Activity Based Client for iOS
Bachelor Thesis

Hildur Uffe Flemberg (270689-3043, hufl@itu.dk),
Niklas Schalck Johansson (121288-2727, nsjo@itu.dk)

Supervised by
Jakob E. Bardram (bardram@itu.dk)

IT University of Copenhagen

Wednesday May 23th

Contents

List of Figures

1 Abstract

TODO

2 Introduction

2.1 Context

In modern times, work practices are often complex entities with respect to the number of resources and work flows involved. The seemingly simple task of doing a standard health check of a patient could easily involve e.g. a doctor, a nurse, a secretary and several documents including the patient record etc. Furthermore, the doctor's office might not be a sufficient location to complete the task - the patient might need to accompany the nurse to the lab to have a blood sample taken, go back to the office to have the doctor test his eyes and initially the doctor might need to get the patients record from the secretary in the reception. Today, computers are heavily used as a tool for optimization when carrying out tasks like these. They solve single tasks on the lowest level like reading e-mails, looking up records in databases and filling out forms very efficiently thus leaving it up to the user to find out for each instance of a work flow which applications, files, documents etc. he needs to prepare to do his job. While this doesn't render the tasks impossible to carry out, it implies some practical overhead in terms of time and mind capacity that moves attention of expert users from things that add value to their work to things that are less important.

2.2 Background

Activity Based Computing is a new paradigm that changes the focus of traditional computing environments from low level tasks like e-mail checking or web browsing to a higher level abstraction in the shape of an activity like "Performing a health check". An activity encapsulates all the smaller tasks and resources that are needed to complete it and provides a manageable unit that can be suspended and resumed, run in parallel with other activities and moved around while it adapts to the new surroundings. The ABC project started in 2003 with it's outset in pervasive computing de- signed for mobile, collaborative and time critical work for clinicians in hospitals. Research on ABC is currently developed and maintained by the ABC research team led by Jakob E. Bardram [1] and is financially funded by The Danish Council for Strategic Research [1]. The research team has developed five versions of the ABC framework, and is currently working on a sixth version. The ABC framework implements services for handling activities and resources. The fifth version in use is a Java based peer-to-peer based on the Aexo infrastructure and is targeted for non-PC devices [1]. The ABC framework will be described in more detail in chapter 2.

2.3 Motivation

With the introduction of touch screen tablet PCs on the market back in 2001 [3], a new family of devices with yet another screen size and touch screen performance matured. Even though the concept of a tablet PC is not a new one, tablets have since 2001 undergone major changes in the sense that they have become common and they have gotten more interactive user interfaces due to improved touch functionality. Bardram [2011] argues that: "Once you move away from the desktop and into a non-office-like environment such as a hospital, the challenges arising from the management of parallel activities and interruption are amplified because multi-tasking is now combined with a high degree of mobility [...]" Given the challenges of mobile work environments, the recent improvement of tablet computers and the fact that the experimentation with iOS as a technically well-suited operating system within ABC has not previously been explored, we will implement an iOS ABC client for the Apple iPad.

2.4 Goals

The purpose of the project is to develop an ABC client for the ABC framework running on the iOS platform with the following goals:

Activity Centered
It must to support the notion of activities.

Activity Awareness
It needs to able to adapt and adjust itself according to its location, mean- ing that the types of resources available and the UI representation is always dependent of the current working context.

Activity Suspend and Resume
It needs to able to save the state of one activity in order to restore and resume another previously suspended activity.

Uniform UI
It must have a uniform UI, meaning that whatever concrete kinds of dis- plays we choose to build these must be the same for any activity as long as the activities are resumed in the same location and under the same conditions.

2.5 Methods

The goals will be achieved through these methods

Investigation - iOS
We will investigate the iOS platform and discuss how we can apply the above mentioned goals. This will be done by following classes on iOS development and reading related articles on the subject.

Investigation - Location tracking
We will investigate what hardware resources is available on the iPad, and discuss which is better for location tracking. We will then discuss how this can be connected to the result of the iOS investigation, and how it will support the ABC principle of activity-awareness.

Implementation
Based on the analysis and discussion of the important elements we will define a list of requirements and implement a client for iOS that supports these requirements. The client will make use of the ABC framework.

Evaluation
We will evaluate the implementation by defining user scenarios that emphasize the mobility challenge in a hospital environment, where a user needs to bring digital resources with them, and test persons complete the scenarios. Afterwards we will have the test persons fill out a questionnaire where they will rate and evaluate the implemented features. Finally we will analyze the results and suggest improvements to our solution.

2.6 Results

TODO

2.7 Overview

TODO

3 Activity Based Computing

In this chapter we will explain the activity-based paradigm further explained. We will explain why adaptation and awareness is important, and introduce two frameworks that will help us create an activity based client.

3.1 Background

As explained in the introduction, activity based computing is a new computer paradigm that moves focus from application based computer interaction into a higher level computational support for human activities. The paradigm has its outset in clinical work on hospitals, and seeks to aggregate resources to activities, instead of specific applications. An example of such an activity could be the development of our proof of concept. Opening an activity will cause the relevant services and resources to become available to the user, and allow to user to more easily switch between activities and all their associated services and resources. Figure 3.1 shows an example of the activity "Proof-of-concept development", its associated services and their resources.

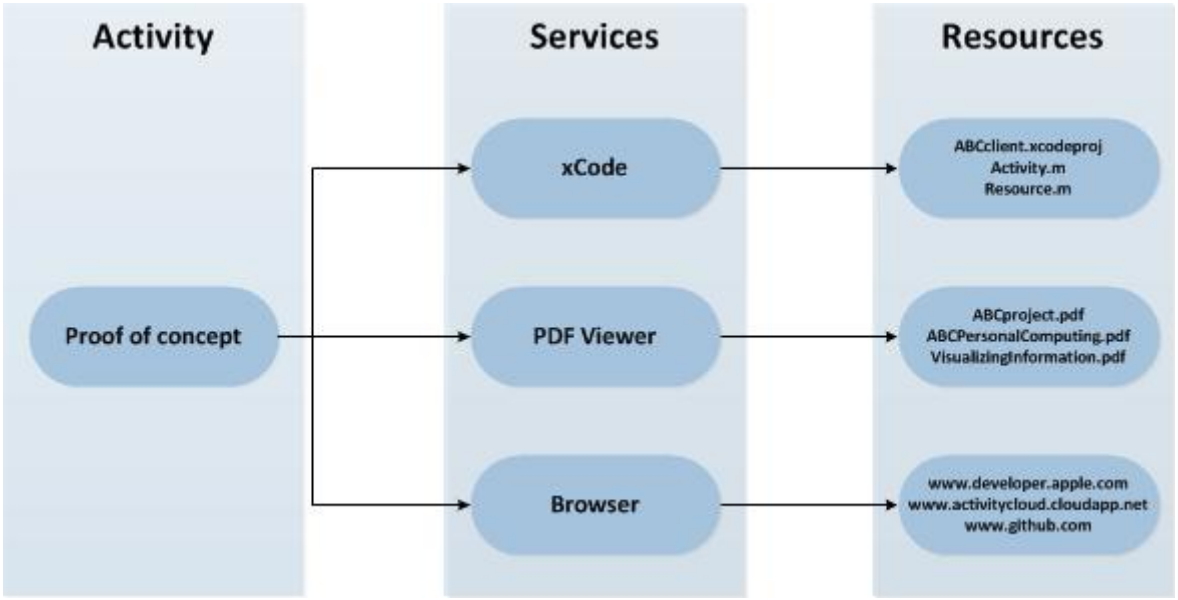

Figure 3.1: *The "Proof of Concept" activity. Illustrates how an activity encapsulates its services and resources*

Bardram [2011] identifies six principles that forms the basis of the activity based computing paradigm, being; *Activity Centered*, *Activity Suspend and Resume*, *Activity Roaming*, *Acitivty Adaptation*, *Activity Sharing* and *Activity Awareness*. Each of these will be further described in the following.

1. Activity Centered

An Activity is a computational unit that encapsulates a set of services and their relevant resources.

An activity therefore encapsulate digital software and data necessary for a user to carry out their work (activity).

2. **Activity Suspend and Resume**
   This allows a user to alternate between several activities and support interruptions that requires the user to perform another task. This is done by suspending the current activity and resuming another.

3. **Activity Roaming**
   This principle enables activities to be stored on an infrastructure, like a server, and allows for activities to be suspended on one device, and then resumed on another, to better support user mobility.

4. **Activity Adaptation**
   An activity can be displayed and handled on very different devices, and should adapt to the resources available at the resumed devices. In this case resources could be CPU power, screen size and network bandwidth.

5. **Activity Sharing**
   Focuses and deals with the collaborative aspect of activities. This principle states that activities are shared among collaborators that appear on a list of participants for any particular activity. If two or more participants engage resumes the same activity they will both be notified and engage in video and audio chat if possible.

6. **Activity Awareness**
   Allows for an activity to be context aware, such that it adapts itself to its current environment and work context. This could be to i.e. adapting the user interface or changing activities and services based on where the device is located.

Implementing all of these features enables computational devices to better support human activities, and allow users to move away from the traditional document -and application centered model on a desktop computer. In this thesis we will focus on how to display activities on the iPad and how to adapt the interface to the orientation of the iPad, how to store activities in an infrastructure and how the iPad can be aware of its surroundings. In this chapter we will further explain the principle of adaptation and awareness.

3.2 Activity Adaptation and Awareness

One of the major challenges in supporting human activities in a computer environment, are when the human activities no longer applies to a stationary device. Many work environments requires the users to move around in order to complete an activity. An environment where this is especially difficult is in hospitals as addressed in Bardram [2009], but even in a familiar environment like the IT-University, one is usually moving between 3 or 4 locations during a day. Usually when one is moving to a new location, the activities that are relevant to you changes, and thus one need not see all available activities, but only a subset of these. This requires the device to be able to detect the current location that the user is currently in, and to update the available activities. Bardram [2009] explains that it is necessary to find a way to notify the user about changes in the context, and that the list of activities has changed based on this new context. Bardram [2009] also argues that based on the evaluation with the clinical personnel, tablets were not deemed proper for mobile work situations based on some physical criteria. The personnel thought that the tablets were to large, meaning that they didnt fit in their large white-coat pockets and that they were to heavy to operate. The research were conducted back in 2009, which is before the iPad

was launched, and compared to tablets that existed back then, the screen is slightly smaller, 9.7 inches compared to the 11-12 inches tablets had back then, which might make it possible to fit it into a pocket, and the weight has decreased a lot, from 1.5 - 2 kg's to rought 600 grams. Especially the decrease in the weight might again make the tablet a good device to use in mobile environments. And with the use of Activity Roaming principle, the user would probably not even need to carry it around, but just pick up a device at the desired location, to resume their last activity as Bardram [2009] argues.

A second issue with mobility also found during evaluation by Bardram [2009] is that of changing screen sizes between devices. When a person moved from a device with a large screen like 1900×1600 to a smaller screen like 800x600, the saved states of each displayed service might not appear on the screen at all when resumed on a smaller device. This shows that it might not be feasible to have a similar implementation between devices with very different screen sizes. The iPad for example has a very distinct screen size, both in terms of physical size, but also in terms of resolution, so to try and imitate a work environment similar to a desktop computer would not be a good idea, because it is nothing like a desktop computer. One would therefore come up with a UI design that is different from a standard desktop UI, to eliminate the issues of positioning on the screen. It is also important to notice that a user would probably not use a small tablet computer for the same tasks that the user uses a large desktop computer for. For a tablet such as the iPad one could also take into account that the two orientations of the device could show a different UI, since it will move from a widescreen view to a portrait view, and thus make the UI more adaptable.

3.3 ABC Framework

4 Mobile Activities

5 Implementation

6 Evaluation

7 Conclusion

Bibliography

A.R.Team. Activity based computing. `http://activitybasedcomputing.com`, 2012.

Jakob E. Bardram. Activity-based computing for medical work in hospitals. *ACM Transactions on Computer-Human Interaction*, Vol. 16(No. 2), 2009.

Jakob E. Bardram. The activity-based computing project. *AAAI Workshop (WS-11-04)*, Techniques and Languages, 2011.

Bardram et al. Support for activity-based computing in a personal computing operating system. *CHI 2006*, 2006.

Bardram et al. Reticularspaces: Activity-based computing support for physically distributed and collaborative smart spaces. *CHI 2012*, 2012.

Luca Chittaro. Visualizing information on mobile devices.

Consolvo et al. Activity sensing in the wild: a field trial of ubifit garden. *CHI 2008*, April 2008.

Froehlich et al.. Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits. *CHI 2009*, April 2009.

Michael B. Skov and Rune Th. Hoeegh. Supporting information access in a hospital ward by a context-aware mobile electronic patient. *Pers. Ubiquit. Comput. 10: 205-214*, October 2006.

Various. Tablet history. `http://en.wikipedia.org/wiki/Microsoft_Tablet_PC`, 2012.