Machine learning and dynamic user interfaces in a context aware nurse application environment

Article *in* Journal of Ambient Intelligence and Humanized Computing · June 2016

DOI: 10.1007/s12652-016-0384-1

3 authors, including:

Amir Dirin
Haaga-Helia ammattikorkeakoulu

Teemu Laine
Luleå University of Technology

Machine Learning and Dynamic User Interfaces in a Context Aware Nurse Application Environment

Nathaniel Ham
Uxit Consulting
Espoo, Finland
ham.nathan88@gmail.com

Amir Dirin
Business Information Technology,
Haaga-Helia University of Applied Sciences
Helsinki, Finland
amir.dirin@haaga-helia.fi

Teemu H. Laine*
Department of Information and Computer Engineering,
Ajou University
Suwon, Republic of Korea
teemu@ubilife.net

Abstract— The increasing usage of smartphones in daily life has received considerable attention in academic and industry driven research to be utilized in the health sector. There has been development of a variety of health-related smartphone applications. Currently, however, there are few to none applications based on nurses' historical or behavioral preferences. Mobile application development for the health care sector requires extensive attention to security, reliability, and accuracy. In nursing applications, the users are often required to navigate in hospital environments, select patients to support, read the patient history and set action points to assist the patient during their shift. Finally, they have to report their performance on patient related activities and other relevant information before they leave for the day. In a working day, a nurse often visits different locations such as the patient's room, different laboratories, and offices for filling reports.

There is still a limited capability to access context relevant information on a smartphone with minimal recourse such as Wi-Fi triangulation. The Wi-Fi triangulation signals fluctuate significantly for indoor location positioning. Therefore, providing relevant location based services to a mobile subscriber has become challenging. This paper addresses this gap by applying machine learning and behavior analysis to anticipate the potential location of the nurse and provide the required services. The application concept was already presented in IMCOM 2015. This paper focuses on the process to ascertain a user's context, the process of analyzing and predicting user behavior, and finally, the process of displaying the information through a dynamically generated UI.

Keywords-component; Mobile Application; Machine Learning; Behavioral Analysis

*Corresponding author contact: Paldal Hall 1012, Ajou University, Worldcup-ro 206, Yeongtong-gu, Suwon, Republic of Korea; Tel. +82-10-219-3549; Fax. +82-31-219-1621

1 Introduction

In today's hospitals, nursing staff are finding themselves increasingly stressed due to various reasons such as over-working and under-staffing (Van Bogaert et al. 2009). As work-related demands continue to increase, nurses must be outfitted with the most current technology in order to best complete their job quickly and efficiently. Research shows that hospital staff would greatly benefit from a mobile application that not only aggregates a majority of the data and services required for daily tasks, but also provides context-aware capabilities based on the high mobility and dynamic nature of the job (Prgomet et al. 2009). Having automatic predictions of user behavior and form generation would reduce the work load of filling reports and allow nurses to spend more time on patient care.

The Context-Aware Nurse Assistant (CANA) project aims to develop an application that is both mobile and context-aware to address the issues faced by the nursing staff. We have previously presented a conceptual design of the application (Dirin et al. 2015). In this paper, we describe the design and implementation of a context-awareness module through which CANA can assist nurses by providing context-sensitive information in any work context.

In order to predict what information would be most useful to a nurse in any given context (location, time, current activity), a machine learning (Alpaydın 2014) algorithm will be employed. This algorithm will be trained by a nurse's past behavior, and based on the training data it will learn what resources are most useful to the nurse when a certain action is performed. This will simplify Human Machine Interactions (HMI) (Cannan and Hu 2011) and allow the nurse to focus primarily on the current patient.

In addition, in order to adapt to each user as a unique individual, there will be a second machine learning server that will create forms and display information dynamically. This will allow each user to have their own customized UI that conforms to their changing usability needs.

In this paper, there is a literature review section which overviews the pros and cons of geolocation methods for determining context, machine learning algorithms for making predictions, and algorithms for developing a dynamic user interface.

The application is being developed in cooperation with Haaga-Helia UAS located in Finland, Ajou University in South Korea, and HES-SO UAS in Switzerland. From the private sector, consulting is provided by UXIT Consulting.

2 Objectives and Scope

The objectives of this paper are:

1) Find a quick and cost effective approach to determining a user's context while indoors. In order to properly implement context-aware services, this system must have a reliable approach to determining the user's physical location through which other context data can be inferred. For outdoor locations, there already exists a wide variety of open source satellite-based technologies that can be leveraged. However, satellite-based technology can be unreliable indoors and alternate approaches must be adopted to ensure consistency.

2) Determine an effective machine learning algorithm for predicting user's behavior patterns based on the context and location data provided from the objective above. After analyzing the data, the system must then be able to make accurate and quick predictions on the user's behavior.

3) After having made an accurate prediction on anticipated user behavior, the system aims to dynamically generate a personalized UI with information that is immediately useful based on the user's current context.

4) After completing the above objectives, the goal is to implement these objectives into a singular, cohesive proof-of-concept application.

Throughout the entire process, it is a goal for the CANA project to adhere to User-Centered Design concepts in order to create an application that meets and adapts to a user's usability needs. To that effect, when at all possible, test users will be incorporated into the testing process to get immediate feedback on newly implemented features.

3 Methodology

3.1 User-Centered Design

The term User-Centered Design (UCD) originated by Don Norman during 1980s after publication of book named: *User-Centered System Design, New Perspective on Human-Computer Interaction* (Norman and Draper 1986). UCD together with participatory design are common methods for designing commercial products (Wilkinson and De Angeli 2014). In addition, UCD is a common method for developing smart products. The UCD approach has also been applied in the healthcare sector for various purposes. For example, Lerouge et al. (Lerouge et al. 2013) conducted a literature review on applied UCD methods and principles to support the development of consumer informatics technology for diabetic patient in various scenarios.

Similarly, Ratwani et al. (Ratwani et al. 2015) analyzed the usability of the UCD processes of eleven electronic health record vendors. They categorized the vendor's UCD practice into three groups: well developed UCD, basic UCD and miscomputations of UCD.

3.2 Concept Development Methodology: the mLUX Framework.

For the CANA application concept development, we applied the mobile application development framework (mLUX) (Dirin and Nieminen 2015). The mLUX framework consists of three layers: 1. The role-players, 2. The process, and 3. The context of use. Figure 1 presents the overview of the mLUX development framework, and Figure 2 describes the three layers in detail.

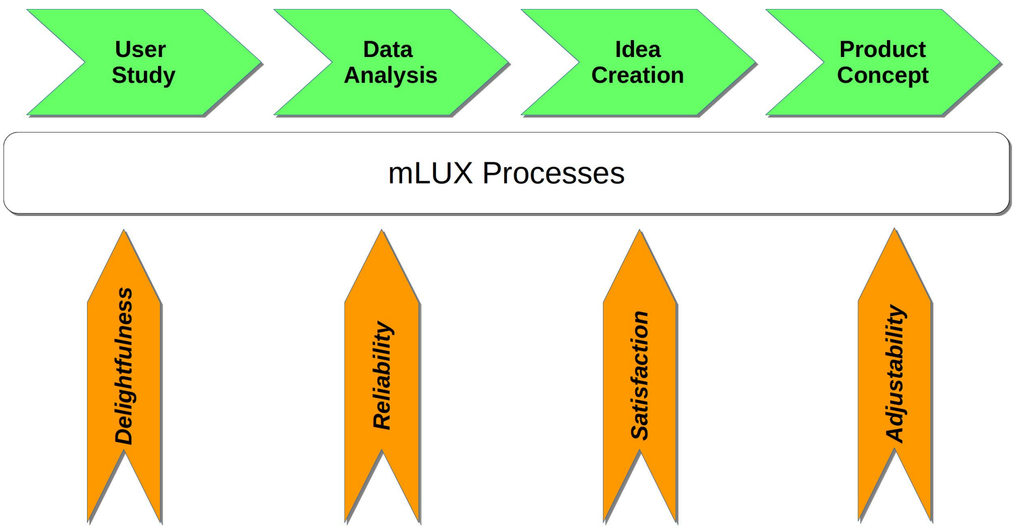

Figure 1. Overview of the mLUX development framework

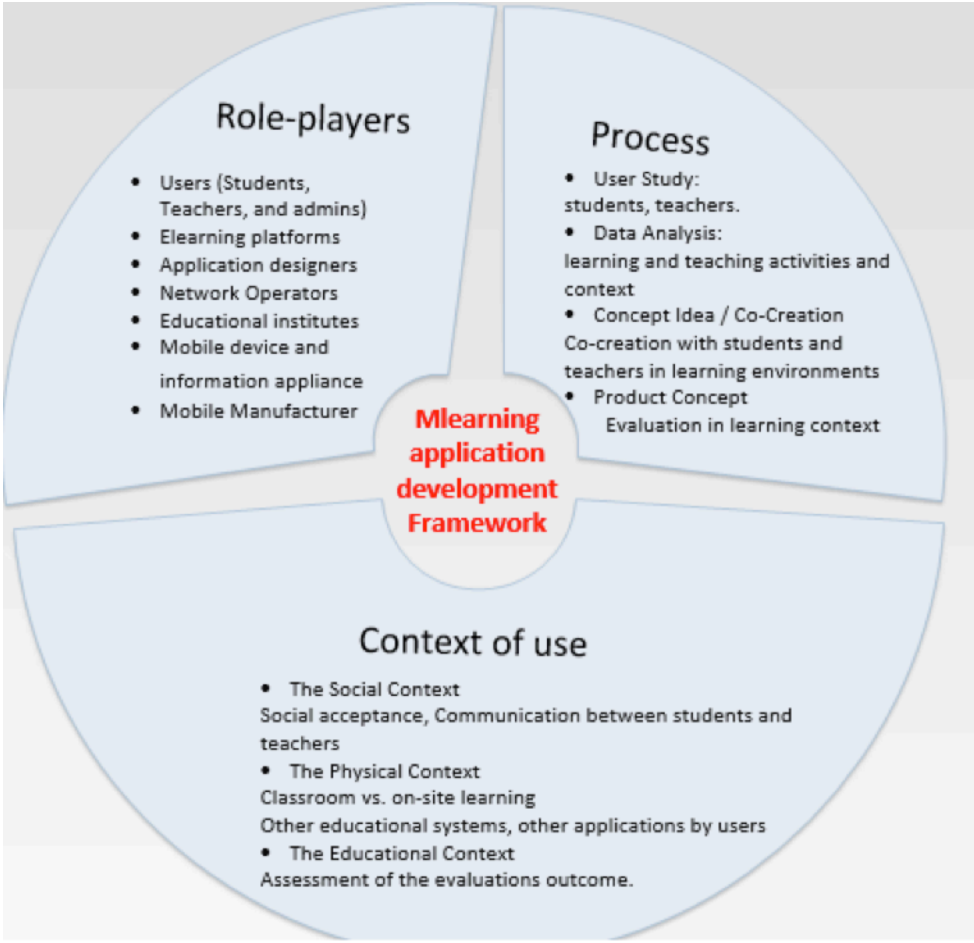

Figure 2. Layers of the mLUX development framework

3.2.1 User Study Phase (Understanding the User)

At this phase the stakeholders of the CANA application are identified and their real needs are analyzed by applying various user study methods such as questionnaires and interviews. The user study methods help the designers to learn about the users and their environments in which they often interact. *Standard* UCD methodology recommends six to twenty target users' involvement at various user study stages (Beyer and Holtzblatt 1998) .

3.2.2 Data Analysis Phase (Identifying the Interactions)

At this phase, the gathered data are analyzed by applying various data analysis methods such as transcript coding, user task and environment analysis and affinity diagram to classify and categorize the users' requirements.

3.2.3 Idea Creation Phase (Producing Design Solution)

At this phase we need to confirm the categorized and prioritized requirements with the potential test users. This confirmation ensures that designers understood the users' needs properly. Additionally, this is yet another opportunity

for target users to impact the proposed requirements. We present the categorized requirements as scenarios. Scenarios are proved to be an appropriate approach that we promote in the mobile learning application concept design method, as the scenario speaks user's language and avoids technical terms and complexities.

3.2.4 Product Concept (Evaluating the Designs)

In this phase, the application concept is modified based on the gathered data in the previous phase. The application concept is now ready to be implemented as a non-functional prototype. The prototype consists of the potential user interface components and the navigations of various screens. In the last stage of users' involvement in the design process, we conduct usability evaluation test for the proposed application prototype. The target application design refinement is based on the usability test results.

The UCD for mobile learning application is an iterative design method. In this method, the concept development mandates the users' involvement at all phases of the mobile learning application development, which minimizes the application failure and maximizes the penetration of the application among the target users.

3.3 Methodology Implementation

The CANA project was started to specifically address the needs of nurses working in hospitals. The goal was to have university students work both on the development process while also conducting research. One of the major focuses of the CANA project was to develop a solution that conforms to UCD principles. Because of this, there was close cooperation with healthcare professionals in both design and user testing phases.

During the requirements phase of the development lifecycle, the scope and objectives were set by the product owner and only revised minimally based on feedback from user testing.

In the autumn of 2014, an initial proof-of-concept was created for the user interface and was tested in a hospital. At the time of this writing, the user interface is available in both Finnish and English.

4 Background Study

The attempt to use algorithms that apply computer learning functionality to perform better has a long history. In his book, Mitchell (1997) studied various machine learning algorithms. The Bayesian method (Tipping 2001) has been very popular both by academicians and practitioners in recent years. Similarly, pattern recognition is one of the applications of machine learning (Bishop 2006).

4.1 Determining User's Context

One of the key components of the CANA project is the ability for the application to determine the user's physical location. Knowing the user's physical location is a large part of a user's current "context". In order to properly supply the machine learning algorithm with helpful data, great care must be taken when implementing the location determining service.

For outdoor geolocation, there are a wide range of solutions that provide a great deal of both speed and accuracy available under open source licensing. However, determining a user's location becomes more difficult when that location is inside a large building, such as a hospital. The metal and concrete in large structures interferes with the GPS signal and the service will be unreliable at best, and impossible at worst.

In order to work around this limitation, there are several alternatives that can be leveraged to still provide accurate, quick and reliable geolocation.

4.1.1 GPS Dead Reckoning

GPS dead reckoning is a modern incarnation of techniques used in navigation aboard naval vessels. The main concept relies on taking an initially known position, and making an educated guess on the user's current position based on initial speed and direction (Ojeda and Borenstein 2007). This method can be used to fill in gaps in traditional GPS services, which frequently occurs in large buildings. This method would conceivably work by taking a user's initial location, speed and bearing at predetermined locations and extrapolate the user's future location with a series of calculations and educated assumptions.

The main issue with GPS dead reckoning is its high level of uncertainty, as all of the results are at best "an educated guess" (Beauregard and Haas 2006). While this has the possibility of being correct and accurate, it also has the possibility of being very inaccurate. As hospitals can be quite busy, it would be troublesome to properly calculate the average length of time it would take to get from point A to point B. In addition, each individual possesses a different pace length and speed. Calibration would be required to properly determine the actual speed of the nurses while in transit between locations.

Since this method would extend the usage of existing GPS features on mobile devices, no extra sensors or devices would have to be installed. This makes the method reasonable from a cost perspective. However, developing algorithms to track and predict user movement would prove both costly and also time consuming for those responsible for the implementation.

4.1.2 Wi-Fi Positioning

Wi-Fi positioning (Yang and Shao 2015) systems are ideal for indoor environments where traditional GPS access is limited or not available. This method leverages the increase in availability of wireless access points in most public and commercial locations. A user's location is determined by the resulting strength of the wireless signal(s) at their current position. As with traditional triangulation techniques, the more reference points being used, the more accurate the location measurements will be (Evennou and Marx 2006). In order to properly employ a wireless access point, a virtual map of sorts will need to be constructed that has the location of each wireless access point, along with its SSID and MAC address. With a carefully laid out virtual structure, a user would then continue their activities as normal, while the mobile device would track its position in relation to the access points currently available.

One of the benefits of implementing this technology is that many, if not most, modern buildings already have wireless access points installed. Although additional access points would still have to be installed to ensure greater accuracy, this would reduce the initial cost and complexity of the installation. Additionally, virtually all modern mobile devices are standardly equipped with wireless access features. Again, this makes this option more attractive as new sensors or devices would not have to be purchased to run the app at a user level.

As accuracy determines heavily on the presence of stable access points, there are several factors that can reduce the accuracy of the triangulation calculations. The first factor that affects positional accuracy is signal strength/availability. In such a system, the presence of humans will greatly affect the signal propagation and therefore strength. On a slow day at a hospital, the available signals will have much more strength as there are fewer bodies to absorb signals. But at a time when there are many people present at the hospital, the signal will be absorbed and behave in a much different way. This would give a user false readings as the signal would not behave in a predictable or reliable manner.

The cost for this technology depends greatly on the size of the facility in question and the existing wireless infrastructure. If a hospital already has an extensive wireless network, additional installation costs may be marginalized. However, regardless of building size, the location of each wireless access point will still have to be determined. As most hospitals tend to be large in nature, the costs involved for installation could be considerable.

In order to turn signal strength and wireless access into a meaningful location, the application must run calculations and algorithms to produce the user's position (Yang and Shao 2015). These calculations contribute to the complexity of implementing this technology. All of the above mentioned factors of signal strength must be taken into consideration and accounted for. During the installation phase, precise calibration and testing would have to be done to determine the location of the wireless access points in relation to each other. This feature would make the technology less appealing to a client as implementation would require a larger cost and complicated algorithms.

Overall, the Wi-Fi positioning method should be considered primarily for hospitals with a large existing wireless network and where a high degree of accuracy is not required. Hospitals of any size could also consider it if they find that human traffic stays relatively consistent (although, again, high accuracy cannot be guaranteed)

4.1.3 Near Field Communication

Newer generations of smart phones and mobile devices have begun to include NFC (Near Field Communication) sensors to enhance networking and inter-device communications. Unlike its predecessor, RFID (Radio Frequency Identification), NFC tags support encryption and are thus much more secure (Madlmayr et al. 2008). As hospitals deal primarily with private personal information, such a technology would lend itself very useful in protecting location and sensitive information. Another feature of NFC is its short range. While this may at first sound like a disadvantage, it again supports security based on confidence of location. For example, it is not possible to access a specific NFC tag without being in basic proximity to it (0.2 m) (Coskun et al. 2013). In application, this technology would work by placing NFC tags at physically meaningful locations, such as one per department, or where more accuracy is required, one per patient room or lab. When a nurse enters the desired area, she would need only to scan her device over the tag and the app would automatically update its location based on the tags predetermined location in relation to the hospital.

As with wireless capabilities, most modern mobile devices are equipped with NFC capabilities. This would mean no special installation would be required client-side. Since the technology is established, major mobile operating systems (Android, iOS, Windows Phone, etc…) provide open APIs to implement these features.

In keeping with the cost constraints, NFC tags have a relatively low cost (approx. 1 Euro per tag at the time of writing). Since there is no cost for client-side sensors or installation, a hospital could conceivably install the NFC tags for under 100 euros, depending on their size and intended level of accuracy.

A particular downside to utilizing the NFC tags is that the nurse would have to physically introduce the mobile device to the tag. While not a particularly complicated gesture, it does require the nurse to remember to swipe the device when location is changing.

The complexity of installing the NFC tags amounts to registering the tag id number and assigning it to a relative location in the hospital. Installation of the tags physically is as simple as attaching a sticker to a surface.

4.2 Analyzing and Predicting User Activity with Machine Learning

4.2.1 What is machine learning?

Machine learning is a specific discipline under artificial intelligence that aims to implement algorithms that can programmatically learn from an input source and provide predictions as output. This will allow the algorithm to adapt to changing datasets automatically without input from the user (Domingos 2012).

4.2.2 Supervised vs Unsupervised Learning

In machine learning, there are two main categories, supervised and unsupervised learning. The first category, supervised learning, gives the algorithm a specific set of instructions along with a definition of what the predictions should look like (James et al. 2013). For example, the algorithm can be shown a data set containing a time of day, weather and a sales amount. The algorithm then knows the format it can expect the data to be in, and what an example of a "good" prediction is. This method is useful in situations when the data to be analyzed is consistent. In unsupervised learning, none of the input data is labeled, and the algorithm must decide what the appropriate output should be based on data clustering from the example data (Ghahramani 2004). This method is useful in scenarios when there would be a wide range of acceptable predictions. A commonly used example would be facial recognition. For the CANA application, supervised learning will be used to help the predictive service determine which outcomes we are looking for and provide it with examples of data.

4.2.3 Classification vs Regression

Another subset of machine learning is classification and regression (Alpaydın 2014). A regression algorithm deals with continuous numeric data analysis, while classification handles nominal labels. Calculating sales figures based on the time of day and weather would be an example of regression, while predicting what movie a user should watch based on past favorite movies would be a classification problem (Kotsiantis 2007). In the case of the CANA project, the service will need to be able to predict a specific activity for the user. This would be an example of classification as nominal values are being used.

4.2.4 Common Classification Approaches

4.2.4.1 Naïve Bayes

A Naïve Bayes classifier calculates the probability that a specific outcome will belong to a certain class (Zhang 2004). In this context, naïve simply means that the algorithm assumes that all input variables are completely independent and do not influence each other in any way. While one of the simpler solutions, Naïve Bayes classifiers are highly scalable and are often one of the most cost effective solutions.

4.2.4.2 Decision Tree Learning

Decision Tree Learning has a fundamentally different approach than the Naïve Bayes. This approach seeks to divide all of the training data into sub-categories until the data either matches the prediction or does not match the prediction. Each "tree" is split according to possible decisions (Su and Zhang 2006). Each node represents a decision to be made, while each branch represents an option. Each end node will represent a classification that can be predicted.

4.2.4.3 SVM (Support Vector Machines)

A support vector machine is a non-probabilistic binary linear classifier (Noble 2006). An SVM splits the data along a hyperplane and effectively divides the data into two sets (binary classifier).

5 RELATED WORK

Context-aware solutions have been widely used in various industries including in the health care sector. Adomavicius et al. (Adomavicius et al. 2011) proposed the Context-Aware Recommender Systems (CARS) which generate a recommendation by adapting them to the specific contextual situation of the users. The contextual information helps to create an intelligent and useful recommendation system. In addition, Rendle et al. (Rendle et al. 2011) suggested that the choice, which is made in a specific situation, is important information for the recommender system. The CARS can use this information to make predictions e.g., Tucker tensor factorization model (Li et al. 2013).

Context-aware applications and solutions in health care sectors have increased significantly by technological standards in recent years. For example, context-aware health self-monitoring (Chen et al. 2011) (or smart health (Solanas et al. 2014)) presents a context-aware health system in a smart city. In another example, Fenza et al. (Fenza et al. 2012) applied fuzzy logic to automatize the context information detection in order to provide the appropriate health care service, which meets the actual user's context. Similarly, De Maio et al. (De Maio et al. 2011) recommended a framework based on Fuzzy Cognitive Maps (FCM) to support knowledge processing and resource discovery based on emergency situations. Surendran et al. (Surendran et al. 2013) proposed a context-aware biomedical robot platform for elderly health care. The robot contains environmental parameter monitoring sensors and it works through voice commands. The system interprets the environmental contexts and makes context-aware decisions based on sensed information. Similarly, the role of smart phones is also increasing in health care sectors such as a context-aware mobile application to monitor diabetes (Preuveneers et al. 2013). World Health Organization (World Health Organization 2011) presented a compact analysis on the role of smart mobiles in the health sector which is aimed to increase mHealth awareness among policy-makers. Finally, Eriksen et al. (Eriksen et al. 2014) studied the importance of usability and user experience in mobile health from patient and health care sectors. They applied participatory design with a multi-disciplinary team to design a mHealth application for diabetes type 2 patient.

6 IMPLEMENTATION

6.1 Architecture

The following sections detail the architectural aspects of the CANA application.

6.1.1 Hardware Requirements

At this stage of development, the mobile client is designed to run on Android devices. CANA's prediction server relies on a Linux server environment, but has the capability to port to a Windows server environment.

6.1.2 Software Requirements

The mobile client is optimized for Android devices. The predictive services and HTTP server run on Python 3.5 in 64-bit environments.

6.1.3 Database Model

During the prototype phase of the project, the machine learning service will rely on a simple CSV file to store and retrieve user behavior. The CSV file is hosted on the application server for easy access by the prediction service.

6.1.4 Physical Architecture Model

The client side features of the CANA application run on a mobile device running an Android OS. The device's NFC reader is also facilitated anytime the device is introduced to an NFC tag. This introduction to an NFC tag deploys a call to the server side features.

6.1.5 Application Architecture Model

The NFC feature of the CANA application runs on the mobile client as a background process. This process will be triggered anytime the mobile device is in proximity of an NFC tag. The NFC tags will be programmed with the hospital's room number. When the process is triggered, it sends an http request to the machine learning service residing on the server containing the user id, time of day, and current location of the user.

The prediction and analysis of user behavior is located on the server and will handle client requests with standard http requests. The process will require the client to provide the current user id, the time of day, and the current location of the user. This data is then put into nominal form for the algorithm to properly make the prediction. Next, the application will load previous stored data into memory and fit the data into a decision tree. At this point, the client's data is then processed according to the previous results and the most likely result is sent to the client. Finally, the process updates the database with the most current user action.

When the client receives the result of the prediction, it will match the result provided from the server with the corresponding form. The UI is then appropriately updated with the information that the user is looking for. As the user continues to use the form, the database is updated dynamically via AJAX methods.

A Windows Communication Foundation (WCF) client is planned to handle all server requests. This will run automatically in the background and will be responsible for fetching predictions and informing the user of UI updates. The WCF will make the query to the machine learning service in order to provide a secure connection. This will ensure that only users with proper access rights can access the sensitive data. Figure 3 outlines the structure of the architecture.

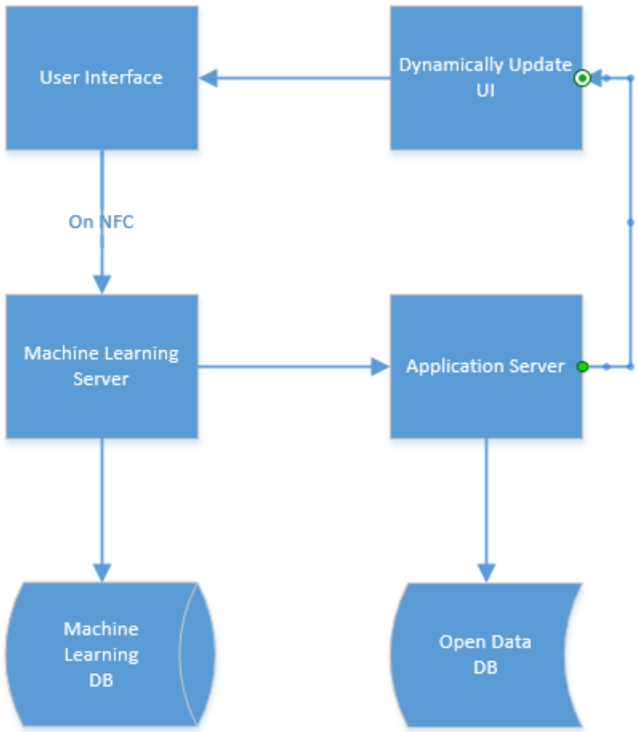

Figure 3. Overview of Client and Server

6.2 Technology

The following tools and technologies were implemented in throughout the development process of the CANA project.

6.2.1 User Positioning

In order to effectively and reliably determine the user's position indoors, the following approaches were used.

6.2.1.1 Hardware

NFC TAGS

In the first two implementations, NFC tags were used to determine a user's location and context. When a user would scan their mobile device to the NFC tag, the CANA project would send the NFC tag's id (which corresponds with a physical location in the database), the current time, and the user id.

ANDROID PHONE

At this phase in the development cycle, the CANA project is being developed solely for Android devices. The device being used for testing is a Samsung S III (I9300) and runs Android OS version 4.0 (Ice Cream Sandwich).

6.2.1.2 Software

PHONEGAP/ANDROID SDK

The open source distribution of Cordova, PhoneGap, was used in conjunction with the Android SDK to deploy solutions to an Android device for testing. Using PhoneGap allowed for the use of NFC libraries written in JavaScript. By using the JavaScript libraries, more focus could be put on testing and development, instead of spending time writing native Java code. PhoneGap also provides CLI tools for deployment which were used to package solutions and transferring them to a mobile device. In addition, Phonegap offers platform dependent support and would allow for the same project to be deployed for iOS, Windows Phone, Android or Tizen.

NFC LIBRARY

Chariot Solutions provides an open source library compatible with PhoneGap that gives access to several key NFC reading features. In the case of the CANA project, the library was used to access the phone's NFC reader. It was on top of this library that the CANA project built its NFC capabilities.

6.2.2 Predicting User Behavior

In order to predict the user's behavior, the following technologies have been used.

6.2.2.1 Software

AMAZON WEB SERVICES (AWS)

Amazon Web Services provides remote computing services, allowing for an implementation of cloud services. AWS was chosen as our cloud server because an academic version was available. In addition, the server is flexible and allows for different versions of Python to be installed and configured. Since the machine learning was written in Python version 3.4 (64-bit), it was imperative to have a server environment with the same version. Finally, AWS also offers SSH connections, which allowed a degree of customization for web services and HTTP hosting.

SCIKIT-LEARN

Scikit-learn is a library built in Python with a wide range of tools focused on machine learning. Scikit-learn was chosen because it offers solutions for algorithms in Naïve Bayes Classifiers, decision trees and support vector machines (SVMs). Using Scikit-learn to implement the algorithms allowed more focus to be placed on the proper application of the algorithm. In addition, Scikit-learn provides graphical features for generating images to represent prediction results and show a graphical implementation of the decision tree.

NUMPY

Numpy is an extension of the Python programming language and allows for n-dimensional arrays which are useful for handling large sets of data. In the CANA project, Numpy is used to efficiently create arrays and store the data in. This provides an efficient manner for storing the data currently being used in memory once it has been loaded from the data source.

FLASK

In order to handle HTTP requests and responses from both the server and the client, Flask is used on the server side to simplify this process. Flask simplifies the URL routing and will handle user authentication during the prototype phase of the CANA project. Flask is intended only for development stage and a more robust HTTP Python server will be used during actual deployment.

ANACONDA

Anaconda is a free distribution of Python, which bundles a wide variety of popular Python packages for ease of install and configuration. This includes Scikit-Learn, Numpy, and Flask.

6.2.3 Displaying Dynamic Form Information

6.2.3.1 Hardware

ANDROID PHONE

As with the NFC portion of the CANA project, an Android phone is used to dynamically display the information predicted by the server.

6.2.3.2 Software

The software portion of generating the user interface will be also handled with Python on the server side. On the client side, Phonegap will again be used to handle requests and responses from the server and updating of the user interface.

6.3 Implementation Process

At the time of this writing, there have been three implementation phases.

6.3.1 First Implementation

The first iteration was focused on generating meaningful test data that could be used to test the validity of the machine learning algorithms. It was important to generate data that reflected the actual daily activities at a hospital. This would ensure that the data are not too random for the algorithm to provide meaningful predictions. The data was generated to simulate a nurse's ordinary shift. For the purposes of the testing, a shift *is defined as an eight-hour period that can take place during one of three times: 7am – 3pm, 3pm – 11pm, and* 11pm – 7am. During this stage of testing, in order to limit scope, the nurse will be responsible for five rooms and perform routine duties throughout the scheduled shift. Generating data in this way will give the machine learning algorithm a chance to develop a basis for normal behavior patterns. For testing purposes, the data is stored in standard CSV files, which are then read and written to by the application. Figure 4 shows a sample of the data collected in the file. In the final production environment, the CSV format will be replaced with an SQL server/database environment.

	A	B	C
1	RoomNumber	TimeOfDay	Activity
2	1	290	1
3	1	393	2
4	2	396	1
5	2	399	2
6	3	402	1
7	2	405	2
8	4	408	1
9	4	411	2
10	5	414	1
11	5	417	2
12	1	420	5
13	2	430	5
14	3	440	5
15	4	450	5
16	5	460	5
17	100	480	2
18	100	480	2
19	100	495	2

Figure 4. Sample of User Behavior Data

6.3.2 Second Implementation

In the second iteration, each separate portion of the CANA application was developed on its own environment. At an early stage, it was important to establish that each individual technology was feasible and capable to implement within scope.

6.3.2.1 User Positioning

The user positioning portion of the application was built specifically for Android mobile platforms. In order to focus on rapid implementation, PhoneGap was used to deploy the code to an Android environment. Since PhoneGap was used, the development for the NFC reader was done using JavaScript.

For testing purposes, NFC tags were programmed with id values to represent room numbers. This was to demonstrate that a mobile device could effectively and quickly scan a tag and display its information.

6.3.2.2 Predicting User Behavior

User behavior was predicted using Python's Scikit-learn library. Scikit-learn provides an interface to implement a variety of common machine learning algorithms. A decision tree algorithm was chosen to handle the interpretation of the user behavior and providing an accurate prediction. The machine learning server receives an HTTP request from the client and proceeds to make a prediction based on the request data.

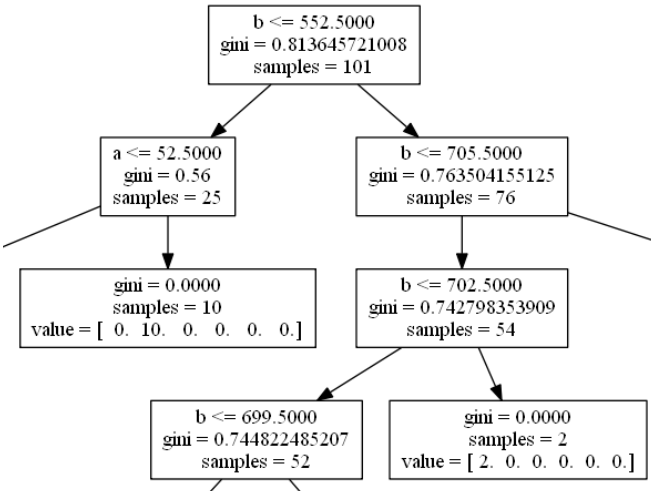

Figure 5. Sample of Decision Tree Data

The data is split automatically according to key features. Each end node on the tree represents a possible outcome. At this stage of development, there are five possible predicted outcomes: take patient measurements, patient activities, physical exam, give prescription, and patient nutrition. The end nodes represent a clustering of data points that match the criteria of a specific outcome. This can be seen in the "value" array in Figure 5. The position of the integer in the array "value" determines which prediction is represented, while the value of the integer shows how many samples are present. For example, value= [0, 10, 0, 0, 0, 0] on a leaf node would mean that 10 samples belong to the second classifier group based on their identifying features.

6.3.2.3 Generating Dynamic User Interface

Once the user's behavior has been predicted, the CANA application will return the predicted action back to the client UI. This is done via a simple HTTP response. The response is handled by the client, which will then display the corresponding form. By updating the UI's current form automatically, this allows the user to focus on the task at hand, rather than on manipulating the UI itself. The forms are updated asynchronously via AJAX methods. This ensures that the entire screen need not be reloaded and gives the user a more consistent experience.

6.3.3 Third Implementation

In the third and final iteration, the focus was on synthesizing each of the individual aspects of the CANA application into one working unit. With all of the elements being deployed, proper testing could be conducted. This allowed a more general, top-level, view to be taken of the CANA application. Finding a suitable server environment to deploy on was crucial at this stage.

Determining user positioning was labeled as a client side feature, while predicting user behavior and displaying dynamic form information were labeled as server side features.

7 Assessment and Analysis

7.1 Test Design

In this phase, the testing focused on how the application handled as a complete environment. In consideration to scope and budget constraints, the testing was conducted in a closed environment to demonstrate a proof-of-concept. Establishing the testing environment involved placing NFC tags around a room. Each tag is clearly labeled as if it were a hospital room number. For the purposes of this test, they were labeled numbers 101 – 105 consecutively. This allows for the simulation of a hospital environment for the test users. The test users are given a list of tasks to perform in several of the rooms.

The users' performance is measured based on their ability to complete their tasks and the time taken to complete the task. At the conclusion of the tasks, the users are asked about their overall impression of the application flow and the accuracy of the predictions.

The tasks are as follows:

1) Go to room number 101 and scan the tag
2) Check/Update Patient Measurements
3) Go to room number 103 and scan the tag
4) Check/Update Patient prescriptions

Follow-up questions:

1) Was the application flow intuitive and easy to use?
2) How accurate were the predictions?
3) Were the correct forms displayed in a meaningful way?

7.2 Test Results

Usability testing is conducted during the design phase and has been published in (Dirin et al. 2015) This paper aims to supplement the research done in that study. Testing was conducted with test users in a closed environment to assess the viability of the proposed methods. The results aim to demonstrate the accuracy of the predictive capabilities as well as test the viability of the entire CANA prototype. The test was conducted with six users and their results were stored in Microsoft Excel format. Table 1 shows the layout of the testing responses.

Table 1. Sample of User Testing Results

	User 1	**User 2**	**User 3**
Successfully Launch App	Yes	Yes	Yes
View All Patients	Yes	Yes	Yes
Scan Tag for room 101	Yes	Yes	Yes
Check/Update Patient Measurements	Yes	Yes	Yes
Was Prediction Accurate?	Yes	No	Yes

As the tests were conducted in a closed environment, the accuracy levels will not completely reflect results taken in a working environment. Although the data was generated to reflect an average work shift, there will be deviation.

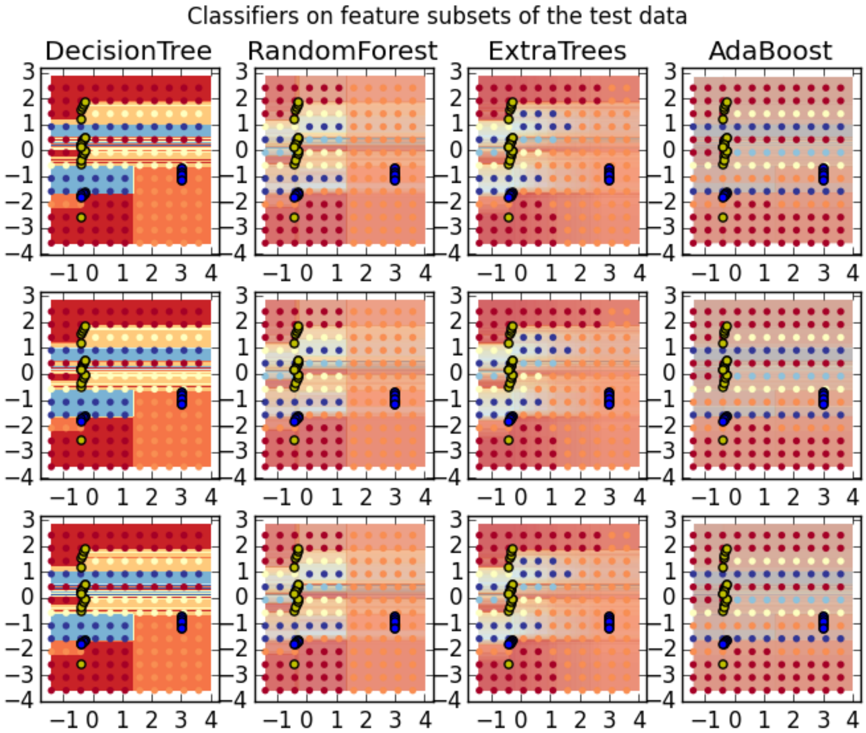

Figure 6. Distribution of Prediction Accuracy

For the users that were tested, the machine learning service provided 83% accuracy with an average run time of 31 milliseconds. The distribution of predictions is graphically represented in Figure 6. Each possible outcome (in this case, a specific classification) is given a separate color to demonstrate the spread of results. Training points are marked by solid colors on the graph, while testing data is partially transparent. The accuracy is based on the end user's experience with the prediction service. Further testing will allow the prediction service to get better, more accurate results as it learns as the data grows.

7.3 Test Results Analysis

The results from the testing phase demonstrate that machine learning predictive services are able to address the user scenarios in the testing phase. The testing phase is being considered successful as it met the requirements detailed in the introduction. This proposed solution is able to determine a user's context and quickly and reliably provide a prediction while being cost effective and adhering to UCD guidelines. These features allow users to bring up a specific patient's information simply based on being in the same room as the patient. When the user is ready to begin focusing on the next patient, updating the UI is as simple as changing their location and scanning the next NFC tag.

In a real world application, we propose that this context-aware system would be able to assist nurses to reduce the amount of overhead created by managing paperwork and aide in determining location in large facilities. This would allow nurses to spend more time focusing on patients and hospitals would be able to assign less resources for more clerical duties.

This phase of testing has not yet taken account for a large amount of concurrent users, and as this project progresses, a new testing and analysis phase will have to be administered in order to determine the scalability of this proposed solution. Furthermore, if the project procures funding during later stages, new approaches, such as Wi-Fi positioning, would also be available for testing that were not previously realistic options.

8 Conclusion

The goal of the CANA project is to deliver a reliable context aware, predictive solution to be deployed in a hospital environment. The current proof of concept prototype demonstrates the CANA application's ability to accomplish the goals established in the Scope. This prototype is capable to provide contextual cues to a predictive service, which will then supply the user interface with the appropriate form.

Considering the level of accuracy and speed of the prediction, the testing phase is being viewed as successful. However, during future implementations, work will be done to further optimize the machine learning algorithm to better fit the data sets. One factor that limits the accuracy of the prediction is the scope of the test data itself. In order to have the most accurate predictions, it is necessary to have a broad range of user behavior stored. Once a normal pattern has been established, the prediction service will have a better model to make predictions.

For the initial stages of the application, Knime services served as a starting point, but licensing fees would have been outside of the project's budget. Therefore, by developing the machine learning aspects of the application in Python, we were able to avoid any of the licensing fees attached with using Knime server in a production environment. In addition, Python showed an increase in speed while allowing us an added degree of flexibility working with analyzing the data.

9 Future Work

The next step for the CANA project is to seek for partners and funding to support further development. This will allow proper testing to be conducted in a live hospital environment. In addition, this would allow for more members to be brought into the project to aid the development and optimization process. The project will require work to be done to implement an interface for the hospital's patient record database. This will also require that diligent steps be taken to provide a secure environment to protect patient's privacy.

More work will need to be done to improve the accuracy of the predictions provided by the application. This will include optimization of the current decision tree or the implementation of a completely new approach such as a neural network classifier.

In the future, connection with the machine learning service will be based on a WCF client. This will allow all server communication to be asynchronous. Utilizing WCF will also ensure secure data connections to further protect patient data, as only users with appropriate access rights will be granted access to the service.

References

Adomavicius G, Mobasher B, Ricci F, Tuzhilin A (2011) Context-Aware Recommender Systems. AI Mag. 32:67–80.

Alpaydın E (2014) Introduction to machine learning. Methods Mol Biol 1107:105–128. doi: 10.1007/978-1-62703-748-8-7

Beauregard S, Haas H (2006) Pedestrian dead reckoning: A basis for personal positioning. Positioning, Navig Commun 27–35.

Beyer H, Holtzblatt K (1998) Contextual design: Defining customer-centered systems.

Bishop CMCCM (2006) Pattern recognition and machine learning. Pattern Recognit 4:738. doi: 10.1117/1.2819119

Cannan J, Hu H (2011) Human-Machine Interaction (HMI): A Survey. Gesture 1–16.

Chen F, Hekler E, Hu J, et al (2011) Designing for context-aware health self-monitoring, feedback, and engagement. Proc ACM 2011 Conf Comput Support Coop Work - CSCW '11 613. doi: 10.1145/1958824.1958927

Coskun V, Ozdenizci B, Ok K (2013) A survey on near field communication (NFC) technology. Wirel. Pers. Commun. 71:2259–2294.

De Maio C, Fenza G, Gaeta M, et al (2011) A knowledge-based framework for emergency DSS. Knowledge-

Based Syst 24:1372–1379. doi: 10.1016/j.knosys.2011.06.011
Dirin A, Nieminen M (2015) mLUX :Usability and User experience development framework for m-learning.
Dirin M, Dirin A, Laine TH (2015) User-Centered Design of a Context-Aware Nurse Assistant (CANA) at Finnish Elderly Houses. In: The 9th International Conference on Ubiquitous Information Management and Communication. The Mulia, Bali, Indonesia,
Domingos P (2012) A few useful things to know about machine learning. Commun ACM 55:78. doi: 10.1145/2347736.2347755
Eriksen S, Georgsson M, Hofflander M, et al (2014) Health in hand: Putting mHealth design in context. In: 2014 IEEE 2nd International Workshop on Usability and Accessibility Focused Requirements Engineering, UsARE 2014 - Proceedings. pp 36–39
Evennou F, Marx F (2006) Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning. EURASIP J Appl Signal Processing. doi: 10.1155/ASP/2006/86706
Fenza G, Furno D, Loia V (2012) Hybrid approach for context-aware service discovery in healthcare domain. J Comput Syst Sci 78:1232–1247. doi: 10.1016/j.jcss.2011.10.011
Ghahramani Z (2004) Unsupervised Learning BT - Advanced Lectures on Machine Learning. Adv Lect Mach Learn 3176:72–112. doi: 10.1007/978-3-540-28650-9_5
James G, Witten D, Hastie T, Tibshirani R (2013) An Introduction to Statistical Learning. Springer New York, New York, NY
Kotsiantis SB (2007) Supervised Machine Learning : A Review of Classification Techniques. Informatica 31:249–268. doi: 10.1115/1.1559160
Lerouge C, Wickramasinghe N, Affiliations A (2013) A Review of User-Centered Design for Diabetes-Related Consumer Health Informatics Technologies. J Diabetes Sci Technol J Diabetes Sci Technol 77:1039–1056. doi: 10.1177/193229681300700429
Li X, Zhou H, Li L (2013) Tucker Tensor Regression and Neuroimaging Analysis. ArXiv 1–28.
Madlmayr G, Langer J, Kantner C, Scharinger J (2008) NFC devices: Security and privacy. In: ARES 2008 - 3rd International Conference on Availability, Security, and Reliability, Proceedings. pp 642–647
Mitchell T (1997) Machine learning. McGraw-Hill, New York
Noble WS (2006) What is a support vector machine? Nat Biotechnol 24:1565–1567. doi: 10.1038/nbt1206-1565
Norman DA, Draper SW (1986) User Centered System Design; New Perspectives on Human-Computer Interaction. L. Erlbaum Associates Inc., Hillsdale, NJ
Ojeda L, Borenstein J (2007) Personal dead-reckoning system for GPS-denied environments. In: SSRR2007 - IEEE International Workshop on Safety, Security and Rescue Robotics Proceedings.
Preuveneers D, Berbers Y, Joosen W (2013) The future of mobile e-health application development: Exploring HTML5 for context-aware diabetes monitoring. In: Procedia Computer Science. pp 351–359
Prgomet M, Georgiou A, Westbrook JI (2009) The Impact of Mobile Handheld Technology on Hospital Physicians' Work Practices and Patient Care: A Systematic Review. J Am Med Informatics Assoc 16:792–801. doi: 10.1197/jamia.M3215
Ratwani RM, Fairbanks RJ, Hettinger AZ, Benda NC (2015) Electronic health record usability: analysis of the user-centered design processes of eleven electronic health record vendors. J Am Med Inform Assoc 22:1179–82. doi: 10.1093/jamia/ocv050
Rendle S, Gantner Z, Freudenthaler C, Schmidt-Thieme L (2011) Fast context-aware recommendations with factorization machines. Proc 34th Int ACM SIGIR Conf Res Dev Inf 635–644. doi: 10.1145/2009916.2010002
Solanas A, Patsakis C, Conti M, et al (2014) Smart health: A context-aware health paradigm within smart cities. IEEE Commun Mag 52:74–81. doi: 10.1109/MCOM.2014.6871673
Su J, Zhang H (2006) A Fast Decision Tree Learning Algorithm. 21st Natl Conf Artif Intell - Vol 1 5:500–505.
Surendran S, Rasamany S, Megalingam RK (2013) Context aware biomedical robotic platform for elderly health care. In: Proceedings of the 8th International Conference on Computer Science and Education, ICCSE 2013. pp 259–263
Tipping ME (2001) Sparse Bayesian Learning and the Relevance Vector Machine. J Mach Learn Res 1:211–245. doi: 10.1162/15324430152748236
Van Bogaert P, Meulemans H, Clarke S, et al (2009) Hospital nurse practice environment, burnout, job outcomes and quality of care: Test of a structural equation model. J Adv Nurs 65:2175–2185. doi:

10.1111/j.1365-2648.2009.05082.x
Wilkinson CR, De Angeli A (2014) Applying user centred and participatory design approaches to commercial product development. Des Stud 35:614–631. doi: 10.1016/j.destud.2014.06.001
World Health Organization (2011) mHealth: New horizons for health through mobile technologies. Glob Obs eHealth Ser. doi: ISBN 978 92 4 156425 0
Yang C, Shao H (2015) WiFi-based indoor positioning. IEEE Commun Mag 53:150–157. doi: 10.1109/MCOM.2015.7060497
Zhang H (2004) The Optimality of Naive Bayes. Proc Seventeenth Int Florida Artif Intell Res Soc Conf FLAIRS 2004 1:1 – 6. doi: 10.1016/j.patrec.2005.12.001