a leaner LO format as well as the integration of ad hoc networks into an M-Learning environment.

In this example, we consider an outdoor informal learning happen around a historical lake such as Hoan Kiem lake in Viet Nam. Around the lake there are several points of interests to numerous people including historians, tourists and teachers.

Therefore, it is possible that a teacher may take her students to the lake for field trip. Also, scientists are conducting experiments and collecting soil and water samples as well as transmitting their findings remotely in real time. In addition, a group of tourists is taking pictures and learning more about the history of this lake area. In the lake there are animas such as turtles, birds etc. The teacher, scientists and tourists all have interest in learning more about this historical lake in a per need basis. Since there are may be an overlap in the material each group is interested in and given that each group wants to acquire the information in real-time and on the spot, the groups can use their cell-phones to exchange and share information. For example, the teacher may receive some environmental facts from the scientists to share with her students. Therefore, one of the possible informal learning ways is using their mobile devices to create and/or join to an ad-hoc network to be able to achieve on-the-fly learning.

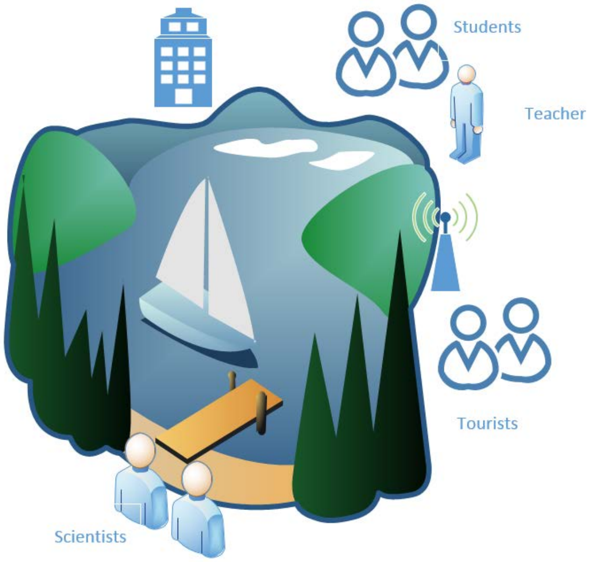


Figure 4.1 On-the-fly Learning Environment around Hoan Kiem lake

The problem explored in this work was to design an architecture and associated mechanisms that support or plug into a mobile, wireless ad-hoc learning environment to support applications such as the one described in this motivating scenario. In what follows we outline these mechanisms.

4.2.2 Architecture of Mobile Wireless Ad-hoc Learning

Our proposed solution is to use and share mobile learning objects among ad-hoc nodes using a lightweight repository system that installed on each node with the ability of ranking and comparing learning object mechanism [12,13]. To achieve this we apply the Bayesian theorem to generate a learning object that its neighbor node maybe interested in. Bayesian theorem is vastly used in science to determine the probability of events. It is a valid measurement in scientific research and learning to verify from theory approaches to sampling marginal and conditional distributions. In addition, Bayesian theorem links the degree of belief to calculate the probability that are causally ‘inverse’, meaning capable to determine the probability of an earlier event given another learning object event is known to have occurred later on in time. Therefore the Bayesian theorem is not only theoretically correct but practical usable in our approach to determine learning object event management. From the perspective of learning object management, each mobile node can receive a high volume of learning object events from it surrounding nodes; however, each node only share a computational global learning object sharable (GLOS) that is publicly available to it neighbor nodes to share.

4.2.3 Conceptual System Architecture

Each mobile device (MD) installs a lightweight application called Mobile Learning App (MLA) that is responsible for preprocessing the entire incoming data stream from surrounding nodes. Data streams can be any kind of data communication and are not limited to learning objects. A mobile node searches for all the learning objects from other nodes that are part of the ad-hoc networks by broadcasting a learning object request. Other nodes receive the request then compare it to their own learning object repository that marks sharable. Then it applies the Bayesian theorem belief to generate a global learning object sharable (GLOS). They reply to the sender with their GLOS and the sender will again use the belief method to decide which learning objects the sender wants to use. (Ad-hoc routing; trust among nodes and that all other ad-hoc network fundamental issues are not the focus of this research). Our primary goal is to identify how to use learning objects concept to solve mobile learning in an ad-hoc network environment.

Each mobile node has one or more learning objects. The learning object format is aligned with the IEEE 1484.12.1 Learning Object Metadata (LOM) [5] that widely adopted since 2002. In the following, we outline the high level description of our algorithm:

Step 1: A Node broadcasts to all other nodes in the ad-hoc network its GLOS that include the specific information to search for any similar learning objects.

Step 2: Each neighboring node that receives the broadcasted request compares the broadcasted GLOS with its learning objects residing in its local repository and calculates the relationship among the broadcasted GLOS with the learning objects residing in its local repository.

Step 3: If a relationship exists between the broadcasted GLOS and local learning objects, the neighboring node generates its own GLOS and sends it to the broadcasting node.

Step 4: When the broadcasting node receives the GLOS' from the neighboring nodes, the broadcast nodes decides which GLOS is interested in and sends a share request to the neighboring node that is associated with the GLOS it is interested.

Table 4.1 High level description of our algorithm

In the above algorithm we outline the process between an ad-hoc node (client) and the nodes participating in the ad-hoc network (viz., Neighboring nodes). To compare relationships among learning objects for step 2 we use the ranking metrics and search guidance for learning object repository methodology used by [12]. To generate a learning object for steps 3, we use Bayesian Theorem. In the next section we describe how to use the Bayesian Theorem to generate a global learning object sharable (GLOS) by an ad-hoc node.

4.2.4 Generating GLOS

One of the advantages of Bayesian theorem is the ability to determine probabilities that are causally 'inverse" meaning it is possible to determine the probability of a particular learning object event that happen earlier given that another event is known to have occurred in a later time. For example, the probability of a downloaded learning object that was not completed from the GLOS of a given neighbor node. From the perspective of a network management view, this is a conditional probability to determine an incompleteness of a particular learning object from a given node. Let we denote E is the download event from a GLOS of a given node. PP is the probable problem from one of the learning objects.

Therefore, the probability P of occurrence of the probable problem PPi given that the event E has already occurred via downloading from the GLOS is given by this formula:

$$P(PP_i\,|E) = \frac{P(E \cap PP_i)}{\sum_j P(E \cap PP_i)}$$

The value on the right hand side of the above equation can be obtained via the counting method experimentally. As sample size j, which denotes number of entities at which correlation between E, and PPi is known increases, conditional probability that E is due to PPi will be more accurate.

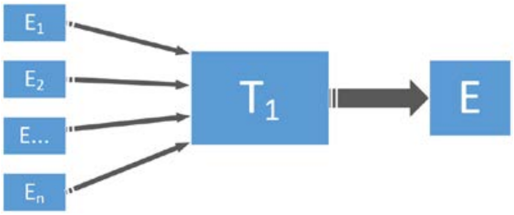


Figure 4.2 Correlation Transformation of Events to GLOS.

Given correlation transformation T_i, we can consider each of the incoming LO of the transformation T is a discrete event E and the output result is a master list of those learning objects which as an input event E for the neighbor node. In theory, we assume that the transformation T is working well and there is no losing any of the incoming E_{1-n}. However the assumption does not hold true in reality when a particular E_{1-n} which is denoted e_j, is missing or corrupted while downloading due to the nature character of ad-hoc network. In this case, we use Bayesian theory correlation transformations T_i, to calculate the conditional probabilities $P(E|e_j)$ for each ej and whenever all ej are known, $P(E|e_j) = P(E)$. Therefore,

$P(PP_i\,|E) = \frac{P(E \cap PP_i)}{\sum_j P(E \cap PP_i)}$ now can be rewritten

As $P(PP_i\ |e_j) = \frac{P(PP_i \cap e_j)}{P(e_j)}$

Because of ej may not contribute to E for some value of j, we have:

$$P(PP_i\ |e_j) = \frac{P\big(PP_i \cap E \cap e_j\big) + P(PP_i \cap E' \cap e_j)}{P(e_j)}$$

Applying Bayes theorem again, we have:

$$P(PP_i\ |e_j) = P(PP_i\ |E \cap e_j) \times P(E\ |e_j) + P(PP_i\ |E' \cap e_j) \times P(E\ '|e_j)$$

If the transformation is designed correctly to be able to detect those ej, meaning we can ignore those events and the equation above now become:

$$P(PP_i\ |e_j) = P(PP_i\ |E) \times P(E\ |e_j) + P(PP_i\ |E') \times P(E\ '|e_j)$$

The methodology presented here is suitable for generating global learning objects that can be shared based on the probability of a particular learning object event that happened earlier given that another event is known to have occurred at a later time. We are currently exploring the simulation environment whereby our ad-hoc network consists of a random collection of participating nodes which relay packets of fixed size.

Mobile learning can augment formal education and bridge the gap between formal and informal education by creating extended learning communities using any digital technology in connected or infrastructure-less environments. With the use of ad hoc networks and mobile authoring tools, we can now create an ‘on-the-fly’ learning scenario, where learners can create, share, and view content from their mobile devices without the need for server–client or infrastructure-based liaisons. Mobile learning, however, is not just about sharing content or learning using mobile, wireless, and portable devices. Rather, it is learning across spatiotemporal contexts that enables learners to form knowledge and understanding in different scenarios. When facing limited resources stemming from the characteristics of mobility and wireless technologies, determining the best practices for content creation and delivery becomes a challenge. This paper describes the architecture of a mobile-focused learning network designed so that learning tools, activities, contexts, and interactions are created as necessary over time and space while adhering to traditional learning object standards.

4.3 Methodology

4.3.1 Approach

The IEEE defines a LO as ‘any entity, digital or non-digital, that may be used for learning, education or training’ [14]. LOs are like building blocks, in that they are digital or non-digital entities that can be grouped and reused or referenced during technology-supported learning. Unfortunately, LO schemas today are too bulky and work as building blocks only for the systems they are created for. A LO made for one LMS is in most cases not portable to another LMS that uses a different schema or M-Learning object management process designed for that system. They also contain large amounts of metadata that are used to manage the object according to some learning pedagogical approach. This makes current LO designs and authoring tools system and learning approach biased. Approaches to LO implementation today are moving away from the building block concept and more towards packages, limiting their interchangeability and portability as well as making them much larger than they need to be. For e-learning-based learning systems, this is no problem as most LOs are hosted in the cloud or local servers that are made to withstand large file sizes and streaming. The majority of M-Learning systems in use today are simply extensions of e-learning systems. Like the desktop/Web-based approach, the mobile device acts as a viewer of the LO and its material. But because of the device form factor, users are limited to being a spectator, and editing and sharing abilities are limited as these systems are not truly designed for native M-Learning. This does not work, for a truly native M-Learning experience should be designed for the mobile work flow. Mobile devices are not designed to handle and manage large files, nor should they, so simply porting current LO and management systems to a mobile platform will not simply suffice.

We decided that it is necessary to take LO development back to its original roots and keep LOs as close as possible to lean, simple components to be used and packaged by a larger system. Taking cues from how users interact with their mobile devices, we also decided that the learning system in which M-Learning objects are created may also need a new approach. Many user expectations in mobile systems are slightly different from those non-mobile platforms. Users expect to be able to search for and retrieve information quickly; time spent trying to break up the information they receive into the parts they need is something users try to keep at a minimum. We also take cues from how learners interact with each other. Take the following anecdote: person A asks person B for directions to a place of interest. Person B does not know where that place of interest is but instead offers to ask person C and person D if they might know, as they are more knowledgeable of such things. Person C and Person D recommend two different ways to the place of interest, and Person B passes back that information to person A. What was just described here is an ad hoc network information exchange and is the way many people naturally inquire and gather information related to their current needs. This gathering of information is a form of learning, and we recognize that with the ubiquity of mobile devices in our lives, the same ad hoc inquiry situation can be used in an M-Learning environment, where now the persons interact with each other through their mobile devices in order to obtain LOs and information.

An ideal informal M-Learning situation should be as portable as the mobile devices it uses. When creating a system that allows informal learning, we want to enhance if not encourage collaboration and sharing in learner–learner and learner–educator relationships. For learning scenarios made ‘on- the-fly’, infrastructure-based networking is not necessary and in fact may slow down the learning process. Instead, embedding ad hoc network capability not only mimics the ‘on-the-fly’ social nature of the learners but can also reduce setup and transfer times, enabling content exchanging to be as natural as social exchanges. Therefore, we propose an alternative solution to M-Learning. An ad hoc integrated, mobile native, LO authoring and management system.

4.3.2 Network Structure

Our network approach will be supported both by an ad hoc as well as infrastructure-based components but will not be dependent on the infrastructure-based component. The ad hoc network component is the main means through which user devices communicate and exchange data in a peer-to-peer manner. This allows for information exchange without needing to be online or connected to the Web and is convenient for quick, ad hoc learning sections. Infrastructure-based communications are more suitable and convenient for more formal scenarios.

Most ‘on-the-fly’ learning situations do not require more than an ad hoc scenario. Ideal situations for an ad hoc M-Learning situation from traditional learning scenarios would be a field trip or study, group study sections, lab sections, and other one-shot or repetitive scenarios.

User devices send queries to other devices nearby or in the same WiFi network in order to share LOs with each other quickly and in an offline manner. User devices may be able to connect to the Internet directly or through other Internet-connected devices via a bridge to obtain LOs stored in the cloud or a familiar server.

Learning objects created during these sections can be kept locally on the device or uploaded to servers or the cloud afterwards for archiving and can be referenced and shared at a later time. They can also be ported to different viewers and/or learning systems for other kinds of tertiary management or lesson plan creation. Because of the building block nature of M-Learning objects, a plethora of learning systems and plans can be made as they are almost atomic in nature and very flexible.

4.3.3 Lightweight System

Mobile devices are powerful tools but face many limitations that non mobile devices do not have, have overcome, or are not pivotal. Battery life, network or data bandwidth, internal storage, memory, and processing speed are factors that greatly vary from mobile device to mobile device. It would not be smart to assume that a cellular phone can do heavy data downloading and processing comparable to that of a desktop without taking a major hit to its system resources. This means that when making a cross-platform application, we must be as flexible as possible to maintain usability and functionality.

Therefore, our system design was made with careful consideration of being as light as possible with system resources.

4.3.4 Mobile Learning Object (mLO)

Mobile devices have limited, most times, unexpandable internal storage. Data compression technologies and cloud-based storage have made this hurdle easier to manage on mobile devices; however, transferring data from device to device in an ad hoc manner remains a resource intensive process.

Learning objects are data heavy, and current LOM schema such as IEEE LOM, SCORM, and CanCore contain a large number of attributes. Most systems that create LOs that follow these standards are created for desktop operating systems and thus are created assuming very large memory and processor thresholds compared with that of a mobile device such as a cell phone. The problem occurs when porting these systems onto a mobile client. Creating a LO that follows any of the afore- mentioned schemas requires a significant amount of processing time and memory and can quickly slow down or halt a system when downloading, creating, or editing multiple LOs consecutively. The solution is to create a lighter LO, one that is more adaptable to mobile devices, so we come up with the mobile LO (mLO).

4.3.4.1 JSON

A mLO is a LO at its most basic, stripped-down form. It contains the raw data that make up the LO and follows a very basic metadata structure. IEEE LOM-based schemas use XML to encode metadata; however, our mLO metadata will be encoded in JavaScript Object Notation (JSON). We choose a different encoding format in order to keep with our lightweight design and to integrate with our system’s storage choice, the document-based database. JSON encoding also keeps up with more recent technologies. JSON has a much smaller grammar and maps more directly onto the data structures used in modern programming languages, therefore making it more flexible than XML and easier to read by machines and humans [citation needed to http://www.json.org/xml.html]. Consider the following comparison between XML and JSON representations of our metadata structure, details of which will be discussed in a later section.

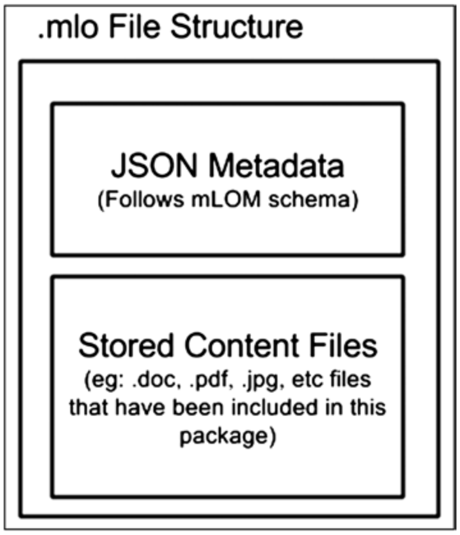

.mlo File Structure
JSON Metadata
(Follows mLOM schema)
Stored Content Files
(eg: .doc, .pdf, .jpg, etc files that have been included in this package)

Figure 4.3: .mlo file structure layout

XML

```
   <?xml  version="1.0"  encoding="UTF-8"?>
 <MLO>
   <General>
      <Identifier>45424259</Identifier>
      <Title>How to write a check</Title>
```

```
  <Description>A simple way to learn the parts of a check and how to
   properly fill one out.</Description>
</General>
<Rights>
  <AuthorIdentifier>455741484D414E</AuthorIdentifier>
  <PrivelegeLevel>00000100</PrivelegeLevel>
</Rights>
<Technical>
  <FilesList>
     <File>
      <FileName>How to write a check in 7 easy steps.txt
      </FileName>
      <FileFormat>txt</FileFormat>
      <FileSize>1133</FileSize>
      <FileDateTimeCreated>2014-07-22T09:43:18-04:00
      </FileDateTimeCreated>
     </File>
     <File>
      <FileName>Check  Diagram.gif</FileName>
      <FileFormat>gif</FileFormat>
      <FileSize>6775</FileSize>
      <FileDateTimeCreated>2002-02-12T22:47:21-04:00
```

```
                </FileDateTimeCreated>
              </File>
          </FilesList>
          <Size>10139</Size>
          <Location> ... </Location>
          <DateTimeCreated>2014-07-22T10:07:33-04:00</DateTimeCreated>
        </Technical>
    </MLO>
```

JSON

```
    {
        "General": {
            "Identifier":          45424259, "Title":
            "How to write a check",
            "Description": "A simple way to learn the parts of a check and how to properly
                 fill one out."
        },
      "Rights": {
        "AuthorIdentifier":  "455741484D414E", "PrivelegeLevel":
        "00000100"
      },
      "Technical": {
          "FilesList":[
            { "FileName": "How to write a check in 7 easy steps.txt", "FileFormat":  "txt",
               "FileSize":  1133,
               "FileDateTimeCreated":  "2014-07-22T09:43:18-04:00"
            },
            { "FileName": "Check Diagram.gif", "FileFormat":  "gif",
               "FileSize":  6775,
               "FileDateTimeCreated":  "2002-02-12T22:47:21-04:00"
            }
         ],
         "Size":  10139,
         "Location":  "...",
         "DateTimeCreated":   "2014-07-22T10:07:33-04:00"
      }
}
```

It may be clear to the reader that JSON is not only more succinct than XML but also easier to read with less opening and closing tag visual clutter. It also represents data in a way both programmers and nonprogrammers can understand and visualize.

4.3.4.2 Document-based database

Document-based databases are faster performing and easier to use than relational databases. They allow for quick data storage and retrieval, are portable, and scalable. Because they receive and send data using JSON-based encoding, we decided to encode our mLOs in JSON. This allows us to use the same document model to store our mLOs on mobile devices, desktop devices, and even on cloud servers.

A document-based database also works ideally for our lightweight storage solution. We can quickly retrieve the metadata entries for the mLOs in our system and package the metadata natively as strings. This is an important factor as the exchange of strings is something every ad hoc network is able to do quickly and efficiently and is a feature we take advantage of in the design of our system with the mLO Index component, to be discussed in a later section.

The process by which mLOs are created in our system and the role of the document-based database in this process is detailed in Algorithm of Mobile Learning Object Creation and Storage

Step 1: Users fill out the mLO creator fields through the GUI and confirm creation of a new mLO

Step 2: Mobile Learning Object Manager creates a new mLO Object and passes it to DMM

Step 3: DMM creates an .mlo file and stores it to a local folder

Step 4: DMM creates a JSON string that describes the mLO Metadata, one of the properties injected into the JSON string is the mLOLocalFilePath, a string that is the local storage path where the file is being stored.

Figure 4.4: Algorithm of Mobile Learning Object Creation and Storage

4.3.5 Metadata

Our lightweight M-Learning object also requires an equally lightweight metadata structure. Starting from the original, nine-category, IEEE standard for LOM, we kept only the most essential categories and subcategories to make our metadata as lean and portable as possible. The result is the metadata structure seen in Figure 2. Three main categories were kept from the original IEEE LOM: general, rights, and technical.

General. This category is used to simply identify the mLO as a whole. The Identifier is a unique value assigned to the mLO when it is added to the user's repository and is modified when the mLO is updated. It is local only to the user's device and thus acts as device-specific hash value. The Identifier property is useful during inquiries and exchanges as it used to match an mLO request with the correct .mLO file version, which could change during the course of a section. The Title is a string provided by the user that acts as the title for the mLO. The Description is also a string provided by the user that acts as the user's description of the mLO.

Rights. This category acts to identify the mLO's creator and any restrictions to future changes and/or sharing they wish to enforce. The Author Identifier is a unique string or integer value assigned to the user based on an identifier they provide and the device they have created the mLO on. Privilege Level is an 8-bit integer value used to identify public, private, or protected- like access.

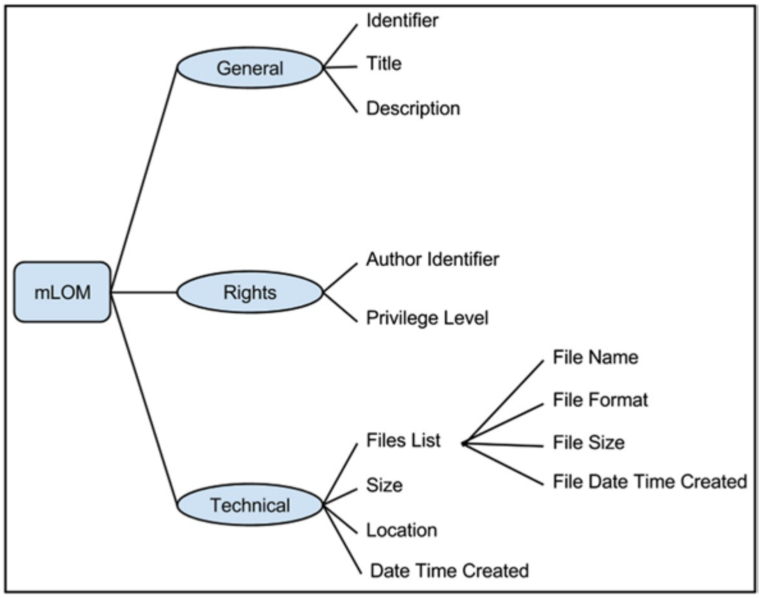


Figure 4.5 Mobile learning object metadata (mLOM).

Technical in this category, mLO file Size and Date Time Created are properties typical of most file metadata structures. We keep the Location property found in the original IEEE LOM to enable location-based or location-sensitive mLOs. We add a new subcategory, Files List, specific to the mLO file structure. Files List is a collection in which each entry is a dictionary, or key-value pairing, of each content file's basic metadata. There are four keys for each content file entry, File Name, File Format (i.e., the file's extension), File Size, and File Date Time Created. This allows for M-Learning tools and users to quickly gain a basic understanding of what files are currently stored in the mLO and gauge whether it meets their criteria.

4.3.6 System Architecture

Figure 3 illustrates a high-level view of our system architecture design. At the most basic level, our system consists simply of user devices connected to each other in an ad hoc network. These devices can be mobile (e.g., tablets, cell phones, and laptops) or stationary (e.g., desktops) as long as they are wireless network capable.

Our M-Learning object system is managed by a cross-platform app located on the devices.

Incoming and outgoing communications are first handled by the app's networking module.

Networking Module. The Networking Module handles the advertising, discovery, and section creation of a user's device with other peer devices. Once peers are connected to each other, data is transferred in two different forms: Resources and Messages. In our system, Resources are the mLO document files, and Messages are small serialized objects that come in three flavors:

- mLO Index messages are objects that consist of a collection of JSON-formatted metadata strings of all the publicly shared mLOs as well as any nonpublic mLOs the receiving peer may have permissions for.
- General Inquiry messages are a unique string that indicate a peer's inquiry for an mLO Index and are used to keep track of peer requests.
- Specific Inquiry messages must be preceded by a General Inquiry message within a certain time period and are objects that consist of the previous General Inquiry's unique string and a JSON- formatted Metadata string identical to one of those in the receiving end's mLO Index.

Algorithm 2 breaks down the Mobile Learning Object Request process. In order to cut down on unnecessary mLO downloading and ease the burden on the device's storage space, we designed a two-factor mLO downloading process. First, the system sends out

General Inquiries to peers in the network. It then receives the mLO Indices and parses through them, pinpointing any mLOs it is interested in downloading. It then sends out Specific Inquiries to peers in the network, requesting to download the mLOs of interest from their device. Second, if the peer devices still contain the mLO in question, a transfer is initiated, and the peer sends the user an mLO resource.

Request and Response Module (RRM). The RRM handles data acquisition and storage requests as passed by the Networking Module. It takes care of the following duties:

- Passes mLO Index objects to the Searching and Ranking Module for parsing;
- Receives and processes requests from the Searching and Ranking Module in order to create General or Specific Inquiries to be sent through the Networking Module;
- Preps any incoming Resources for integration into the user's device; and
- Verifies and packages any outgoing mLO Index or Resource data before passing them to the Network Module to be sent off to other peers.

Searching and Ranking Module (SRM). The SRM handles mLO searching-related and receiving- related logic. When a user requests to search their peers for an mLO, the SRM handles processing the user's request. The SRM works with the Data Management Module (DMM) and RRM in order to collect mLO Index data, parse through it, and find mLOs related to the current inquiry. If the mLO data is already stored locally, the SRM notifies the DDM which mLO to pass back to the Mobile Learning Object Manager. If the mLO data are not stored locally but is identified to be among peers, then the SRM communicates with the RRM to begin the mLO request and transfer process.

The DMM acts as the main interface for sending, packaging, and receiving data from the application’s internally hosted storage, the Neighbor Profiles Cachem, and the mLO Repository.

Neighbor Profiles. This cache stores data about peers the user’s device has encountered. Basic user data received from the Network Module and that user’s most recent mLO Index data are stored in this cache. The cache is managed by the DMM, which updates user entries or removes entries that have not been used after a certain time frame.

mLO Repository. This repository stores all local M-Learning object data. This includes mLOs the user has created and as well kept on their devices.

Mobile Learning Object Manager. The Mobile Learning Object handles inputs from the user through a Graphical User Interface (GUI) in order to create, edit search for, and display mLOs.

GUI. The user interacts with the Mobile Learning Object Manager through a GUI. GUI layouts and design differ based on the user’s device type and operating system.

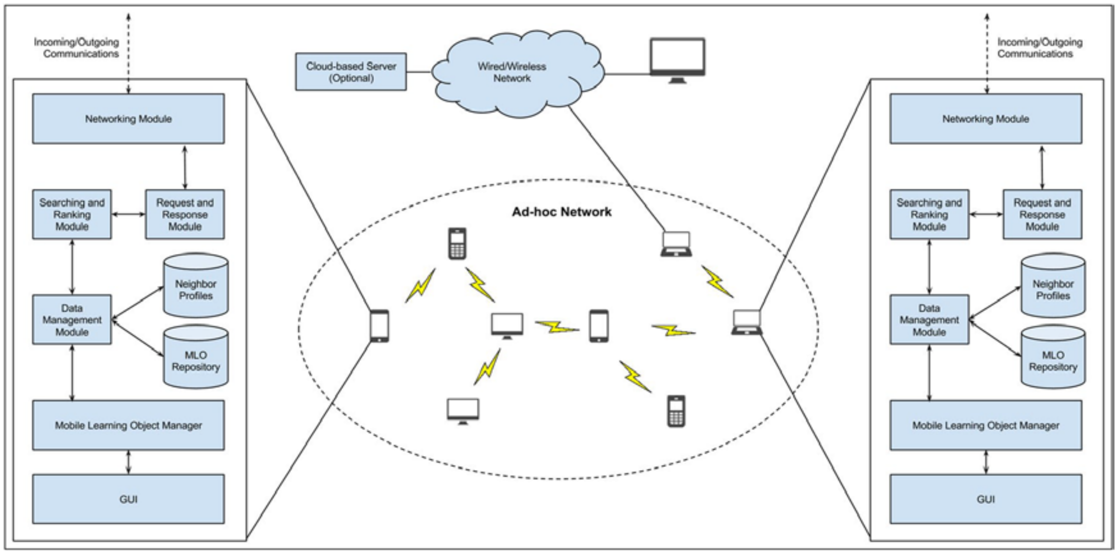

Incoming/Outgoing Communications
Networking Module
Searching and Ranking Module
Request and Response Module
Data Management Module
Neighbor Profiles
MLO Repository
Mobile Learning Object Manager
GUI
Cloud-based Server (Optional)
Wired/Wireless Network
Ad-hoc Network
Incoming/Outgoing Communications
Networking Module
Searching and Ranking Module
Request and Response Module
Data Management Module
Neighbor Profiles
MLO Repository
Mobile Learning Object Manager
GUI

Figure 4.6 High-level system architecture. mLO, mobile learning object.

The algorithm below to describe how mobile learning object send a request to other mobile node

```
User creates a semantic inquiry for relevant mLOs through the Mobile Leaming Object Manager's GUI.
Mobile Leaming Object Manager sends mLORequest to DMM
DMM Passes User's mLOindex to SRM
if User has indicated network inclusive search then
  SRM receives command to search peers for relevant mLOs
  SRM tells RRM to send a General Inquiry to peers
  RRM creates a General Inquiry and passes it to the Networking Module for outgoing broadcast
  if Network Module receives responses then
    SRM clears the Neighbor Profiles Cache and then adds all new incoming mLOIndices received from this request section.
    SRM Creates an empty FoundmLOsColleetion to store and relevant mLO JSON Metadata strings it finds.
    for all mLOindex in Neighbor Profiles Cache do
      if SRM finds an mLO entry of interest then
        SRM adds mLO Metadata entry to it's FoundmLOsC olleetion
      end if
    end for
    if FoundmLOsColleetian contains entries then
      for all mLOEntry in FoundmLOsCollection
```

```
            do
                SRM tells RRM to make a Specific Inquiry to peer for *FoundmLO*
                RRM creates Specific Inquiry object and sends it out to peers through Networking Module
                **if** Network Module successfully finds peer **then**
                    Network Module initiates file transfer re- quest and receives mLO Resource
                    mLO Resource is passed through RRM and SRM to the DMM for storage
                    Mobile Learning Object Manager notifies the user of the newly received mLO files through GUI
                **end if**
            **end for**
        **else**
            Exit process, alert Mobile Leaming Object that no mLOs were found
        **end if**
    **end if**
else
    Exit process, alert Mobile Leaming Object that no mLOs were found
**end if**
```

Figure 4.7 Mobile Leaning Object Request Algorithm

4.4 Implementation

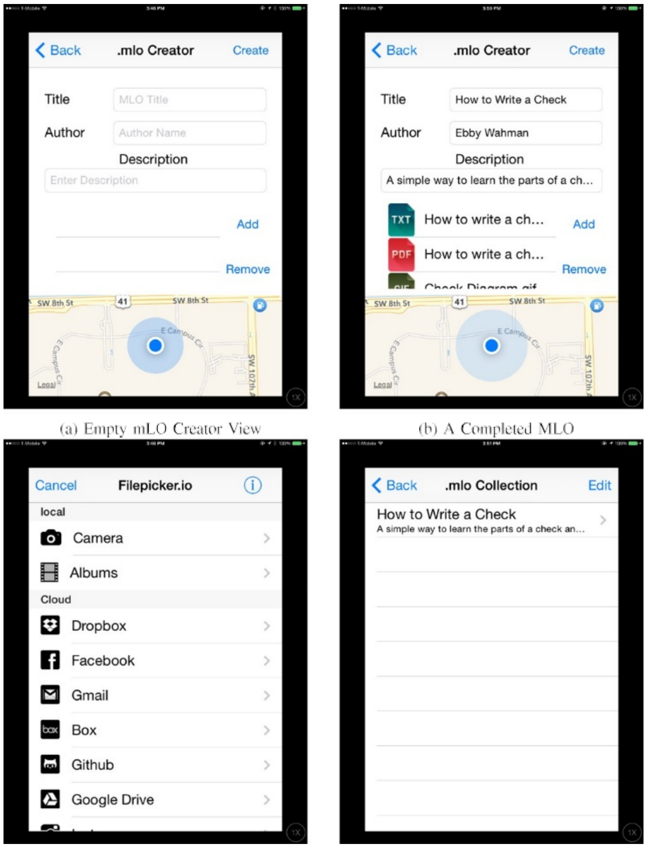
This section details how one would implement an M-Learning object author. For quick prototyping, we decided to develop the application on Apple's iOS mobile operating system because, as of version 7, it enables ad hoc communications between device applications natively through its Multi-peers Connectivity API. We discuss frameworks and design approaches that we used and that have the potential for use in our system implementation. We also go into detail into how a cross-platform application could be created and what frameworks that could be used in order to implement it.

4.4.1 iOS mobile application prototype

Using Xcode, Apple (Cupertino, CA, USA) integrated development environment for its desktop and mobile operating systems, OS X, and iOS, we built a prototype app using Objective-C.

The user GUI was created using Xcode's storyboard designer, which allows creators to drag and drop elements onto a blank canvas in order to create quick, responsive GUIs that can properly fit any of Apple's different-sized mobile or desktop devices. Our prototype application supported three 'views' or screens: the mLO creator, the mLO viewer, and the mLO repository browser; a fourth view, the mLO discovery view, was not implemented because of reasons that will be discussed later.

Figure 4a shows a screen shot of an empty creator view. For the purposes of our prototype, our creator view had the following fields that correspond to the following mLOM schema:



Title: MLO Title
Author: Author Name
Description: Enter Description

(a) Empty mLO Creator View

Title: How to Write a Check
Author: Ebby Wahman
Description: A simple way to learn the parts of a ch...
TXT How to write a ch...
PDF How to write a ch...
GIF Check Diagram.gif

(b) A Completed MLO

local
Camera
Albums
Cloud
Dropbox
Facebook
Gmail
Box
Github
Google Drive

How to Write a Check
A simple way to learn the parts of a check an...

Figure 4.8 Prototype iOS App screenshots. (a) empty mobile learning object (mLO) Creator View, (b) a completed mLO, (c) File Picker View, and (d) mLO Repository View.

- Title → the General/Identifier property.
- Author → the Rights/Author Identifier property. This was merely for the purpose of the proto- type. In future implementations of the app will not need this field to be entered by the user and instead will use user profile information filled out by the user when the app is first opened after installation and that can be edited by the user in a settings screen.
- Description → the General/Description property. This is a user-entered description of the mLO so that other users may read and the system can be used for semantic searching.

- A TableView of the files added to this mLO → the Technical/FilesList property. Each content file's filename, file format, size, and date created are added to our mLO's metadata.
- A MapView of the user's current location → the Technical/Location property. This is pulled from the system's geolocation data and cannot be manipulated by the user through the GUI.

From this screen, after users have filled out the appropriate fields and added at least one content file, a new mLO can be created by pressing/tapping the 'Create' button. This new mLO is then stored in the mLO Repository and is immediately available for viewing in the user's mLO Repository Viewer. Not included in this screen is the ability to allow users to make the files public, private, or protected, and therefore, by default, all files in this prototype are private until the user decides to share the object.

An example of a filled-out mLO in the creator view is shown in Figure 4b. The files list is populated by tapping on the 'Add' button, which pulls up a file picker view as shown in Figure 4c. For this prototype, we use the filepicker.io API, a cross-platform API that allows users to choose and view files stored locally on the device, from other apps on the user's device, from popular cloud storage services the user may have an account for, or from the Web. Filepicker.io was chosen because of its cross-platform ability, so that we may be able to reuse the same method across different platforms without much difference. Users can also remove files they have added to the files list by tapping on the 'Remove' button, which lets users swipe away the file row they wish to delete.

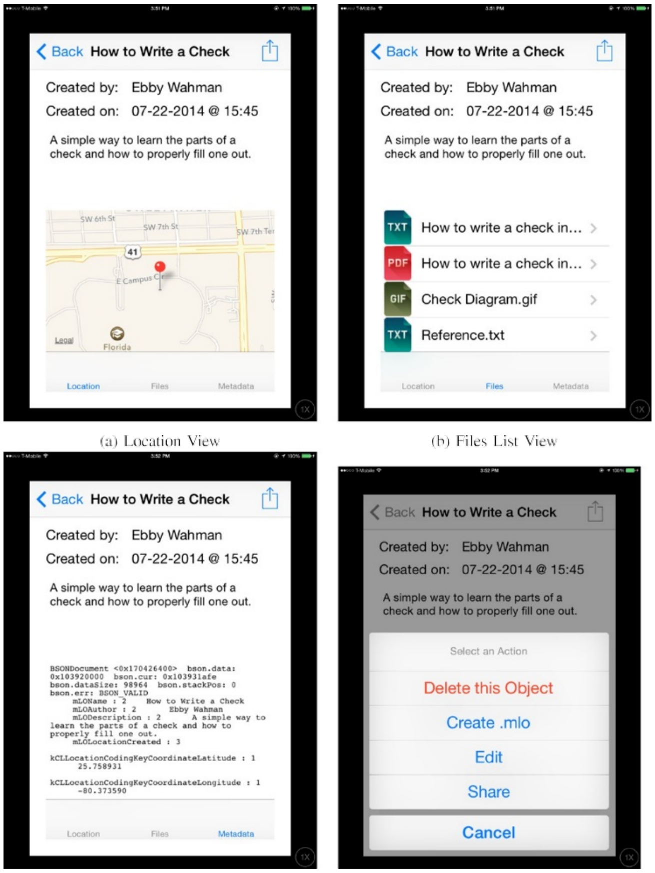
Figure 4d demonstrates our prototype’s table view of the user’s mLO Repository. Here, users can select to which mLO files they wish to view and are then taken to the mLO Viewer screen.

Two options were originally considered for the ad hoc, peer to peer connectivity on our iOS app, the first being the Multi-peers Connectivity API, available natively in iOS version 7 and up and the second being the AllJoyn framework, because of its cross-platform capability. The AllJoyn frame- work, although promising on paper, is a difficult framework to integrate in reality and may require more time for development. Although we were able to compile and run their sample applications that demonstrate their cross-platform communications ability, we were unable to successfully integrate it into our own application. Fragmented and outdated documentation and a less-than-friendly API made it difficult to integrate into our prototype, so further development with AllJoyn was put on hold.

Quick tests for portability were able to be checked by sending the .mlo files from one device to another and viewing the content in the newly acquired .mlo. Figure 5 shows how our prototype views mLO files.

The mLO Viewer has three tabs that reveal different information about the mLO. Regardless of tab, general information about the mLO is displayed at the stop of the screen. In our prototype, the title of the mLO is displayed in the navigation bar. Its creator, date time created, and the mLO’s description ad are shown below the navigation bar. Figure 5a shows the mLO Viewer with Location tab selected. This gives us a map view of where the mLO was created. This can become handy for LOs where location is vital to the learning process. Figure 5b shows the mLO Viewer with Files tab selected. This presents a table view that allows the user to select a file row and preview the associated file or open in the app of their choice. Figure 5c shows the mLO Viewer with Metadata tab selected. This was included only for test purposes so that we could have a clear visual of what metadata the application sends between devices.

From this mLO Viewer screen, users can also tap on the top-right action button, which brings up an action view as shown in Figure 5d. The ‘Delete this Object’ option allows users to permanently delete the mLO object from their app. In our prototype, mLO objects were created and stored in the system cache for that section; it is not until the ‘Create .mlo’ button is pressed that a .mlo file is created and stored onto disk. This was merely for test purposes, and in a full implementation, when tapping the ‘Create’ button in the mLO Creator view, the .mlo file will automatically be created. The ‘Edit’ action brings up the mLO Creator view but with the fields filled out with the mLO’s data and allows you to edit those fields; the ‘Create’ button turns into a ‘Save’ button, and tapping it confirms changes to the mLO. When an mLO is changed, its date time created field is modified, as well as its General/Identifier property.



Created by: Ebby Wahman
Created on: 07-22-2014 @ 15:45

A simple way to learn the parts of a check and how to properly fill one out.

(a) Location View

Created by: Ebby Wahman
Created on: 07-22-2014 @ 15:45

A simple way to learn the parts of a check and how to properly fill one out.

TXT How to write a check in...
PDF How to write a check in...
GIF Check Diagram.gif
TXT Reference.txt
Location
Files
Metadata

(b) Files List View

Created by: Ebby Wahman
Created on: 07-22-2014 @ 15:45

A simple way to learn the parts of a check and how to properly fill one out.

```
BSONDocument <0x170426400>  bson.data:
0x103920000  bson.cur: 0x103931afe
bson.dataSize: 98964  bson.stackPos: 0
bson.err: BSON_VALID
     mLOName : 2    How to Write a Check
     mLOAuthor : 2       Ebby Wahman
     mLODescription : 2       A simple way to
learn the parts of a check and how to
properly fill one out.
     mLOLocationCreated : 3

kCLLocationCodingKeyCoordinateLatitude : 1
      25.758931

kCLLocationCodingKeyCoordinateLongitude : 1
      -80.373590
```

Created by: Ebby Wahman
Created on: 07-22-2014 @ 15:45

A simple way to learn the parts of a check and how to properly fill one out.

Figure 4.9: Metadata View and mLO Options View

4.4.2 Cross-platform application

Ultimately, our goal is to implement a cross-platform authoring application that runs on both mobile and desktop environments without the need for an infrastructure-based network.

This way, learners can use the same application to create, send, and view mLO files to each other regardless of platform and on the same network. Users on a laptop can interact with users on cellular devices or users on tablets without the need for converting files or switching devices to enable new features. This creates an easy to use environment that supports most methods and devices for learning without the need for much setup while enabling true usability in an informal or ‘on-the-fly’ learning section.

Currently, there are many technologies available that allow for cross-platform application development. During our research, we experimented with Motorola’s RhoMobile Suite, Appcelerator’s Titanium, and Digia’s Qt frameworks to gauge them for the potential of future implementations. Each system has their strengths and drawbacks, but they, including many other cross-platform development frameworks currently in the market, either lack or have a weak cross-platform ad hoc network capability. This means that we may either need to develop our own ad hoc network component for these development platforms or proceed with developing each platform’s application individually in its native code and then using AllJoyn’s or another ad hoc framework to incorporate our peer-to-peer network features in order to keep the same or similar codebase and ease the process for updates.

4.5 Conclusions

Mobile learning is here to stay. As mobile devices become more and more ingrained the 21st century consumer’s life, and as technologies continue to move to a mobile form factor, it is evident that learners will demand that their learning technology become just as mobile as the rest of the technology in their lives. The challenges reside in how we can design a system that truly works with not only the learner in mind but also the way learners utilize mobile devices. Our M-Learning object authoring tool and management system proposal aims to take the M-Learning design and implementation topic into a mobile computing path and create discussions on how we should see M-Learning as less of an e-learning extension and more as its own independent learning approach.

In this chapter, we proposed the integration of ad hoc networks into our system to take advantage of mobile device technologies and to imitate a more natural form of information learning and exchange. We detailed a metadata schema for LOs that are made with mobile devices in mind, the mLO. We also discussed the necessary components and processes for a system that uses the mLO and enables ad hoc sharing and communications. We discussed the implementation of a mLO authoring tool and made clear our intentions for a cross-platform application. In future works, we plan to implement the full system as cross-platform application and run analysis on the lightness and efficiency of such a system.

CHAPTER 5: MOBILE LEARNING OBJECT AND M-LEARNING SYSTEM IMPLEMENTATION

This chapter provides a detailed implementation of our proposed mobile learning system. First, we introduce the design goals in section 5.1, then specification and design of the proposed mobile learning system in section 5.2. In section 5.3, we present the low level design and implementation of the system. Testing and validation is presented in section 5.4. Lastly, we discuss our results and conclusions in section 5.5.

5.1 Introduction

In this chapter, we provide a detailed implement of our proposed M-Learning system discussed in Chapter 3 as a proof of concept. The design concept proposed uses Bluetooth connectivity to allow users to share reusable mobile learning objects between devices in an ad-hoc network environment. The proposed solution should also allow users in a group to search for, share and download files, and assign different levels of privacy within that group. Another feature of our system is a peer-to-peer chat which uses the Bluetooth chat messaging feature using MLO structure. We also implement our extended solution for the hybrid mobile ad-hoc and wireless networks. The high level solution specification and system design is described in the next section.

5.2 System Design and Specification

The system consists of mobile ad-hoc nodes that are connected via Bluetooth. Each node runs the mobile learning application software that we built to provide the user interface to provide interaction among the mobile nodes. This mobile app uses the mobile learning objects to send request to the neighbor nodes and receive from them. The communication process between mobile nodes is shown in figure 5.1. First, the system initializes the Bluetooth connection. The sender node creates a request in the form of MLO format and stores it at its local storage device. Then the sender node sends that MLO over the ad-hoc network to reach the intended receiving node. The receiving node reads that MLO, analyzes the given query request within the MLO, and executes the query. The result is now being wrapped into a new MLO object. It then stores that MLO object to its local storage which will be sent back to the sender node. The sender node then extracts its response from the receiver's replied MLO object.

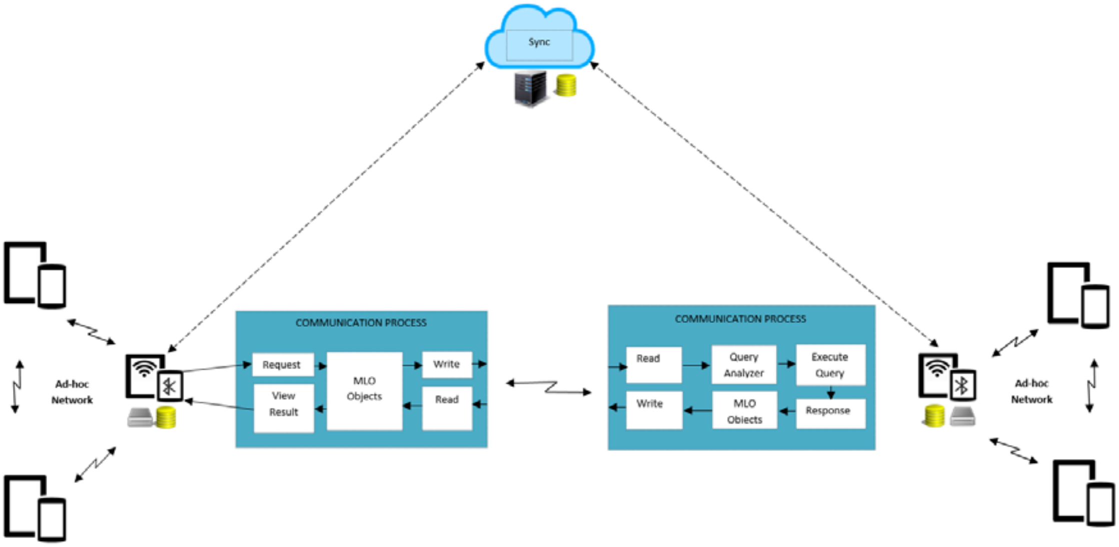


Figure 5.1: Overview of a communication between sender and receiver nodes

In case of availability of WiFi or cellular network connections, those MLO objects could be stored in the cloud-based document database called Couchbase which has already been installed on the Amazon EC2 web server. Also Couchbase Sync Gateway already has been installed on the server to allow mobile nodes to sync the MLO objects to the server so that those mobile nodes can be synced offline. In the next section, we will explain further details of the mobile learning object construction which is one of the core development concept in our proposed system architecture.

5.2.1 Mobile Learning Object

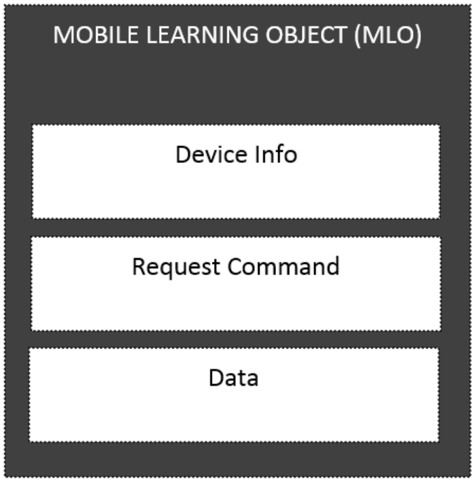


Figure 5.2: Mobile Learning Object (MLO) Structure

The communication between mobile devices are done using mobile learning objects. We modeled the MLO object based on the object oriented design (OOD) principles of the Java programming language. The MLO structure has three parts: Device Information (DI), Request Command (RC), and the Data. The DI includes device name, Android unique and secure identification number, and device's MAC address. The RC is the command that the receiving node uses to execute in order to obtain the required information about the sender's request. The RC is not limited to include a command to search for a file, send results, request downloads and send messages. The Data is the actual information contained within the MLO to be used by the receiving nodes. For example, the sender node can send an MLO object to search for a given file on the receiving node.

The MLO object information would be:

- Device Information: sender name, sender MAC address, Android secure id
- Request Command: search file
- Data: The keyword search, title, location.

The MLO is the primary object that the system uses as an essential piece of data communication between mobile nodes. In the following section, we describe how we handle the privacy issues regarding to files and MLOs and what access level mobile nodes can have.

5.2.2 Privacy & Group Policy

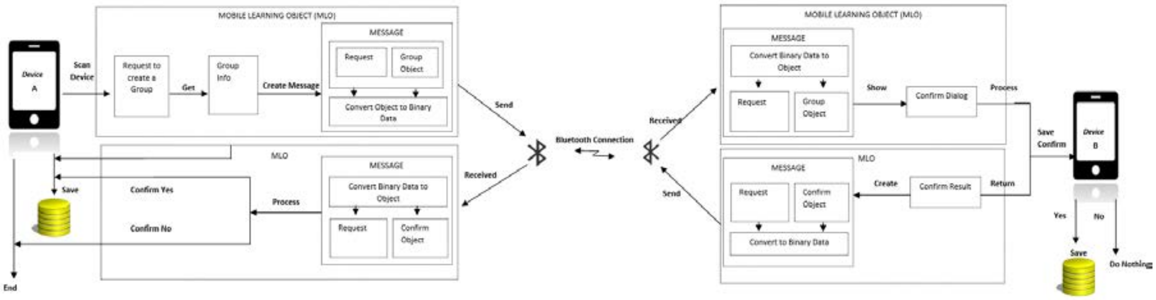


Figure 5.3: Privacy and Group Policy Assignment Diagram

Our solution approach limits the amount of privacy issues our users would face by implementing a group policy. Mobile devices are connected in the ad-hoc network to request and share files via the MLO objects. This group policy includes 3 different types of access rights called private, public and protected. The figure 5.3 depicts the group policy assignment diagram of the solution. Private files allow access only to the device owner. Public files are accessible to all paired devices. Protected files allow access to owner specified paired devices. To create group access, the mobile devices must be paired and a group name must be assigned. Then any MLOs made available to group access will become visible to all members of the group. The MLOs access right can be changed and updated by the mobile device owners whenever they want. In the following section, we will describe how to share files among mobile devices in the group or in the ad-hoc network.

5.2.3 File Sharing

Host mobile devices select files to share and grant them specific access rights. The sharable file information is encapsulated into the MLO object which is stored onto the local mobile device local storage and then will be sent to the intended paired device. In the case that mobile devices have internet or cellular network connection, those MLOs are synced into the cloud-based database server. Once sharable, the paired devices are able to view MLOs containing the file sharing information. The paired devices can then use these MLOs to access the files from the host device as shown in figure 5.4. Once the files are configured to be shared, other mobile nodes can search those files using keyword search or location attribute. The search files will be explained in more details in the next section.

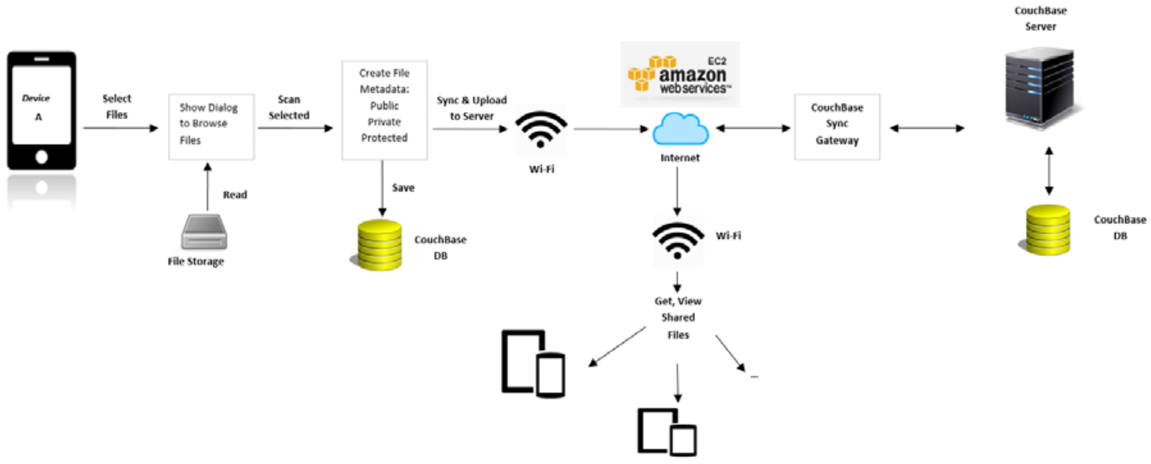


Figure 5.4: File Sharing Diagram

5.2.4 Search Files

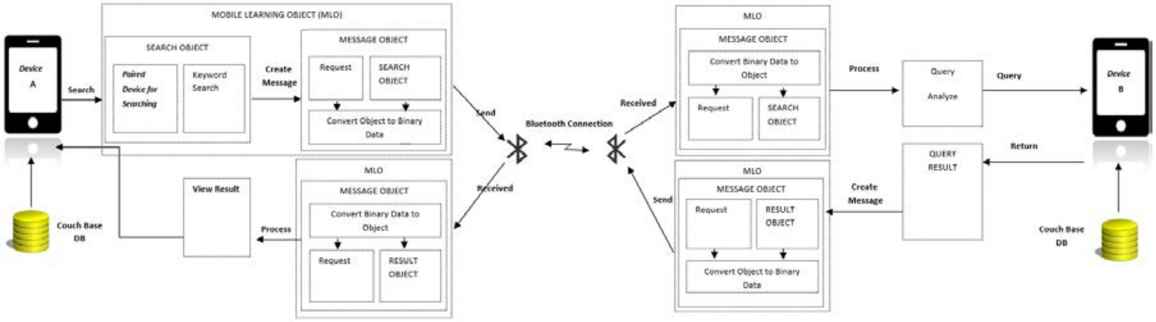


Figure 5.5: Searching Files Diagram

Figure 5.5 shows the diagram of mobile device A (the source node) which is searching for a file on a paired Bluetooth mobile device B (the destination node). Node A initiates the search and enters the search criteria such as keyword search or location information. The search command and data is encapsulated in a MLO object to be sent to the destination mobile node B. Node B reads the MLO and processes the query then return the result in a MLO object to node A. Node A receives and reads the response MLO object from node B. A file searching feature is one of the useful tools in our system. Once the mobile nodes search and receive the search result, the users can choose which file to download. In the next section, we will describe the process of downloading a file.

5.2.5 File Download

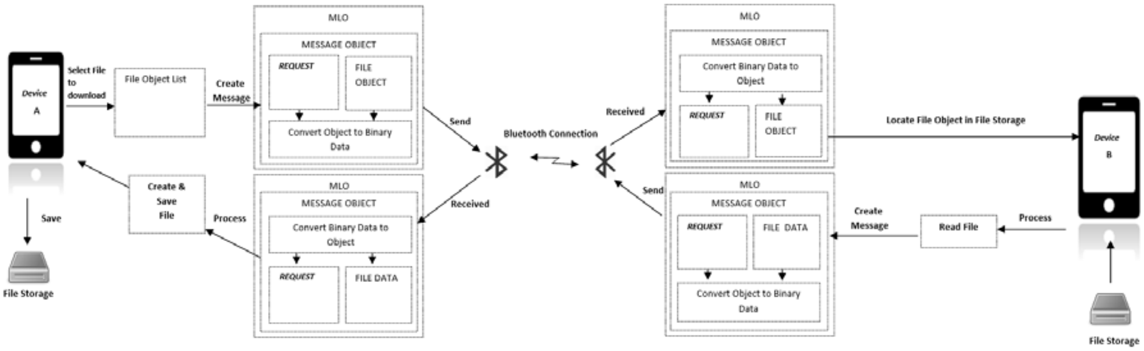


Figure 5.6: Downloading a File Diagram

Figure 5.6 shows the diagram of the downloading process by which mobile nodes can send a download request to the file's origin node after selecting a file from the available download list. The download request command is encapsulated in a MLO object and sent to source node B. Node B reads the incoming MLO object, executes the request, and then locates the file in its local storage. Node B continues to read that file and creates a new response MLO that includes the actual file data within the object. Node A receives the response MLO object from Node B and is then able to read, create, and save the new file to its local storage. Besides using a searching and downloading a file feature, mobile ad-hoc nodes/users can use the chat feature to communicate directly to their peers by using the Bluetooth ad-hoc messaging feature that we will explain in the next section.

5.2.6 Bluetooth Ad-hoc Messaging

We incorporated a mobile peer-to-peer messaging feature in an ad-hoc network environment in our system by means of Bluetooth technology that allows mobile users to chat among themselves. Even though peer-to-peer networking using Bluetooth technology is not new, the use of sending chat messages and encapsulating them into a MLO object format is a unique approach that was provided by our solution. Any paired mobile devices can create and send messages under the MLO format. The figure 5.7 shows the diagram of how our Bluetooth ad-hoc messaging function works. First, the mobile device A creates a text message and then it is converted into the binary data that is encapsulated inside the MLO object. The MLO is sent to the intended device B. Once received, device B reads the text message from the MLO and reply. This process can be continued until mobile device users decide to stop it.

We have described the unique approach which uses MLO as the central communication object among ad-hoc nodes. We also described essential features that we include in the implementation such as group policy, searching and downloading files/MLOs, and P2P chatting. We research and extend our work further by including the hybrid solution that is explained in the section 5.2.7.

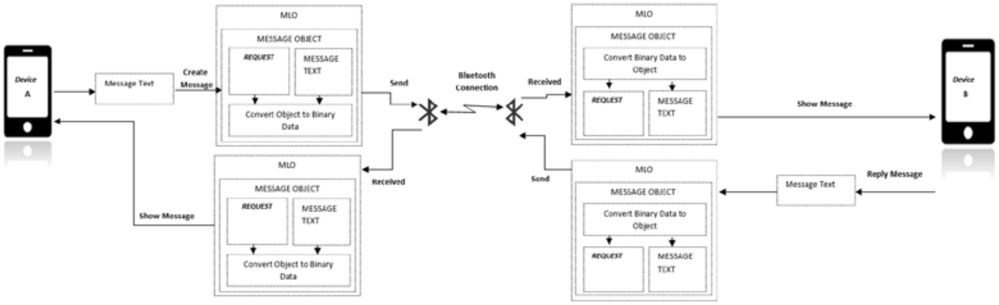


Figure 5.7: Mobile Ad-hoc Chat/Messaging via Bluetooth Diagram

5.2.7 Hybrid Solution

The primary focus of the proposed M-Learning system was to implement the design concept of the ad-hoc networks on the fly learning on Android mobile devices via Bluetooth connectivity. We have also extended our work to include the hybrid solution which allows the MLO objects to be stored on a cloud-based Couchbase server which provides for offline synchronization among mobile nodes.

The Couchbase server is a well-recognized, distributable and scalable, open-source, and NoSQL document-oriented database. Its architecture is shown in figure 5.8 which includes the Couchbase Servers, Couchbase Sync Gateways, and Couchbase Lite Application that runs on iOS or Android OS. The Couchbase server architecture is an innovative solution approach that makes offline synchronization implementation possible for our system.

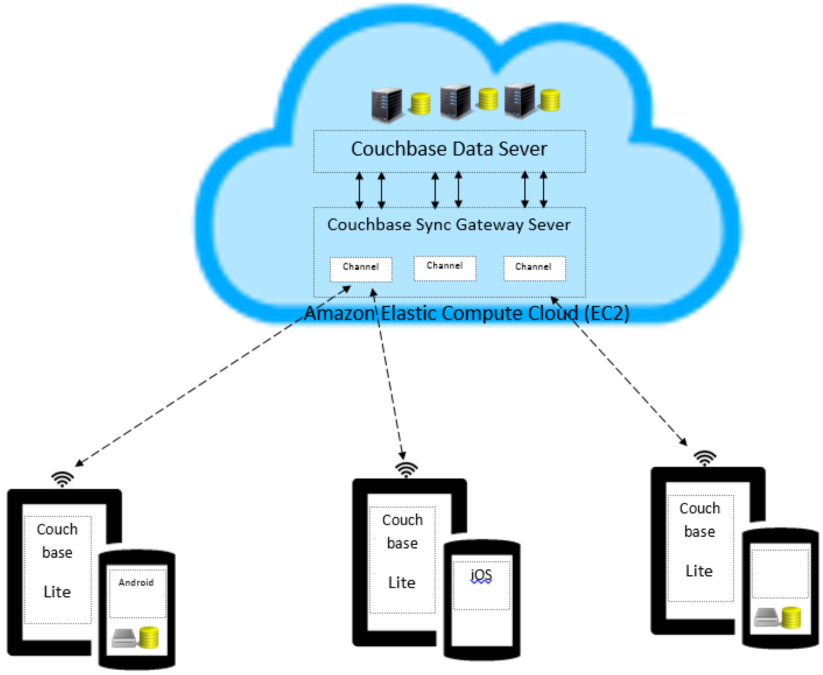


Figure 5.8: The Couchbase Mobile Solution for iOS and Android

Installing Couchbase server and Couchbase sync gateway onto the Amazon EC2 cloud-based web service was our solution approach depicted in figure 5.9. The sync gateway is a server that enables the Couchbase server to act as a replication endpoint between the Couchbase Lite that runs on the client mobile devices. User documents are stored in the server under the server bucket as persistent storage.

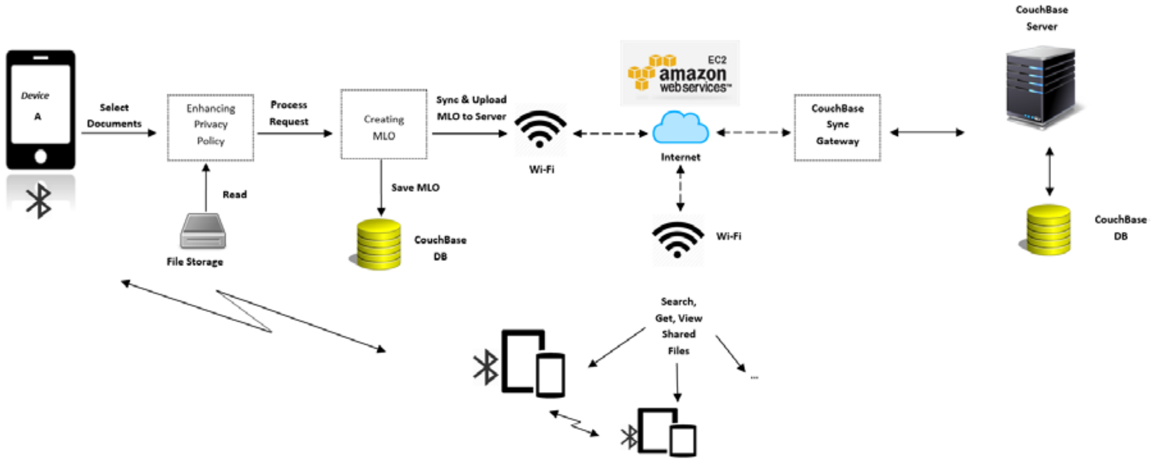


Figure 5.9: Hybrid Solution for using MLO in Ad-hoc and Wireless Networks

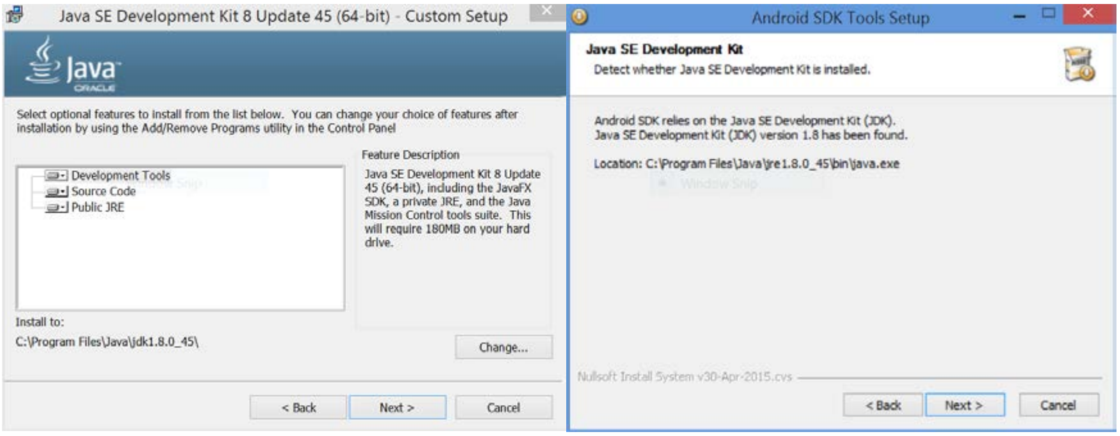
5.2.8 The Assumptions Used in the Implementation

We assume that the implemented mobile application will be installed and will be fully functioning on the Android devices. Mobile devices must be Bluetooth enabled and devices must be paired before using this mobile learning object sharing application. Also, we assume the mobile devices have WiFi or cellular network connectivity in order to use the offline synchronization feature.

5.2.9 The Implementation Platform

To demonstrate that mobile learning objects can be shared amongst all nodes participating in the ad-hoc network, we assume that each mobile device runs on the Android platform. We set up a programming environment using the flowing steps as depicted in figure 5.10:

1. Download and install the Java Development Kit (JDK) that is released by Oracle.
2. Install the Android Software Development Kit (SDK),
3. Download and install the Android Development Tools (ADT) plug-in for Eclipse and finally we need Eclipse IDE to write, compile and run Android applications. To install the ADT plugin for Eclipse IDE, start Eclipse and click "Install New Software." We then click "Add" repository followed by a name such as "Android Plugin" and enter this URL: https://dl-ssl.google.com/android/eclipse/. Continue to click Next, OK to complete and restart Eclipse IDE



Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel

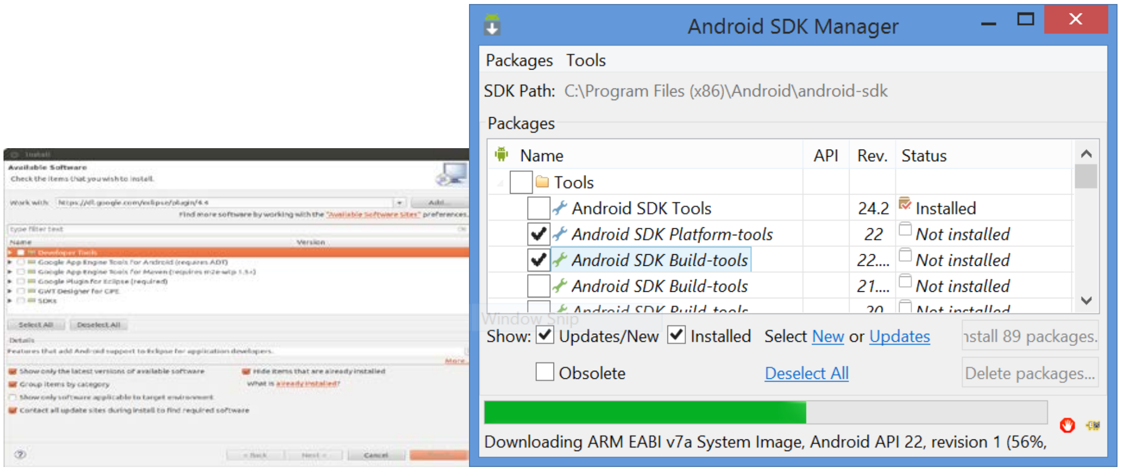
Java SE Development Kit 8 Update 45 (64-bit), including the JavaFX SDK, a private JRE, and the Java Mission Control tools suite. This will require 180MB on your hard drive.

Install to:
C:\Program Files\Java\jdk1.8.0_45\

Java SE Development Kit
Detect whether Java SE Development Kit is installed.

Android SDK relies on the Java SE Development Kit (JDK).
Java SE Development Kit (JDK) version 1.8 has been found.

Location: C:\Program Files\Java\jre1.8.0_45\bin\java.exe



Available Software

Check the items that you wish to install.

Work with: https://dl.google.com/eclipse/plugin/4.4

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
▸ □ Developer Tools	
▸ □ Google App Engine Tools for Android (requires ADT)	
▸ □ Google App Engine Tools for Maven (requires m2e-wtp 1.5+)	
▸ □ Google Plugin for Eclipse (required)	
▸ □ GWT Designer for GPE	
▸ □ SDKs	

Details

Features that add Android support to Eclipse for application developers.

☑ Show only the latest versions of available software ☑ Hide items that are already installed

☑ Group items by category What is already installed?

□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

SDK Path: C:\Program Files (x86)\Android\android-sdk

Packages

Name	API	Rev.	Status
□ Tools			
□ Android SDK Tools		24.2	Installed
☑ Android SDK Platform-tools		22	Not installed
☑ Android SDK Build-tools		22....	Not installed
□ Android SDK Build-tools		21....	Not installed
□ Android SDK Build-tools		20	Not installed

Show: ☑ Updates/New ☑ Installed Select New or Updates

□ Obsolete Deselect All

Downloading ARM EABI v7a System Image, Android API 22, revision 1 (56%,

Figure 5.10: Software Installation and Configuration

During the implementation of this solution, we aim to demonstrate our unique solution of mobile learning objects that is much different than mobile learning objects in existing E-Learning systems. The application end users are mobile device users who would use their mobile devices to connect with other mobile devices in the mobile ad-hoc networks via Bluetooth connection and we also implement the extension solution that includes the hybrid concept which allows mobile devices connect and share files via Bluetooth as well as Wi-Fi and sync the Mobile Learning Object list among the mobile nodes.

Although we have many options, in our choice of operating systems for mobile devices such as Apple iOS, Windows Mobile OS, BlackBerry OS we opted to work with the Android OS. This is because Android is an open source software that is dominant in the mobile market.

5.3 Low Level Implementation

5.3.1 Mobile Learning Object

The MLO object class was implemented as in the segment of the file class source code of table 5.1. The object data contains the object’s properties such as current keyword, current location, MAC address, device id, file name, share option, file path, description, group, object timestamp. The request message command is stored in a string format value to be used by the receiving node for example: CHOOSE_FILE_TO_UPLOAD, REQUEST_SEARCH, SEND_RESULT, REQUEST_DOWNLOAD, SEND_FILE_DOWNLOAD, SAVE_GROUP_TO_DEVICE, SEND_A_MESSAGE, and REPLY_TO_UPDATE_GROUP. We describe the MLO design in section 5.2.1. The source code implementation of a MLO object is shown in table 5.1.

```
public class MobileLearningObject {

        private Integer _id;
        private String filename;
        private String keyword;
        private String share_option; // public/private/protected;
        private String uri; // local file path
        private String loc; // location
        private String description;
        private String group;
        private Boolean check;
        private String create_at; // date-time create
        private String identifier;
        private String title;
        private String authorID;
        private String fileFortmatType;
        private String size;

  // OBJECT  data
  byte[] byte_data=null;
        public MobileLearningObject(Integer _id, String filename, String keyword,String share_option,
String uri, String loc, String description,String group, Boolean check, String create_at) {
                super();
                this._id = _id;
                this.filename = filename;
                this.keyword = keyword;
                this.share_option = share_option;
        this.uri = uri;
        this.loc = loc;
        this.description = description;
        this.group = group;
        this.check = check;
        this.create_at = create_at;
    }

    public Integer get_id() {
        return _id;
    }

    public void set_id(Integer _id) {
        this._id = _id;
    }

    public String getFilename() {
        return filename;
    }

    public void setFilename(String filename) {
        this.filename = filename;
    }

    public String getKeyword() {
        return keyword;
    }

    public void setKeyword(String keyword) {
        this.keyword = keyword;
    }

    public String getShare_option() {
        return share_option;
    }

    public void setShare_option(String share_option) {
        this.share_option = share_option;
    }

    public String getUri() {
        return uri;
    }

    public void setUri(String uri) {
        this.uri = uri;
    }

    public String getLoc() {
        return loc;
    }

    public void setLoc(String loc) {
        this.loc = loc;
```

```
	}

	public String getDescription() {
		return description;
	}

	public void setDescription(String description) {
		this.description = description;
	}

	public String getGroup() {
		return group;
	}

	public void setGroup(String group) {
		this.group = group;
	}

	public Boolean getCheck() {
		return check;
	}

	public void setCheck(Boolean check) {
		this.check = check;
	}

	public String getCreate_at() {
		return create_at;
	}

	public void setCreate_at(String create_at) {
		this.create_at = create_at;
	}

	public String getIdentifier() {
		return identifier;
	}

	public void setIdentifier(String identifier) {
		this.identifier = identifier;
	}

	public String getTitle() {
		return title;
	}

	public void setTitle(String title) {
		this.title = title;
	}

	public String getAuthorID() {
		return authorID;
	}
```

```
    public void setAuthorID(String authorID) {
        this.authorID = authorID;
    }

    public String getFileFortmatType() {
        return fileFortmatType;
    }

    public void setFileFortmatType(String fileFortmatType) {
        this.fileFortmatType = fileFortmatType;
    }

    public String getSize() {
        return size;
    }

    public void setSize(String size) {
        this.size = size;
    }

    public byte[] getByte_data() {
        return byte_data;
    }

    public void setByte_data(byte[] byte_data) {
        this.byte_data = byte_data;
    }
    @Override
    public String toString() {
        return "MobileLearningObject [_id=" + _id + ", filename=" + filename
                        + ", keyword=" + keyword + ", share_option=" + share_option
                        + ", uri=" + uri + ", loc=" + loc + ", description="
                        + description + ", group=" + group + ", check=" + check
                        + ", create_at=" + create_at + ", identifier=" + identifier
                        + ", title=" + title + ", authorID=" + authorID
                        + ", fileFortmatType=" + fileFortmatType + ", size=" + size
                        + ", byte_data=" + Arrays.toString(byte_data) + "]";
    }


}
```

Table 5.1: Mobile Learning Object Class

The MLO object is central object which is carrying the data and command request that other objects use as reference object in their class implementation. A detailed source code implementation is included in the appendices 2 of this dissertation. The next implementation section is about the privacy and group policy that we make sure paired mobile device users have access right to their neighbors' nodes.

5.3.2 Privacy and Group Policy

Mobile devices are connected to an ad-hoc network using the Bluetooth communication. Once mobile node is paired, it can also be configured and granted permission to a specific device by creating group policy via a message object. A message object has the search option and group object to serve as the query to transfer a message from sender device to receiver device. The flowchart below shows all the steps required to create groups and grant permission to the group to access the system.

The table 5.2 is a high level description of the procedure for creating a group permission from Device A to Device B

The Algorithm:

```
Step 1: Start
Step 2: Turn on Bluetooth, scan and pair the neighbor devices
Step 3: Create communication socket to listen to incoming connections
Step 4: Device A create a group name and its device information and generate the MLO group object.
Step 5: If the connection exists between Device A and Device B
                Then Device A sends the MLO object request to Device B
        Else
                Go to step 9 and stop.
Step 6:
        6a. Device B reads the MLO request that is come from Device A and confirms the decision made by Device B
        6b. If the answer is yes, then device B send a response as a MLO object to Device A
        6c. Else, Device B send a confirmation deninal response as MLO object to Device A
```

```
Step 7:
        7a. Device A reads the response MLO object from Device B
        7b. View and save the result stored in that MLO object
                If Device B confirms yes, then Device A save the group action
                Else go to step 9 and stop.
Step 8: If Device A wants to continue to create a new group, then go to step 4
        Else, go to step 9 and stop
Step 9: Stop
```

Table 5.2: Creating Group Permission from Device A to Device B

Figure 5.11 shows a flowchart to create group, access right, or permission from two paired mobile devices. The partial source code reflecting the implementation of this group policy feature is shown in table 5.3. The group policy object includes private data to store information of the device id, group name, device information, description, and date/time created for that group.

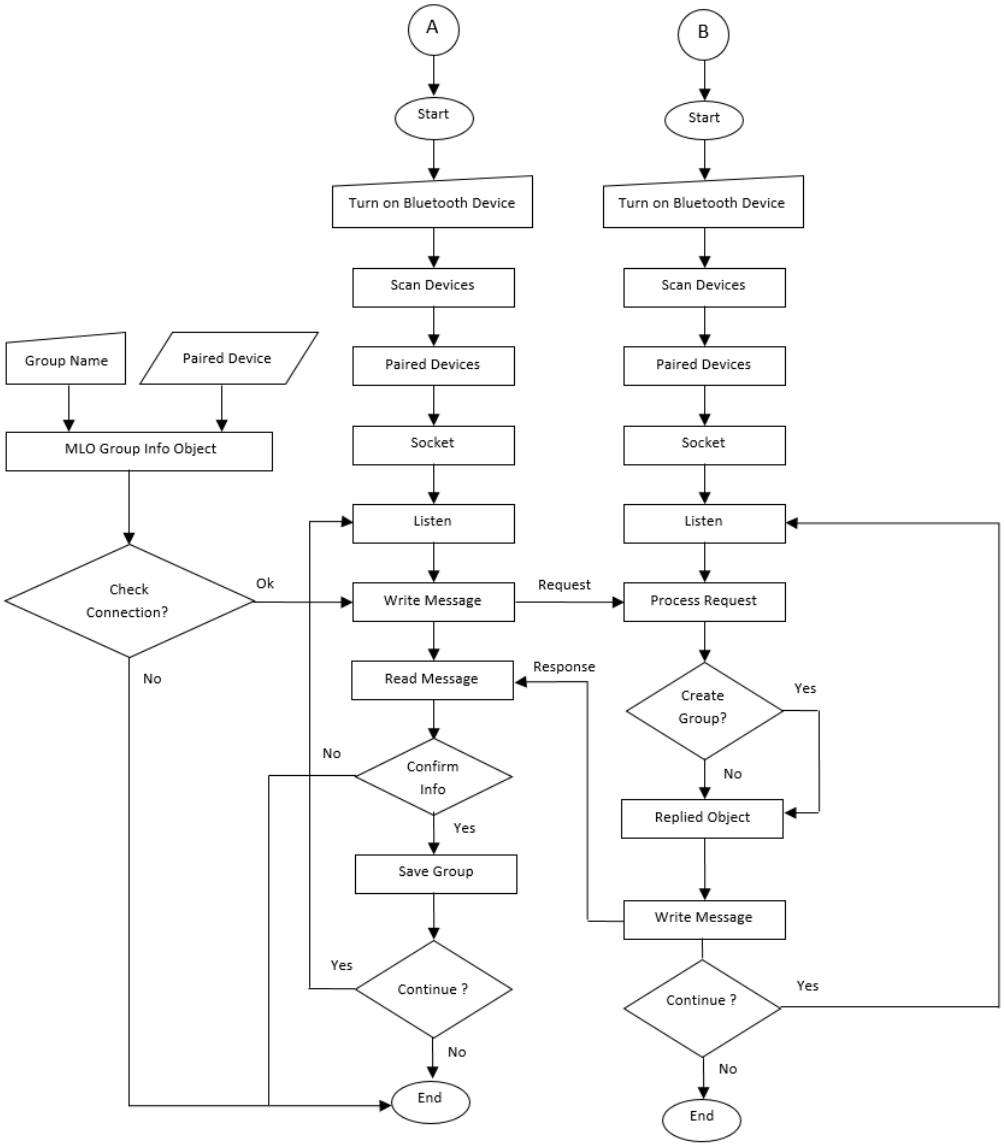


Figure 5.11: Creating Group and Permission Flowchart

```
import java.io.ByteArrayOutputStream;
import java.io.DataOutputStream;
import java.util.HashMap;
import java.util.Map;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Message;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import com.Couchbase.lite.CouchbaseLiteException;
import com.Couchbase.lite.Document;
import com.Couchbase.lite.SavedRevision;

public class Group_Policy {

    private String mac;
    private Integer id;
    private String group_name;
    private String device;
    private String description;
    private Date dateCreated;

    public Group_Policy(String mac, Integer id, String group_name,
                String device, String description, Date dateCreated) {
        super();
        this.mac = mac;
        this.id = id;
        this.group_name = group_name;
        this.device = device;
        this.description = description;
        this.dateCreated = dateCreated;
    }

    public static void requestToCreateGroup() {
        String keyName = "SAVE_GROUP";
        byte[] keyNameData = keyName.getBytes();

        byte[] imei = GROUP_CREATE_BY_IMEI.getBytes();

        byte[] group_name = GROUP_NAME_SAVE_TO_DEVICE.getBytes();

        ByteArrayOutputStream byteOs = new ByteArrayOutputStream();
        DataOutputStream out = new DataOutputStream(byteOs);

        try {
            out.writeInt(keyNameData.length);
            out.write(keyNameData);

            out.writeInt(imei.length);
            out.write(imei);

            out.writeInt(keyValueMac.length);
            out.write(keyValueMac);

            out.writeInt(group_name.length);
            out.write(group_name);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        byte[] allWrittenBytes = byteOs.toByteArray();

        // Invoke client thread to send
        Message message = new Message();
        // message.obj = compressedImage;

        message.obj = allWrittenBytes;

        MainApplication.clientThread.incomingHandler.sendMessage(message);
    }

    public static void save_group(final Activity activity) {
        String docID = deviceId;
        // retrieve the document from the database
        Document retrievedDocument = database.getDocument(docID);
        // display the retrieved document
        Log.e("RETRIVE DOCUMENT: ",
                "retrievedDocument="
                        +
String.valueOf(retrievedDocument.getProperties()));

        EditText editGroup = (EditText) activity.findViewById(R.id.editGroup);
        String myGroupName = editGroup.getText().toString();
        // update the document
        Map<String, Object> updatedProperties = new HashMap<String, Object>();
        updatedProperties.putAll(retrievedDocument.getProperties());
        updatedProperties.put("group_name", myGroupName);
        SavedRevision currentRevision = retrievedDocument.getCurrentRevision();
        try {
            retrievedDocument.putProperties(updatedProperties);
        } catch (CouchbaseLiteException e) {
            // Log.e (TAG, "Cannot update document", e);
        }

    }

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_device);
        Intent myIntent = getIntent();
        String sMAC = myIntent.getStringExtra("MAC_ADDRESS_PARA");
        String sGROUP = myIntent.getStringExtra("GROUP_NAME_PARA");
        String sDeviceName = myIntent.getStringExtra("DEVICE_NAME");
        deviceId = myIntent.getStringExtra("GROUP_ID_PARA");
        TextView mac = (TextView) findViewById(R.id.textMac);
        mac.setText(sMAC);
        TextView device_name = (TextView) findViewById(R.id.textDeviceName);
        device_name.setText(sDeviceName);
        EditText group = (EditText) findViewById(R.id.editGroup);
        group.setText(sGROUP);
        Button btnSave = (Button) findViewById(R.id.btnGroupSave);
        btnSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                MainLearningObjectActivity.CHILD_ACTIVITY
= DeviceGroupActivity.this;
                EditText editGroup = (EditText) findViewById(R.id.editGroup);
                String myGroupName = editGroup.getText().toString();
                // Call save to device via bluetooth
                MainLearningObjectActivity.CURRENT_FUNCTION =
"SAVE_GROUP_TO_DEVICE";
                MainLearningObjectActivity.GROUP_CREATE_BY_IMEI =
MainLearningObjectActivity.IMEI;

                MainLearningObjectActivity.GROUP_NAME_SAVE_TO_DEVICE =
myGroupName;
                MainLearningObjectActivity
                        .do_thread_start("SAVE_GROUP_TO_DEVICE");
                finish();

            }
        });

        Button btnClose = (Button) findViewById(R.id.btnGroupClose);

        btnClose.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                finish();

            }
        });
    }

    public String getMac() {
        return mac;
    }

    public void setMac(String mac) {
        this.mac = mac;
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getGroup_name() {
        return group_name;
    }
    public void setGroup_name(String group_name) {
        this.group_name = group_name;
    }

    public String getDevice() {
        return device;
    }

    public void setDevice(String device) {
        this.device = device;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public Date getDateCreated() {
        return dateCreated;
    }

    public void setDateCreated(Date dateCreated) {
        this.dateCreated = dateCreated;
    }

    @Override
    public String toString() {
        return "Group_Policy [mac=" + mac + ", id=" + id + ", group_name="
                + group_name + ", device=" + device + ", description="
                + description + "]";
    }

}
```

Table 5.3: The Group_Policy Class

Group policy is one of the important features that we intent to solve regrading privacy issue. The implementation of the shared file feature is presented in the next section

5.3.3 Shared Files

The high level design of the file sharing feature was described in section 5.2.3. We now further explain the implementation of the file sharing feature in this section. Mobile users can select the files to be shared. Then the system will walk them through the steps to create the metadata for those files and then stored them into the local Couchbase database.

If a mobile device has the internet connection via Wi-Fi, then users can share and sync those files to the cloud which can be shared offline once they are connected. If the mobile users choose not to share and sync files to the cloud, then they can share files to others connected mobile nodes directly via Bluetooth connection. Figure 5.12 shows a flowchart of steps involved to share a file in the system.

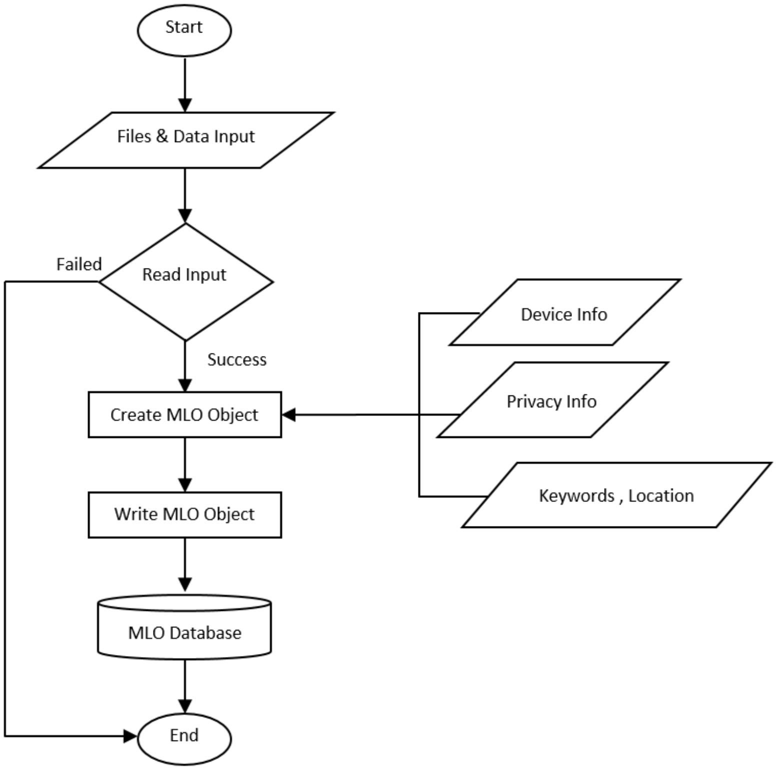


Figure 5.12: Shared Files or Documents Flowchart

Table 5.4 describes the steps involved for sharing a file. In addition, table 5.5 shows a partial source code of the implementation.

The Algorithm:

Step 1: Start

Step 2: Select data files/documents to be shared

Step 3:

```
        3a. If a source is found, then do the following:
                Update device information
                Update privacy option: public/private/protected
                        If public option is selected, then update the privacy variable
                        Else if protected is selected, then create and update group information
                        Else update the privacy variable as private.
                Else update the keyword, location, description
        3b. If the source is not found, then display message and go to step 6 and stop
Step 4: Create MLO Object
Step 5: Update the MLO to the local database
Step 6: Stop
```

Table 5.4: Shared Files

```java
private Document createFileShareItem(String text, String keyword,
                    String uri, String loc, String des, String mac) throws
Exception {
            SimpleDateFormat dateFormatter = new SimpleDateFormat(
                        "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'");
            UUID uuid = UUID.randomUUID();
            Calendar calendar = GregorianCalendar.getInstance();
            long currentTime = calendar.getTimeInMillis();
            String currentTimeString =
dateFormatter.format(calendar.getTime());
            String id = currentTime + "-" + uuid.toString();
            String file_path_convert_from_uri =
java.net.URLDecoder.decode(uri,
                        "UTF-8");
            Document document = database.createDocument();
            Map<String, Object> properties = new HashMap<String, Object>();
            properties.put("_id", id);
            properties.put("filename", text);
            properties.put("keyword", keyword);
            properties.put("share_option", "private");
            properties.put("uri", file_path_convert_from_uri);
            properties.put("loc", loc);
            properties.put("description", des);
            properties.put("mac", IMEI);
            properties.put("group", "");
            properties.put("check", Boolean.FALSE);
```

```
        properties.put("created_at", currentTimeString);
        String s = new String("");
        File file = new File(file_path_convert_from_uri);
        FileInputStream fin = null;
        try {
            fin = new FileInputStream(file);
            byte fileContent[] = new byte[(int) file.length()];
            properties.put("filelenght", file.length());
            document.putProperties(properties);
            fin.read(fileContent);
            InputStream inputStream = new
ByteArrayInputStream(fileContent);
            fin.close();
            UnsavedRevision newRev;
            try {
                newRev =
document.getCurrentRevision().createRevision();
                newRev.setAttachment("profileImage", "image/jpg",
inputStream);
                newRev.save();
                MainLearningObjectActivity.syncArrayAdapter
                        .notifyDataSetChanged();
            } catch (CouchbaseLiteException e) {
                e.printStackTrace();
            }

    MainLearningObjectActivity.syncArrayAdapter.notifyDataSetChanged();
        } catch (Exception ex) {
            ex.printStackTrace();
            Log.e("READ FILE ERROR", ex.getMessage());
            Toast.makeText(getApplicationContext(), "New file fail",
                    Toast.LENGTH_SHORT).show();
        }
        return document;
```

}

Table 5.5 Source Code to Share a File

5.3.4 Search Files

The next feature that we implement based on the designed in section 5.2.4 is the file searching feature in our system. First, sender device generates the search MLO to be send to the receiver device. Then, the received device begins to execute the query and then creates a new MLO result to be sent back to the sender device whose is in the shared file group. Now the sender user decides which files to be retrieved to be downloaded to its local device. The flowchart in figure 5.13 details the logic order that involve the steps to search a file in device B that device A is initiated.

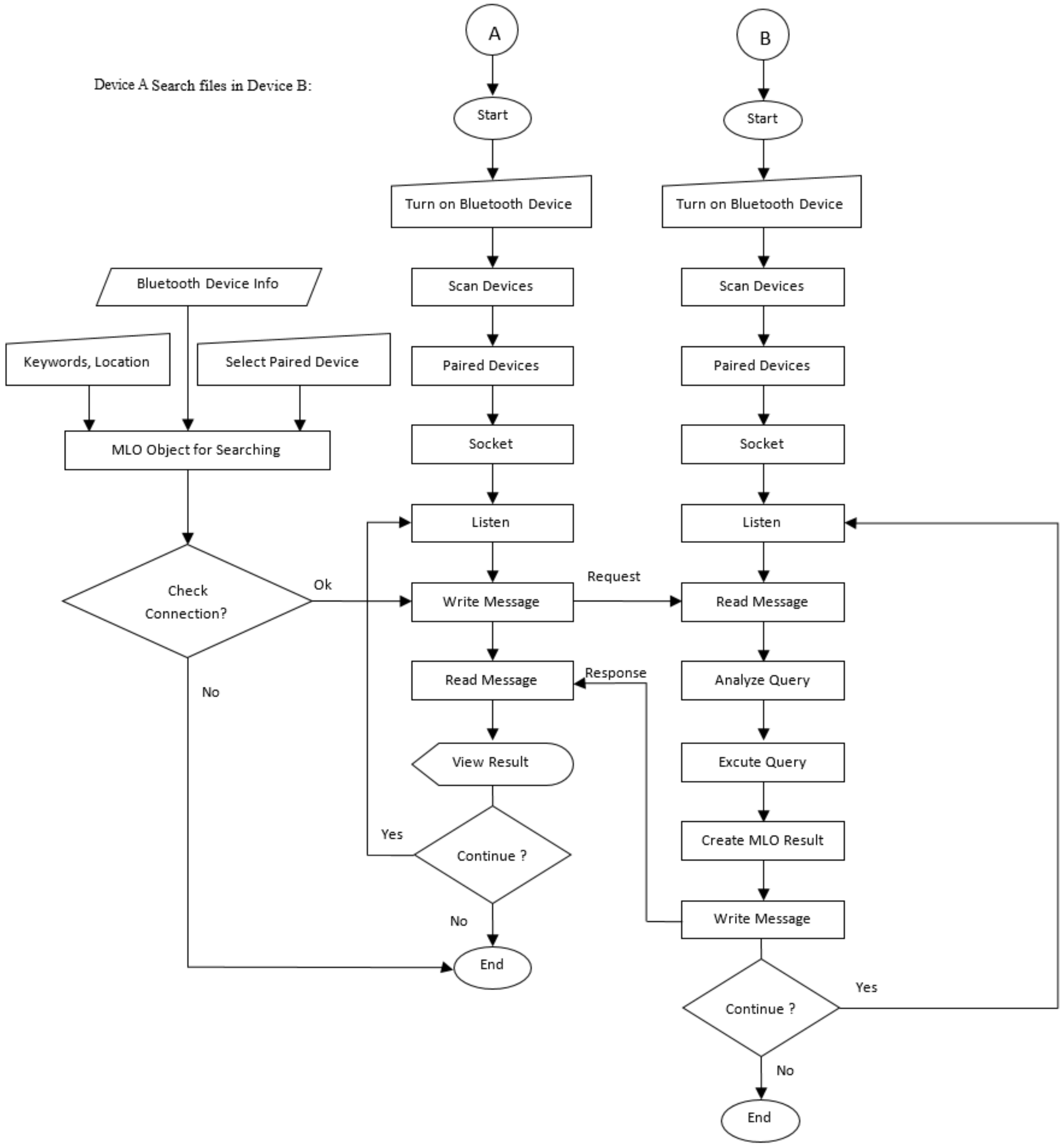


Figure 5.13: Searching Files or Document Flowchart

The follow table 5.6 shows the steps to search for a file from mobile device A to mobile device B.

```
Step 1: Start
Step 2: Turn on Bluetooth, scan and pair the neighbor devices
Step 3: Create communication socket and listening for incoming connections
Step 4: Device A update search requirement: keyword, location…and generate the MLO object.
Step 5: If the connection is existed and connected
                Device A send the MLO object request to Device B
        Else
                Go to step 8 and stop.
Step 6:
        6a. Device B read, analyze and execute the requested MLO on device B.
        6b. Create the MLO result
        6c. Send the response MLO object to Device A
Step 7:
        7a. Device A read the response MLO object from Device B
        7b. View the result
                If Device A wants to search again, then go to step 4
                Else go to step 8 and stop.
Step 8: Stop
```

Table 5.6: Searching File

The follow code in this table below show the partial code that mobile device select the paired devices from the device list before initiate the search.

```
ArrayList<DeviceData> deviceDataList = new ArrayList<DeviceData>();
        Integer i = 0;
        for (BluetoothDevice device : MainApplication.pairedDevices) {
                        deviceDataList.add(new DeviceData(device.getName(), device
                                        .getAddress()));
                }
ArrayAdapter<DeviceData> deviceArrayAdapter = new ArrayAdapter<DeviceData>(
                                MainLearningObjectActivity.this,
                                android.R.layout.simple_spinner_item, deviceDataList);
        deviceArrayAdapter
        .setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
                deviceSpinner = (Spinner) findViewById(R.id.deviceSpinner);
                deviceSpinner.setAdapter(deviceArrayAdapter);
```

Table 5.7: Display Paired Device List

The source code in table 5.8 is shown the implantation of the method called searchKeyword that allow the users to enter the keyword search or location as a text input. The system search for the matched keyword based on the metadata and response the result back either it found or not found.

```
protected void searchKeyword(String myKeyword, String senderMac)
                throws Exception {
        manager = new Manager(new AndroidContext(this), Manager.DEFAULT_OPTIONS);
        String[] arrayKeywords = myKeyword.split(",");
        String ikeyword = "";
        String ilocation = "";
        if (arrayKeywords.length > 0) {
                try {
                        ikeyword = arrayKeywords[0].trim();
                        ilocation = arrayKeywords[1].trim();
                } catch (Exception e) {
                }
            }
        CURRENT_KEYWORD = ikeyword.toLowerCase();
        CURRENT_LOCATION = ilocation.toLowerCase();
        Toast.makeText(getApplicationContext(),
                        "Keyword: " + ikeyword + " Location: " + ilocation,
                        Toast.LENGTH_LONG).show();
        Log.e(TAG, "SEARCH KEYWORD: " + CURRENT_KEYWORD);
        String docname = "query1";
        String docfilter = "filter1";
        database = manager.getDatabase(DATABASE_NAME);
        long time = System.currentTimeMillis();
        com.Couchbase.lite.View viewItemsByDate = database.getView(String
                        .format("%s/%s", docname, docfilter));
        viewItemsByDate.setMap(new Mapper() {
                @Override
        public void map(Map<String, Object> document, Emitter emitter) {
                        Object keyword = document.get("keyword");
                        Object location = document.get("location");
                        Object secue = document.get("share_option");

        if (secue != null && !secue.toString().contains("private")) {
            if ((keyword != null && keyword.toString().toLowerCase()
            .contains(CURRENT_KEYWORD))&& (location != null &&
location.toString().toLowerCase().contains(CURRENT_LOCATION)))
{
emitter.emit(keyword.toString().toLowerCase(), null);
                }
            }

        }
    }, "2." + time);
    Query query = viewItemsByDate.createQuery();
    query.setLimit(100);
    QueryEnumerator result = query.run();
    String sResult = "";
    CURRENT_RESULT = "";
    for (Iterator<QueryRow> it = result; it.hasNext();) {
        QueryRow row = it.next();
        SavedRevision currentRevision = row.getDocument()
                .getCurrentRevision();
        String id = (String) currentRevision.getProperty("_id");
        String filename = (String) currentRevision.getProperty("filename");
        String keyword = (String) currentRevision.getProperty("keyword");
        String share_option = (String) currentRevision
                .getProperty("share_option");

        String uri = (String) currentRevision.getProperty("uri");
        String des = (String) currentRevision.getProperty("des");
        String mac = (String) currentRevision.getProperty("mac");
        String group = (String) currentRevision.getProperty("group");
        JSONObject obj = new JSONObject();
        obj.put("filename", filename);
        obj.put("keyword", keyword);
        obj.put("share_option", share_option);
        obj.put("uri", uri);
        obj.put("mac", mac);
        obj.put("des", des);
        obj.put("group", group);
        sResult += (filename + " --- " + keyword + "\n");
        if (CURRENT_RESULT != "")
            CURRENT_RESULT += "<-->" + obj.toString();
        else
            CURRENT_RESULT = obj.toString();
    }
    CURRENT_FUNCTION = "SEND_RESULT";
    for (BluetoothDevice device : MainApplication.adapter
            .getBondedDevices()) {
        if (device.getAddress().contains(senderMac)) {
            Log.v(TAG, "Starting client thread");
            if (MainApplication.clientThread != null) {
                MainApplication.clientThread.cancel();
            }
            MainApplication.clientThread = new ClientThread(device,
                    MainApplication.clientHandler);
            MainApplication.clientThread.start();
        }
    }
```

```
        }
```

Table 5.8: Search File Source Code

Once the search complete and it shows the found result in the list. Users can select the specific shown files to download. The implementation of the file download is explained in the next section.

5.3.5 File Download

As we explained the design of this file download feature in the section 5.2.5 above. We show the implementation and our logic design in this section via the flowchart in figure 5.14. User decides whichever files to be downloaded, then the system would create the MLO to be sent to the receiver whose has that files in his/her local storage. Now, the files would be packed in the binary format inside the result MLO to be sent back to the requested mobile user for retrieving.

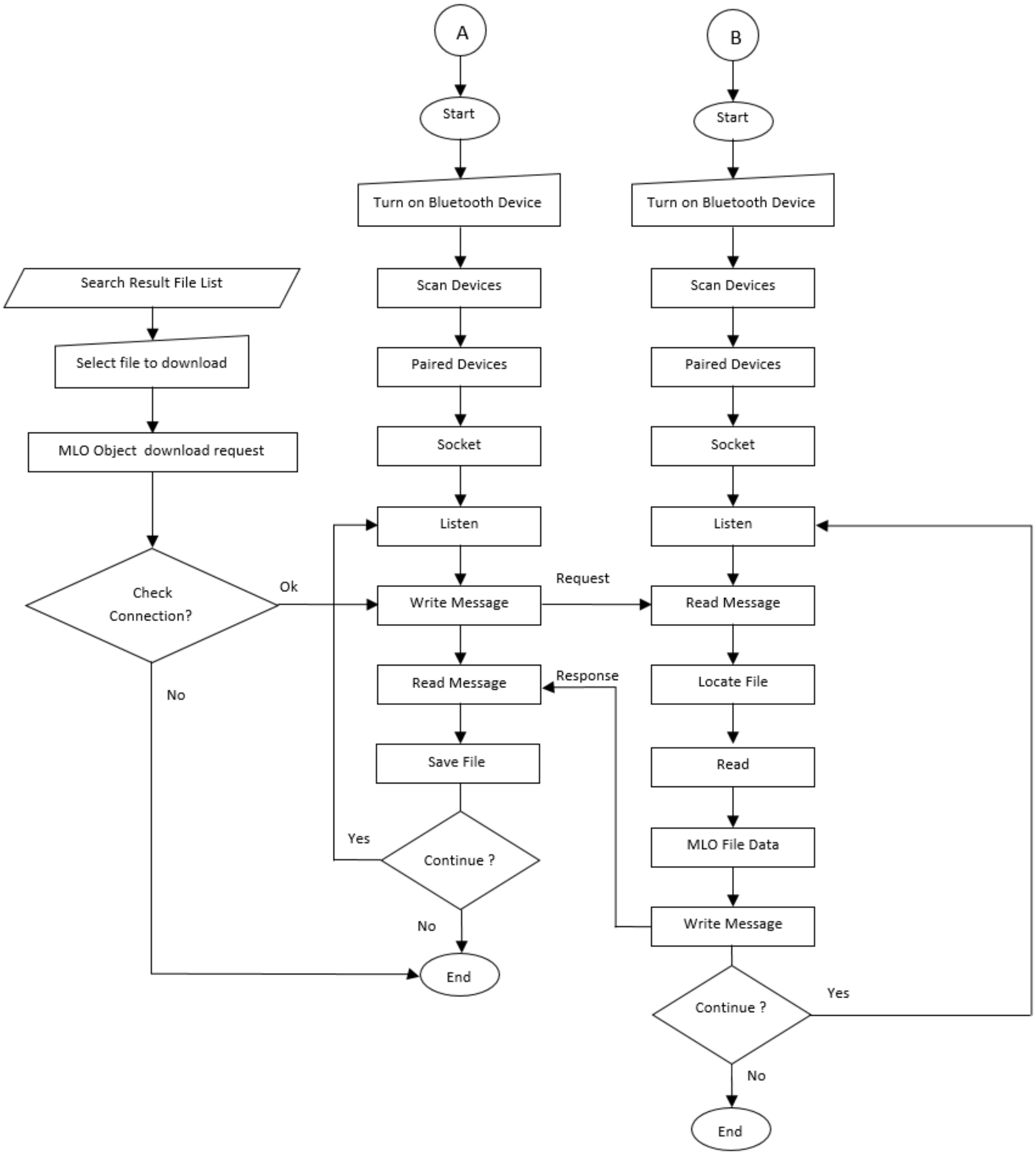


Figure 5.14: Request to Download Files or Document Flowchart

The procedure to download files from mobile device A to mobile device B is showed in the table 5.9 below.

```
Step 1: Start
Step 2: Turn on Bluetooth, scan and pair the neighbor devices
Step 3: Create communication socket and listening for incoming connections
Step 4: Device A view the research result list and select file to be download and generate the MLO object.
Step 5: If the connection is existed and connected
                Device A send the MLO object download request to Device B
        Else
                Go to last step and stop.
Step 6:
        6a. Device B read, execute the requested by locate the file and read files.
        6b. Create the MLO result
        6c. Send the response MLO object to Device A
Step 7:
        7a. Device A read the response MLO object from Device B
        7b. View and save the result
                Device A save the file
                If Device A wants to search again, then go to step 4
                Else go to step 8 and stop.
Step 8: Stop
```

Table 5.9: Download File

The source code in the table 5.10 and 5.11 below is shown the send command request to download a file and the response to download a file.

```
String keyName = "RECEIVE_FILE"; // send funciton
        byte[] keyNameData = keyName.getBytes();
        String filename = CURRENT_REQUEST_DOWNLOAD_URI;
        int cut = filename.lastIndexOf('/');
        if (cut != -1) {
            filename = filename.substring(cut + 1);
        }
        byte[] keyFileName = filename.getBytes();
        JSONObject obj = new JSONObject();
        byte[] byte_data = null;
        try {
            File file = new File(CURRENT_REQUEST_DOWNLOAD_URI);
            byte_data = new byte[(int) file.length()];
            new FileInputStream(file).read(byte_data);
            obj.put("filename", filename);
            obj.put("data", byte_data);

        } catch (Exception ex) {
            ex.printStackTrace();
        }
        byte[] keyValueData = byte_data;
        ByteArrayOutputStream byteOs = new ByteArrayOutputStream();
        DataOutputStream out = new DataOutputStream(byteOs);
        try {
            out.writeInt(keyNameData.length);
            out.write(keyNameData);
            out.writeInt(keyValueData.length);
            out.write(keyValueData);
            out.writeInt(keyFileName.length);
            out.write(keyFileName);
            out.writeInt(keyValueMac.length);
            out.write(keyValueMac);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        byte[] allWrittenBytes = byteOs.toByteArray();
        // Invoke client thread to send
        Message message = new Message();
        message.obj = allWrittenBytes;
        MainApplication.clientThread.incomingHandler.sendMessage(message);
    }
```

Table 5.10: Send a command request to download a file

This is the source code to download the file that was implemented in the method read_message_file_download.

```
public void read_message_file_download(byte[] args) throws Exception {

DataInputStream in = new DataInputStream(new ByteArrayInputStream(args));
int sizeFunction = in.readInt();
byte[] arrFunction;
arrFunction = new byte[sizeFunction];
in.read(arrFunction, 0, sizeFunction);
int sizeFileData = in.readInt();
byte[] arrFileData;
arrFileData = new byte[sizeFileData];
in.read(arrFileData, 0, sizeFileData);
int sizeFileName = in.readInt();
byte[] arrFileName;
arrFileName = new byte[sizeFileName];
in.read(arrFileName, 0, sizeFileName);
String myFileName = new String(arrFileName);
String path = "/sdcard/" + myFileName;

try {
writeFile(arrFileData, path);
} catch (Exception e) {
}
} // end read message

public void writeFile(byte[] data, String fileName) throws IOException {

FileOutputStream out = new FileOutputStream(fileName);
out.write(data);
out.close();

}
```

Table 5.11: Read to download file and write to store file

5.3.6 Bluetooth Ad-hoc Messaging

In this section, we implement the ad-hoc messaging feature that commonly known as peer-to-peer chatting. Our implementation is different since we encapsulated the chat message as string datatype into the MLO object. We use Bluetooth 4.0 as the ad-hoc communication version between mobile nodes. The flowchart in figure 5.15 is shown the procedure from connecting, send and reply a chat message in our system.

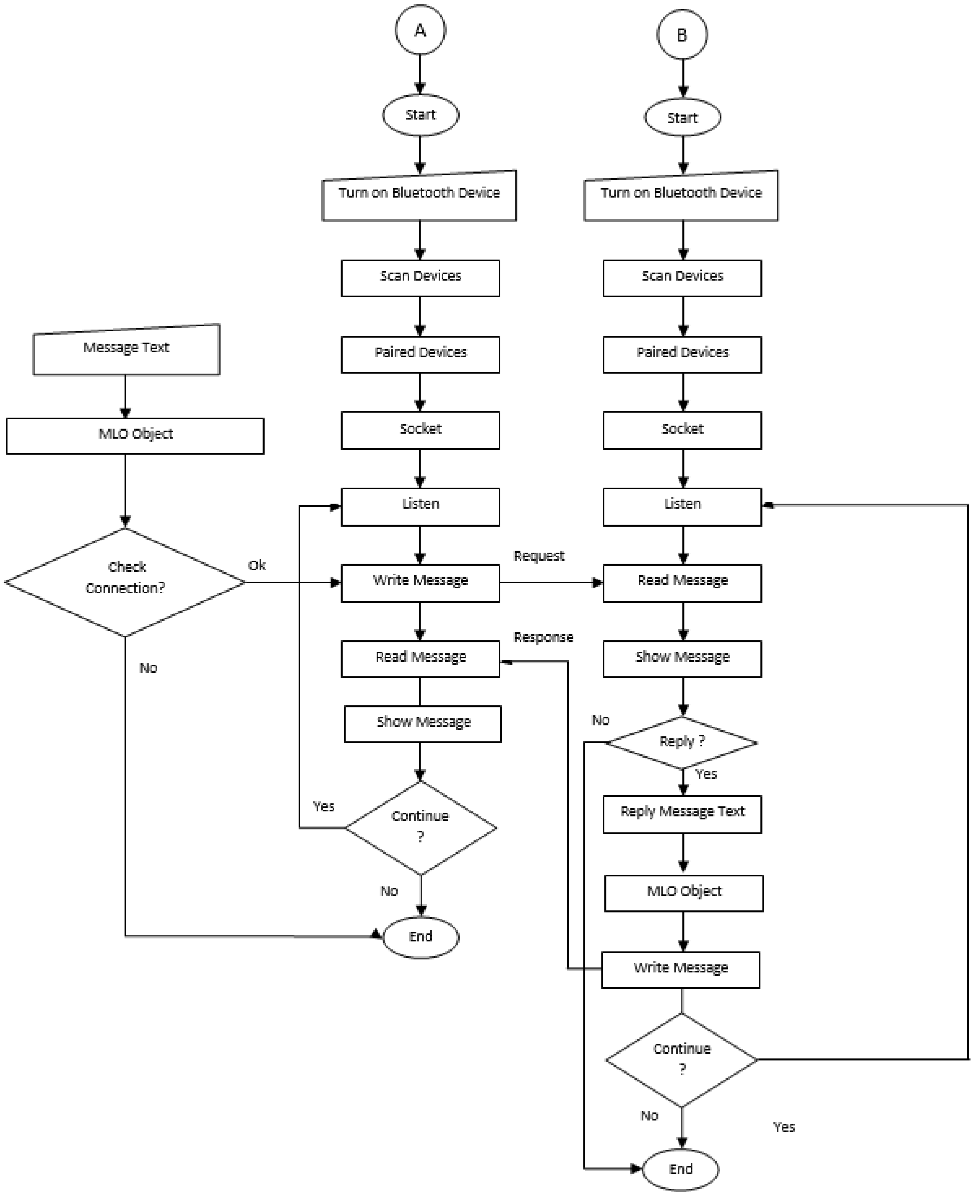


Figure 5.15: Bluetooth Messaging Flowchart

We use the high level description in this table 5.12 as the logical steps to implement this feature. The implemented source code is shown the Bluetooth chat messaging feature initiate from Device A to Device B in table 5.13 and 5.14. Both tables are shown the method to send a message and the method to show the reply message from the paired P2P mobile devices.

```
Step 1: Start
Step 2: Turn on Bluetooth, scan and pair the neighbor devices
Step 3: Create communication socket and listening for incoming connections
Step 4: Device A create a chat message and encapsulated in a MLO object
Step 5: If the connection is existed and connected
                Device A send the MLO to Device B
        Else
                Go to last step and stop.
Step 6: Device B receives, read MLO and display the message
Step 7: If Device B decides to create a reply message, then go to step 4
        Else go to step 8
Step 8: Stop
```

Table 5.12: Bluetooth Chat

SEND MESSAGE:

```
public void send_a_message()
{

        final EditText txtMsg = new EditText( MainLearningObjectActivity.this);
                        txtMsg.setHint("");


                new AlertDialog.Builder( MainLearningObjectActivity.this)
.setTitle("Send Message")
.setMessage("Enter your message text here")
.setView(txtMsg)
.setPositiveButton("Send", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int whichButton) {
          String text = txtMsg.getText().toString();
          if (text!=""){


           CURRENT_FUNCTION="SEND_A_MESSAGE";
           CURRENT_SEND_A_MESSAGE_TEXT=text;
```

```
            DeviceData deviceData = (DeviceData) deviceSpinner.getSelectedItem();
                                            for          (BluetoothDevice          device          :
MainApplication.adapter.getBondedDevices()) {
                                       if (device.getAddress().contains(deviceData.getValue())) {
                                          Log.v(TAG, "Starting client thread");
                                          if (MainApplication.clientThread != null) {
                                             MainApplication.clientThread.cancel();
                                          }
                                          MainApplication.clientThread              =              new
ClientThread(device, MainApplication.clientHandler);
                                          MainApplication.clientThread.start();
                                       }
                                    }// end if
          }
  }
})
.setNegativeButton("Close", new DialogInterface.OnClickListener() {
  public void onClick(DialogInterface dialog, int whichButton) {

  }
})
.show();

}
```

Table 5.13: Sending a Chat Message Source Code

SHOW CHAT MESSAGE AND REPLY

```
public void show_message_and_reply()

{

         if (paraFunction.toString().contentEquals("SEND_A_MESSAGE"))
                                {
         String receive_message=new String(arrKeyValue);

         final EditText txtBN = new EditText( MainLearningObjectActivity.this);

         BluetoothDevice devc =
BluetoothAdapter.getDefaultAdapter().getRemoteDevice(PARAMAC);
         txtBN.setHint("Reply here");
         new AlertDialog.Builder( MainLearningObjectActivity.this)
                  .setTitle("Message from device: [" + devc.getName() + "]")
                  .setMessage(receive_message)
```

```
                    .setView(txtBN)
                    .setPositiveButton("Reply", new DialogInterface.OnClickListener() {
                                public void onClick(DialogInterface dialog, int whichButton) {
                                        String text = txtBN.getText().toString();
                                                      if (text!="")
                                                         {


CURRENT_FUNCTION="SEND_A_MESSAGE";
CURRENT_SEND_A_MESSAGE_TEXT=text;

for (BluetoothDevice device : MainApplication.adapter.getBondedDevices()) {
   if (device.getAddress().contains(PARAMAC)) {
          Log.v(TAG, "Starting client thread");
          if (MainApplication.clientThread != null) {
                    MainApplication.clientThread.cancel();
                                                            }
            MainApplication.clientThread = new ClientThread(device,
MainApplication.clientHandler);
                                                            MainApplication.clientThread.start();
                                                         }
                                                      }// end fo
                                                         }
                                                   }
                                                 })
          .setNegativeButton("Close", new DialogInterface.OnClickListener() {
                                                   public void onClick(DialogInterface dialog, int
whichButton) {

                                                   }
                                                 })
                                                 .show();

                                 }
}
```

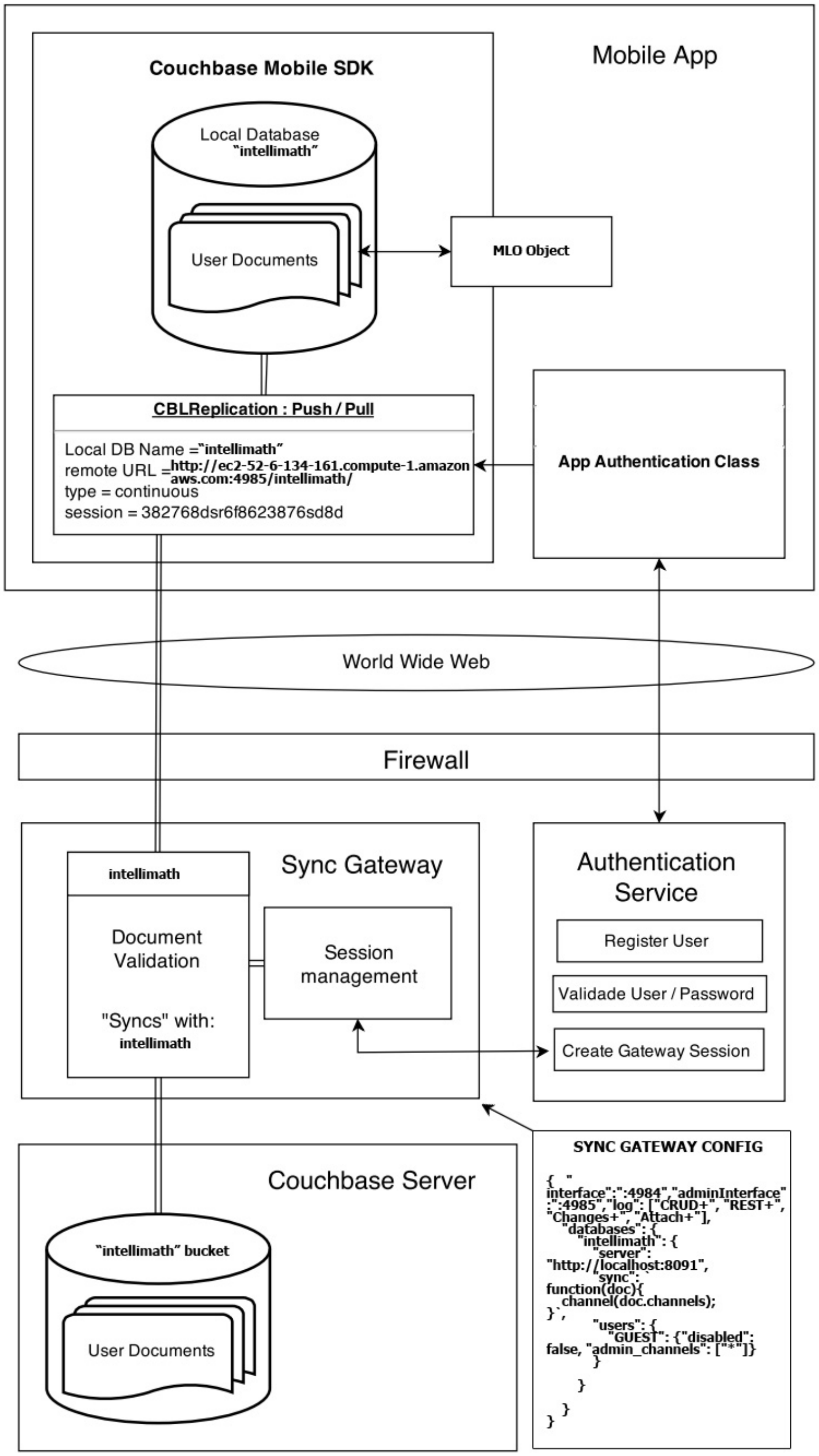
Table 5.14: Show and Reply a Chat Message Source Code

5.3.7 Hybrid Solution

The hybrid solution implementation was done by installing Couchbase server and the sync gateway server on the Amazon EC2 cloud-based services. Figure 5.16 depicts the complete Couchbase mobile synchronization architecture. It has three main components: the mobile clients, the database server and the mobile sync gateway server. Mobile clients is the mobile app that we implement in our solution. The database we used to store user documents is the embedded lightweight, document-oriented (NoSQL) database with the ability to sync the data to the server called Couchbase Lite. At the lowest level, the database is an embedded JSON database that can work standalone, in a peer-to-peer network, or as a remote endpoint for Couchbase Server. JSON is a simple way to store and exchange data using the key and value pairs. This document-based layout database is a free-form layout that uses the map reduce query engine to execute query efficiently over large data sets regardless of how users structure their data. The Couchbase Lite provides native APIs for the iOS and Android platform. It also supports replication with compatible database servers and peer-to-peer replication as well.

The second component of the architecture is the Couchbase Sync Gateway server which is the interface between Couchbase Lite and Couchbase Database Server. Its communication between mobile client nodes and server nodes is the common hypertext transfer protocol that allows the client nodes to handles access control and data routing while a single large Couchbase Server cluster can manage data for multiple users. In order to sync data, mobile clients and the server must pass the authentication, authorization and data validation steps. The authentication service checks and validates users' information and their passwords. Once verified, it creates a new gateway section to listen for client requests for synchronization services. The sync gateway service can be executed using command line or by running the sync gateway configuration file manually which includes the server and user information, port information, bucket name, and the sync function.

The third component of the architecture is the Couchbase database server which is a high-performance NoSQL distributed and scalable database with a flexible data model. Once the server components are installed, we are able to configure the gateway information to be synced to the client mobile devices. The information needed by the client devices includes the server URL, port information, bucket name information, user’s information, and the methods to be used to invoke the database. Also, the client devices would use the push and pull methods to execute synchronization calls. The push method is used to sync data from the clients to the server while the pull method is used to sync data from the server to the clients. The Couchbase Lite API included the class CBLReplication which allows the creation of a replication object that is responsible for listening and updating data changes between the client and the server.



```
{   "
interface":":4984","adminInterface"
:":4985","log": ["CRUD+", "REST+",
"Changes+", "Attach+"],
    "databases": {
        "intellimath": {
            "server":
"http://localhost:8091",
            "sync":
function(doc){
    channel(doc.channels);
}`,
            "users": {
                "GUEST": {"disabled":
false, "admin_channels": ["*"]}
            }

        }

    }
}
```

Figure 5.16: Couchbase Mobile Synchronization Architecture (adapted from http://ti.eng.br/connecting-Couchbase-lite-database-to-sync-gateway/)

INITIALIZE COUCHBASE

```
public void do_init()
{

    // create a manager
    // Manager manager;
    try {
        manager = new Manager(new AndroidContext(this),
Manager.DEFAULT_OPTIONS);
        Log.d (TAG, "Manager created");
    } catch (IOException e) {
        Log.e(TAG, "Cannot create manager object");
        return;
    }

    // create a name for the database and make sure the name is legal
    if (!Manager.isValidDatabaseName(DATABASE_NAME)) {
        Log.e(TAG, "Bad database name");
        return;
    }

    // create a new database
    // Database database;
    try {
        database = manager.getDatabase(DATABASE_NAME);
        Log.d (TAG, "Database created");
    } catch (CouchbaseLiteException e) {
        Log.e(TAG, "Cannot get database");
        return;
    }
}
```

Table 5.15: Chat Message Source Code (Initialize Couchbase)

REPLICATION CHANGE EVENT

```
@Override
public void changed(Replication.ChangeEvent event) {

    Replication replication = event.getSource();
    Log.d(TAG, "Replication : " + replication + " changed.");
```

```
      if (!replication.isRunning()) {
        String msg = String.format("Replicator %s not running", replication);
        Log.d(TAG, msg);
      }
      else {
        int processed = replication.getCompletedChangesCount();
        int total = replication.getChangesCount();
        String msg = String.format("Replicator processed %d / %d", processed, total);
        Log.d(TAG, msg);
      }

      if (event.getError() != null) {

        Toast.makeText(getApplicationContext(),  "Sync  error:  "+  event.getError(),
Toast.LENGTH_SHORT);
      }

    }
```

Table 5.16: Chat Message Source Code (Replication Change Event)

COUCHBASE SYNCSCHRONIZE

```
  private void initSync() {

      URL syncUrl;

      try {
        syncUrl = new URL(SYNC_URL);
      } catch (MalformedURLException e) {
        throw new RuntimeException(e);
      }

      pullReplication = database.createPullReplication(syncUrl);
      pullReplication.setContinuous(true);

      pushReplication = database.createPushReplication(syncUrl);
      pushReplication.setContinuous(true);

      pullReplication.addChangeListener(this);
      pushReplication.addChangeListener(this);

    }
```

```
private void startSync() {

    pullReplication.start();
    pushReplication.start();
}

private void stopSync() {

    pullReplication.stop();
    pushReplication.stop();
}
```

Table 5.17: Show and Reply a Chat Message Source Code

5.4 Testing and Validation

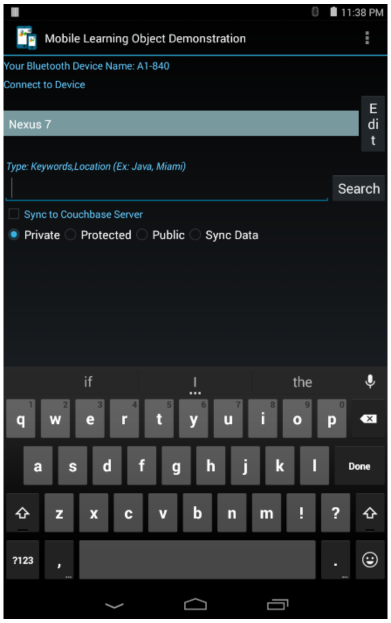
We implemented a prototype as a proof of concept based on the proposed designed in Chapter 3 focus primary on the mobile learning object and its reusability among mobile nodes in ad-hoc and wireless networks. The mobile learning application was tested on mobile phones and tablets that Google Android OS was preinstalled, specifically the Google Nexus 5, Google Nexus 7 and HP 7" tablet. We carefully tested in an ad-hoc environment by turning off the cellular connectivity on the phone and also temporarily disable the WiFi connection on those devices. We only turned on the Bluetooth on the mobile phone and tablets. Once they were paired, we tested those featured successfully as described in section 5.2.1 to 5.2.7 and 5.3.1 to 5.3.7 of this Chapter. We checked the functional requirements and tested the system against those requirements to make sure it met those intended purposes such as the mobile learning object, the MLO communication only in truly ad-hoc network environment without the use of wireless infrastructure, the ability to send and receive MLO objects from mobile sender nodes to receiver nodes, the ability to create group policy among those connected nodes, the ability to share, search and download files. During the course of software application development, we followed the procedure to perform the unit testing, border testing and system testing as well as shown in the table 5.1 below:

Features	Expected	Test Result
Constructing MLO	Create a MLO object to store node information, command request and actual MLO data.	Completed
Create Group Policy	Create group name, send request to the assigned mobile node and receive the confirmation.	Completed
Share File	Select file, then choose share type: public, private, protected.	Completed
Search File	Enter search criteria, search and receive result: either show the list or not found message.	Completed

Download File	Select a file to download. Download complete and confirm message.	Completed
Bluetooth Chat	Send, receive and reply chat messages.	Completed
Hybrid Solution using Couchbase Database	Installed, configure Couchbase Lite, Couchbase Sync Gateway, Couchbase Server	Completed

Table 5.18: Mobile Application Features Testing

Here are the screenshot of the mobile application demonstration features.



Your Bluetooth Device Name: A1-840

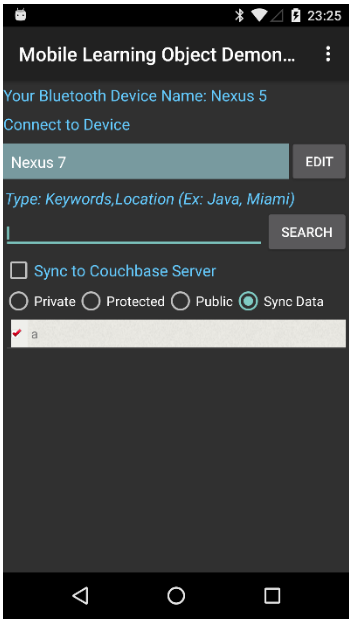
Connect to Device

Nexus 7

Type: Keywords,Location (Ex: Java, Miami)

□ Sync to Couchbase Server

● Private ○ Protected ○ Public ○ Sync Data



Your Bluetooth Device Name: Nexus 5

Connect to Device

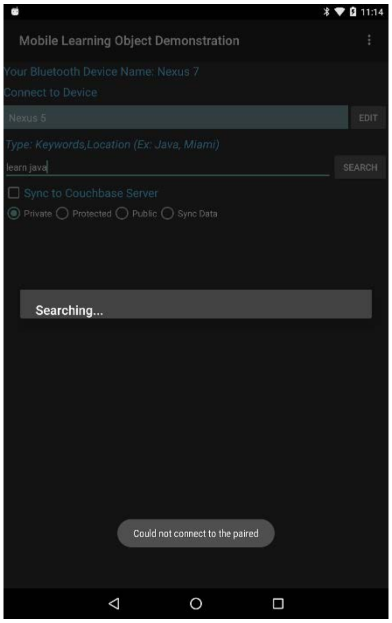
Nexus 7

Type: Keywords,Location (Ex: Java, Miami)

□ Sync to Couchbase Server

○ Private ○ Protected ○ Public ● Sync Data

✓ a



Your Bluetooth Device Name: Nexus 7

Connect to Device

Nexus 5

Type: Keywords,Location (Ex: Java, Miami)

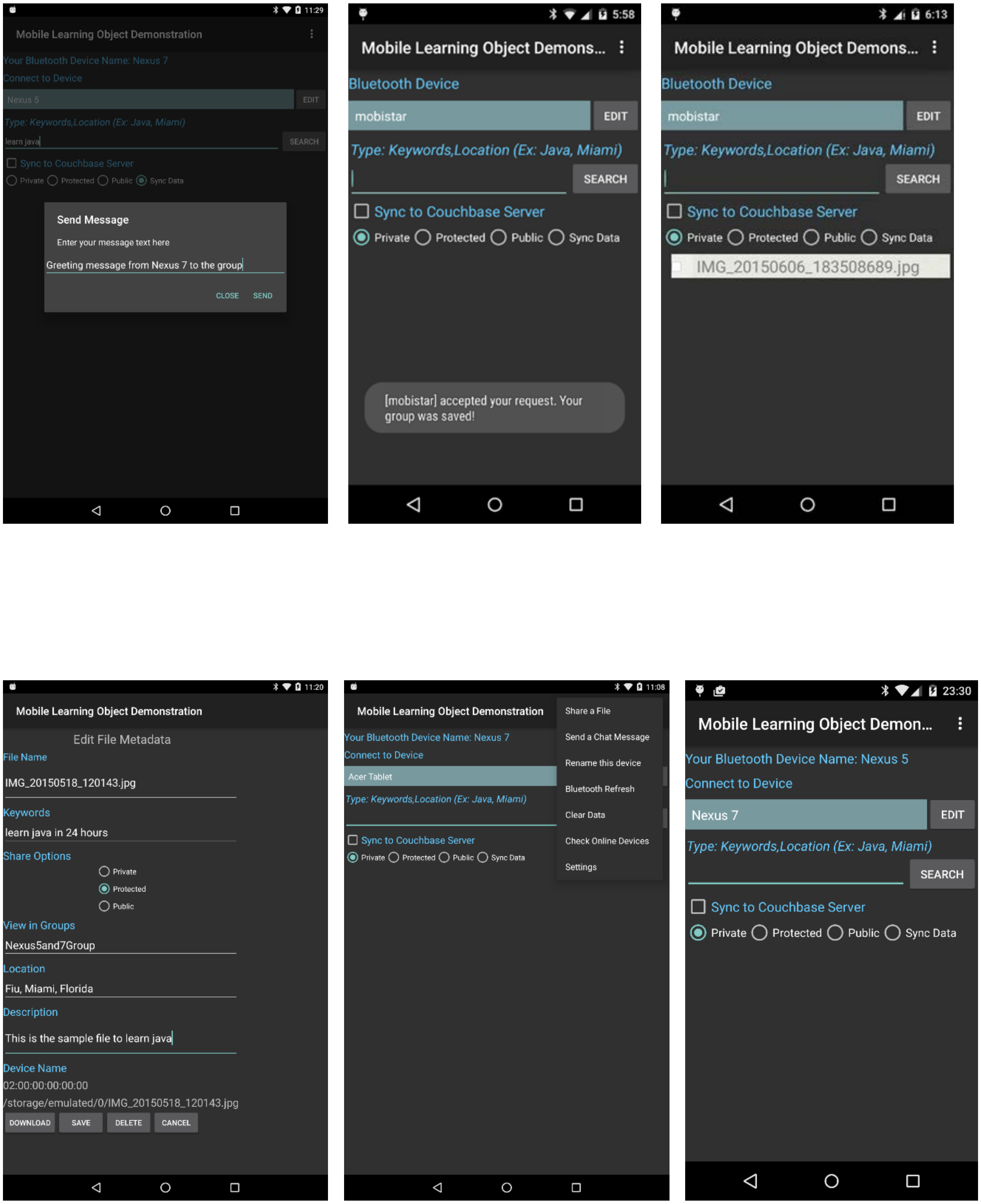
learn java

□ Sync to Couchbase Server

● Private ○ Protected ○ Public ○ Sync Data

Searching...

Could not connect to the paired



11:29

Mobile Learning Object Demonstration

Your Bluetooth Device Name: Nexus 7

Connect to Device

Nexus 5 EDIT

Type: Keywords,Location (Ex: Java, Miami)

learn java SEARCH

□ Sync to Couchbase Server

○ Private ○ Protected ○ Public ● Sync Data

Send Message

Enter your message text here

Greeting message from Nexus 7 to the group

CLOSE SEND

5:58

Mobile Learning Object Demons...

Bluetooth Device

mobistar EDIT

Type: Keywords,Location (Ex: Java, Miami)

SEARCH

□ Sync to Couchbase Server

● Private ○ Protected ○ Public ○ Sync Data

[mobistar] accepted your request. Your group was saved!

6:13

Mobile Learning Object Demons...

Bluetooth Device

mobistar EDIT

Type: Keywords,Location (Ex: Java, Miami)

SEARCH

□ Sync to Couchbase Server

● Private ○ Protected ○ Public ○ Sync Data

IMG_20150606_183508689.jpg

11:20

Mobile Learning Object Demonstration

Edit File Metadata

File Name

IMG_20150518_120143.jpg

Keywords

learn java in 24 hours

Share Options

○ Private
● Protected
○ Public

View in Groups

Nexus5and7Group

Location

Fiu, Miami, Florida

Description

This is the sample file to learn java

Device Name

02:00:00:00:00:00

/storage/emulated/0/IMG_20150518_120143.jpg

DOWNLOAD SAVE DELETE CANCEL

11:08

Mobile Learning Object Demonstration

Your Bluetooth Device Name: Nexus 7

Connect to Device

Acer Tablet

Type: Keywords,Location (Ex: Java, Miami)

□ Sync to Couchbase Server

● Private ○ Protected ○ Public ○ Sync Data

23:30

Mobile Learning Object Demon...

Your Bluetooth Device Name: Nexus 5

Connect to Device

Nexus 7 EDIT

Type: Keywords,Location (Ex: Java, Miami)

SEARCH

□ Sync to Couchbase Server

● Private ○ Protected ○ Public ○ Sync Data

Figure 5.17: Mobile Application Feature Demonstration Screenshots

5.5 Result Discussion and Conclusion

Our Android application implementation allows mobile devices to share and communicate via Bluetooth using mobile learning objects in an ad-hoc network environment. The mobile devices can communicate and share files assigned different levels of privacy: public, private or protected via group sharing or directly. Once paired and/or grouped, mobile users can search for specific MLO criteria within the group. In addition, mobile users can send file download requests to paired devices using our Mobile Learning Object method to improve efficiency of communication and reduce the overhead cost of traditional file sharing methods. The mobile application also features a Bluetooth chat service using the same peer-to-peer networking concept but our novel approach incorporates the use of MLO objects as a means of a messaging system.

Our solution also includes the use of a Couchbase cloud-based server that enables user devices to synchronize MLOs while offline as long as the devices have WiFi or cellular network connectivity. By using a more scalable and modern document-based database, our hybrid system solution is a dynamic solution that allows offline synchronization as well as reusable MLOs. In conclusion, this innovative mobile learning system proves efficient in sharing reusable mobile learning objects that work in mobile ad-hoc networks as well as in traditional mobile wireless networks.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

This chapter provides a summary of the research presented in this dissertation and discusses the future work in the areas of M-Learning in ad-hoc and wireless networks. In summary, we present the literature reviews, background and related work on M-Learning, the proposed model and design of an ad-hoc mobile learning system, the development of the mobile learning object authoring tool, the proof of concept of sharing a mobile learning object in infrastructure-less platform and extended solution to the hybrid M-Learning in ad-hoc and wireless networks. The future work section describes several topics concentrating on complexity design and integration of sensor network, cross platform implementation. This final chapter contains both conclusions and possible future work

6.1 Conclusions

Mobile learning is in high demand and is among the top 14 challenging research topics in the 21st century. This dissertation offers a cutting edge context aware and ubiquitous self-directed learning methodologies using ad-hoc and sensor networks. Mobile learning is a new idea for learning because of the ubiquitous presence of mobile devices in people’s lives and the possibility of learning anywhere from anyone it allows access to. M-learning uses mobile devices, sensors and wireless transmissions to provide anytime and anywhere access to content as well as just in-time training or review of content. It delivers anytime, at anywhere using any mobile computing platform learning. Due to the rapid increase in the number of mobile devices available to consumers, there is a need to evaluate these devices and their various features to see where they fit into the current idea of mobile learning. The emergence of lifelong learners has created a need for a new approach to learning so that learning can occur easily in a non-standard classroom environment. Many problems exist with m-learning that have prevented its widespread adoption, outside of tests and limited classroom use that have been researched but never resolved. This dissertation focuses on modeling, designing, and implementing an ad-hoc M-Learning platform that integrates sensory data to support ubiquitous learning. The emphasis of our work is on defining appropriate mobile learning objects and their adaptation descriptions as well as providing mechanisms for ad-hoc service discovery and developing concepts for the seamless integration of the learning objects and their contents with a particular focus on preserving data and privacy. We modeled mobile, "on-the-fly" learning object metadata based on the IEEE standards which characterized by its content, context, structure and categories. We also designed a mechanism for composing real-life mobile and "on-the-fly" learning objects that were characterized by mobility, context awareness and privacy. Furthermore, we implemented our proposed M-Learning system as a proof of concept to demonstrate the use of mobile learning objects for real-life mobile and "on-the-fly" learning object composition that allow mobile ad-hoc nodes to share, search and download reusable MLO objects. This research also involves a combination of model development, design and rapid prototyping of context-aware, privacy and learning objects for an ad-hoc networking platform.

6.2 Future Work

The research presented in this dissertation provides the foundation for methodology and system development in the area of ad-hoc M-Learning where mobile devices do not have or rely on the infrastructure network connectivity. It is expected that our research will shed light on how ad-hoc networking and wearable sensing technology can enable new forms of informal educational practices that actively engages student-centered learning. The design concept proposed in this dissertation are modelled based on real world interactivities such as how knowledge being shared among users on an airplane, or group of users in the remote location where no internet connection is available.

Our research will lead to technological advancements that will allow more personalized learning experiences and allow access to learning resources anytime and anywhere. Since our research is one of the first m-learning works that focuses on content exchange among the mobile ad-hoc nodes to ensure consistency and availability of information. It also provides an on-the-fly content service discovery, query request, and retrieving data from mobile nodes and sensors. Our research is also expected to show how emerging technologies can enable new forms of educational practices. One of the unique concept proposed here is the use of mobile learning objects with the ability to adapt to ad-hoc networks and mobile devices.

As mobile devices become more ubiquitous and their capabilities increase, the demand for instant communication and access to learning materials will rise and modern education must meet that challenge. More research needs to be done to design a complete and thorough system that works across different platforms. Thus, we believe that research should continue to be devoted to developing practical applications that could be used on a variety of mobile devices and supported by an abundance of mobile platforms and functions. User's adaptation is also needed to improve this system to work better to support truly ad-hoc learning anywhere at any time using any mobile devices. Also, further research efforts are required to address the challenges around security and privacy issues in the mobile learning setting and beyond. Also, as the cloud technologies are largely studied and mobile technologies are evolving, new research is needed for the development of mobile learning tools that can be deployed on cloud. The development of such learning tools requires specific development strategies for an effective abstracting of pedagogical principles at the software design and implementation level.

REFERENCES

[1] Perry, William, and Alec Broers. "Grand Challenges for Engineering." Grand Challenges for Engineering. National Science Foundation, 2008.Web. 25 May 2013. <www.engineeringchallenges.org>.

[2] " Advanced Distributed Learning” Advanced Distributed Learning Initiative RSS. Web. 01 June 2013. <http://www.adlnet.org/>.

[3] "IEEE LTSC | WG12." IEEE LTSC | WG12.Web. 01 June 2013. <http://ltsc.ieee.org/wg12/>.

[4] "Learning in the Palm of Your Hand." Welcome to TBR Mobilization & Emerging Technology. Web. 25 May 2013. <http://emergingtech.tbr.edu/learning-palm-your-hand>.

[5] Xinyou Zhao; Okamoto, T., "A Device-Independent System Architecture for Adaptive Mobile Learning," Advanced Learning Technologies, 2008. ICALT '08. Eighth IEEE International Conference, vol., no., pp.23,25, 1-5 July 2008

[6] A. Syvänen, R. Beale, M. Sharples, M. Ahonen, and P. Lonsdale, "Supporting pervasive learning environments: Adaptability and context awareness in mobile learning," Proceedings of the Third IEEE International Workshop on Wireless and Mobile Technologies in Education 2005. November 28-30, 2005, Tokushima, Japan, H. Ogata, M. Sharples, Kinshuk, and Y. Yano, Los Alamitos: IEEE Computer Society, 2005, pp. 251-253.

[7] A. Zimmermann, A. Lorenz, and R. Oppermann, "An operational definition of context," Paper presented at the Sixth International and Interdisciplinary Conference on Modeling and Using Context – The Context 07, Denmark, 20–24 August 2007, 2007.

[8] Abozeid, Amr, Mohammed A. Razek, Hosam F. El-Sofany, and Fayed F. Ghaleb. "Mobile Interactive Blended Learning System."IEEE. IEEE Education Society Students Activities Committee, Dec. 2010. Web. 25 May 2013.

[9] Ainsworth, S., Honey, M., Johnson, W. L., Koedinger, K., Muramatsu, B., Pea, R., Recker, M. & Weimar, S. (2005). Cyberinfrastructure for Education and Learning for the Future: A Vision and Research Agenda: Computing Research Association. Retrieved from http://www.cra.org/reports/ cyberinfrastructure.pdf on 8 June 2008

[10] Atkins, D. E., Brown, J. S. & Hammond, A. (2007). A review of the Open Education Resources Movement: Achievements, challenges, and new opportunities. William and Flora Hewlett Foundation.Retrieved from http://www.hewlett.org/NR/rdonlyres/5D2E3386-3974-4314-8F67-

5C2F22EC4F9B/0/AReviewoftheOpenEducationalResource sOERMovement_BlogLink.pdf on 8 June 2008.

[11] Ayala, G.; Castillo, S., "Towards Computational Models for Mobile Learning Objects," Wireless, Mobile, and Ubiquitous Technology in Education, 2008. WMUTE 2008. Fifth IEEE International Conference, vol., no., pp.153,157, 23-26 March 2008

[12] Battistella, Paulo E., Aldo V. Wangenheim, and Christiane G. Wangeheim. "Evaluation of Free Authoring Tools for Producing SCORM-Conform Learning Objects." IEEE Technology and Engineering Education. IEEE MULTIDISCIPLINARY ENGINEERING EDUCATION MAGAZINE, Dec. 2010. Web. 25 May 2013. <http://www.ewh.ieee.org/soc/e/sac/itee/index.php/meem/article/view/122>.

[13] Bishop, Barry, AtanasKiryakov, DamyanOgnyanoff, Ivan Peikov, ZdravkoTashev, and RuslanVelkov. "OWLIM: A Family of Scalable Semantic Repositories." Semantic Web – Interoperability, Usability, Applicability an IOS Press Journal. Semantic Web – Interoperability, Usability, Applicability an IOS Press Journal, Aug. 2010. Web. 25 May 2013. <http://www.semantic-web-journal.net/content/owlim-family-scalable-semantic-repositories>.

[14] Boneu, Josep. "Survey on Learning Content Management Systems - Springer."Survey on Learning Content Management Systems - Springer.Web. 01 June 2013. <http://link.springer.com/chapter/10.1007/978-1-4419-6959-0_7>.

[15] Brezillon, Patrick. "Context in Artificial Intelligence: I. A Survey of the Literature." Web. 25 May 2013. <http://www-poleia.lip6.fr/~brezil/Pages2/Publications/CAI1-99.pdf>.

[16] Brown, Elizabeth, Dirk Borner, Mike Sharples, Christian Glahn, Tim D. Jong, and Marcus Specht. "Deliverable Contribution Small Scale Study." Web. 25 May 2013. <http://www.stellarnet.eu/d/1/2/images/2/23/Sss6.pdf>.

[17] Brown, Elizabeth; Borner, Dirk; Sharples, Mike; Glahn, Christian; de Jong, Tim and Specht, Marcus(2010). Location-based and contextual mobile learning.A STELLAR Small-Scale Study.STELLAR EuropeanNetwork of Excellence in TEL (EU).

[18] CLARK, RUTH. "Six Principles of Effective E-Learning: What Works and Why." THE E LEARING DEVELOPERS' JOURNAL (2002): 1-9. David Holcombe. Web. 25 May 2013. <www.eLearningGuild.com>.

[19] FitzGerald, Elizabeth, Mike Sharples, Robert Jones, and Gary Priestnall. "Guidelines for the Design of Location-based Audio for Mobile Learning." International Journal of Mobile and Blended Learning: 70-85. Web. 25 May 2013.

<http://www.open.ac.uk/personalpages/mike.sharples/documents/Preprint_IJMBL_audio_guidelines.pdf>.

[20] Fudickar, Sebastian, and Klaus Rebensburg. "Mobile Group-Based Multimedia Cooperation System for MANETS." IEEE MULTIDISCIPLINARY ENGINEERING EDUCATION MAGAZINE. IEEE Education Society Students Activities Committee, June 2007. Web. 25 May 2013.

[21] Ganchev, Ivan, StanimirStoyanov, MairtinO'Droma, VeselinaValkanova, and Damien Meere. Pervasive InfoStation-based MLearning System.Proc. of Fifth International Conference on Systems and Networks Communications.IEEE Computer Society, n.d. Web. 25 May 2013.

[22] Gonzalez, L., and W. Ruggiero. "Collaborative E-learning and Learning Objects." IEEE Latin America Transactions 7.5 (2009): 569-77. Print.

[23] H. Bristow, C. Baber, J. Cross, J. Knight, and S.I. Woolley, "Defining and evaluating context for wearable computers," International Journal of Human Computer Studies, vol. 60, 2004, pp. 798-819.

[24] Harman, K., and A. Koohang, eds. Learning Objects: Standards, Metadata, Repositories, and LCMS. Santa Rosa, California: Informing Science, 2007. Print.

[25] Hunaiti, Z., S. Almasri, E. Sedoyeka, N. Matar, and A. Fenton. "Location Based Guided Tour M-Learning." IEEE MULTIDISCIPLINARY ENGINEERING EDUCATION MAGAZINE.N.p., Dec. 2008. Web. 25 May 2013.

[26] J. Taylor, J. Slay, and F. Kurzel, “An ontological approach to learning objects,” in Learning Objects: Applications, Implications, & Future Directions, K. Harman and A. Koohang, Eds. Santa Rosa, California: Informing Science Press, 2006, pp. 35-62.

[27] Jovanovic, Jelena, DraganGasevic, Colin Knight, and Griff Richards. "Ontologies for Effective Use of Context in E-Learning Settings." Journal of Educational Technology & Society (2007): 47-59. Journal of Educational Technology & Society.International Forum of Educational Technology & Society.Web. 25 May 2013. <http://www.ifets.info/>.

[28] Scott, K.; Benlamri, R., "Context-Aware Services for Smart Learning Spaces," Learning Technologies, IEEE Transactions, vol.3, no.3, pp.214,227, July-Sept. 2010

[29] Konig-Ries, Birgitta, and Michael Klen. "CiteSeerX — Information Services to Support E-Learning in Ad-hoc Networks."CiteSeerX — Information Services

to Support E-Learning in Ad-hoc Networks. First International Workshop on Wireless Information Systems, 2002.Web. 25 May 2013.

[30] Lin, Freya H., and Timothy K. Shih. "Automatic Trap Detection: A Debugging Mechanism for Abnormal Specification in the IMS Sequencing Controls." IEEE Transactions on Learning Technologies 1.3 (2008): n. pag. IEEE CS & ES.Web. 25 May 2013.

[31] McGreal, Rory, Billy Cheung, Tony Tin, and Steve Schafer. "Implementing Mobile Environments Using Learning Objects: The Athabasca University Digital Reading Room."2005 IEEE International Workshop on Wireless and Mobile Technologies in Education. Proc. of Implementing Mobile Environments Using Learning Objects: The Athabasca University Digital Reading Room. IEEE Computer Society, 2005.Web. 25 May 2013.

[32] Mierlus-Mazilu, Ion. 2010 International Conference on Electronics and Information Engineering. Proc. of M-learning Objects. IEEE, 2010.Web. 25 May 2013.

[33] Muyinda, Paul B. "Deploying and Utilizing Learning Objects on Mobile Phones." Diss. Makerere University, 2010.Deploying and Utilizing Learning Objects on Mobile Phones. Nov. 2010. Web. 25 May 2013. <http://mak.ac.ug/documents/Makfiles/theses/Muyinda%20Paul%20Birevu.pdf>.

[34] PaulDourish, "What We Talk About When We Talk About Context," Personal and Ubiquitous Computing, vol. 8, 2004, pp. 19-30.

[35] Pathmeswaran, R., and V. Ahmed. "SWmLOR: Technologies for Developing Semantic Web Based Mobile Learning Object Repository." The Built & Human Environment Review 2.1 (2009): Web. 25 May 2013.

[36] Qu, Changtao, and Wolfgang Nejdl. "Toward Interoperability and Reusability of Learning Resources: A SCORM Conformant Courseware for Computer Science Education." IEEE Technical Committee on Learning Technology, 2002.Web. 25 May 2013. <http://lttf.ieee.org/icalt2002/proceedings/t1603_icalt024_End.pdf>.

[37] Quinn, Clark. Designing MLearning: Tapping Into the Mobile Revolution for Organizational Performance. 1st ed.San Francisco: Pfeiffer, 2011. Print.

[38] Quinn, ClarkN. Mobile Learning: Landscape and Trends. Ed. Temple Smolen. Santa Rosa: ELearning Guid, 2011. Print.

[39] Redeker, G.H.J., "An educational taxonomy for learning objects," Advanced Learning Technologies, 2003. Proceedings. The 3rd IEEE International Conference, vol., no., pp.250,251, 9-11 July 2003

[40] S.Abiteboul, O. Benjelloun, I. Manolescu, T.Milo, and R. Weber, “Active XML: peer-to-peer data and web services integration,” in Proceedings of VLDB 2002, Hong Kong, 2002, pp. 1087-1090.

[41] Sharples, Mike, Inmaculada A. Sanchez, Marcelo Milrad, and GiasemiVavoula. "Chapter 14: Mobile Learning, Small Devices, Big Issues." KAL Legacy Mobile Learning. Print.

[42] Sharples, Mike, Josie Taylor, and GiasemiVavoula. "Towards a Theory of Mobile Learning."World Conference on Mobile and Contextual Learning.Web. 25 May 2013.

[43] Shemshadi, A.; Soroor, J.; Tarokh, M.J., "An Innovative Framework for the New Generation of SCORM 2004 Conformant E-Learning Systems," Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference, vol., no., pp.949,954, 7-9 April 2008

[44] Siadaty, M., Torniai, C., Gasevic, D., Jovanovic, J., Eap, T., Hatala, M. m-LOCO: An Ontology-based Framework for Context-Aware Mobile Learning. In Proceedings of the 6th International Workshop on Ontologies and Semantic Web for Intelligent Educational Systems at 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada, June 2008

[45] Sung, Michael, Jonathan Gips, Nathan Eagle, AnmolMadan, Ron Caneel, Rich DeVaul, JoostBonsen, and Sandy Pentland. "MIT.EDU: M-learning Application for Classroom Settings."http://web.media.mit.edu. Massachusetts Institute of Technology.Web. 25 May 2013.

[46] United States of America. National Science Foundation.Task Force on Cyberlearning.National Science Foundation. By Christine Borgman, Hal Abelson, Lee Dirks, Roberta Johnson, Kenneth R. Koedinger, Marcia C. Linn, Clifford A. Lynch, Diana G. Oblinger, Roy D. Pea, Katie Salen, Marshall S. Smith, and Alex Szalay. National Science Foundation, Web. 25 May 2013. <http://www.nsf.gov/pubs/2008/nsf08204/nsf08204.pdf>.

[47] Yau, Jane Y. "A Mobile Context-aware Learning Schedule Framework with Java Learning Objects | Jane Yin-Kim Yau - Academia.edu." A Mobile Context-aware Learning Schedule Framework with Java Learning Objects | Jane Yin-Kim Yau - Academia.edu, Apr. 2011. Web. 25 May 2013. <http://www.academia.edu/691000/A_mobile_context-aware_learning_schedule_framework_with_Java_learning_objects>.

[48] Yen, Neil Y.; Shih, T.K.; Chao, L.R.; Qun Jin, "Ranking Metrics and Search Guidance for Learning Object Repository," Learning Technologies, IEEE Transactions on , vol.3, no.3, pp.250,264, July-Sept. 2010

[49] Zavitsanos, E.; Paliouras, G.; Vouros, G.A., "Gold Standard Evaluation of Ontology Learning Methods through Ontology Transformation and Alignment," Knowledge and Data Engineering, IEEE Transactions on , vol.23, no.11, pp.1635,1648, Nov. 2011

[50] Zouaq, A.; Nkambou, R., "Building Domain Ontologies from Text for Educational Purposes," Learning Technologies, IEEE Transactions on, vol.1, no.1, pp.49,62, Jan.-March 2008

[51] NikiPissinou, and Kia Makki. Bringing Knowledge to your Palm: New Venue in Cooperative Information Access for Wireless Education, National Science Foundation proposal submission 2001

[52] H. Peng, Y. Su, C. Tsai "Ubiquitous knowledge construction: mobile learning re-defined and a conceptual framework" Innovations in Educations and Teaching International, Vol. 46, No. 2, May 2009.

[53] A. Elmorshidy "Mobile Learning – A New Success Model" The Journal of Global Business Management, Vol. 8, No. 2, August 2012.

[54] Y. Liu, H. Li "What drives M-Learning success? Drawing insights from self-directed learning theory" Pacific Asia Conference on Information Systems Proceedings, August 2009.

[55] G. Hartmann, G. Stead, A. DeGani "Cross-platform mobile development" Tribal, 2011.

[56] R. Bhalalusesa, M. Arshad "Possibilities and Challenges of deploying effective learning materials in mobile learning" International Journal of Information Technology & Computer Science, Vol. 7, No. 1, January/February 2013.

[57] Y. Park "A Pedagogical Framework for Mobile Learning: Categorizing Education Applications of Mobile Technologies into Four Types" The International Review of Research in Open and Distance Learning, Vol. 12, No. 2, 2011.

[58] P. Muyinda, J. Lubega, K. Lynch, T.Weide "A Framework for Instantiating Pedagogic MLearning Objects Applications" 2011.

[59] P. Zervas, S. Ardila, R. Fabregat, D. Sampson "Designing Tools for Context-Aware Adaptive Mobile Learning" European, Mediterranean & Middle Eastern Conference on Information Systems 2011.

[60] R. Pathmeswaran, V. Ahmed "SWmLOR: Technologies for Developing Semantic Web based Mobile Learning Object Repository" The Built & Human Environment Review, Special Issue, Vol. 1, 2009.

[61] P. Muyinda, J. Lubega, K. Lynch “Mobile Learning Objects Deployment and Utilization in Developing Countries” International Journal of Computing and ITC Research, Special Issue, Vol. 4, No. 1, October 2010.

[62] J. Agaba, B. Kanagwa “On Addressing Challenges in Instruction Design using Learning Objects” Strengthening the Role of ICT in Development, 2011.

[63] P.Muyinda, E. Mugisa, K. Lynch “M-Learning: The Education Use of Mobile Communication Devices” Advances in Systems Modeling and ICT Applications, 2007.

[64] M. Milrad, D. Spikol “Anytime, Anywhere Learning Supported by Smart Phones: Experiences and Results from the MUSIS Project” Educational Technology & Society, Vol. 10, No.4, 2007.

[65] Balci, B.; Inceoglu, M., "A web-based learning content design platform and metadata editing," Frontiers in Education Conference - Global Engineering: Knowledge without Borders, Opportunities without Passports, 2007. FIE '07. 37th Annual, vol., no., pp.F1H-13,F1H-18, 10-13 Oct. 2007

[66] Tavangarian D, Leypold ME, Nölting K, Röser M, Voigt D. “Is e-learning the solution for individual learning” Electronic Journal of E-learning 2004; 2(2):273–280.

[67] Xu S, Li Shaozi. The SCORM standard and its application in Web-based education resources building. In IT in Medicine and Education, 2008. ITME 2008. IEEE International Symposium on, Xiamen, China, 2008; 212–215.

[68] Cruz RL. Ad-hoc networks at global scale. In International Conference on Computing, Networking and Communications (ICNC), 2013, San Diego, USA, 2013; 813–817.

[69] Poltrack J, Hruska N, Johnson A, Haag Jason. The next generation of SCORM: innovation for the global force. In The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), volume 2012, Orlando, Florida, USA, 2012. NTSA.

[70] Koohang A, Floyd KS, Stewart C. Design of an open source learning objects authoring tool – the LO creator. Interdisciplinary Journal of E-Learning and Learning Objects (IJELLO) 2011; 7:111–123.

[71] Yen NY, Shih TK, Chao LR, Jin Q. Ranking metrics and search guidance for learning object repository. IEEE Transactions on Learning Technologies 2010; 3(3):250–264.

[72] Kortuem. G. Proem: A peer-to-peer computing platform for mobile ad-hoc networks. In Advanced Topic Workshop Middleware for Mobile Computing, Linkoping, Sweden, 2001.

[73] De Marcos L, Hilera JR, Gutiérrez JA, Pagés C, Martínez J-J. Implementing learning objects repositories for mobile devices. In Proceedings of the I International Conference on Ubiquitous Computing: Applications, Technology and Social Issues, ICUC 2006 Ubiquitous Computing, Alcal de Henares, Madrid, Spain, 2006.

[74] Marinai, S., "Metadata Extraction from PDF Papers for Digital Library Ingest," in Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference, vol., no., pp.251-255, 26-29 July 2009

APPENDICES

Appendix 1: MainApplication.java

MainApplication Class (MainApplication.java)

```
package com.example.mobilelearningobjectdemotration;
import android.app.Activity;
import android.app.Application;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.os.Handler;
import android.util.Log;
import android.widget.Toast;
import com.example.mobilelearningobjectdemotration.ClientThread;
import com.example.mobilelearningobjectdemotration.ProgressData;
import com.example.mobilelearningobjectdemotration.ServerThread;
import java.util.ArrayList;
import java.util.List;
import java.util.Set;

public class MainApplication extends Application {
      private static String TAG = "MOBILE_LEARNING_OBJECT/MainApplication";
      protected static BluetoothAdapter adapter;
      protected static Set<BluetoothDevice> pairedDevices;
      protected static Handler clientHandler;
      protected static Handler serverHandler;
      protected static ClientThread clientThread;
      protected static ServerThread serverThread;
      protected static ProgressData progressData = new ProgressData();
      protected static final String TEMP_IMAGE_FILE_NAME = "btimage.jpg";
      protected static final int PICTURE_RESULT_CODE = 1234;
      protected static final int IMAGE_QUALITY = 100;
      protected static final int CHOOSE_FILE_TO_UPLOAD = 101;
      protected static final int REQUEST_SEARCH = 102;
      protected static final int SEND_RESULT = 103;
      protected static final int REQUEST_DOWNLOAD = 104;
      protected static final int SEND_FILE_DOWNLOAD = 105;
      protected static final int SAVE_GROUP_TO_DEVICE = 106;
      protected static final int SEND_A_MESSAGE = 107;
      protected static final int REPLY_TO_UPDATE_GROUP = 108;
      protected static final int CHECK_ONLINE_DEVICE = 109;
      protected static final int REPLY_CHECK_ONLINE_DEVICE = 110;
      int isAppBackgrounded = 0;

      @Override
      public void onCreate() {
            super.onCreate();
            adapter = BluetoothAdapter.getDefaultAdapter();
            if (adapter != null) {
                  if (adapter.isEnabled()) {
```

```
                pairedDevices = adapter.getBondedDevices();
            } else {
                Log.e(TAG, "Bluetooth is not enabled");
            }
        } else {
            Log.e(TAG, "Bluetooth is not supported on this device");
        }
    }
    public static void device_refresh() {
        adapter = BluetoothAdapter.getDefaultAdapter();
        if (adapter != null) {
            if (adapter.isEnabled()) {
                pairedDevices = adapter.getBondedDevices();
            } else {
                Log.e(TAG, "Bluetooth is not enabled");
            }
        } else {
            Log.e(TAG, "Bluetooth is not supported on this device");
        }
    }
}
```

Appendix 2: MainLearningObjectActivity.java

```java
public class MainLearningObjectActivity extends Activity implements
             Replication.ChangeListener, OnKeyListener, OnItemClickListener,
             OnItemLongClickListener {
// Variable Declaration
      private static final String TAG = "MOBILE_LEARNING_OBJECT/MainActivity";
      public static Spinner deviceSpinner;
      private ProgressDialog progressDialog;
      public static String MAC_ADDRESS = "";
      public static String MOBI_NAME = "";
      public static String MAC_NAME = "";
      public static String IMEI = "";
      // CouchBase
      protected static Manager manager;
      private Database database;
      private LiveQuery liveQuery;
      public static SyncArrayAdapter syncArrayAdapter;
      public static final String DATABASE_NAME = "mobile-object-1";
      public static final String designDocName = "mobile-object-type";
      public static final String byDateViewName = "byDate";
      public static final String SYNC_URL = "http://ec2-52-6-134-161.compute-1.amazonaws.com:4985/intellimath/";
      protected static String GROUP_LOOKUP_VALUE = "";
      protected static String IMEI_LOOKUP_VALUE = "";
      public static String CURRENT_FUNCTION = "";
      public static String CURRENT_REQUEST_DOWNLOAD_URI = "";
      public static android.content.Context CURRENT_CONTEXT;
      protected static String CURRENT_KEYWORD = "";
      protected static String CURRENT_KEYWORD_VALUE = "";
      protected static String CURRENT_RESULT = "";
      protected static String CURRENT_LOCATION = "";
      protected static String CURRENT_SEND_A_MESSAGE_TEXT = "";
      protected static Integer CURRENT_DEVICE_INDEX = 0;
      public static String MAC_GROUP = "";
      public static String GROUP_CREATE_BY_IMEI = "";
      public static String GROUP_NAME_SAVE_TO_DEVICE = "";
      protected static String MY_GROUP_DEFINE = "";
      // main screen
      protected EditText addItemEditText;
      protected ListView itemListView;
      public static Context myContext;
      protected Replication pullReplication;
      protected Replication pushReplication;
      // -----------------------------
      public static String PARAMAC = "";
      public static Activity CHILD_ACTIVITY;
      protected static boolean ALERT_DIALOG_RETURN = false;
      protected static Document RETRIEVED_DOCUMENT;
      protected static Map<String, Object> UPDATE_PROPERTIES;
      protected static String RECEIVE_GROUP_STRING;
      private Vector<AlertDialog> dialogs = new Vector<AlertDialog>();
      protected static String OPTION_GROUP_VIEW_VALUE = "private";
```

```
    public static List<BluetoothDevice> mybt = new
ArrayList<BluetoothDevice>();
    public static Message CURRENT_CLIENT_MESSAGE;
    public static Integer PAIRED_ERROR = 0;
    public static String COLLECT_RESULT = "";
    protected Dialog myDialog = null;
    protected Stack<String> STACK_MESSAGE_DEVICE = new Stack<String>();
    protected Queue<String> QUEUE_SEARCH_DEVICE = new LinkedList<String>();
    protected List<CharSequence> ONLINE_DEVICE_LIST = new
ArrayList<CharSequence>();

@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_learning_object);
        BTStateChangedBroadcastReceiver.mlo = this;
        registerReceiver(new BTStateChangedBroadcastReceiver(),
                new
IntentFilter(BluetoothAdapter.ACTION_STATE_CHANGED));
        mybt.clear();
        BluetoothAdapter myBluetoothAdapter = BluetoothAdapter
                .getDefaultAdapter();
        Set<BluetoothDevice> pairedDevices = myBluetoothAdapter
                .getBondedDevices();
        List<String> s = new ArrayList<String>();
        for (BluetoothDevice bt : pairedDevices) {
            s.add(bt.getName());
            mybt.add(bt);
        }
        itemListView = (ListView) findViewById(R.id.itemListView);
        addItemEditText = (EditText) findViewById(R.id.addItemEditText);
        addItemEditText.setOnKeyListener(this);
        IMEI = getIMEI();
        initItemListAdapter();
        RadioGroup rbtgroup = (RadioGroup)
findViewById(R.id.rdgViewOption);
        rbtgroup.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup arg0, int arg1) {
                RadioButton rb = (RadioButton)
arg0.findViewById(arg1);
                if (true) {
                    OPTION_GROUP_VIEW_VALUE =
rb.getTag().toString();
                    try {

    MainLearningObjectActivity.this.startCBLite();

    itemListView.setAdapter(syncArrayAdapter);
                    } catch (Exception e) {
                        Log.e("Log", "Error");
                    }
                }
            }
        });
        Button btnEditGroup = (Button) findViewById(R.id.editGroup);
        btnEditGroup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                DeviceGroupActivity.database = database;
                DeviceGroupActivity.manager = manager;
                DeviceData deviceData = (DeviceData) deviceSpinner
                        .getSelectedItem();
                Intent i = new Intent(getApplicationContext(),
                        DeviceGroupActivity.class);
                i.putExtra("MAC_ADDRESS_PARA",
deviceData.getValue());
                String text =
deviceSpinner.getSelectedItem().toString();
                JSONObject obj = new JSONObject();
                try {
                    obj = group_lookup("mac_address",
deviceData.getValue()
                            .toString());
                    i.putExtra("DEVICE_NAME", text);
                    i.putExtra("GROUP_NAME_PARA",
obj.getString("group_name")
                            .toString());
                    i.putExtra("GROUP_ID_PARA",
obj.getString("_id").toString());
                    startActivityForResult(i, 1);
                } catch (Exception e) {
                    Toast.makeText(getApplicationContext(),
                            "Error lookup: " +
e.getMessage(),
                            Toast.LENGTH_SHORT).show();
                }
            }
        });
        CheckBox chkSync = (CheckBox) findViewById(R.id.checkSync);
        chkSync.setOnCheckedChangeListener(new OnCheckedChangeListener()
{
            @Override
            public void onCheckedChanged(CompoundButton buttonView,
                    boolean isChecked) {
                if (isChecked == true) {
                    startSync();
                    Toast.makeText(getApplicationContext(),
                            "Start sync to couchbase
server!",
                            Toast.LENGTH_LONG).show();
                } else {
                    stopSync();
                    Toast.makeText(getApplicationContext(),
                                              "Stop sync to couchbase
server!", Toast.LENGTH_LONG)
                                              .show();
                          }
                    }
             });
             try {
                    startCBLite();
             } catch (Exception e) {
                    Log.e(TAG, "Error initializing CBLite", e);
             }
```

```
// Client Handler Function
```

```
             MainApplication.clientHandler = new Handler() {
                    @Override
                    public void handleMessage(Message message) {
                          CURRENT_CLIENT_MESSAGE = message;
                          byte[] paraMac = MAC_ADDRESS.getBytes();
                          switch (message.what) {
                          case MessageType.READY_FOR_DATA: {
                                 PAIRED_ERROR = 0;
                                 if (CURRENT_FUNCTION ==
"REPLY_CHECK_ONLINE_DEVICE") {
                                       Intent i = new
Intent(getApplicationContext(),
                                                    TemplateActivity.class);
                                       startActivityForResult(i,

      MainApplication.REPLY_CHECK_ONLINE_DEVICE);
                                       break;
                                 }
                                 if (CURRENT_FUNCTION ==
"REPLY_TO_UPDATE_GROUP") {
                                       Intent i = new
Intent(getApplicationContext(),
                                                    TemplateActivity.class);
                                       startActivityForResult(i,

      MainApplication.REPLY_TO_UPDATE_GROUP);
                                       break;
                                 }
                                 if (CURRENT_FUNCTION ==
"SAVE_GROUP_TO_DEVICE") {
                                       Intent i = new
Intent(getApplicationContext(),
                                                    TemplateActivity.class);
                                       startActivityForResult(i,

      MainApplication.SAVE_GROUP_TO_DEVICE);
                                       break;
                                 }
                                 if (CURRENT_FUNCTION == "SEND_A_MESSAGE") {
                                       Intent i = new
Intent(getApplicationContext(),
                                TemplateActivity.class);
                        startActivityForResult(i,

    MainApplication.SEND_A_MESSAGE);
                        break;
                    }
                    if (CURRENT_FUNCTION ==
"CHECK_ONLINE_DEVICE") {
                        Intent i = new
Intent(getApplicationContext(),
                                TemplateActivity.class);
                        startActivityForResult(i,

    MainApplication.CHECK_ONLINE_DEVICE);
                        break;
                    }
                    if (CURRENT_FUNCTION ==
"SEND_REQUEST_DOWNLOAD") {
                        String keyName = "DOWNLOAD";
                        byte[] keyNameData =
keyName.getBytes();
                        String keyValue =
CURRENT_REQUEST_DOWNLOAD_URI;
                        byte[] keyValueData =
keyValue.getBytes();
                        ByteArrayOutputStream byteOs = new
ByteArrayOutputStream();
                        DataOutputStream out = new
DataOutputStream(byteOs);
                        try {

    out.writeInt(keyNameData.length);
                            out.write(keyNameData);

    out.writeInt(keyValueData.length);
                            out.write(keyValueData);
                            out.writeInt(paraMac.length);
                            out.write(paraMac);
                        } catch (Exception ex) {
                            ex.printStackTrace();
                        }
                        byte[] allWrittenBytes =
byteOs.toByteArray();
                        message = new Message();
                        message.obj = allWrittenBytes;

    MainApplication.clientThread.incomingHandler
                                .sendMessage(message);
                        break;
                    }
                    if (CURRENT_FUNCTION == "SEND_FILE") {
                        Intent i = new
Intent(getApplicationContext(),
                                                     TemplateActivity.class);
                                        i.putExtra("welcome", "I - SEND FILE");
                                        startActivityForResult(i,

       MainApplication.PICTURE_RESULT_CODE);
                                        break;
                                 }
                                 if (CURRENT_FUNCTION == "SEND_SEARCH") {
                                        Intent i = new
Intent(getApplicationContext(),
                                                     TemplateActivity.class);
                                        i.putExtra("welcome", "I - SEND
SERACH");
                                        startActivityForResult(i,

       MainApplication.REQUEST_SEARCH);
                                        break;
                                 }
                                 if (CURRENT_FUNCTION == "SEND_RESULT") {
                                        byte[] keyValueMac =
MAC_ADDRESS.getBytes();
                                        String keyName = "GET_RESULT";
                                        byte[] keyNameData =
keyName.getBytes();
                                        String keyValue = CURRENT_RESULT;
                                        byte[] keyValueData =
keyValue.getBytes();
                                        ByteArrayOutputStream byteOs = new
ByteArrayOutputStream();
                                        DataOutputStream out = new
DataOutputStream(byteOs);
                                        try {

       out.writeInt(keyNameData.length);
                                              out.write(keyNameData);

       out.writeInt(keyValueData.length);
                                              out.write(keyValueData);

       out.writeInt(keyValueMac.length);
                                              out.write(keyValueMac);
                                        } catch (Exception ex) {
                                              ex.printStackTrace();
                                        }
                                        byte[] allWrittenBytes =
byteOs.toByteArray();
                                        Message message1 = new Message();
                                        message1.obj = allWrittenBytes;

       MainApplication.clientThread.incomingHandler
                                                     .sendMessage(message1);
                                        break;
                                 }
                                 if (CURRENT_FUNCTION == "SEND_FILE_DOWNLOAD")
//
                                 {
                                        Intent i = new
Intent(getApplicationContext(),
                                                     TemplateActivity.class);
                                        i.putExtra("welcome", "I - SEND
RESULT");
                                        startActivityForResult(i,

       MainApplication.SEND_FILE_DOWNLOAD);
                                        break;
                                 }
                           }
                           case MessageType.COULD_NOT_CONNECT: {
                                 PAIRED_ERROR = 1;

       Toast.makeText(MainLearningObjectActivity.this,
                                              "Could not connect to the
paired",
                                              Toast.LENGTH_SHORT).show();
                                 Log.e(TAG, "COULD NOT CONNECT");
                                 if (CURRENT_DEVICE_INDEX < mybt.size()
                                              && CURRENT_DEVICE_INDEX != -1
                                              && CURRENT_FUNCTION ==
"SEND_A_MESSAGE") {
                                        CURRENT_DEVICE_INDEX++;

       send_message_to_device(CURRENT_DEVICE_INDEX,

       CURRENT_SEND_A_MESSAGE_TEXT);
                                 }
                                 if (CURRENT_DEVICE_INDEX < mybt.size()
                                              && CURRENT_DEVICE_INDEX != -1
                                              && CURRENT_FUNCTION ==
"CHECK_ONLINE_DEVICE") {
                                        CURRENT_DEVICE_INDEX++;

       send_check_online_device(CURRENT_DEVICE_INDEX);
                                 }
                                 if (CURRENT_DEVICE_INDEX < mybt.size()
                                              && CURRENT_DEVICE_INDEX != -1
                                              && CURRENT_FUNCTION ==
"SEND_SEARCH") {
                                        CURRENT_DEVICE_INDEX++;
                                        search_to_device(CURRENT_DEVICE_INDEX);
                                 }
                                 if (CURRENT_DEVICE_INDEX >= mybt.size()
                                              && CURRENT_FUNCTION ==
"CHECK_ONLINE_DEVICE") {
                                        show_online_device();
                                 }
                                 if (CURRENT_DEVICE_INDEX >= mybt.size()
                                              && CURRENT_FUNCTION ==
"SEND_SEARCH") {
                                       if (myDialog != null)
                                              myDialog.dismiss();
                                       if (COLLECT_RESULT.toString() != "") {
                                              SearchResultForm.main_activity =
MainLearningObjectActivity.this;
                                              Intent myIntent = new Intent(

       MainLearningObjectActivity.this,

       SearchResultForm.class);
                                              myIntent.putExtra("result",
COLLECT_RESULT);
                                              startActivity(myIntent);
                                       } else {

       Toast.makeText(MainLearningObjectActivity.this,
                                                           "No file found
matched", Toast.LENGTH_SHORT)
                                                           .show();
                                       }
                                 }
                                 break;
                          }
                          case MessageType.SENDING_DATA: {
                                 Log.e(TAG, "SHOW PROJECT");
                                 if (PAIRED_ERROR == 0) {
                                       progressDialog = new ProgressDialog(

       MainLearningObjectActivity.this);
                                       progressDialog.setMessage("Sending
request...");
                                       progressDialog

       .setProgressStyle(ProgressDialog.STYLE_SPINNER);
                                       progressDialog.show();
                                 } else {

       Toast.makeText(MainLearningObjectActivity.this,
                                                     "No Show progress",
Toast.LENGTH_SHORT).show();
                                 }
                                 break;
                          }
                          case MessageType.DATA_SENT_OK: {
                                 if (progressDialog != null) {
                                       progressDialog.dismiss();
                                       progressDialog = null;
                                 }
                                 if (CURRENT_DEVICE_INDEX < mybt.size()
                                              && CURRENT_DEVICE_INDEX != -1
```

```
                            && CURRENT_FUNCTION ==
"SEND_A_MESSAGE") {
                        CURRENT_DEVICE_INDEX++;

    send_message_to_device(CURRENT_DEVICE_INDEX,

    CURRENT_SEND_A_MESSAGE_TEXT);
                    }
                    if (CURRENT_DEVICE_INDEX < mybt.size()
                            && CURRENT_DEVICE_INDEX != -1
                            && CURRENT_FUNCTION ==
"CHECK_ONLINE_DEVICE") {
                        CURRENT_DEVICE_INDEX++;

    send_check_online_device(CURRENT_DEVICE_INDEX);
                    }
                    if (CURRENT_DEVICE_INDEX < mybt.size()
                            && CURRENT_DEVICE_INDEX != -1
                            && CURRENT_FUNCTION ==
"SEND_SEARCH") {
                        CURRENT_DEVICE_INDEX++;
                        search_to_device(CURRENT_DEVICE_INDEX);
                    }
                    break;
                }
                case MessageType.DIGEST_DID_NOT_MATCH: {

    Toast.makeText(MainLearningObjectActivity.this,
                            "File was sent, but didn't go
through correctly",
                            Toast.LENGTH_SHORT).show();
                    break;
                }
                }
            }
        };
```

VITA

HIEN NGUYEN

2015	Doctoral Candidate in Computer Science
	Florida International University, Miami, Florida
2003	Master of Science in Telecommunications and Networking
	Florida International University, Miami, Florida
2002	Bachelor of Science in Computer Science
	Florida International University, Miami, Florida

PUBLICATIONS AND PRESENTATIONS

1. H. Nguyen ,M. Adjouadi, M.L. Lucas, E. L. Pozo, K. Maynard, S. Thomas, A. Barreto, S. Graham, N. Rishe. "Content-Based Image Retrieval." Proceedings of the 2007 International Conference on Enterprise Information Systems and Web Technologies (EISWT-07), Orlando, FL, USA., pp. 26-31, July 9-12, 2007.

2. Nguyen, H.; Pissinou, N.; Iyengar, S.S., "Enabling on-the-fly learning for mobile ad-hoc wireless networks using Bayesian theory," Computing, Management and Telecommunications (ComManTel), 2014 International Conference on , vol., no., pp.223,227, 27-29 April 2014.

3. Nguyen, H., Wahman, E., Pissinou, N., Iyengar, S.S., and Makki, K. "Mobile learning object authoring tool and management system for mobile ad-hoc wireless network", International Journal of Communication Systems. May 2015.