ABSTRACT

Telephony has become an integral part in the day to day communication and new telephony services are quickly being deployed in the industry. There is a need for users to be provided with new services on the fly; these services can be composed from existing services to provide an added-value service. The vision is to allow ordinary people, who are the end users, to easily compose a set of available services and run them on their devices while they are on the move without requiring specialized IT or telecom skills.

An end user service composition approach is followed that reduces the composition complexity and difficulty from the end user perspective. The approach enables the end users to personalize the compositions with a powerful presentation and supporting the end users to dynamically customize the service composition.

A scenario based approach is followed whereby different practical composition scenarios are explored to shed light on several aspects of how the end users can personalize the composition process using the tool that has been presented by creating compositions that create added value services for the scenarios looked into.

PREFACE

This project report is written at the Department of Telematics in the Norwegian University of Science and Technology (NTNU) during the Autumn Semester, 2010. The project is submitted as a fulfillment of the graduation requirements of my Master degree in Telematics.

The theme of this project work was defined by UbiCompForAll - Ubiquitous service Composition For All users. UbiCompForAll is a research project founded by the Norwegian Research Council. The project involves SINTEF, NTNU, Gintel, Tellu and Wireless Trondheim. The focus being on end user service composition of telephony services, messaging services and other personal management services such as appointment organizer, calendar, customized search, location handling, reservation services, etc.

A part of this project report is based on the work carried out by Jens Einar in his Masters Thesis whereby he created a tool 'EasyComposer' for end-user service composition that is made to work with another tool the 'EasyDroid' developed by John Edvard Reiten. After studying the two set of tools, I extended the EasyComposer to include a calendar as well as add more components on the EasyDroid.

I would like to thank my Supervisor, Mazen Malek Shiaa for giving me good and comprehensive advice and guiding me through the project. In addition I would like to thank Jens Einar and John Edvard for their support and cooperation throughout my project work.

Frank K. Mbaabu

January 31, 2011

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Service composition involves the development of added value services by allowing the orchestration of already existing services into an added value service which is needed at a particular instance. In this chapter we look into the background of service composition what it generally entails in section 1.1, at the same time looking at why service composition is of importance. In the section 1.2 that follows we look at the problem to be addressed in general and how service composition comes into play with regards to the problem. Finally section 1.3 looks at the structure of the report.

1.1 Motivation and Background

Service Composition involves the development of customized services by orchestrating already existing services into one or more new services in order to achieve certain objectives. Although current composition styles need professional expertise such as programming skills to achieve this, end user service composition allows end users without this expertise to easily compose their own services.

In the wake of ubiquitous computing and advances in mobile device design, devices all around a person will provide an array of services that users might want to use. It is therefore highly recommended that the user makes use of these services made available with ease and with minimal effort when detecting, invoking and coordinating them. The introduction of smart phones in the market is also revolutionizing the telephony industry, therefore the focus is on telephony services, messaging services and other personal management services such as appointment organizer, calendar, customized search, location handling, reservation services, etc thus making service composition in mobile environments possible.

1.2 What is the problem to be addressed?

Telephony has become an integral part in the day to day communication and new telephony services are quickly being deployed in the industry. There is a need for users to come up with new services on the fly; these services can be composed from existing services to provide an added-value service. The vision is to allow ordinary people, who are the end users, to easily compose a set of available services and run them on their devices while they are on the move without requiring specialized IT or telecom skills.

In traditional networks the user has to subscribe to the service offered by the network operator or services already preprogrammed in applications within the device. The user has no way of composing new services from the existing service and this makes it difficult for the user to coordinate the services offered in order to accomplish his/her particular tasks on the fly. Some technologies have emerged, such as the Easy Designer, which enable for service composition but not at the end user level but by the operators.

Service composition is left to the service provider rather than the users who are in better position to define what he or she needs. The intention is to allow the user meet their requirements by composing their own services, rather than using services offered by the service providers that are not fully tailored to his/her needs. This added value will enrich the user experience. The users can take up services that are made available and orchestrate a new service. The focus of the project is on telephony services, messaging services and other personal management services such as appointment organizer, calendar, customized search, location handling, reservation services, etc.

1.3 Structure of the Report

Chapter 1 introduces the project by looking at the motivation and the problem to be solved. In chapter 2 the report looks at service oriented computing and supporting architectures, programming models and techniques required to implement service composition in mobile environments. Chapter 3 then looks at the problem description looking at the challenges in end user service composition and the proposed approach to end user service composition. Chapter 4 focuses on the service composition tool that has been developed. Chapter 5 focuses on the assessment of the EasyComposer calendar tool that has been developed for end user service composition applying its usage in several scenarios to identify if it meets its objectives. Chapter 6 discusses the outcome of the project and any future work that can be done on the tool. Finally chapter 7 looks at the conclusion.

CHAPTER 2: BACKGROUND

Service oriented computing focuses on four related research themes: service foundations, service composition, service management and monitoring, and service-oriented engineering [1]. This report focuses on service composition and this chapter looks in detail at the background of service composition.

In order to understand service composition better section 2.1 looks into service oriented computing, as it is where service composition stems from. Service Oriented Computing views business functions as services and the associated Service Oriented Architecture (SOA) described in section 2.1.1, establishes a relationship between such services. Further section 2.1.2 focuses on Service Component Architecture (SCA) which provides a programming model for building applications and solutions based on a Service Oriented Architecture. Finally under service oriented computing, service composition is described in section 2.1.3.

Current researches mainly focus on service composition technologies based on wired networks. Since the project focuses on telephony services section 2.2 focuses on service composition for mobile networks and finally section 2.3 looks into context aware mobile service composition and focuses on how an abundance of services can be accessed from the context.

2.1 Service Oriented Computing

Service-oriented computing (SOC) focuses on utilizing services when developing applications. Its main emphasis is the development of rapid, low-cost, interoperable, evolvable, and massively distributed applications by code and application component reuse. In simpler terms it reflects a “service-oriented” approach to programming where

applications are composed by discovering and invoking available services to accomplish some task. In addition this approach is meant to be independent of specific programming language or operating systems. In order to achieve this, application components should be assembled into loosely coupled network of services that can be used to create agile applications[2].

Web services are currently the most promising SOC based technology. They use the Internet as the communication medium and open Internet-based standards, including the Simple Object Access Protocol (SOAP) for transmitting data, the Web Services Description Language (WSDL) for defining services, and the Business Process Execution Language for Web Services (BPEL4WS) for orchestrating services.

2.1.1 Service Oriented Architecture (SOA)

The SOA architecture packages functionality as a suite of interoperable services that can be used within multiple separate systems which in turn can be used from several business domains. Thus the power of SOA stems from its ability to enable business agility through business process integration and reuse.

The figure 2.1 by Papazoglou and Georgakopoulos [3] shows the software oriented architecture pyramid. The basic services constitute the SOA foundation, while the higher layers in the pyramid provide additional support required for service composition and service management.

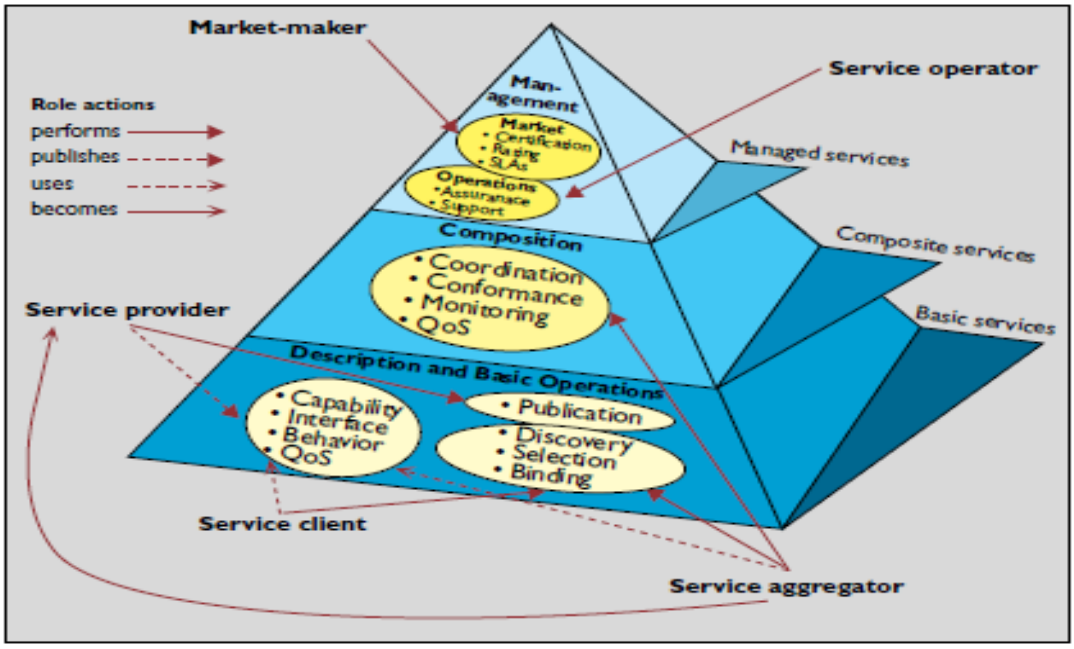

Figure 2.1 Service Oriented Architecture.

Papazoglou and Georgakopoulos further look at the different layers, their functions and roles which include [3]:

- **The Basic services layer** constitutes the SOA foundation. It offers the basic services, their descriptions, and basic operations, which are publication, discovery, selection, and binding, which produce or utilize such descriptions.
- **The service composition layer** encompasses necessary roles and functionality for the consolidation of multiple services into a single composite service. The resulting composite services may be used by service aggregators as components in further service compositions or may be utilized as applications/solutions by service clients.
- **The Service management layer** provides managed services by managing critical applications/solutions and specific markets. SOA's operations management functionality is aimed at supporting critical applications that require management of

the service platform, the deployment of services and the applications. Another aim of SOA's service management layer is to provide support for open service marketplaces.

2.1.2 Service Component Architecture

Service Component Architecture (SCA) provides a programming model for building applications and solutions based on a Service Oriented Architecture. It is based on the idea that business functions are provided as a series of services, which are assembled together to create solutions that serve a particular business need. These composite applications can contain both new services created specifically for the application and also business function from existing systems and applications, reused as part of the composition. SCA provides a model both for the composition of services and for the creation of service components, including the reuse of existing application function within SCA composites.

SCA is a model that aims to encompass a wide range of technologies for service components and for the access methods which are used to connect them. For components, this includes not only different programming languages, but also frameworks and environments commonly used with those languages. For access methods, SCA compositions allow for the use of various communication and service access technologies that are in common use, including, for example, Web services, Messaging systems and Remote Procedure Call (RPC) [4].

Therefore SCA gives a simplified implementation of SOA using components and compositions, following the principle of loose coupling by having loosely coupled components to support agility at the same time handling event driven behavior through a comprehensive invocation model and Separating development and assembly to allow technology-agnostic composition.

2.1.3 Service Composition

Services are autonomous, platform-independent entities that can be described, published, discovered, and loosely coupled. Service composition creates new customized services by discovering, integrating multiple services into a single composite service which in turn can be used as basic service in further service compositions or offered as service to service users.

Papazoglou et al define the process of service composition as having three phases:

1. The planning phase; where the candidate services are discovered and checked for composability and conformance.

2. The definition phase; which generates the actual composition structure.

3. The implementation phase; which implements the composite service bindings based on the service composition specification.

In addition for a complete solution for service composition to be developed the following elements are needed: service request description, service matching and compatibility checking, description of service composition, and service execution monitoring and coordination.

Papazoglou et al further identify the following types of service composition which are used throughout these phases of service composition [5]:

- **Explorative composition:** service composition is generated on the fly based on a service client request. The desired service is described and the service broker then compares the desired composite service features with potentially matching published constituent service specifications and may generate feasible service composition plans.

These plans result in alternative service compositions that can be ranked or chosen by service customers depending on a criterion such as availability, cost and performance.

- **Semi-fixed composition:** Here some of the actual service bindings are decided at run time. When a composite service is invoked, the actual composition plan will be generated based on a matching between the constituent services specified in the composition and the possible available services.
- **Fixed composition:** a fixed composite service synthesizes fixed (pre-specified) constituent services. The composition structure and the component services are statically bound. Requests to such composite services are performed by sending sub-requests to constituent services.

2.2 Service Composition in Mobile Environments

Current researches mainly focus on service composition technologies based on wired networks. Service composition architecture is generally centre-based. In the wake of ubiquitous computing and advances in mobile device design, devices all around a person will provide an array of services that users might want to use. The architectures for service composition based on the wired networks cannot fulfill service composition requirements in this kind of environment. Dynamic topology changes, limited bandwidth, limited terminal capability, and central node failure are some of the reasons why the wired networks infrastructure cannot be implemented in the non wired environments.

Further mobile environments can be categorized into infrastructure based and infrastructure-less or ad-hoc environments. Infrastructure-based mobile environments have access to services in the wired infrastructure through gateways, proxies and base

stations e.g. wireless office networks and the WiFi. Infrastructure-less mobile environments, e.g. warfront activities, ad-hoc sensor networks, walkways in cities that are created on-the-fly call for an alternate design approach of service composition architectures as they are not able to leverage from centralized composition managers to some extent.

Some issues of concern when implementing service composition in mobile environments [6]:

- **Distributed Coordination Model:** Service Composition involves the use of a coordinating entity referred to as broker, which manages the discovery, integration and execution of a composite service. Coordination and management need to be primarily adaptable to the topology changes of the mobile environment and handle mobility, network disconnections and other types of node or service failures. This creates the need for composition architectures in ME to have a distributed coordination model.
- **Resource Heterogeneity and Awareness:** Devices with varying resources and capabilities exist in the geographical vicinity of one another. Resources refer to either services or computation power that are available and vary amongst devices. An efficient composition protocol should ideally be able to compose a service with the most efficient use of the surrounding environment and with minimum network overhead.
- **Mobility Management and Adaptability:** Service composition protocols should be able to able to adapt themselves to the changing neighborhood. In particular, it should be able to utilize newly arrived services that were not available before. Composition protocols for ME should also be able to adapt their execution model to suit the changing service topology.

- **Fault Tolerance and Reliability:** A robust composition protocol should be tolerant towards failures and degrade gracefully with increasing service/resource unavailability. Faults may range from network level disconnections, service discovery failures to node failures and service execution failures.
- **Efficient Network Utilization:** By default, service discovery and composition protocols are usually broadcast-based and essentially impose a relatively high network load. An efficient composition protocol should try to find a way to utilize the network efficiently in such an environment, since reducing network wide searches may reduce the chances of a service being discovered and hence reduce composition efficiency.

2.3 Context Aware Mobile service composition

In the recent past mobiles have had preprogrammed settings to suite a particular environment. For instance the sony Ericson T-28 has profiles that suite a particular environment for instance when a person goes to a meeting he/she can select the meeting profile which has the optimal settings for a meeting scenario whereby the ring volume is set off [7].

In the context of future mobile communications the users will be able to access an abundance of services typically developed by many co-operating entities in diverse service access environments and different mobile technologies. This will place a requirement for applications to be optimally delivered and executed over a large diversity of infrastructures and configurations, as well as for dynamic adaptability of services to changing conditions and contexts. The current preprogrammed profiles will not be able to meet these changing conditions and contexts as it would not be feasible to develop separate versions for different execution contexts; applications should be to a large extent aware of the

environment they run on. Like in the case of the Ericson T-28 where you place your phone on the hands free unit in the car and the In Car profile is activated, intelligent mechanisms should exist for identifying the context and the particular high-level requirements of an application and mapping them to appropriate reconfiguration operations on the underlying hardware and software infrastructure. One approach is to encode contextual information in the user profile, which consists of user preferences, terminal, ambient, network, and service profiles which describes the content elements and objects composing the service [8].

CHAPTER 3: PROBLEM DESCRIPTION

As mentioned earlier telephony has become an integral part in the day to day communication and new telephony services are quickly being deployed in the industry, thus the need for service composition to provide users added value services on the fly composed from the existing services that are deployed. In addition ordinary people, who are the end users, should be able to easily compose a set of available services and run them on their devices without requiring specialized IT or telecom skills rather than the service provider composing these services as the users are in better position to define their needs. Therefore in order to highlight the existing problem section 3.1 looks at the end user service composition and currently existing approaches to end user service composition. Section 3.2 then looks generally at some of the challenges that limit the end users from composing services. Section 3.3 further identifies strengths and weaknesses in the approaches mentioned in section 3.1 in an effort to come up with a proposed solution which is stated in section 3.4. The intention is to allow the user meet their requirements by composing their own services in the easiest way possible and when necessary.

3.1 End-User Service Composition

End user service composition is an initiative to bring the advantages of service composition to users who necessarily don't have computing skills. Despite the advantages only a small proportion of users can construct service-based applications as they require considerable modeling and programming skills. This limitation can be attributed to the complexity of the existing composition approaches and the limited technical knowledge of end users. Thus the challenge lays in simplifying service composition. Usman et al while

working for the SOA4ALL project, identify three approaches aimed at simplifying service composition as outlined below [9]:

3.1.1 Control Flow-based Composition Approach

This approach follows a sequential process when composing services, where by a task /service has to be completed before the next task / service is executed. In this approach what is of importance is the order in which atomic services are executed. Apart from the sequence the various useful relations between services are defined; these include unconditional branching, conditional branching, iterative execution and unconditional stopping. Figure 2.2 shows how the control flow-based composition approach works.

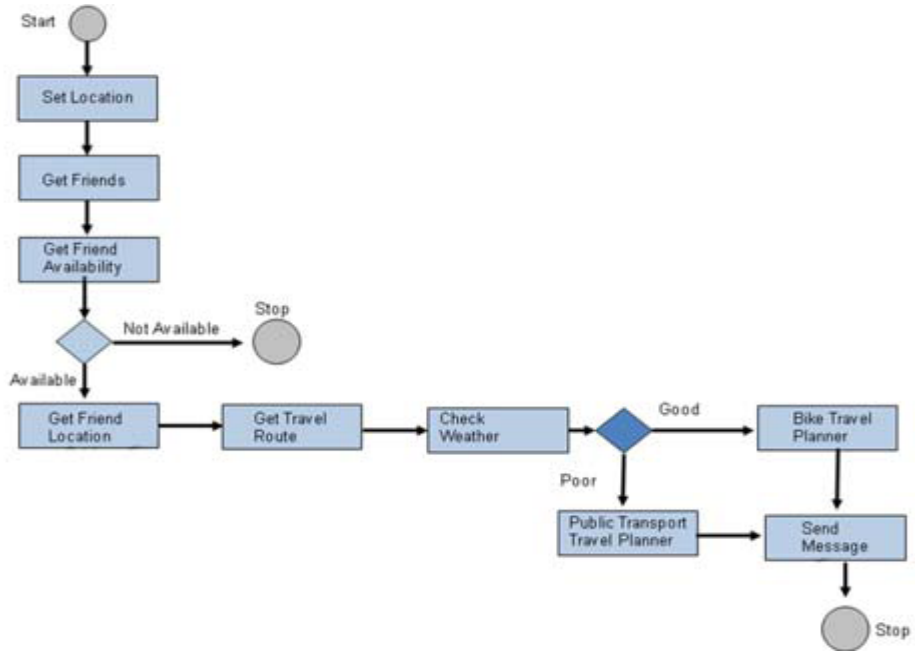

Figure 2.2: Control Flow-based Composition Approach

3.1.2 Data Flow-based Composition Approach

In this approach the focus is on the data passed between multiple services rather than the sequence. It can be looked at as an information-oriented view of service composition. This approach enables users to define how data flows from source service(s)

to destination service(s) and does not consider the information about service execution order and conditions. Figure 2.3 shows how data is passed between multiple services.

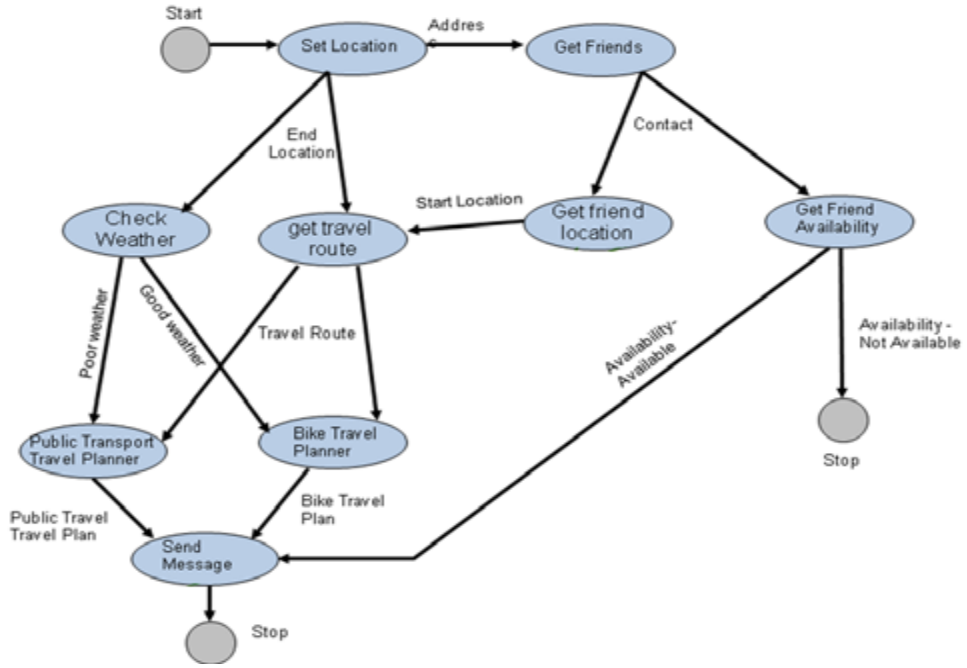

Figure 2.3: Data Flow-based Composition Approach

3.1.3 Assisted Composition Approach

This approach is drawn from human computer interface design methods and principles, whereby users choose individual services from a wide range of available services in the composition template. In addition there is a computational algorithm that manages the interoperability and compatibility of services therefore the users can select from a wide range of customizable system templates and do not have to define control and data flow among services as these aspects are managed automatically. An example of the assisted composition approach is shown in figure 2.4.

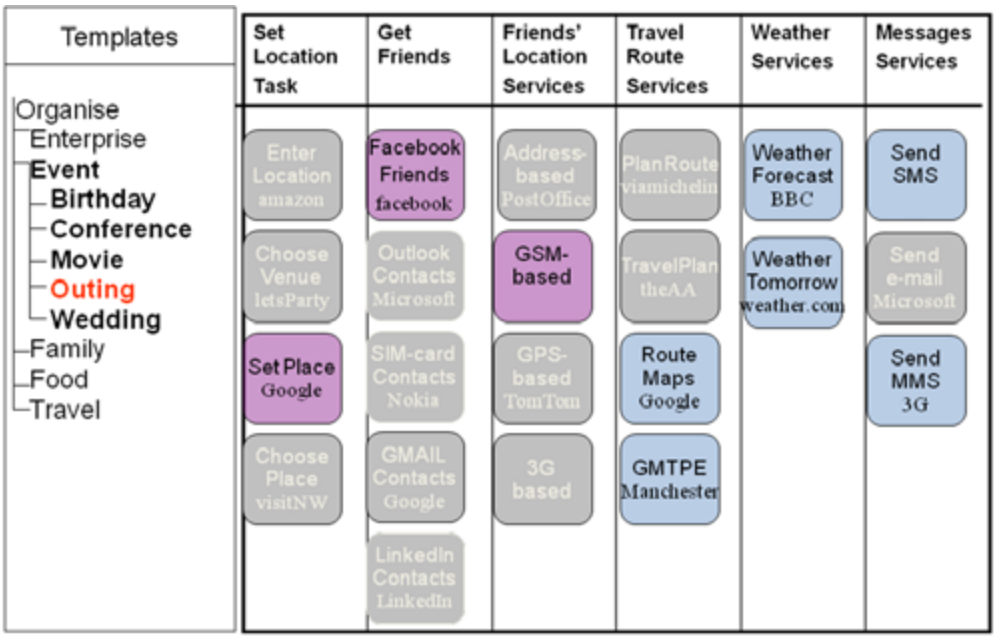

Figure 2.4: Assisted Composition Approach

3.2 challenges in End user service composition.

Some of the challenges that limit the end users from composing their services include:

1. **Complexity in service description and discovery:** The existing techniques for service description and discovery are too complex for users without programming and modeling knowledge. For instance the Web Service Description Language (WSDL) which is commonly used to define the programming interface of a service is intended for SOA experts to understand the interface and parameters of a Web service and to correctly invoke a service. This therefore makes it difficult for end users to understand the functionality of a service to discover an appropriate service when composing services.
2. **Complexity in composing services on the fly:** Once a service is deployed it also becomes difficult for end users to customize the composed services as this will

involve a long lifecycle from design, development, design and finally deployment. This is done by experts where they orchestrate services through the BPEL (Business Process Execution Language) which would be almost impossible for someone without this knowledge.

3.3 Challenges of Existing End User Service Composition Approaches.

In the quest to solve the above mention problem three end user service composition approaches are identified in section 3.1 above. These techniques have their strengths and also have inherent problems which include:

3.3.1 Control Flow-based Composition Approach

Advantages:

- It is easy to use as one follows certain logic when composing services.
- It also allows for flexibility in that logic can be created for many situations when the need arises.
- Also it can take up a lot of services.

Disadvantages:

- At times understanding the logic required for a particular situation maybe a difficult task as it somehow requires some modeling techniques.
- Additionally, due to the sequential nature of a control flow a composition involving various services can get too complex for users to keep track of.

3.3.2 Data Flow-based Composition Approach

Advantages:

- Data flow-based approach is highly flexible as it allows for composition of many situations.

Disadvantages:

- On the other hand it is a difficult approach for non-programmers due to the underlying complexity of managing different data sources in that there might be a lot of inconsistencies of the data from the different sources.
- Moreover, modeling skills are required where such modeling technique as the use of data flow diagrams is used when coming up with complex compositions.
- The complexity can also lead composing services wrongly

3.3.3 Assisted Composition Approach

Advantages:

- The assisted composition approach is an easy approach to use.
- It reduces the complexity of service composition by providing the user with a set of options that make it easy to compose services without any programming or modeling knowledge.

Disadvantages:

- The limitation of this approach is its scalability and flexibility. For instance, one cannot customize or create their own compositions and extending the list of all available services might be impossible as it may take up a lot of resources.

3.4 Selected Approach/Proposed Solution.

The control flow based composition approach seems more applicable in composing of telephony services. Combining this with visual programming will make the end user experience much more easier than dealing with the control flow constructs like for

instance the for loops. In this case the approach seeks to combine the strengths of the three approaches. The selected approach will:

1. Follow a control based approach to service composition, whereby the services will be executed in a sequential manner where one service has to be completed before the next service is executed.
2. The user interface design will assist the user in composing his/her services. This will be achieved by having language that is familiar to the user as well as graphical widgets such as icons that a user can relate to.
3. In addition the compositions are to be stored as data in order to achieve a high flexibility as it allows for composition of many situations.
4. In addition context aware service discovery should be integrated in the approach whereby the users' context which denotes the end users environment, will be attached to the service thus reducing the amount of information required from the end user when composing services.
5. The approach should also contain the functionality to allow users compose their services on the fly, therefore eliminating the need to conform to only a set of services probably composed by the network operators.

In order to achieve this, the existing tool for end user service composition that was developed by Gintel AS called the EasyComposer will be studied and extended by incorporating the above mentioned approach. In addition a set of service composition scenarios will be worked out that will guide and demonstrate service composition using the approach identified above.

CHAPTER 4: SERVICE COMPOSITION TOOL

The main focus of the project is a tool that enables the end users to compose their customized services. This chapter is divided into two; the first part section 4.1 identifies the composition infrastructure already in existence that partially can compose some telephony and messaging services. The existing tool is developed by Gintel AS a software vendor and was also developed as part of previous Master Thesis work at NTNU. The second part, section 4.2, focuses on the further development done on the EasyComposer and EasyDroid

The tool has evolved from the Easy Designer described in section 4.1.1 to the EasyDroid and the EasyComposer described in section 4.1.2 and 4.1.3 respectively, which are two separate applications that work hand in hand to help in end user service composition. They follow the control flow-based composition approach to end user service composition, mentioned earlier in chapter 3. Section 4.1.4 describes the different components found in both theEasyDroid and the EasyComposer and section 4.1.5 looks into the xml that represents the compositions.

The EasyComposer calendar is described in section 4.2.1, which put things into context that all users can whereby the users can log activities that they choose to carry out at certain times and in addition user profiles can be attached to these logged activities. These profiles have service compositions encoded in them. Section 4.2.2 looks at the additional components developed for the EasyDroid and finally section 4.2.3 looks the new XML format that will represent the compositions.

4.1 Existing Tool

The existing tools include the EasyDesigner, EasyComposer and EasyDroid. The other part describes further development to the EasyComposer and EasyDroid. The EasyComposer allows for the end user to compose their services using service components and export them as an xml that is deployed on the EasyDroid as compositions.

4.1.1 Easy Designer

Easy designer is a service composition tool that allows Operators and service providers to assemble services from a comprehensive palette of core capabilities where service flows can be created or modified to meet the needs of each client. The operators and service providers are able to offer service customization capabilities to meet the individual needs of their clients. The java based application can design and assemble new services on-the-fly, within hours, not days or weeks, greatly reducing time to market and enhancing operator agility. Operators can both innovate within their markets and respond more rapidly to demand and competitive threats. New services can be launched rapidly and at low cost, reducing the risk of service delivery. The limitation of the easy designer is that it is intended for service providers and operators. End users can not compose their services without going through the operators

4.1.2 EasyDroid

The Easy droid is an application developed for the android phones. The EasyDroids architecture shown in figure 4.1 shows how the different parts work together. It is made up of different components that make use of services available in the android environment. It contains different components such as the Notification. These components make use of the inherent services in the android software stack, such as the GPS. The components are

further combined to form composite services as defined by the XML developed by the EasyComposer which is described in the next section. The Engine controls how the services are invoked and coordinated.

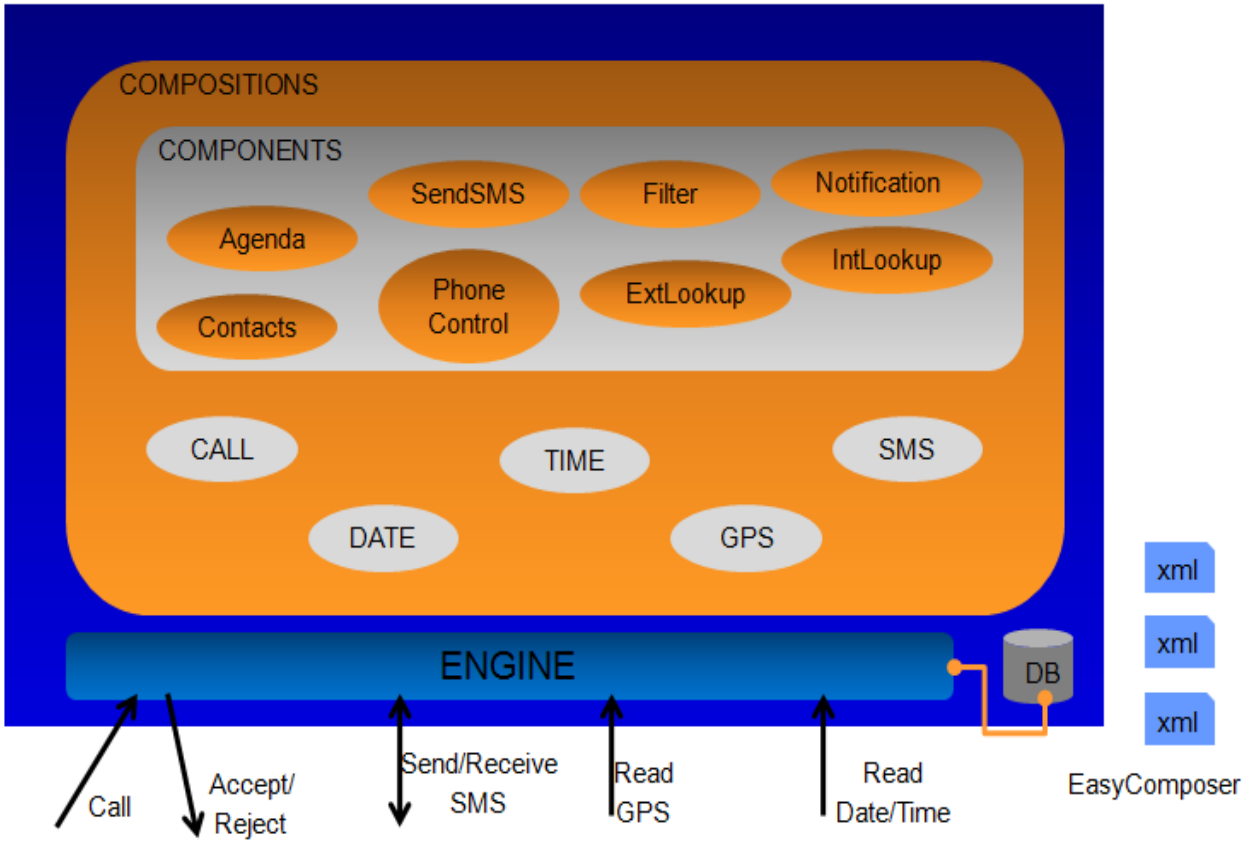

Figure 4.1 Easy Droid Architecture.

4.1.3 EasyComposer

The EasyComposer brings more power to the end user by allowing him/her to customize services as needed, removing reliance on the operator or service provider. It is a graphical tool that allows for creation and modification of services. It contains a palette of components that allow for Service composition by creating service flows as shown in figure 4.2. The components in the palette are components that exist in the EasyDroid.

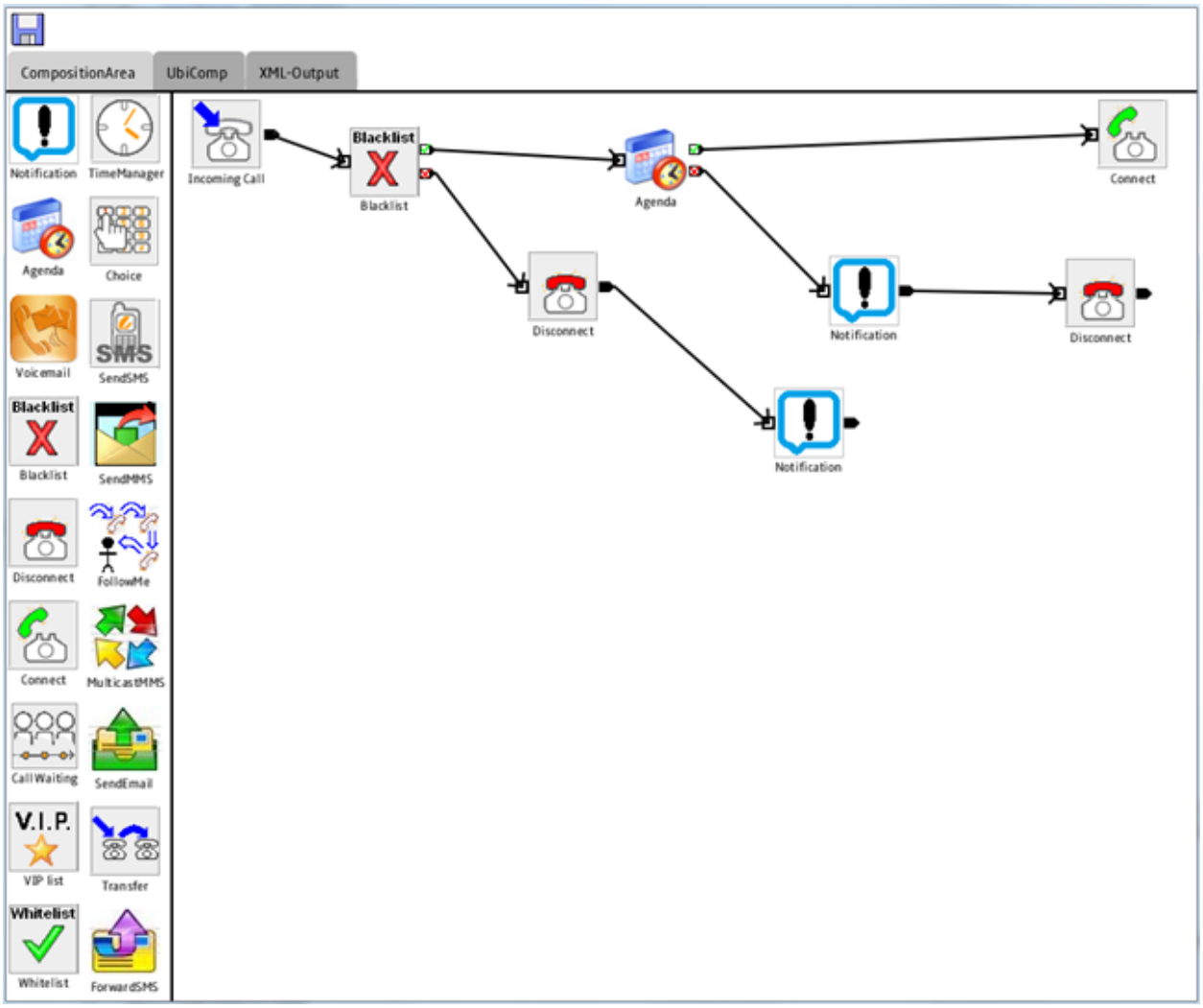

Figure 4.2 EasyComposer

Therefore the Easy Composer allows for end user service composition and then these compositions can be instantly deployed to the android phone that is running the Easy droid.

4.1.4 Basic Components

Initially, several basic components have been developed which basically allow composing basic telephony services. The major components include:

- **Agenda component:** which allow for users to specify the availability of users according to the activities they would be carrying out at the particular time. For instance if the

user is in a meeting the incoming calls will be disconnected while allowing certain numbers that are placed in the white list.

- **Send SMS component:** allows for users to send SMS messages as and when needed. For instance in the case a caller is disconnected, the SMS component sends a message to this caller.
- **Filter Components:** These are the black list component and the V.I.P list component. The blacklist contains all the numbers which should be disconnected when they try to connect with the user. On the other hand the V.I.P list is a list of all numbers that will go through when the user is in mode which does not allow any incoming calls.
- **Notification component:** notifies the user of certain actions that have taken place in the background. For instance if a call is disconnected, the user will not know that this has happened. The notification component allows the user to know of these events.

4.1.5 XML representation

When the compositions are created they are exported as an XML that is deployed to the Easy droid. This XML file is read by the engine and executed. A structure was developed of how this XML should look like. The structure of the XML is as shown below:

```
<Composition type="1" firstCid="1">
<Component cid="4" name="Disconnect" type="2" nextCid="-1" action="1" />
<Component cid="5" name="Connect" type="2" action="4" nextCid="-1" />
<Component cid="1" name="Blacklist" type="1" nextCidIfMatch="3" nextCidIfNoMatch="2" matchType="1" />
<Component cid="3" name="Disconnect" type="2" nextCid="6" action="1" />
<Component cid="2" name="Connect" type="2" action="4" nextCid="-1" />
<Component cid="6" name="SendSMS" type="6" nextCid="-1" textMessage="Compose your own message" toggleButtonIsChecked="1" />
</Composition>
```

Whereby the composition is surrounded by the composition tags and the components within the composition are placed in component tags within these composition tags. Within the component tags are attributes that show how the different components relate with each other

4.2 Further Development of the Tool.

The EasyComposer and EasyDroid make the composition of services by end users much easier and friendlier when compared to other existing end user service composition tools. The only limitation is that the Easy composer is not attached to a context in which end users can relate to when in their day to day activities and environments. In addition these tools focus on very few services whereby new services can be developed to enrich the user experience and create richer services. Thus a calendar is incorporated in the EasyComposer as well as two new components for integration in the EasyDroid. The XML is also improved to take these developments into account.

4.2.1 EasyComposer Calendar

In order to put things into context that all users can relate to a calendar application is developed whereby the users can log activities that they choose to carry out at certain times and in addition user profiles can be attached to these logged activities. The idea is to encode contextual information in the user profile, which consists of user preferences and service compositions as earlier mentioned in chapter 2 under context aware mobile service composition.

Therefore the user in this case will be presented with an application which contains a calendar tab in addition to the composition area tab that was present in Easy Composer as shown in figure 4.3.

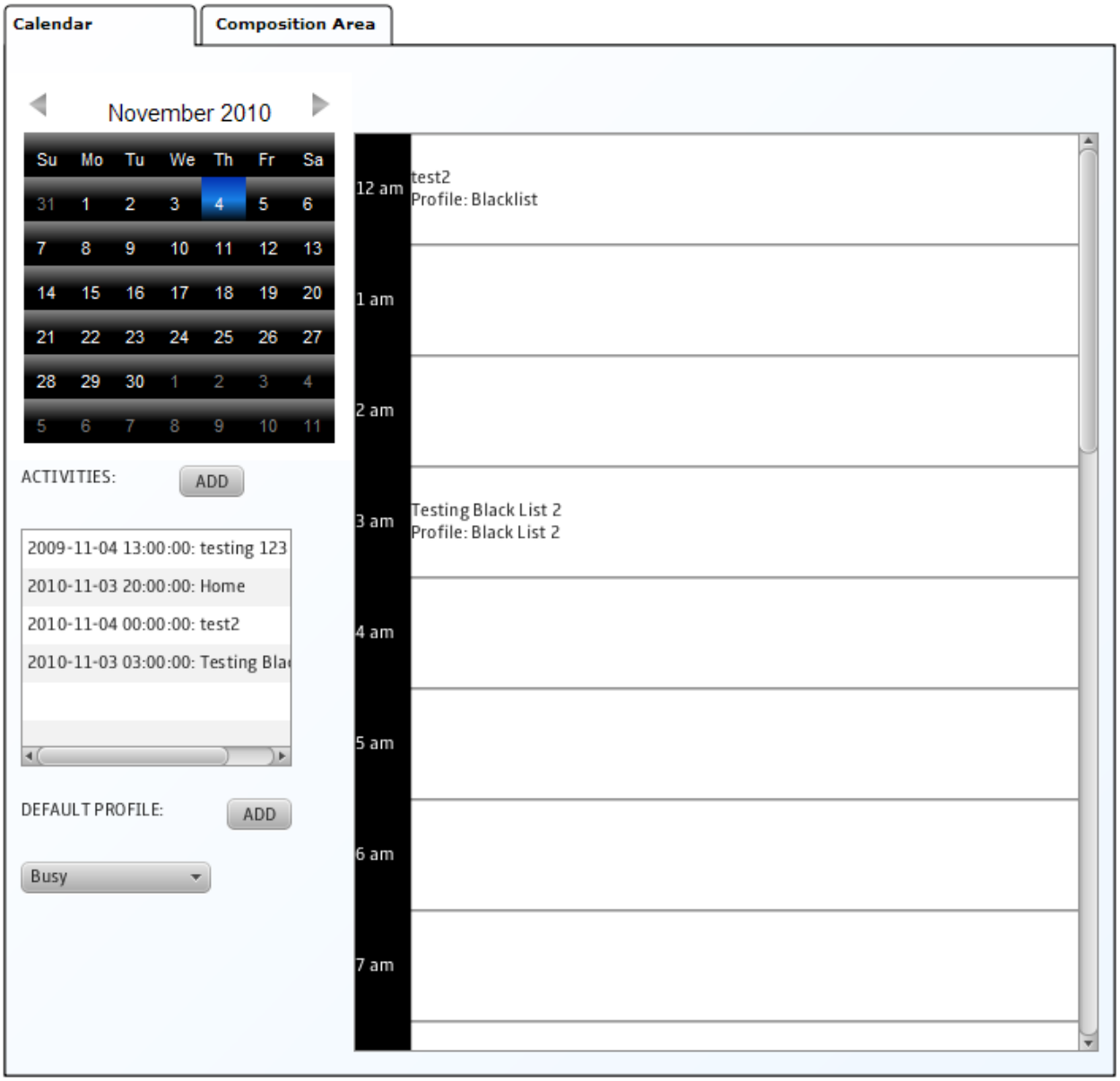

Figure 4.3 Easy composer calendar

The figure shows an entry of activities that the user has entered and the profiles that the user has attached to the particular activity which will be activated at that particular time. In addition the user also selects the default profile which is the composition which will be

running at all other times when there is no activity scheduled. The user composes his/her services at the composition area as shown in figure 4.4 and saves the XML output as a profile.

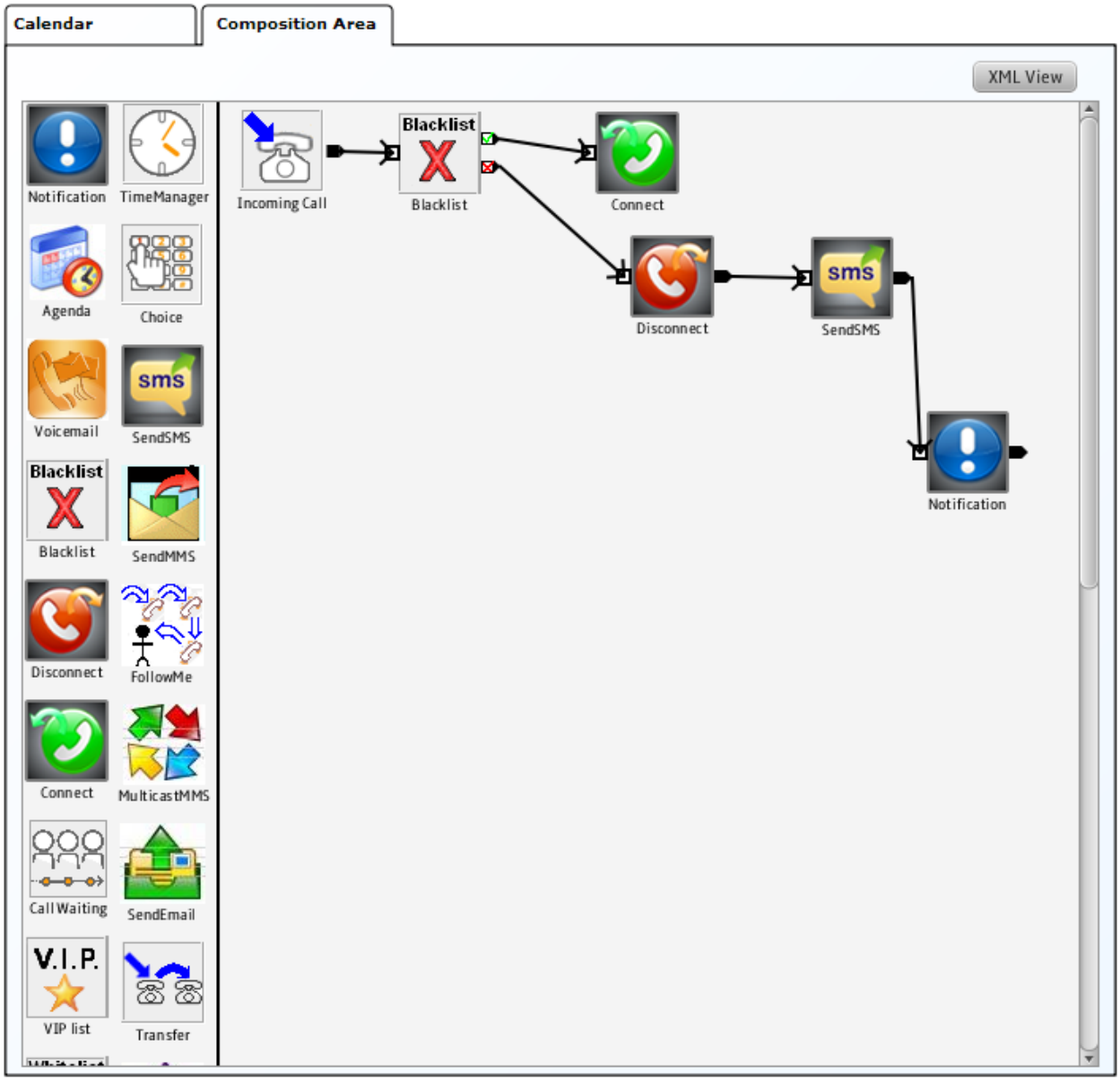

Figure 4.4 Service Composition

When the user has finally finished composing all the services and entering the activities and their active profiles he/she will then export an XML file that contains all the activities and compositions at the specified times. This will then be fed into the Easy Droid and the

engine should be able to coordinate the different compositions to run at different times as specified in the XML output.

4.2.2 Additional Components

In order for the contextual information to be captured effectively two other components will be added to the EasyDroid these components are also represented in the EasyComposer. These components will capture the terminal settings and also allow for location based service composition. The settings component will enable the user to set the terminals settings. The other component is to deal with user locations.

- **Settings Component:** This component is simply meant to capture the settings that the user wishes to be active when a particular service profile is being executed. For instance in figure 4.5 we can change the ring volume level, select the mode depicting if vibrate is on or off and so on. This allows for the control of the terminal, therefore this can be looked at as the terminal profile.

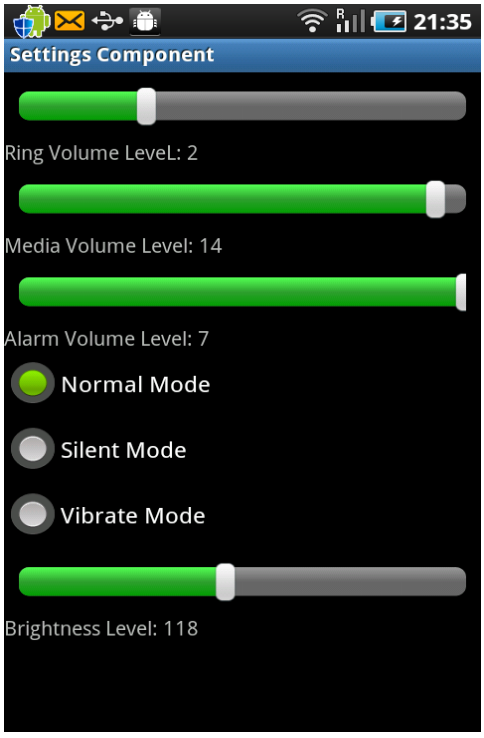

Figure 4.5 settings component

- **Location Component:** This component is meant to allow for location based service composition whereby the component is to be fed locations and proximities within these locations which enables for certain services to be activated or deactivated when the user is within the proximity of the location. For instance, if a user enters a certain location notifications or SMS is sent to the user. Figure 4.6 shows how this component looks like.

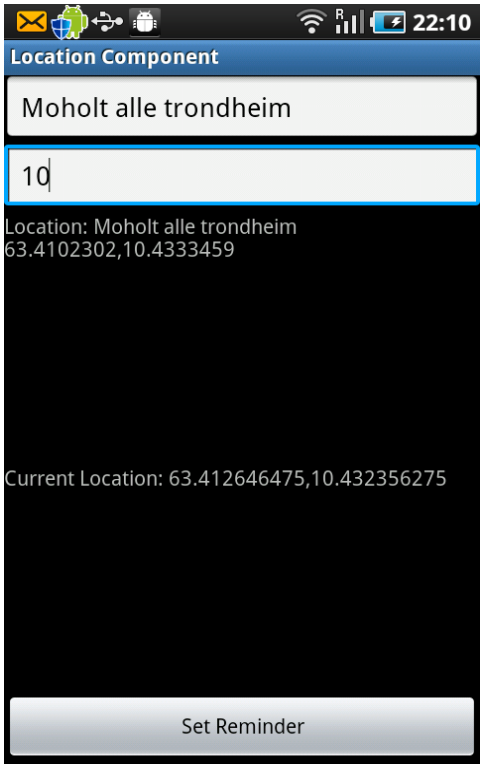

Figure 4.6 Location component

4.2.3 XML Representation

Having added the calendar and service profiles the XML created by the EasyComposer has to be extended to include this information. Therefore new XML developed is as follows:

```
<Compositions>
<Composition type="1" firstCid="0" time="0800-1600" id="1" date="0" latitude="..." longitude="..." radius="...">
<Component type="1" cid="0" name="Blacklist" nextCidIfMatch="1" nextCidIfNoMatch="2" matchType="1" />
<Component type="2" cid="2" name="Let Phone Ring" nextCid="-1" action="4" />
<Component type="2" cid="1" name="Reject" nextCid="3" action="1" />
<Component type="4" cid="3" name="Filter Notification" nextCid="-1" tickerText="Call rejected" contentTitle="Call rejected" contentText="@a was rejected by blacklist" />
<Component type="4" cid="4" name="Agenda Notification" nextCid="-1"tickerText="Call rejected" contentTitle="Call rejected" contentText="@a was rejected by agenda" />
<Component type="3" cid="5" name="Agenda Component" nextCidIfBusy="6" nextCidIfAvailable="2" />
<Component type="2" cid="6" name="Reject" nextCid="4" action="1" />
</Composition>

<Composition type="4" firstCid="0" time="0" id="6" date="0" latitide="0" longitude="0" radius="0">
 <Component type="1" cid="0"  ..... />
</Composition>
</Compositions>
```

The changes implemented are that another new tag called compositions is added. This new tag is the overall tag. This tag will contain all the compositions that have been created. These compositions have the time and date attribute which show when they are to be implemented. In addition the longitude and latitude attributes are added to dictate the locations where these compositions are to be executed.

CHAPTER 5: SCENARIO-BASED ASSESSMENT

The Scenario based assessment will assess the capability of the software tool developed with respect to performance objectives of the tool that is if it the Composition tool allows for end users to compose value added services from existing services. The assessment approach is based on the PASA method [10] which aims at assessing the EasyComposer calendar, described in chapter 4, to see if will be able to achieve the task of end user service composition. Three scenarios are identified and discussed in this chapter, which include the doctor's scenario, student scenario and the travelling businessman scenario.

The approach followed is the IBMs model for designing SOA solutions [11], where each scenario is subjected to several steps in order for the perceived end users in the scenarios can come up with compositions. The process starts by focusing on the goals and objectives that are to be achieved. Next the processes and tasks that have to be done in order to achieve these goals and objectives are explored. This will provide the functional requirements that the SOA solution has to meet. The modeled tasks and processes will then be used to help identify services that will be useful in creating an SOA solution that meets the requirements. Service specifications and implementations will then be created that fulfill these requirements. Therefore in each scenario the process will follow the following steps: service identification, service specification, service realization, service composition and service implementation. While these process is being followed a more detailed discussion of the composition tool (EasyComposer calendar) and the specific features that support the scenarios is carried out.

In addition the results are explored pointing out the problems found in more depth giving alternatives for meeting performance objectives identified and an analysis is carried out to determine whether it will support the general performance objective of end user service composition. The chapter that follows presents recommendations on future work that can be carried out to improve the composition tool.

5.1 Doctor's Scenario

5.1.1 Summary

A doctor creates a service to help patients, in distress, that call in. The service is meant to ensure that patients get help even when he/she is out of reach. There is also the need to ensure that interruptions by the phone are avoided during consultations with patients. A step by step approach on how to compose a service on the EasyComposer Calendar is looked at to meet the doctor's needs.

5.1.2 Problem description

When patients get ill they call the doctor in order to request for a meeting with the doctor. In some cases the patients might not be able to reach the doctor. Concerns arise when a patient who is in a critical condition tries to reach a doctor and the doctor is unavailable and have no one else to turn to. This means that the doctor should always have his/her phone on at all times. But in some cases this would not be appropriate, as some of the incoming calls might not be urgent and might lead to undesirable disruptions, for example when in a meeting with another patient.

In essence a way for the doctor to manage his appointments without any interruptions from the phone and at the same time come up with a solution for urgent

issues is important. In that someone else can receive the call, for instance the nurse and determine if the case requires urgent attention or not and if need be alert the doctor.

5.1.3 Main actor (s)

- The doctor the patients can call when in distress. He/she has a mobile phone that he carries with him that he can receive calls from patients.
- The nurses who are called incase the doctor is out of reach and deal with the issue at hand and determine on the best way to help the patient.
- The patients that call in.

5.1.4 Composition Scenario

A patient calls the doctor, but it's after hours or the doctor is currently with another patient, and the doctor is out of reach. Instead of taking a message and saving it for when the doctor is available, the call is forwarded to a bank of nurses. The nurse receives the call and listens to the patients problems, and determines the urgency of the case at hand. If it is urgent, the nurse contacts the doctor on behalf of the patient in certain cases the nurse might even organize on how to get the patient to the emergency rooms or offer any assistance the patient might need before getting medical care. This means the nurses will be able to contact the doctor even when he is out of reach. Otherwise, the nurse saves the message for when the doctor is available and passes it on to him/her. At each step in the process, the nurse sends a text message back to the patient to let him know what's happening, e.g. "We have asked the doctor to speak to you on the phone; stand by and he'll be calling you soon."

On the other hand the doctor could be in with a patient in an ongoing appointment. The phone is on and while the doctor examines the patient the phone rings four times, disrupting the examination and finally picks up the phone just to find out the case was just a cancellation. This gets the doctor agitated. In order to avoid discomforts to the patients and allow the doctor carry out his job with the keenness required the phone should not be allowed to ring when in appointments. The doctor has the option to keep on switching on and off the phone during the appointments; another option is to switch between flight mode or silent mode and normal mode. In the case where the phone is off there is still the case where the doctor misses an important call. Therefore the doctors phone should contain all his appointment information; this could be in a calendar. In the case where the appointment begins the doctors phone should automatically switch to one of the customized profile that will allow the nurses to contact him/her in times of dire consequences and minimize unnecessary disruptions.

5.1.5 Service Identification and specification

In order for the doctor to come up with the required solutions he has to identify the particular services that should be connected in order to meet the need. The starting point is by capturing the goals and objectives necessary to realize the need. The next step is to model the processes that are necessary to meet the goals and objectives. Finally the process help identify the required services and the potential relationships between them.

5.1.5.1 Goals and objectives:

In the above scenario the goal is to ensure that the patients always get a response when they call in. In addition the doctor's phone should react according to a particular

context as appropriate, thereby allowing for him/her to be able to avoid unnecessary disruptions and at the same time giving the callers an alternative.

5.1.5.2 Service Behavior:

Figure 5.1 shows the process involved in the doctor scenario, in this case there will be an incoming call to the doctors phone if the doctor is available he receives the call otherwise the message is forwarded to the nurses. The nurse receives the call and determines whether the matter is urgent or not if it is then the doctor is called, if it is not message about the case is sent to the doctor.

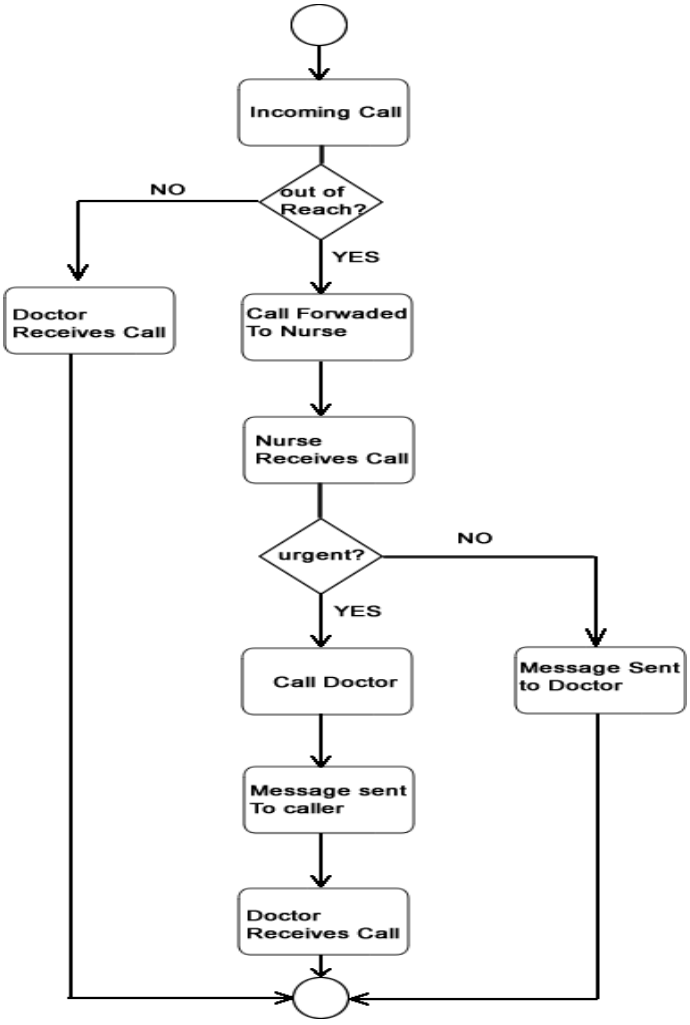

Figure 5.1 processes in the doctor scenario

5.1.5.3 Required Functionality (Services):

From the process above the services that will be required include the following:

1. A service that allows for call forwarding, this is in case the doctor is not available. The call forwarding service will redirect the caller to a different number that is set by the doctor.
2. Appointment service
3. In addition a messaging service is required to send the messages to the intended recipients. In this case the message can be a voice message or a text message.
4. Finally a filtering service is required. This will filter the caller numbers, in that the doctor can receive calls from certain numbers and not others. This service will allow the nurse to get through to the doctor while the patient's call is forwarded, when the doctor is out of reach.

In this case the doctor should be able to filter out numbers and specify numbers that should be able to reach him at all times, such as the nurses line. Any other number should be forwarded to the nurses' number. Furthermore the doctor should have the call that has been allowed to choose whether to leave a voice message or whether to talk to him. In addition the nurse can send a text message to the doctor.

5.1.6 Service Components for Realization

The next step is to determine how the services identified can be realized. In this case we identify the service components that will be able to achieve the need. Each of these components provides services and capabilities according to the service specifications. Each provided service component has a method that describes how the service is actually implemented.

The following components are identified:

1. CallTransfer component: This component allows for the forwarding calls to another number. This allows for the patients call to be forwarded to the nurse when the doctor is out of reach.
2. SendSMS component: this component allows for a set message to be sent from one party to another when triggered. In this case the sendSMS component can allow for the nurses to send text messages to the doctor about a certain case. This can be in the case when the voice mail is full or when text messaging is appropriate.
3. Agenda component: this component consults the agenda to check if you are busy at the moment or not. In this case the agenda component will check if the doctor is in an appointment or not.
4. Notification component: this component gives the user information about an ongoing back ground process. In this case it lets the doctor know whether he/she received a voicemail message and also that a call was forwarded when he/she was busy.
5. VoiceMail component: this component allows for a voice message to be placed for the doctor.
6. V.I.P component: This is a filter component that allows for the doctor to receive calls only from certain numbers while others will be forwarded to the nurse.
7. Choice component: this component allows for the nurse to choose whether to directly speak to the doctor or leave a voicemail message.

5.1.7 Service Implementation

The final step is to assemble the service components identified to use their capabilities in the creation of a new service composition. That is, we will be creating a new service from the composition of other services. Using the EasyComposer calendar the doctor can carry out this process of service composition. In the first step the doctor will compose the telephony service from the composition area which contains the EasyComposer, then saves this as a service profile. In the following step the doctor enters the appointments in the calendar and attaches the service profile to be activated at the times when the service is meant to work. Finally the doctor exports an xml that is deployed to his/her android phone that will be running the Easydroid. These service profiles will be activated according to the appointments scheduled and work as usual all other times.

5.1.7.1 Service Composition:

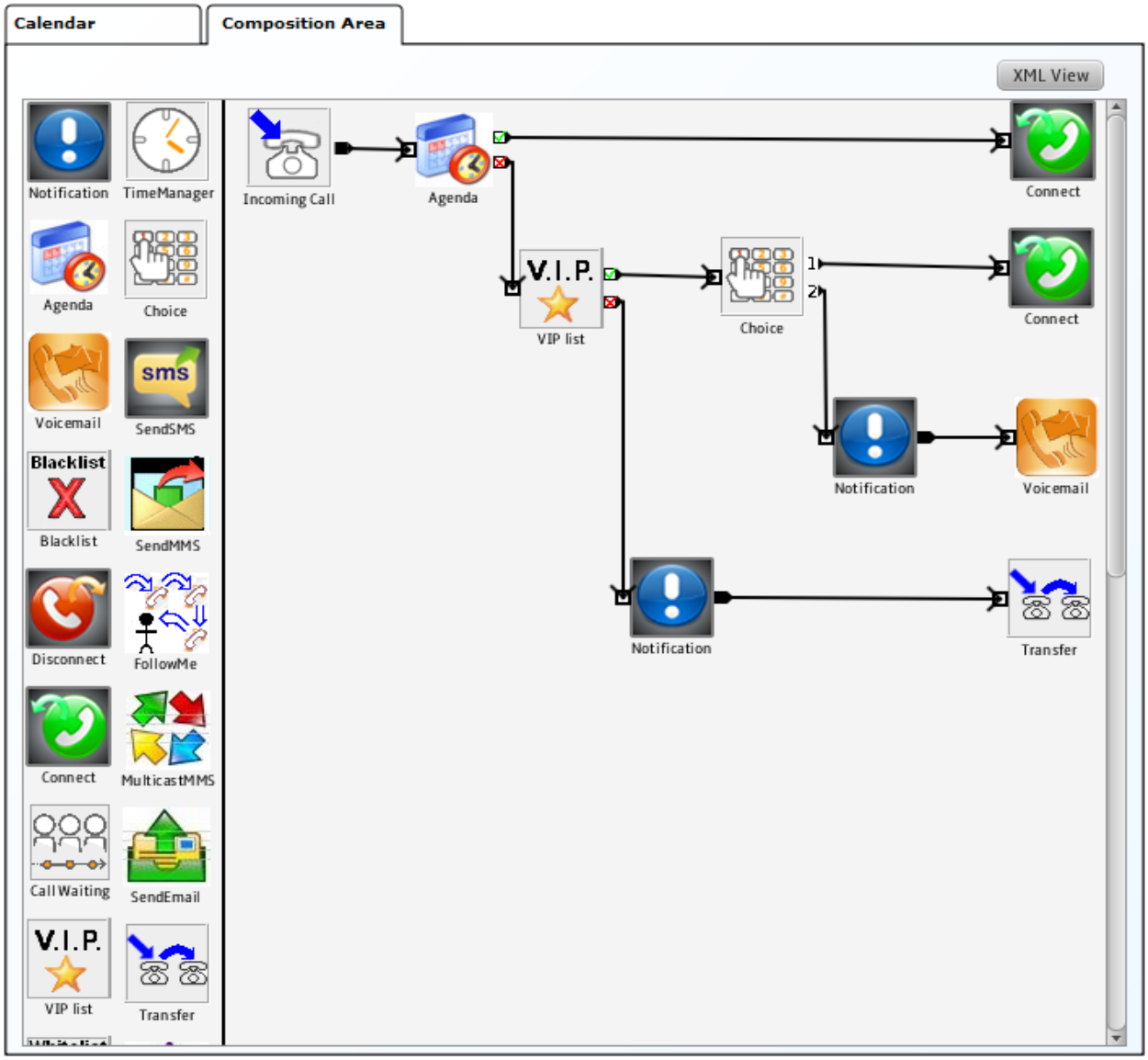

Figure 5.2 Doctor's scenario service composition

From the figure 5.2 it is clear to see how the doctor composes his/her service. When a call is incoming the agenda component is invoked which checks whether the doctor is in an appointment or not. If not the call is connected otherwise the V.I.P component is invoked to check whether the caller is on the filter list. If the caller is on the list the choice component is invoked which gives the caller a choice to either leave a message or get connected. If the caller wants to leave a message they are directed to the voicemail and a

notification sent to the called party by the notification component, otherwise the call is connected. On the other hand if the caller number is not on the filter list the call is transferred by the call transfer component and the called party is notified.

This is then saved as a service profile which can be used by the doctor when scheduling the appointments in the calendar. Many more compositions can be created for instance for the case when the doctor is at home.

5.1.7.2 Scheduling activities and attaching service profiles:

In order to allow the phone to react to the different contexts the doctor has to attach the service profiles to the activities to be scheduled.

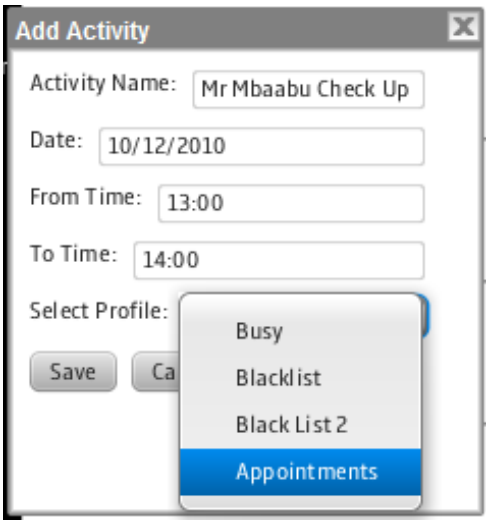

Figure 5.3 Adding activities with profiles

Here the doctor enters the activity and in addition selects the profile that matches the specific activity as shown in figure 5.3. The service composed is named as appointments so when the doctor is adding the activity for “Mr. Mbaabu Check Up” he/she will also attach the appointments profile to it.

The idea here is to attach the contextual information to allow give the system enough information for it to react when the time for the patients appointment is reached.

Once the doctor is done adding all the appointments into the calendar he/she should be presented with the calendar as shown in figure 5.4. Notice that on this calendar the doctor also can select a service profile that can be set as the default profile that will run at all other times that are not scheduled in the calendar for instance the doctor can be on the busy profile which shows that he/she does not wish to be disturbed at the moment.

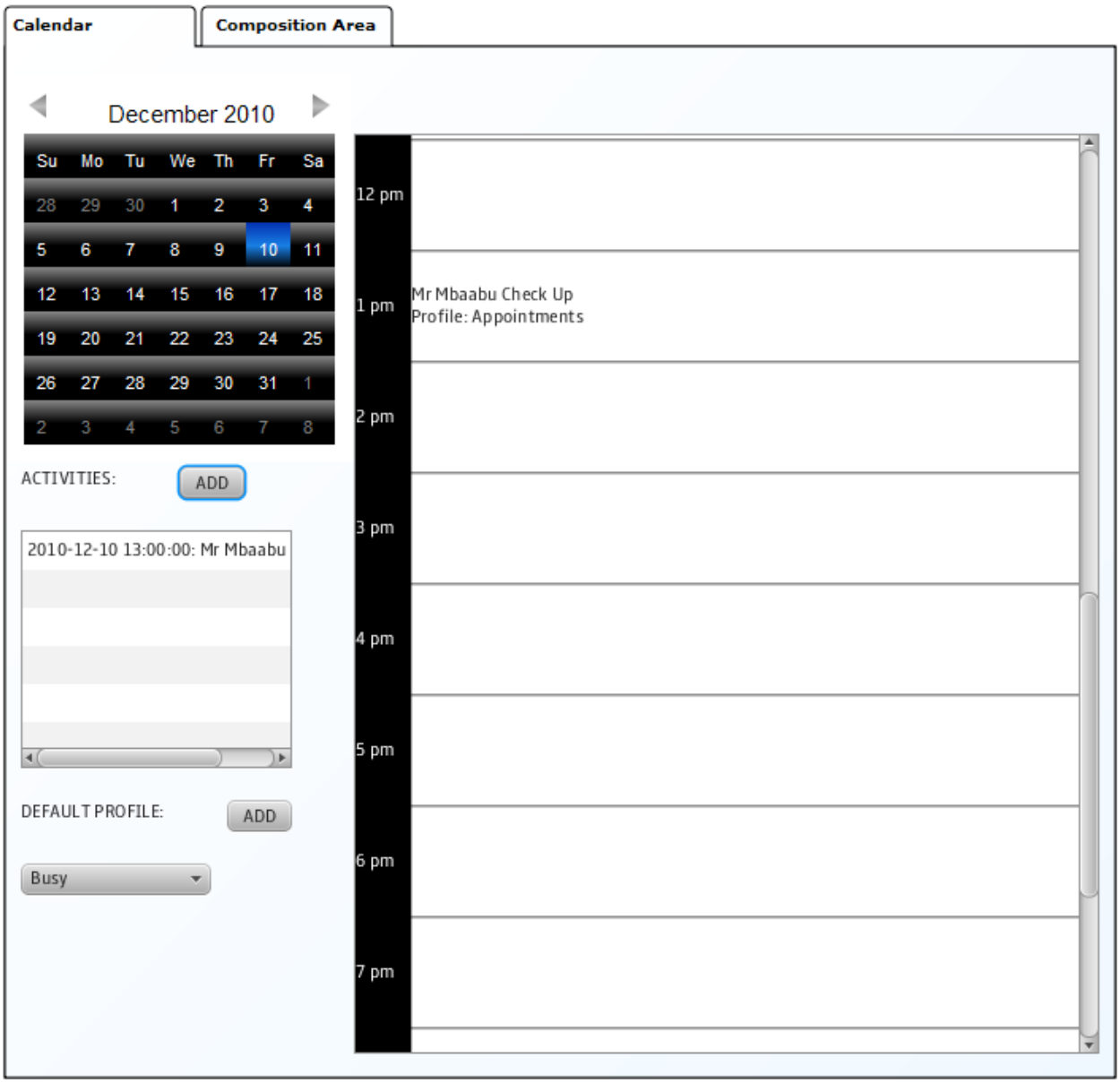

Figure 5.4: View of the calendar with the scheduled activity.

5.1.7.3 Exporting the XML

Once the doctor is done with the composition and calendar scheduling activities he/she has to export the XML that will be used on his/her android phone. This is done by simply pressing the export button. Finally the doctor deploys this XML to the android phone and the Easydroid engine will work out when and when not to activate the service compositions. The exported XML can be viewed in appendix B1.

5.2 Student Scenario

5.2.1 Summary

A student creates a service to help in planning for the lectures. A mobile phone is an important tool for communication and other personal management services such as having alarms, calendar, etc. With all its advantages most users don't make the full use of this technology. In the case of this scenario the tool should help the student manage and plan his daily student related activities. The point in case is whereby a class in progress is once in a while disrupted by ringing phones, because some students forget to change the settings. A step by step approach is followed to compose services that will help the students manage his/her lectures with reminders and also avoiding disruptions in class.

5.2.2 Problem description

Even though mobile phones are useful tool for communication in certain areas it might be highly unsuitable due to disruptions. In many cases schools have banned mobile phones. Still we have to come up with solutions that make the use of these powerful tools minimizing their negative aspects. Many a times students need a tool to manage their time. This is probably due to the high bulk of work that the students have. In this scenario the

idea is to show how the disruptive nature and bring about the benefits that students can get from using the mobile phone as a planning tool.

5.2.3 Main actor (s)

- The Student who requires planning his/her study schedule on the phone and manage his calls at the different times.

5.2.4 Composition Scenario

While in a lecture at the university the student phone starts ringing loudly. Everybody turns back looking at him, the lecturer stops lecturing and says “please make sure that all phones are switched off in my class”. Embarrassed the student quickly switches off the phone thinking “this is like the third time this has happened to me this week”. The lecturer finds it difficult to take off from where he left off as he has forgotten where he had left off.

Once the class is over the student is excited that the day’s activities are over and quickly heads home. While in the bus he remembers that he forgot to turn on his phone. On turning it on several messages stream in from his several of his classmates asking where he was. Shortly one of the student’s friend, fellow group member in the semester’s assignment, calls him asking “where are you?” the student answers confidently “on the way home”. His friend cannot hide his disappointment as they were meant to have a group meeting to finish up on the assignment as it was due the next day. After the call the student remembers that he had another assignment that was due on the same day, which he had planned to do but had forgotten.

It is clear that the student has a forgetful nature and has issues with managing his time. He requires a tool that can help him manage his study schedule as well as give him

reminders of due dates and other important information. His phone seems the best option as he has it always with him and can set reminders for the different tasks that he wishes to accomplish which go off at a time before the due date or time. But this brings about another problem because he has to have his phone on at all times, but with his forgetful nature he might forget to put the phone to silent mode. Therefore he requires that all calls be automatically disconnected when in class to avoid unnecessary disruptions when in lectures at the same time the caller should also be notified that the student is currently in a lecture by a text message. This means that the phone should contain the students lecture schedule. Also the reminders should go off but in a very low tone or in vibration mode when in lectures thus the volume settings should be on a low profile or off with vibration being on when in lectures. This will allow the student not to miss any reminders.

The student also has been planning on visiting the library to pick up a book for the Norwegian course that he is currently pursuing. The library is on his way home but always forgets to pass by the library, probably because he is tired from the day's events and just wants to go home and rest. The phone should also contain reminders that are activated whenever he is within the vicinity of the library reminding him to pick up the book. This should allow the student to log in any location reminders that he needs and gets reminded when they reach the location.

5.2.5 Service Identification and specification

In the same way as the doctor scenario the first step to identify the particular services that should be connected in order to meet the need. The student creates the goals and objectives necessary to realize his/her need. The student further models the processes

that are necessary to meet the goals and objectives. These processes help in identify the required services and the relationships between them.

5.2.5 .1 Goals and objectives:

In the above scenario the goal is to ensure that the student plans his/her time wisely. This by allowing the student know where he should be at what time and do certain things important to his/her study such as reminder of deadlines as well as reminders when in certain locations to carry out certain activities. In addition the student's phone should react according to a particular context as appropriate, thereby allowing for him/her to be able to avoid unnecessary disruptions when in class and at the same time giving the required notifications.

5.2.5 .2 Service Behavior:

Figure 5.5 shows the processes involved in the student scenario. In this case we have three processes that can occur concurrently. The first is when a call is incoming and the first step is to check if the student is in a lecture if so then the call is disconnected and the caller is sent a reject message, otherwise if the student is not in a lecture the student can receive the call. The other process is when a reminder to go off is activated when the set time is reached in this case if the student is in a lecture the settings are activated for a low profile. The reminder could also be because the student stepped in a particular location. If the student is in a lecture situation the settings are set to low profile as well. In the case the student is not in a lecture the reminders go off as usual.

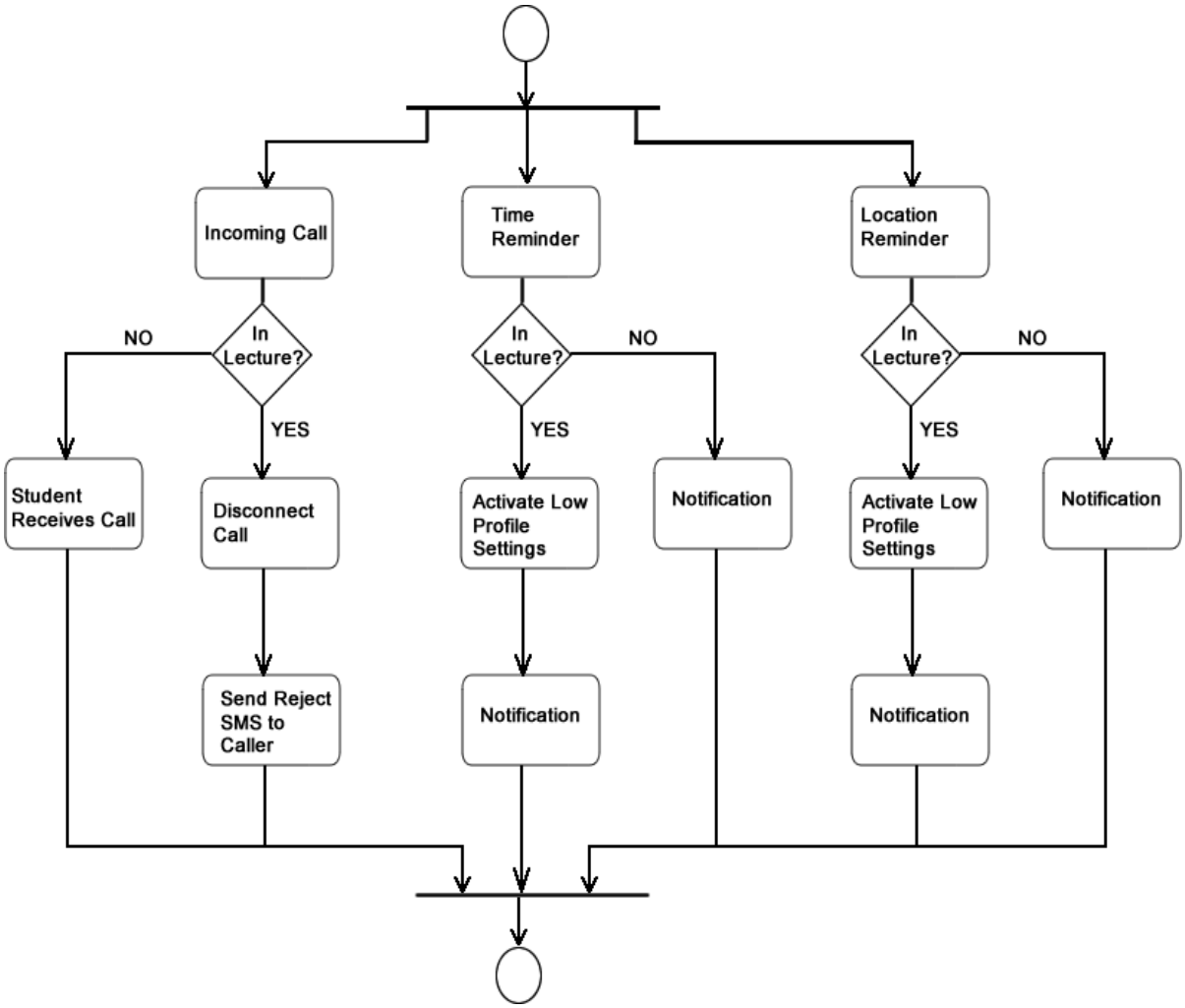

Figure 5.5 processes in the Student scenario

5.2.5 .3 Required Functionality (Services):

From the process model in figure 5.5 the services that will be required include the following:

- A service that allows for the Scheduling information about the student to be stored and will be able to answer the question whether the student is in a lecture or not.
- In order for the students to be notified of the different due dates and locations entered a reminder service is also required.
- Settings service will play the role of defining which settings need to be activated at a particular instance, for instance when in a lecture the volume should be at level 1.

- SMS service will come in handy when sending rejection text messages when in lectures to inform the caller that the student is not available at the moment.

5.2.6 Service Components for Realization

In order to realize the services identified above the student should determine which components to use. In this case we identify the service components that will be able to achieve the need. Each of these components provides services and capabilities according to the service specifications.

The following components are identified:

1. Agenda component: this component consults the agenda to check if the student is in a lecture at the moment or not.
2. Settings component: This component keeps track of all the settings at any particular time. In our scenario this will play a big role in ensuring that the appropriate settings are activated whenever the student is in a lecture.
3. SendSMS component: this component allows for a set message to be sent from one party to another when triggered. In this case the sendSMS component can allow for the student to send text messages to his/her callers when he is not available, that is when in a lecture.

5.2.7 Service Implementation

The final step is to assemble the service components identified to use their capabilities in the creation of a new service composition. That is, we will be creating a new service from the composition of other services. Using the EasyComposer calendar the doctor can carry out this process of service composition. In the first step the doctor will compose the telephony service from the composition area which contains the

EasyComposer, then saves this as a service profile. In the following step the doctor enters the appointments in the calendar and attaches the service profile to be activated at the times when the service is meant to work. Finally the doctor exports an xml that is deployed to his/her android phone that will be running the Easydroid. These service profiles will be activated according to the appointments scheduled and work as usual all other times.

5.2.7.1 Service Composition:

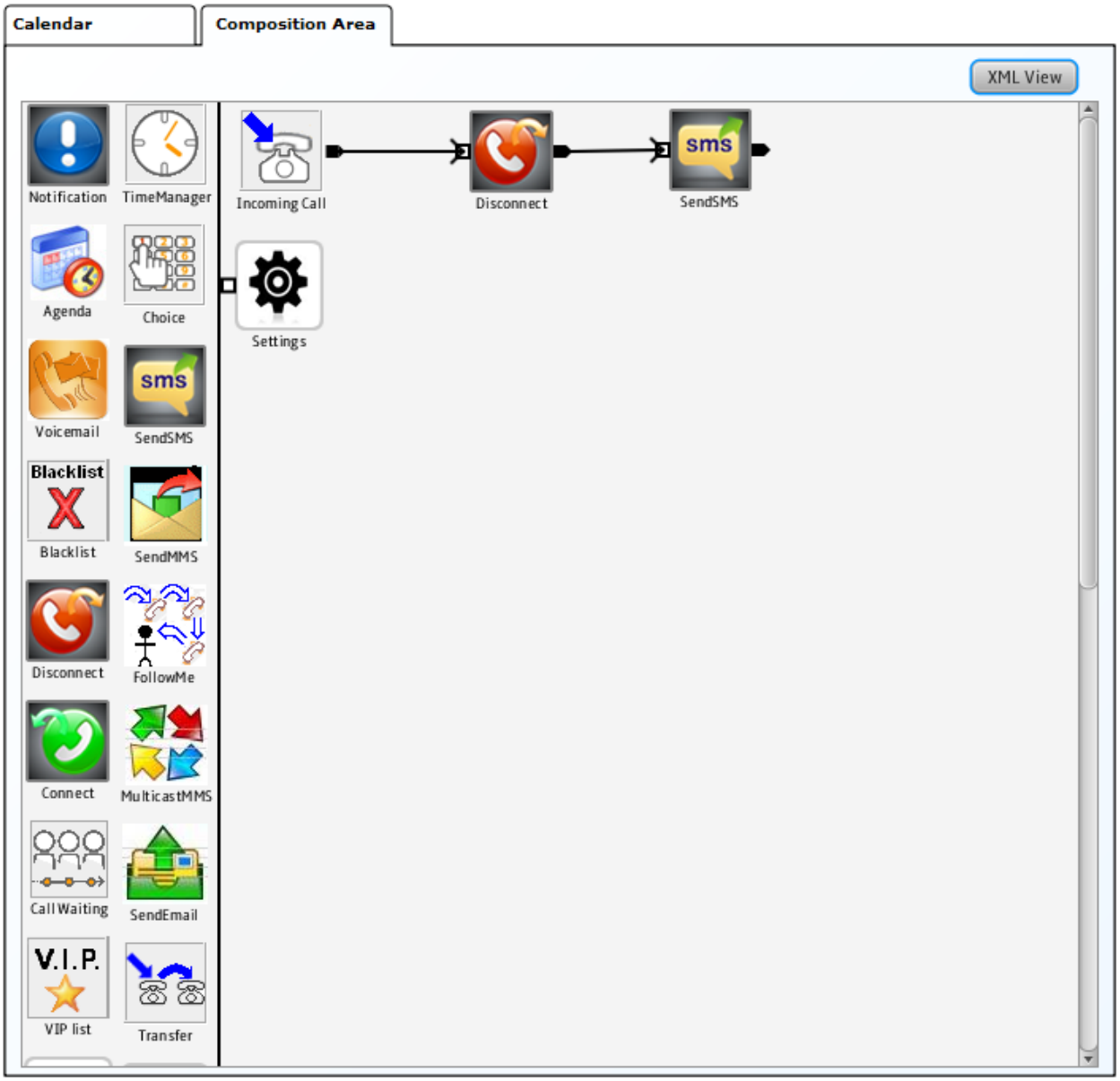

Figure 5.6 Service composition of doctor's scenario

From the figure 5.6 it the student creates a simple service which should be able to handle the problem of disruptions whereby when a call is incoming the call is simply disconnected and a reject text message is sent to the caller informing the caller that the student is in a lecture, in addition settings suitable for the lecture in the case of reminders. This is then saved as a service profile which can be used by the student when scheduling the lectures in the calendar. Many more compositions can be created for instance for the case when the student is in a group meeting.

5.2.7.2 Scheduling activities and attaching service profiles:

In order to allow the phone to react to the different contexts the student will attach the service profiles to the activities to be scheduled.

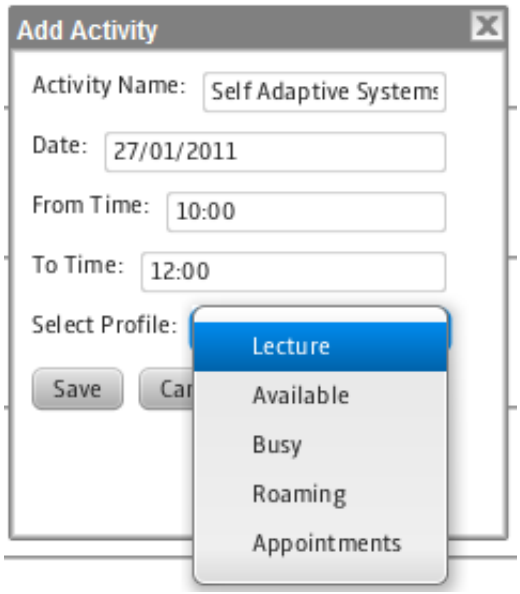

Figure 5.7 Adding activities with profiles

The student will go about this by entering the lecture within the time in which it falls in and in addition selects the profile that will be suitable for the lecture as shown in figure 5.7. The service composed is named as Lecture so when the student is adding the self adaptive systems lecture he/she will also attach the Lecture profile to it.

This will allow the student to attach the contextual information to allow the system enough information for it to react when the lecture time is reached.

Once the student is done adding all the lectures into the calendar he/she should be presented with the calendar as shown in figure 5.8. In addition the student selects the default profile that will be active whenever the student is not in the lecture.

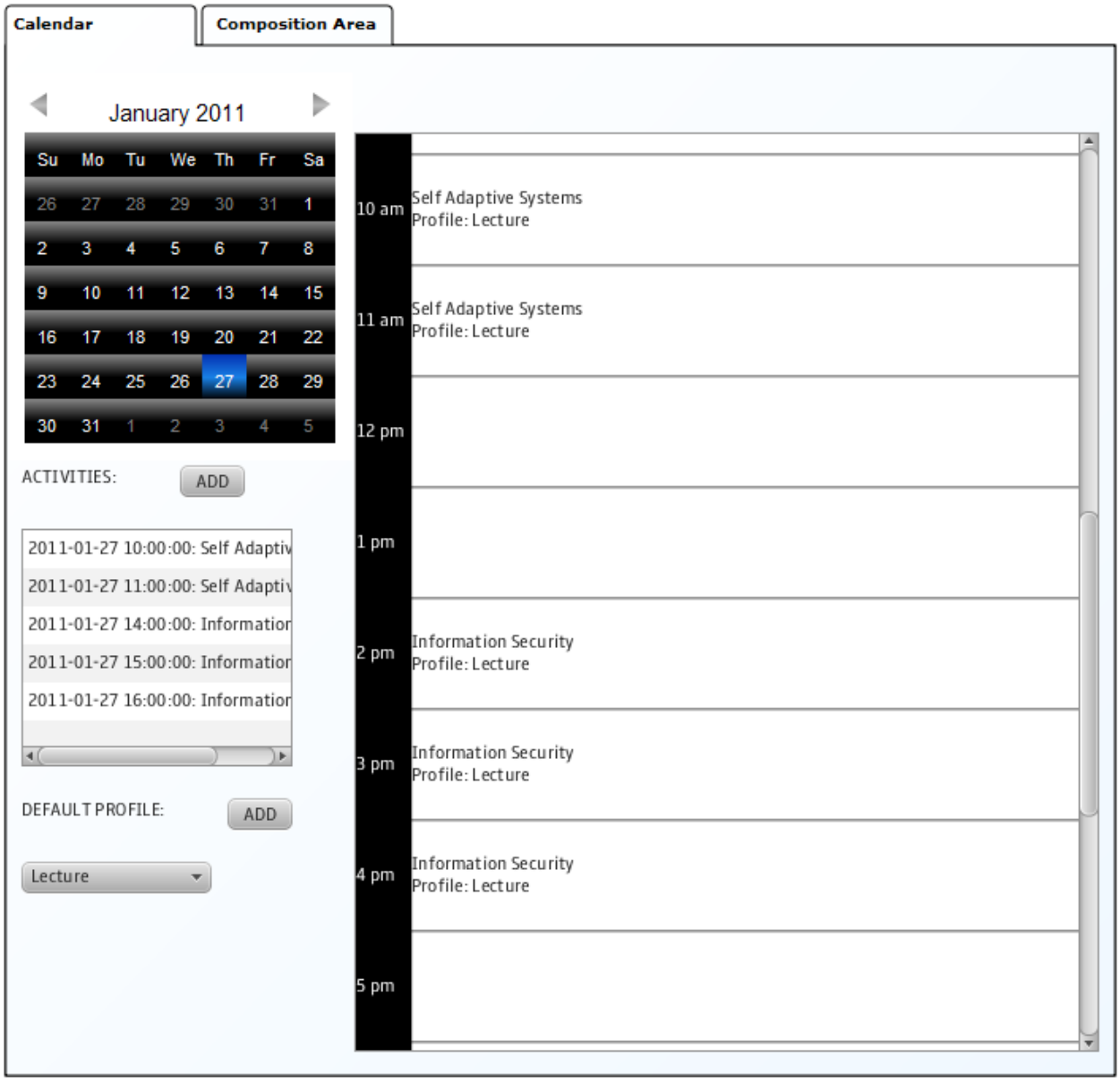

Figure 5.8: View of the calendar with the scheduled lectures.

5.2.7.3 Exporting the XML

As with the doctor's scenario when the student is done with the composition and calendar scheduling activities he/she has to export the XML that will be used on his/her android phone. This XML will be deployed to the android phone and the Easydroid engine will work out when and when not to activate the service compositions. The exported XML can be viewed in appendix B2.

5.3 Travelling Businessman Scenario

5.3.1 Summary

Drake is a business man who deals with supplies all over Europe. The business is run online on his web page. In cases where there's anything on the line, if the relationship seems strained with a client, or if he needs to explain something complicated, or introducing himself for the first time he travels to the client's site. He often finds difficulty in managing his travel arrangements and would be interested in a tool that can help him attend all his meetings with his clients as well as minimize some of the costs he incurs.

5.3.2 Problem description

Drake has a regular routine that he follows, whereby he has a list of clients and visits each one of them periodically, delivering supplies. He usually telephones ahead in order to book appointments with his clients as there is a chance that the client may be out of town, or during a busy season, not available. Since he is a busy man Drake is always on the move and making and receiving calls to plan for his meetings. In most cases Drake finds himself poorly planning for the meetings in that he could book two appointments at the same time. Sometimes when he counts on staying an hour at a client's, he finds himself loosing track of

time and spending more time than intended at the clients office and misses all other appointments he had made for the day or ends up late for the other meetings scheduled which could be a bad first impression especially with the new clients.

He is also moving a lot and when outside Norway his phone is on roaming mode and pays a lot of money as he receives many calls that in most cases he deems unnecessary. On the other hand he cannot do without the phone because he has to coordinate his meetings with the clients. Some of the major costs that he incurs have to do with receiving of unnecessary calls, voicemail costs that he is charged double as the phone is not on unconditional call forwarding. In addition he incurs costs from the internet which is streaming data backward and forward while abroad.

5.3.3 Main Actor(s)

- Drake who owns the company and travels all over Europe carrying out his supplies business
- The clients who buy supplies from Drake, they could be old clients that Drake has been in business with for a long time or prospective clients.
- Collaborating companies that collaborate with Drake in the different countries, who help out the clients when in need if they have the capacity to do so. When he is not within the country in which they operate the clients can contact these companies.

5.3.4 Composition Scenario

At this particular time Drake has several old clients that he is visiting in Sweden. He has one or two new prospects that he intends to get into business with him and he would like to meet. During his visit he wants to meet one of the companies that he collaborates

with that helps him run minor issues on the ground, while he is back in Norway. In the case when Drake is outside of Norway all calls will be disconnected apart from calls that are on his very important list at the same time the phones internet connection should be turned off since information on the internet is accessible virtually anywhere, anytime from hotel computers to Internet cafés.

While Drake is in Sweden moving from client to client, a client from Spain calls Drake, but Drake is not available, and he is out of reach because he is roaming and has set the phone to disconnect instead of going to voicemail. Instead of taking a message and saving it for when the Drake is available, the call is forwarded to a collaborating company that is within the client's country. This in turn will avoid the double charge for the voicemails. Therefore the client's calls should be forwarded to the collaborating company when Drake is out of reach.

The company receives the call and listens to the clients problems, and determines the urgency of the case at hand. If it is urgent, the company contacts Drake and informs him of the problem. In the case where Drake might be in an ongoing meeting with a client, the call is forwarded to the voicemail where the company can leave a message of the case and the particulars of the client so that Drake can contact the client to see how he can solve the problem. This means the company will be able to get through to Drake or his voicemail in the cases when he is busy. Otherwise, the company will have the capacity to help the clients and thus will not have to contact Drake.

While Drake is in an ongoing appointment, the phone should be on but in order to avoid disruptions during the meeting the phone should not be allowed to ring when in appointments. Drake has the option to switch between flight mode or silent mode and

normal mode but there is still the chance where the he misses important calls which may lead to customer dissatisfaction. A more reliable approach is where all his appointment information is stored in the phones calendar. In the case where the appointment begins the phone should automatically switch to one of the customized profile that will which is appropriate for that particular moment, such as the phone being on vibrate when in meetings and ignoring calls and at the same time forwarding important calls to the voicemail. Notifications that a company called in and was redirected to the voicemail should also be sent to the user. In addition notifications should be sent out when the meeting is about to begin and when it should be about to end in order to avoid the case where Drake overstays in the meeting messing up with the schedule.

5.3.5 Service Identification and specification

As Drake has identified a problem that he wants to solve he has to identify the particular services that should be connected in order to meet the need. The starting point is by capturing the goals and objectives necessary to solve the problem at hand. The next step is to model the processes that are necessary to meet the goals and objectives. Finally the process help identify the required services and the potential relationships between them.

5.3.5.1 Goals and objectives:

The identified goals here are to ensure that Drake is able to manage and plan meetings with his clients well and at the same time ensure customer satisfaction by giving customer service when needed. In addition Drake phone expenses should be minimized by having the phone react according to particular contexts and minimize roaming charges as

appropriate, thereby allowing for him/her to be able to avoid unnecessary charges and at the same time giving the callers an alternative.

5.3.5.2 Service Behavior:

Figure 5.9 shows the processes in the travelling businessman scenario. In this case when there is an incoming call there is a check if the businessman is roaming or not. If the businessman is roaming then the call is forwarded to a collaborating company who listen to the case. If the case is complex the company calls Drake. If Drake is busy the call is forwarded to the voicemail otherwise drake picks the call. Also when Drake is roaming the internet connection is disconnected.

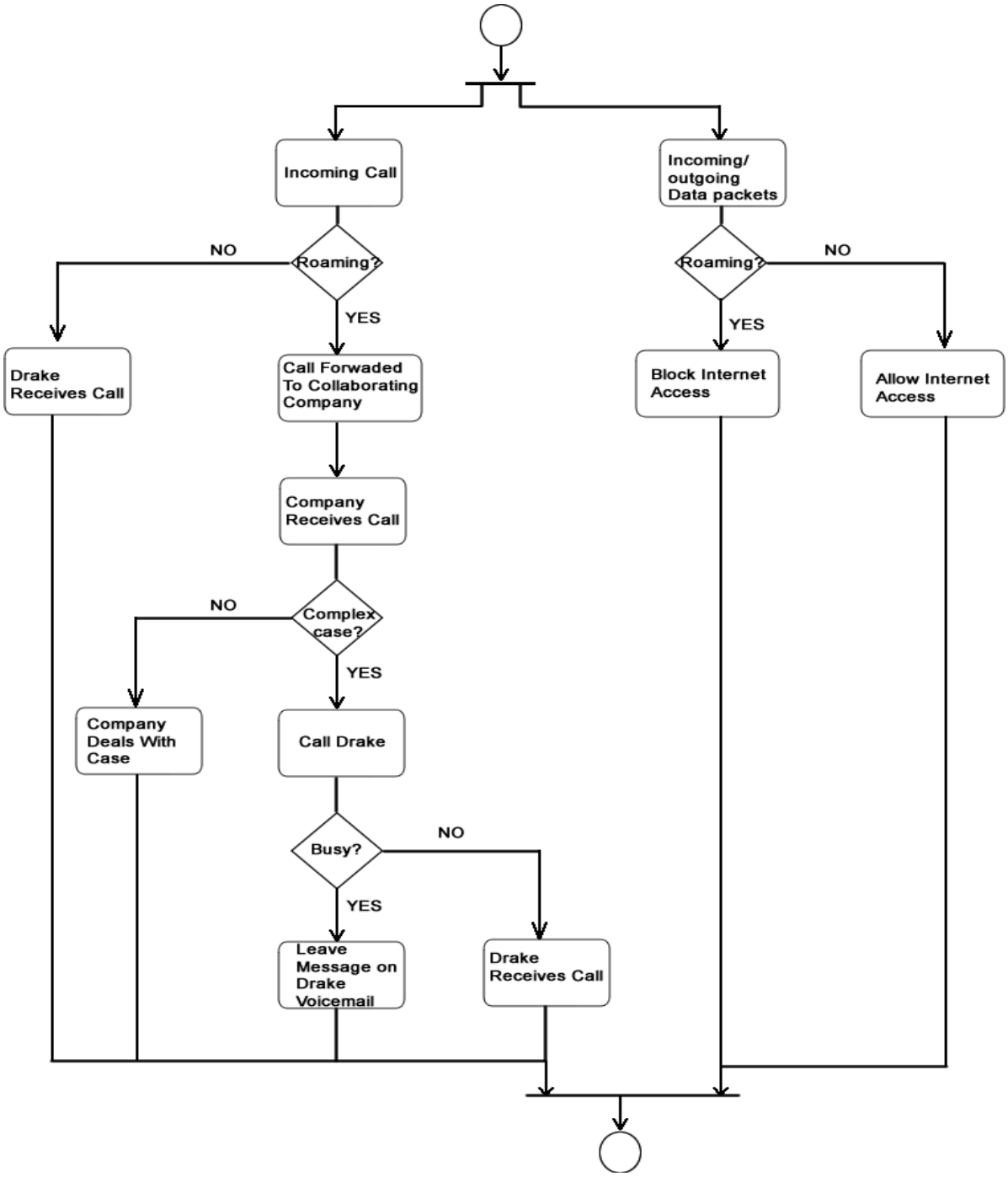

Figure 5.9 processes in the travelling businessman scenario

5.3.5.3 Required Functionality (Services):

From figure 5.9 above of the processes, the following services will be required:

- A service that allows for call forwarding, this is in case that Drake is roaming. The call forwarding service will redirect the caller to a different number that is set by Drake for the collaborating company.
- A service which allows Drake to manage all his planned meetings and will allow for Drake to know when a meeting starts and when a certain meeting should end in order to be able to plan for his meetings.
- Settings service which should allow Drake to turn on and off his GPRS thus avoiding the internet charges when he is roaming.
- A voicemail service is also required which allows callers to leave messages for Drake when he is unavailable.
- Finally a filtering service is required. This will filter the caller numbers, in that the collaborating companies can be able to get through to either Drake or his Voicemail but all the other users will not be able to get through to Drake in the case he is roaming.

5.3.6 Service Components for Realization

Similarly the next step is to determine how the services identified can be realized. In this case we identify the service components that will be able to achieve the need. Each of these components provides services and capabilities according to the service specifications The following components are identified:

1. CallTransfer component: This component allows for the forwarding calls to another number. This allows for the clients call to be forwarded to a collaborating company when Drake is roaming.
2. Agenda component: this component consults the agenda to check if you are busy at the moment or not. In this case the agenda component will check if Drake is currently in a meeting or not.
3. VoiceMail component: this component allows for a voice message to be placed for Drake when he is unavailable.
4. V.I.P component: This is a filter component that allows for Drake to receive calls only from certain numbers while others will be forwarded. In this case Drake will get calls from the collaborating companies who are on the list when roaming.
5. Settings Component, that manages some of the phone settings on Drakes phone at any one particular time. It will be able to turn the GPRS on and off when appropriate.

5.3.7 Service Implementation

As in all other scenarios in this step in this step the service components identified are assembled to use their capabilities in the creation of a new service through service composition. Using the EasyComposer calendar Drake carries out this process of service composition. In the first step Drake composes the telephony services from the composition area which contains the EasyComposer, then saves this as a service profile. In the step that follows Drakes enters the meetings in the EasyComposer calendar and attaches the service profile to be activated at the times when the service is meant to work. Therefore for all the meetings that are outside of Norway will have a service profile suitable for roaming. Finally

Drake exports an xml that is deployed to his/her android phone that will be running the Easydroid. These service profiles will be activated according to the meetings scheduled and work according to the default profile all other times.

5.3.7.1 Service Composition:

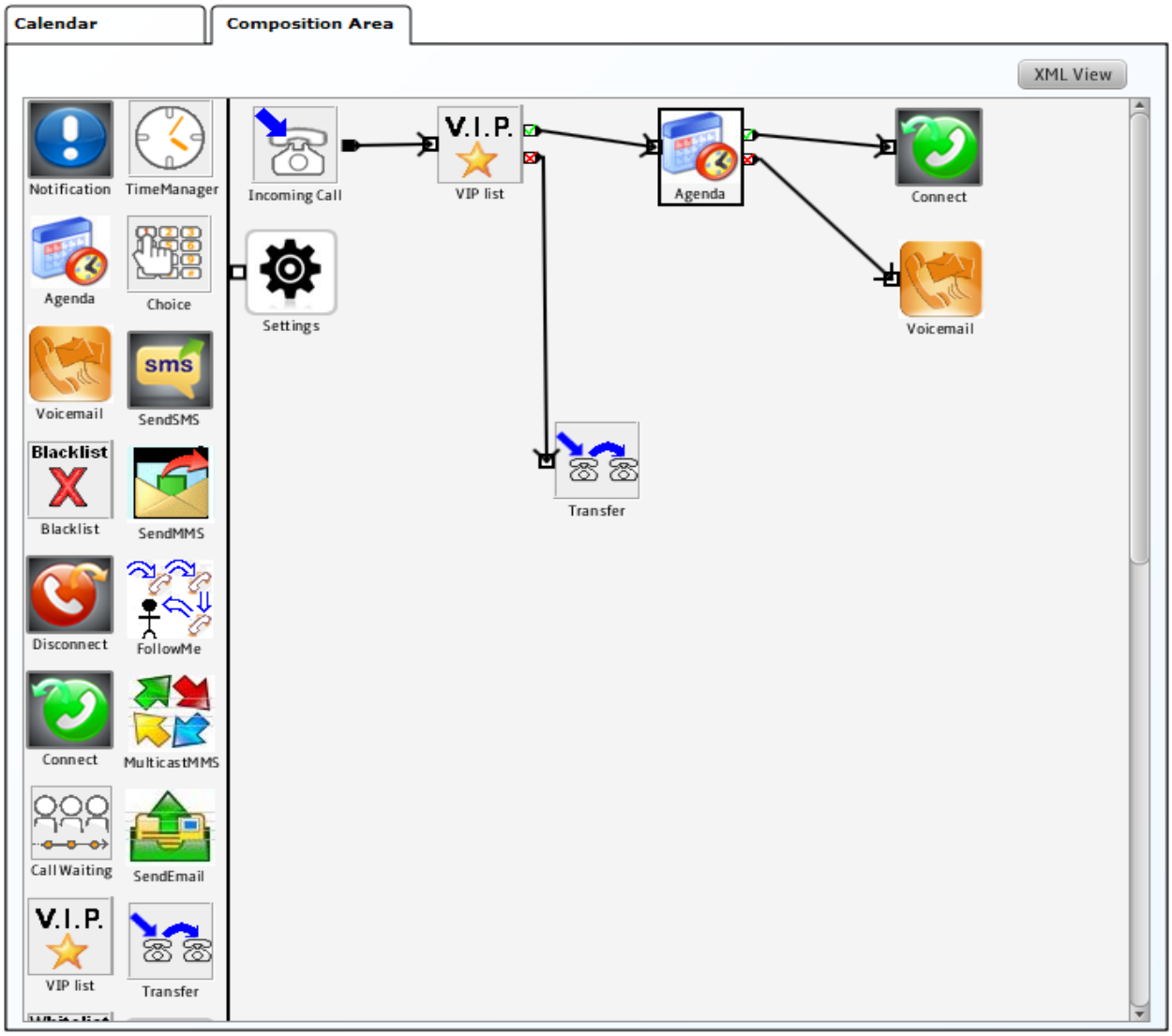

Figure 5.10 Travelling businessman scenario service composition

From the figure 5.10 Drake composes a service whereby when he is roaming and a call is incoming the filter component is invoked which checks whether the incoming call is from a number not on the V.I.P list the call is forwarded to the specified number. Otherwise

the Agenda component is invoked which checks whether he is in a meeting, if in a meeting the call is forwarded to his voicemail. If Drake is not in a meeting the call is connected.

This is then saved as a service profile which can be used by Drake when scheduling the meetings which are outside of Norway in the calendar. Many more compositions can be created for instance for the case when the Drake is in Norway where the composition can be simply when he is in a meeting all calls are forwarded to his voicemail.

5.3.7.2 Scheduling activities and attaching service profiles:

In order to allow the phone to react to the different contexts Drake has to attach the service profiles to the meetings to be scheduled.

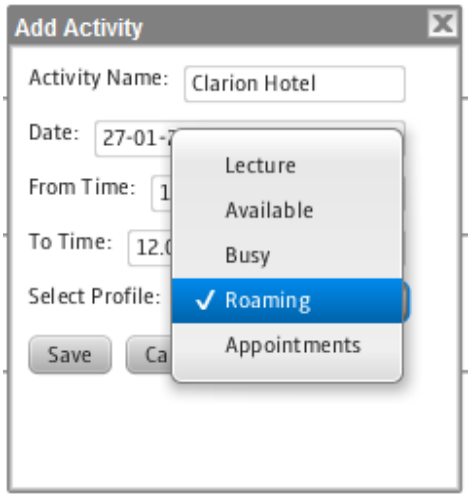

Figure 5.11 Adding meetings with profiles

In figure 5.11 Drake enters the meetings and in addition selects the profile that matches the specific meeting. The service composed is named as Roaming so when Drake is adding the Clarion Hotel meeting he will also attach the Roaming profile to it.

Since the contextual information is attached the system will have enough information that allows it to react when the time for the meeting is reached.

Once Drake is done adding all the meetings into the calendar he should be presented with the calendar as shown in figure 5.12. Notice that on this interface Drake can select a service profile that can be set as the default profile that will run at all other times that are not scheduled in the calendar for instance Drake can be on the Available profile.

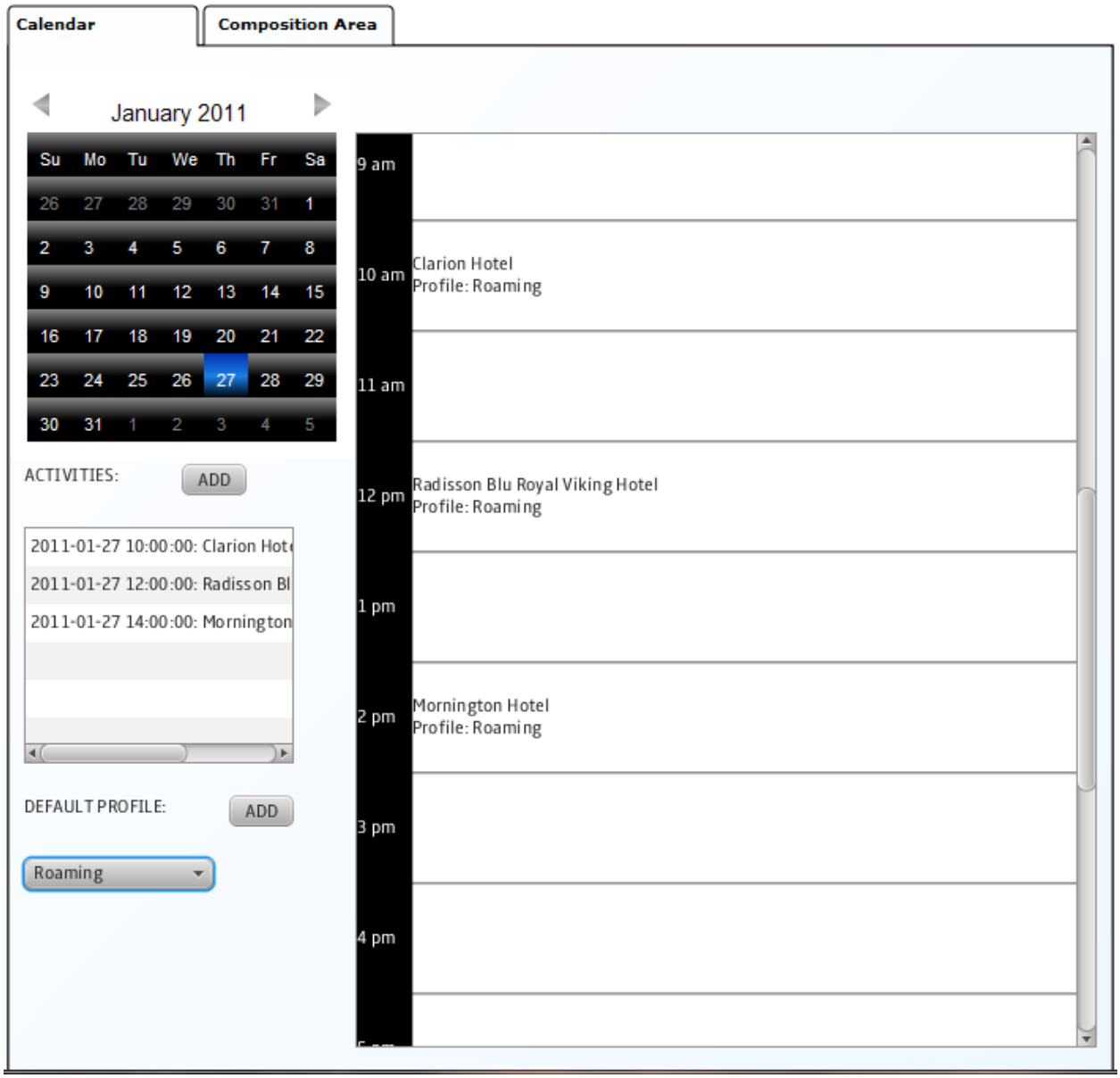

Figure 5.12: View of the calendar with the scheduled meetings.

5.3.7.3 Exporting the XML

Once Drake is done with the composition and calendar scheduling activities he has to export the XML that will be used on his android smart phone. This is done by simply pressing the export button and this XML is deployed on the android phone and the Easydroid engine will work out when and when not to activate the service compositions. The exported XML can be viewed in appendix B3.

5.4 Assessment Results

From the assessment carried with the tool we can fairly say that the end user service composition can be carried out to a high level of satisfaction. In the doctor's scenario, the doctor is able to compose the required composition that allows for his/her patients to always get help when in distress.

In the student scenario, the student is able to compose the service required to help in scheduling the days activities and adding the profiles as required. The problem identified in this scenario is that the student is not able to incorporate compositions that are location based, that will allow him/her set up reminders that give an alert when a certain location is reached. In order to accommodate for this the composition tool should be extended to include location based compositions.

The travelling business man scenario also allows for composition of his required services. The problem that arises is similar to the one identified in the students scenario, whereby the composition is only limited to time. The businessman could intend for a detection of location to identify when he is roaming rather than having it to be explicitly stated that he will be roaming at certain times. This allows for flexibility for instance in the case where the businessman makes an unplanned travel outside the country.

The assessment can be carried out considering the intended achievements of the selected approach in chapter 3 in section 3.4, whereby the following conclusions can be reached:

1. The tool follows a control based approach to service composition, allowing for services to be organized in a sequential manner as seen in the XML representations seen in Appendix B.
2. The user interface design assists the user in composing his/her services. This is seen by the use of the calendar which in essence gives the user an approach to service composition that they can relate to as most users make use of calendars such as Google calendar or the Microsoft Outlook. Further the icons representing the components give a graphical representation of the service thus the user has an easy time when selecting the services required for composition.
3. The compositions are also stored as profiles. This allows the users to use their compositions with ease in way that is much similar to the old fashioned use of profiles in mobile phones.
4. The idea of context aware service discovery is incorporated in the calendar in that when a particular time is reached a particular service composed is activated as set by the user. This is only limited to time and does not consider other context information such as the location and device features. Much more context information can be incorporated.
5. Users have the ability of composing services with the EasyComposer calendar and storing this compositions for a long period in time. The setback to some extent is that the user has always to go back to the composer to make new compositions. The

EasyComposer calendar should not only run on the desktop but should also run on the phone environment where it is meant to be used.

In general the composition process is quite simple thus the objective of end user service composition is met. The problem arises with incorporation of service compositions that involve location which is not quite straight forward. Another problem that arises is one whereby the compositions can only be composed on the desktop application. This should be possible also in the android environment to make the notion of composing of services on the fly complete.

6.1 Achievements

Value-added services by means of service composition have been made possible through the implementation of a tool that makes it easy for end users to compose telephony services. Thus these users can come up with new services on the fly without involving network operators. Having been presented with a composition tool that followed Control Flow-based Composition Approach, containing several components the idea was to find a way in which the users could make use of the tool as it was deemed too technical for the trivial user. In this case a few concepts were introduced in order to make this tool less technical and more usable. The context aware service composition was the main focus whereby service profiles would be the main focus, whereby a calendar is implemented that allows for the user to specify when and where the service profiles should be implemented. These service profiles are services that have been created by the existing composition tool where a set of self contained services is defined as components. The advantage with this is that service compositions that are commonly used can be integrated in the environment easily without undergoing through the whole process of service composition. These services are then used to create a number of composition scenarios. Additional components are also created for the android application that also existed which extends additional functionalities that have been introduced in the composition tool.

In order to demonstrate and investigate end user service composition a few scenarios looked into. The first Scenario looked into was the doctor's scenario which looks into how a doctor manages his appointments and composes a service that helps him/her do that. In addition a student scenario is looked at whereby the student composes a

scenario that helps manage his studies, with his phone and at the same time avoiding the interruptions that come about with mobile phones. Finally the travelling businessman scenario is looked at where apart from managing the meetings cuts down on costs that come about because of roaming.

6.2 Recommendations for Future Work

Based on the work carried out on the project the following can be areas of future development that can make the tool more usable in end user service composition.

First and foremost the user interface can be further polished in order to make it more user-friendly and easy to use. The calendar should be polished in order to compete with current calendars in use in the market such as the Google calendar. This further reflects on how the two applications work together. More work can be carried out to make it more fluid.

The composition tool can further be developed to incorporate the other forms of context apart from time. In this case Location based compositions can also be incorporated as well as take into consideration other factors such as the state the phone is in for instance low battery level or lack of Mobile network.

Further the two different tools, that is the EasyComposer and EasyDroid should be made to work well together without any inconsistencies in the design of both their logics. In addition the composition tool can be extended to include other services that might be useful to end users. In that it should capture a complete picture of the telephony services that exist thus catering for the needs of more users.

The application also is based on a structure that is separated into two parts. A user interface for mobile devices that allows for service composition should be developed so

that the end-users will be able to change the behavior on the go thus making the main goal of service composition on the fly more realistic instead of having to switch from one application to another which in essence might also solve some errors that might occur due inconsistency that arise from this separation.

Since the application is to be used by users experimenting the tool in a real environment may also play an insight on what improvements need to be made. It can also make an insight onto the user behaviors and reactions on the field of end user service composition in light to the current trends where all composition is left for the operators.

CHAPTER 7: CONCLUSION

The project main focus was on service composition of telephony services, messaging services and other personal management services such as appointment organizer, calendar, customized search, location handling, reservation services, etc. With the aim of identifying approaches which can enable ordinary users who have no significant modeling and programming skills to compose services. A tool that followed control flow based approach of service composition was extended to incorporate contextual information whereby services composed were saved as profiles which are used in a calendar when scheduling activities thus making the service composition context aware.

With the wake of service oriented computing designing tools that realize the benefits of end-user based service composition has been a field of great interest. The tool created aids the end user service composition in creating value added services, this has been demonstrated by the use of several scenarios that show the tool can be used by different users and can be used to create many different value added services on the fly that fit into the users current needs.

Even though the approach presented seems to get the end user composing his/her services quite a few issues should be addressed before the tool is fully functional in that the different tools should be consistent with one another to make the process of the service composition more complete.

APPENDIX A: REFERENCES

[1] Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann, Service-Oriented Computing: State of the Art and Research Challenges, IEEE Computer Societ y, 2007.

[2] H. Luthria and F. Rabhi, Service Oriented Computing in Practice – An Agenda for Research into the Factors Influencing the Organizational Adoption of Service Oriented Architectures, Journal of Theoretical and Applied Electronic Commerce Research, VOL 4 / ISSUE 1 / APRIL 2009 / 39-56

[3] M.P. Papazoglou, D. Georgakopoulos, Service-Oriented Computing, Communications of the ACM - Service-oriented computing Volume 46 Issue 10, October 2003

[4] SCA Service Component Architecture Assembly Model Specification Version 1.00, http://www.osoa.org/display/Main/Service+Component+Architecture+Specifications,Mar ch 15 2007

[5] Jian Yang and Mike. P. Papazoglou, Web Component: A Substrate for Web Service Reuse and Composition, Proceeding CAiSE '02 Proceedings of the 14th International Conference on Advanced Information Systems Engineering, 2002

[6] Dipanjan Chakraborty, Anupam Joshi, Tim Finin, Yelena Yesha, Service Composition for Mobile Environments, Mobile Networks and Applications Volume 10 Issue 4, August 2005

[7] User Guide for Sony Ericsson T28, http://www.sonyericsson.com/...G_R1a_AE.pdf

[8] 3rd Generation Partnership Project (3GPP) TS 23.240: "3GPP Generic User Profile-architecture; Stage 2", version 6.2.0, 2003-12.

[9] Usman Wajid,Abdallah Namoune, Nikolay Mehandjiev, A comparison of three service composition approaches for end users, AVI '10 Proceedings of the International Conference on Advanced Visual Interfaces, 2010

[10] Williams, L.G. and C.U. Smith, "PASA: A Method for the Performance Assessment of Software Architecture,"*Proc. of the 3rd Workshop on Software Performance*. 2002.Rome, Italy.

[11] Jim Amsden, Modeling SOA, http://www.ibm.com/developerworks/rational/library/07/1002_amsden/ ,02 Oct 2007

B1: Doctor's Scenario Exported XML

<Composition type="1" firstCid="3">
<Component cid="9" name="Disconnect" type="2" nextCid="-1" action="1" />
<Component cid="10" name="Connect" type="2" action="4" nextCid="-1" />
<Component cid="3" name="Agenda" type="3" nextCidIfBusy="4" nextCidIfAvailable="5" />
<Component cid="5" name="Connect" type="2" action="4" nextCid="-1" />
<Component cid="7" name="Connect" type="2" action="4" nextCid="-1" />
<Component cid="1" name="Notification" type="4" nextCid="2" tickerText="This is the notification tikker text" contentTitle="Notification title" contentText="This is the notification text" />
<Component cid="2" name="Voicemail" />
<Component cid="0" name="Choice" />
<Component cid="4" name="VIP list" type="1" nextCidIfMatch="0" nextCidIfNoMatch="6" matchType="1" />
<Component cid="6" name="Notification" type="4" nextCid="11" tickerText="This is the notification tikker text" contentTitle="Notification title" contentText="This is the notification text" /> <Component cid="11" name="Transfer" />
</Composition>

B2: Student Scenario Exported XML

<Composition type="1" firstCid="1">
<Component cid="2" name="Disconnect" type="2" nextCid="-1" action="1" />
<Component cid="3" name="Connect" type="2" action="4" nextCid="-1" />
<Component cid="4" name="Settings" />
<Component cid="1" name="Disconnect" type="2" nextCid="5" action="1" />
<Component cid="5" name="SendSMS" type="6" nextCid="-1" textMessage="Compose your own message" toggleButtonIsChecked="1" />
</Composition>

B3: Travelling Businessman Scenario Exported XML

<Composition type="1" firstCid="6">
<Component cid="2" name="Disconnect" type="2" nextCid="-1" action="1" />
<Component cid="3" name="Connect" type="2" action="4" nextCid="-1" />
<Component cid="8" name="Settings" />
<Component cid="4" name="Agenda" type="3" nextCidIfBusy="5" nextCidIfAvailable="7" />
<Component cid="6" name="VIP list" type="1" nextCidIfMatch="4" nextCidIfNoMatch="9" matchType="1" />
<Component cid="9" name="Transfer" />
<Component cid="5" name="Voicemail" />
<Component cid="7" name="Connect" type="2" action="4" nextCid="-1" />
</Composition>