NTNU
Norwegian University of
Science and Technology

Interruption Management in ubiquitous collaborative environments

developing suitable interaction mechanisms for ASTRA

Tara Nath Subedi

Master in Information Systems
Submission date: June 2009
Supervisor: Monica Divitini, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

ASTRA is a project that aims at studying awareness systems that help people to feel in touch with family and friends even when they are away from their computers and the existing widespread tools of today (IM, skype,
twitter,...) are not enough. In this context, this thesis aims at addressing the problem of Interruption management. The focus will be on the design and development of different interaction mechanisms to support the definition of interruption rules as needed in ASTRA. This will require the extension of the existing Interruption Manager as well as the development of suitable user interfaces, also considering physical interfaces when appropriate. A limited evaluation of the proposed solutions will be evaluated with users.

Assignment given: 15. January 2009
Supervisor: Monica Divitini, IDI

Interruption Management in ubiquitous collaborative environments

Tara Nath Subedi

Master Thesis
June 15, 2009

Project Supervisor
Monica Divitini

Norwegian University of Science and Technology,
Department of Computer and Information Science

Abstract

ASTRA is a project that aims at studying awareness systems that help people to feel in touch with family and friends even when they are away from their computers and the existing widespread tools of today (IM, skype, twitter etc.) are not enough. In this context, this thesis aims at addressing the problem of Interruption Management. The focus will be on the design and development of different interaction mechanisms to support the definition of interruption rules as needed in ASTRA. This will require the extension of the existing Interruption Manager as well as the development of suitable user interfaces, also considering physical interfaces when appropriate. A limited evaluation of the proposed solutions will be evaluated with users.

Preface

This report is the result of the master thesis at the Norwegian University of Science and Technology (NTNU). This is for the last semester in the Master of Information Systems degree at the Department of Computer and Information Science (IDI).

The study is part of ASTRA project. ASTRA project is funded by the European Community under the "Information Society Technologies" programme.

I would like to thank my university supervisor, professor Monica Divitini for helping me to find an interesting topic and providing good advice during the thesis.

Trondheim, June 15, 2009

Tara Nath Subedi

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This introduction sets the stage for the report, describing the motivation followed by the work context, research questions, research method and report structure.

1.1 Motivation

ASTRA [1] is a project that aims at studying awareness systems that help people to feel in touch with family and friends even when they are away from their computers and the existing widespread tools of today (IM, skype,twitter etc.) are not enough. In this context, this thesis aims at addressing the problem of Interruption Management. Interruption Management is important since increased numbers of mobile computing devices and other social context aware software and devices are contributing information overload and interruption irritability. Interruption management model has already been incorporated inside ASTRA system on a previous project entitled "Interruption Management in ASTRA" [2]. It consisted of interruption manager with rule engine support for different rules. As end user tool, user is presented with a web interface where they can define rules. This research aims at extending interruption management to have one more interface with tangible interface.

The aim of the area of "tangible interfaces" [3] is to move beyond the current dominant model of direct manipulation in Graphical User Interfaces, where computers typically use a rectangular display, windows, a mouse and a keyboard. The idea is to remove the limited communication channels and explore new channels of interaction, such as gestures or touch. Following this idea new types of interfaces have been built, and in particular interfaces with synthetic characters. Objects in the real world can be given extended capabilities that allow users to merge the real world with the virtual world where synthetic characters exist.

The research focus will be on the design and development of different interaction mechanisms to support the definition of interruption rules as needed in ASTRA. I find this field of study interesting because it is applicable to most domains and is challenging problem.

[1] http://www.astra-project.net/

1.2 Context

The context of this report is a master thesis as a part of a Master degree in computer science at the Department of Computer and Information Science at NTNU in Trondheim. Research focuses on extension of the existing Interruption Manager in ASTRA as well as the development of suitable user interfaces, also considering physical interfaces when appropriate. ASTRA develops a framework for supporting the design of Pervasive Awareness Systems intended to support social relationships. A limited evaluation of the proposed solutions will be evaluated with users.

1.3 Research Questions

1.3.1 Main Research Questions

- How to solve the problem of interruption management in pervasive awareness systems ?

1.3.2 Sub Research Questions

- What are the different approaches in general for interruption management ?
- What are the relevant approaches for pervasive awareness system ?
- What support need to be provided by platform for the approach ?
- What support need to be provided by End User Tools for the approach ?
- Design and implementation of tangible interface for interruption management in platform and End User Tools.

1.4 Research Method

State-of-the-art for tangible interface and for ASTRA system will be carried out with literature review and analysis of the existing system. Current interruption management module inside ASTRA system don't have support for community handling. So this also need to be extended inside ASTRA system. With this background information, I need to do COTS (Component Off-the-self) analysis for the tool / technology that I am going to use. The tangible system will be designed and implemented after the COTS analysis. The overall design and implementation need to be compatible with the current ASTRA principles. A limited evaluation of the proposed solutions will be evaluated with users.

1.5 Contributions

This report will present the following set of contributions made by the work.

- state-of-the-art analysis for interruption management and rule related tangible interface.
- Tangible Interface for the system.
- Evaluation results from the comparison of web interface and tangible interface

1.6 Report Outline

Chapter 2 - State of the Art

This chapter motivates the thesis with the state-of-the-art for interruption management and tangible interface.

Chapter 3 - Interruption Management in ASTRA : state of the art

This chapter describes ASTRA project related aspects to the interruption management.

Chapter 4 - Community Extension

This chapter presents the design and implementation for extending community support in current ASTRA system.

Chapter 5 - Design of tangible interface

This chapter describes design for the tangible interface supporting interruption rules in the context of my proof-of-concept implementation.

Chapter 6 - Implementation of Tangible Interface

This chapter presents implementation details for the design depicted in Chapter 5.

Chapter 7 - Evaluation

This chapter evaluates the tangible interface and compares with web interface. This will have results and discussion from the user evaluation.

Chapter 8 - Conclusions

This chapter concludes by summarizing important aspects done and points some future extensions.

Chapter 2

State of the Art

Interruption Management is not new topic, there are lots of work going on. Research done in other projects related to interruption management will guide to design it in ASTRA system. Different interaction mechanisms like physical interfaces is the focus of this project to support the definition of interruption rules as needed in ASTRA. It is necessary to have literature review for framing the problem domain and to present possible approaches to the solutions.

2.1 Interruption Management

Increased number of mobile computing devices and lots of other social context aware softwares are contributing to feelings of information overload and to "interruption irritability" [1]. New sensor-enabled mobile devices will put opportunities for innovative applications that proactively deliver information to people when and where they need it. Mobile computing devices will increasingly deliver phone calls, reminders, email, task lists, instant messages, news, and other time and/or place-based informations. Delivering the proactive interruptions the user wants with the help of context-aware applications such as location and activity based friend-finders for example, therefore will inevitably increase the number of unwanted interruptions the user must endure.

Each time a device proactively provides information, it is competing for the user's attention and possibly interrupting the ongoing tasks. Although computing power will continue to improve, permitting more powerful mobile devices, human attention is a limited resource [4].

Determining a good time to interrupt requires a complex assessment of context and message content. For example, consider an office worker sitting at a desk discussing a report with a supervisor. If the phone rings and it is a coworker with updated information for the report, the office worker is likely to be receptive to the phone call. However, if the phone call is from a friend to discuss plans for the weekend, then the office worker is likely to be less receptive. On the other hand, the office worker might be receptive to the phone call from the friend if the phone displays the message visually instead of using the ring to signal the interruption. The visual notification is less likely to disrupt the flow of the current conversation, perhaps lowering the perceived burden of the interruption for both people in the room. [1]

Factor	Description of the Factor
Activity of the user	The activity the user was engaged in during the interruption
Utility of message	The importance of the message to the user
Emotional state of the user	The mindset of the user, the time of disruption, and the relationship the user has with the interrupting interface or device.
Modality of interruption	The medium of delivery, or choice of interface
Frequency of interruption	The rate at which interruptions are occurring
Task efficiency rate	The time it takes to comprehend the interruption task and the expected length of the task
Authority level	The perceived control a user has over the interface or device
Previous and future activities	The tasks the user was previously involved in and might engage in during the future
Social engagement of the user	The user's role in the current activity
Social expectation of group behaviour	Activities and expected reaction to interruption of nearby people
History and likelihood of response	The type of pattern the user follows when an interruption occurs.

Table 2.1: Eleven factors that influence a person's interruptibility at a given moment. [1]

According to paper [1], there are at least 11 factors that impact the perceived burden of an interruption, as listed in Table 2.1. So an exhaustive model of interruptility would include a weighted sum of these factors. Some of the metrics that have been used to evaluate interruptibility are listed in Table 2.2. Applications that can infer interruptibility from sensors can defer non-time-critical prompts to the times that are likely to be least disruptive. One suggested strategy to minimize the perceived burden of an interruption is to present reminders immediately following the completion of some actions [5]. The assumption is that at activity transitions, memory load may be low, because a person may be between evaluation of the last activity and formation of a new goal. An alternative approach would be to use activity transitions to compute a priority for incoming messages, perhaps in combination with other information about the message content and user context. The priority score could be used by an application to perform negotiation-based coordination.

The strategy of using activity transitions could be adopted as a trigger for non-time-critical interruption to potentially reduce feelings of information overload.

The picture 2.1 depicts overload and irritating to the user by the increased number of devices and demands interruption management to tackle.

Interruption management model has already been incorporated inside ASTRA system on a previous project entitled "Interruption Management in ASTRA" [2]. ASTRA is a project that aims at studying awareness systems that help people to feel in touch with family and friends even when they are away from their computers and the existing widespread tools of today (IM,

Definition of Interruptibility	Measure of Interruptibility
Waiting for an opportune moment to avoid disruption on the primary task	The amount of time necessary to complete the interruption task and the original task while maintaining accuracy
Cost of interruption based on the user's model of attention, such as high-focus solo activity	Willingness to pay to avoid the disruption
Perceived burden of interruption	Self-reports of interruptibility (scale 1-5)
Cognitive limitations to work during an interruption	Completion time, performance accuracy, and number of task switches
Value of the notification	Self-annotation of the value of a notification
Unwanted distraction to primary task	Accuracy
Ability to facilitate decision making	Performance on decisions
Cognitive activity disruption	Accuracy and reaction time

Table 2.2: Definitions of interruptibility and evaluation metrics used in some recent work [1]

Figure 2.1: Overload and irritating by increased number of devices

skype, twitter etc.) are not enough. In this context, this thesis aims at addressing the problem of Interruption Management. The focus will be on the design and development of different interaction mechanisms to support the definition of interruption rules as needed in ASTRA. This will require the extension of the existing Interruption Manager as well as the development of suitable user interfaces, also considering physical interfaces when appropriate.

2.2 Tangible interaction

For many years the development of user interfaces to design applications has focussed on the use of screen real estate and more effective use of menus, palettes, and increasing functionality. Recent developments in HCI technologies have lead to an exploration of a variety of devices to enhance the design experience with tangible and multiple visual interfaces. In recent times, interfaces have been developed which use physical objects as representation and controls for digital information for the purpose of making our interaction with computing devices more natural. These interfaces have been aptly named tangible user interfaces by Ullmer and Ishii in their paper "Tangible Bits" [3]. The main characteristic of tangible user interfaces is its intuitive interaction with the digital world using familiar physical objects, surfaces and spaces. The focus is on direct physical interaction with objects as an element of tangible user interfaces as opposed to input done only through a mouse and keyboard. A central characteristic of tangible interfaces is the seamless integration of representation and control, with physical objects being both representation of information and as physical controls for directly manipulating their underlying associations. Input and Output devices fall together. Tangible interfaces rely on a balance between physical and digital representations. There are 4 characteristics concerning representation and control:

- Physical representations are computationally coupled to underlying digital information.
- Physical representations embody mechanisms for interactive control.
- Physical representations are perceptually coupled to actively mediated digital representations.
- Physical state of tangible embodies key aspects of the digital state of a system.

A Graspable user interface design provides users concurrent access to multiple, specialized input devices which can serve as dedicated physical interface widgets, affording physical manipulation and spatial arrangements. Hence input control can be "space-multiplexed" [6]. That is, different devices can be attached to different functions, each independently (but possibly simultaneously) accessible. This, then affords the capability to take advantage of the shape, size and position of the physical controller to increase functionality and decrease complexity. It also means that the potential persistence of attachment of a device to a function can be increased. By using physical objects, we not only allow users to employ a larger expressive range of gestures and grasping behaviors but also to leverage off of a user's innate spatial reasoning skills and everyday knowledge of object manipulations. These physical artifacts are essentially "graspable

functions" – input devices which can be tightly coupled or "attached" to virtual objects for manipulation, or for expressing actions. These artifacts need to have spatially-aware computational devices.

The benefits of tangible interfaces have been well described elsewhere [3]. Specifically, tangible interactors allow for collaborative and two-handed interaction, require less dexterity than traditional input, and better preserve spatial relationships between virtual objects and their real-world counterparts. This research focuses on tangible interaction interface for end-user programming of interruption management.

2.3 Tangible Interface Examples

Credits should be given to John Frazer and Robert Aish, who were the first to (in parallel) develop the idea and first implementations of a tangible construction kit for creating virtual models at the end of the 70s and beginning of the 80s. This work, coming from the areas of architecture and CAD, has only recently been rediscovered by people interested in tangible interfaces.

George W. Fitzmaurice together with Hiroshii Ishii and William Buxton 1995 introduced Bricks, tangible new input devices that allow direct control of electronic or virtual objects through physical handles for control, and coined the term "Graspable User Interface".

Authors [7] describe the design and implementation of a system, SiteView, for creating and viewing automation control rules for active environments those with sensing and actuation capabilities. Site View has an intuitive tangible interaction method for creating control rules and enhances user understanding of the system by appropriately exposing internal state. SiteView also supports users' visualization of the active environment outside the programming environment through a photographic display keyed to control rule conditions. Site-View programs consist of rules with a simple conjunctive predicate and one or more consequent actions. Users create rules by manipulating tangible interactors representing sensed conditions and automated actions within a world-in-miniature (WIM) model. World-in-miniature is a small-scale representation of the active environment and condition composer is an area that senses and structures the user's specification of rule conditions. Environment display shows what the environment will look like when a rule is activated and rules display interactively shows the rule as it is created and shows other rules applicable for the given set of conditions. To specify rule conditions, they created interactors representing weather, time-of-day, and day-of-week condition categories. Weather interactors depicted sunny, partly cloudy, overcast, and rainy states; time -of-day interactors depicted each of the twenty-four hours of the day, as well as morning, midday, afternoon, evening, and night; and day-of-week interactors depicted each of the seven days of the week as well as weekdays and weekends. These interactors have been implemented as foam-core cards and embedded with RFID transponders. The cards provide tactile feedback about their condition category through a unique shape on their upper edge, which also serves as a physical constraint within the condition composer. To specify the rule actions, they built tightly-coupled interactors representing both binary light state and a thermostat. They built four light state interactors, two for each controllable lamp in the active environment. A physical slider Phidget [8] was used for temperature range for thermostat.

An affective tangible interface (SenToy) that allows a player to influence the emotions of his character (avatar) in a 3-D computer game have been made [9]. SenToy works as an interface to the role playing game where players must exhibit a particular set of emotions and perform a set of actions as a way to evolve in the game [10]. Emotions play a central role in the game, since it is through mastering the emotion expression of the controlled characters in the game that players can advance in the game and win battles against their opponents. The aim of SenToy is to "pull the user into the game" through the use of a physical, touchable, affective interface. With sensors in its limbs, sensitive to movement and acceleration, SenToy captures certain pattern of movements that are associated with particular emotional expressions.

Chapter 3

Interruption Management in ASTRA : state of the art

ASTRA is a project that aims at studying awareness systems that help people to feel in touch with family and friends even when they are away from their computers and the existing widespread tools of today (IM, skype, twitter etc.) are not enough. This chapter details on current state of ASTRA which are taken from ASTRA technical deliverable, whenever otherwise not mentioned. Interruption management model has already been incorporated inside ASTRA system on a previous project entitled "Interruption Management in ASTRA" [2], whose current state will be detailed in last section.

3.1 ASTRA Overview

ASTRA develops a framework for supporting the design of Pervasive Awareness Systems intended to support social relationships. The framework consists of

- A Theory to guide the design and the evaluation of pervasive awareness systems
- supporting technologies:

  service oriented architecture

  end - user tools that support users to appropriate the Awareness applications

Pervasive Awareness Systems(PAS) is awareness systems that use mobile and ambient devices for collecting or presenting awareness information and operate within an Ambient Intelligence (AmI) Environment.

Here is one typical scenario of ASTRA use. This is from ASTRA Movie (http://www.astra-project.net/) of connected life of father (Vic), son (Pavan) and daugher(Eleftheria). When vic is at home and he wants to talk with pavan. But since pavan is cooking, he don't want to talk at that moment. Pavan has published his awareness state "not-available" through nimbus application whenever he is cooking. Vic has set his focus application to Pavan's "not-available"

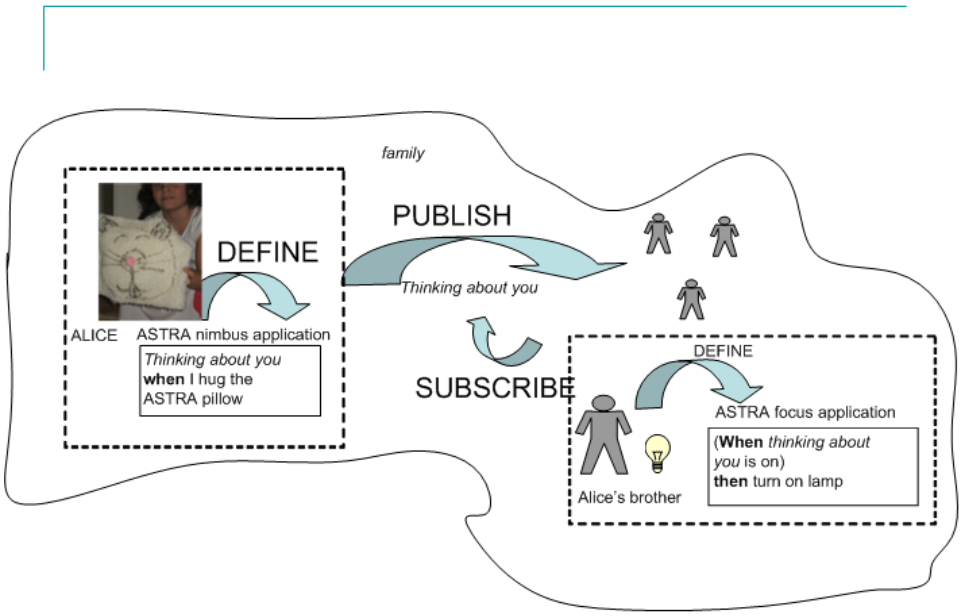


Figure 3.1: ASTRA Awareness System

awareness state and this is represented by red colour in picture frame. When Pavan is not cooking and available, Pavan's another application notify this awareness state to focused application of father and the picture frame will be green notifying to father Pavan is available for talk. This model provides a basis for describing the design space of awareness systems, in terms of content exchanged, elementary user behaviors pertaining to sharing information about themselves or perceiving information about others. Figure 3.1 depicts the tactics of awareness in ASTRA.

3.2 Platform

An open source OSGI [11] implementation called knopflerfish [12] is used as deployment framework for ASTRA. It is very elegant and easy way of deploying services. ASTRA is designed from Service oriented architecture (SOA) environment. In an SOA environment, resources on a network are made available as independent services that can be accessed without knowledge of their underlying platform implementation. SOA can also be regarded as a style of Information Systems architecture that enables the creation of applications that are build by combining loosely coupled and inter operable services. The current overall SOA architecture design of ASTRA is shown in figure 3.2. This includes the platform for awareness services, the ontology, ontology management, context management, service and other necessary modules. There are 3 main types of deployment of the ASTRA SOA and architecture, which are described in following sections.

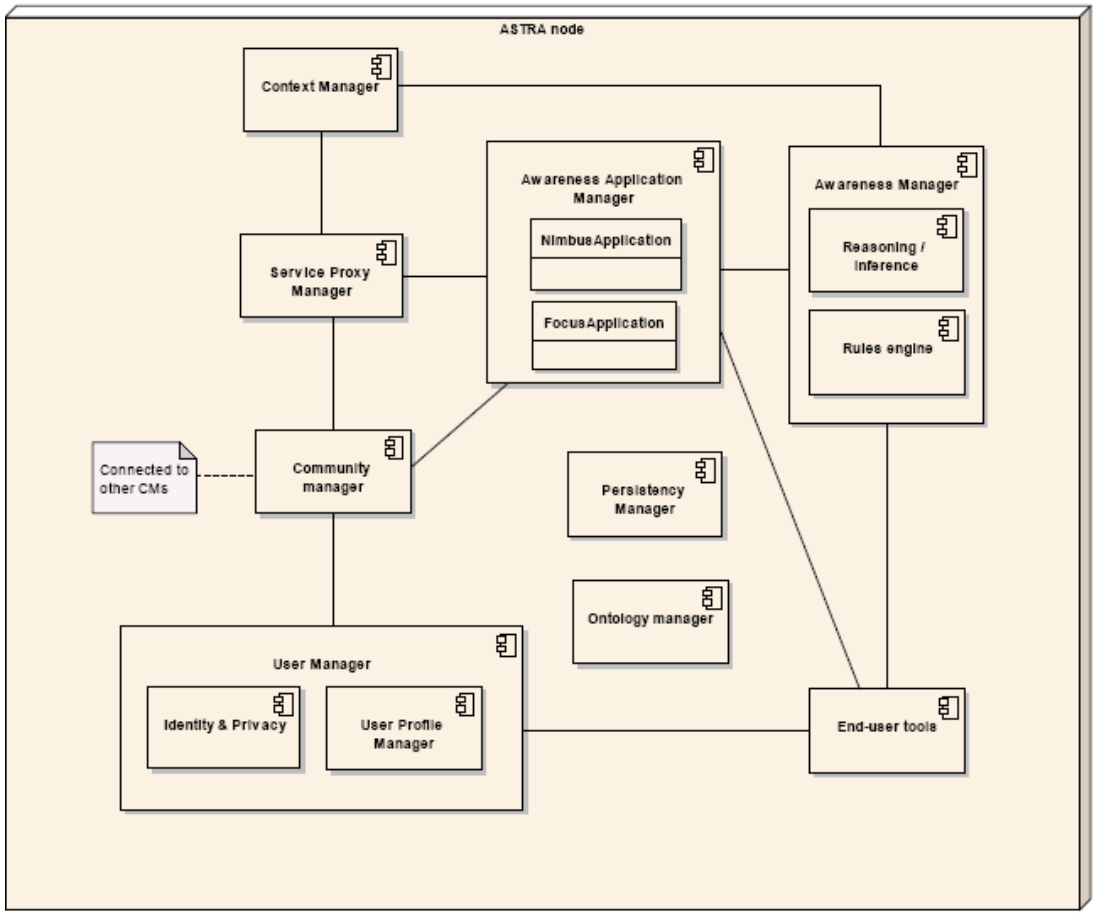


Figure 3.2: Overall SOA Component Architecture of ASTRA

3.2.1 BackEnd

Identities are dependent on an identity provider in the "back-end" of the system, which will manage and authenticate identities across different nodes. For a user to trust the identity of another user, a trusted third party is required to provide identity management. The back-end also takes care of synchronizing the different ASTRA-nodes and also provides persistency to the user manager and the community manager.

Nimbus application from Node updates its application states to this BackEnd and focus application of another Node polls this BackEnd to receive the changed state information.

3.2.2 Node

This is the main deployment unit. The components of the SOA are integral parts of an autonomous deployment on a node - an ASTRA node. This node can for instance be the local deployment in one's home, at the office, at a university or in the car. The nodes work together on peer-to-peer approach and their common channel of communication is through Community Manager - different users collaborate with each other (i.e. sharing information) through so-called collaboration instances of the Community Manager; this is where services and attributes are exposed to other users in the context of a shared community. Considered a part of this ASTRA node are all the awareness devices installed at one location.

It includes the platform for awareness services,context management,service discovery and other necessary modules.

Context Manager

Context Manager is a unified repository of the collected local discovered services. Context Manager is (a) responsible to gather any change in the local environment and inform the other components about it, and (b) to handle incoming requests for activating an actuating device (through a change on a local focus for example). It is responsible then to find the correct Service proxy manager to give the message to forward to the appropriate service in order to actuate it.

Awareness Manager

The awareness manager twofold operation intermediates between the low level user-system interactions and the high level concepts connected with. It is connected to the rule engine kernel which processes (maps) a) actions-situations to nimbus alike concepts and b)focus alike concepts towards particular actuation (reverse of a).

Awareness Application Manager

The awareness application manager (AAM) is responsible for storing and managing the local awareness applications. An awareness application is a mapping of an awareness state to a service that can be made available to a community, in other words it is a service representation of a specific awareness state. Every local awareness application is stored here and the AAM is in turn

associated with a community of the Community Manager, which in turn is the communication point with other nodes. Whenever an awareness state is changed, the Awareness Manager informs the AAM, which in turn checks to see if any communities need to be informed. The AAM at the same time monitors communities for such state changes that are of interest. The AAM provides two types of awareness state applications: nimbus applications and focus applications. These applications are mappings of the focus and nimbus concepts, which will be detailed in next section. Basically whenever a user wants to share her awareness state, she must create a nimbus application and publish it. Every other member can choose to focus on that awareness state by creating focus application.

focus and nimbus

ASTRA supports not only at individual users, but also at communities or social networks. Awareness might be mediated within different types of communities, for example co-located and remote families, communities of friends, colleagues, neighborhoods. In general, each individual can be a member of multiple communities. Within each community, diverse rules and conventions might influence how issues such as awareness service provisioning, sharing and tailoring should be facilitated by the system [13]. The theoretical background for ASTRA awareness applications is the focus-nimbus model, originally described by Benford et al [14]. The authors use room metaphors as the basis for a spatial model to support communication between participants in virtual rooms. The basic idea is that people can not only visit different virtual rooms but they can move around in these different rooms, and the (modeled) spatial characteristics of the rooms mediate the communication between different persons in the room. Two concepts are introduced; the focus represents a space in the room where a person targets his attention. People are more aware of objects in the focus than those outside. The nimbus is the counterpart, representing where the person locates himself in the room. Objects are more aware of a person if the object is located in the person's nimbus than when is located outside [15]. Awareness is defined through the interaction of focus and nimbus. The model has been generalized by Rodden [16]. He extends the notions of focus and nimbus towards application areas without an explicit notion of spatial relations. Metaxas and Markopoulos have later presented a formal model which concentrates on the communication aspects of the focus-nimbus model [17]. Their model addresses issues of privacy by allowing for plausible deni-ability and deception. For e.g., Alice would make her wish for walk available by placing it on her nimbus. She can control to which community of users she publishes this message. Bob, on the other hand, would have a focus on this Alice's message [13].

3.2.3 EUT

Users access the SOA via the tools available on a user device, be it a computer, PDA or mobile phone. These devices offer tools and applications to the end-user. No SOA parts are actually found here (except for a service discovery client), but the user access the SOA functionalities through a set of tools on this device.

From End User Tools one can register as user, create community, join to community, publish application, subscribe to application, have rules on publish and how to get notification of awareness. This is just interface which allows user to communicate with SOA.

3.3 Interruption Management

This section details Interruption management model that has already been incorporated inside ASTRA system on a previous project [2]. There are many interests and study on interruption management. From literature review, the summarized model is as shown in figure 3.3. As ASTRA intends to support social relationship, it will inherit the interruption problems and opportunities from real life, which it has to tackle. Since the interruption properties by ASTRA are subclass of interruption properties of big real life model, it was used to have initial model for interruption management in ASTRA.

The interruption manager is modeled as in figure 3.4, which follows the findings from state of art and hypothetical requirement analysis aimed to ASTRA project. The interruption manager intercepts the incoming messages and is aware of interruptee's context and also of interrupter message. The interrupt manager decides what to do with the interruption for e.g. accept or deny or differ.

User is able to control interruption based on interrupter like interrupting person, interrupter's particular application and the time of interruption; and based on interruptee like device context and awareness state. Since only the application for which she is focusing on can interrupt her, so owners of those application only are probable interrupter. The interruption manager decides whether to accept, deny or differ based on the different rules defined by interruptee. This interruption management model is rule based decision support. Inference engine get updated with interrupter information at the time of interruption; and context of interruptee instantly get updated as facts each time it get change. Through end user tools, user sets the interruption rules and that get updated in inference engine as rules. The decision of inference engine, will be accept, deny or differ based on rules and facts. Interruption manager supports to this inference decision and decides what to do next and logging of denied and differed decisions. In case of accept decision, it allows the interruption to happen. In case of deny decision, it blocks the interruption from happening and log, for future display to user. User is presented web interface to set interruption rules by EUT. EUT accesses the platform component's web services to access, set rules. The platform component implements rule inference engine, what to do after decision and lots of methods. Editing rules interface looks like as in figure 3.5. One single rule looks like as in figure 3.6. This interruption management module don't have support for having interruption rules based on community of the interrupter. This will be detailed in next chapter. The next chapter extends the interruption management module so that to have community support.

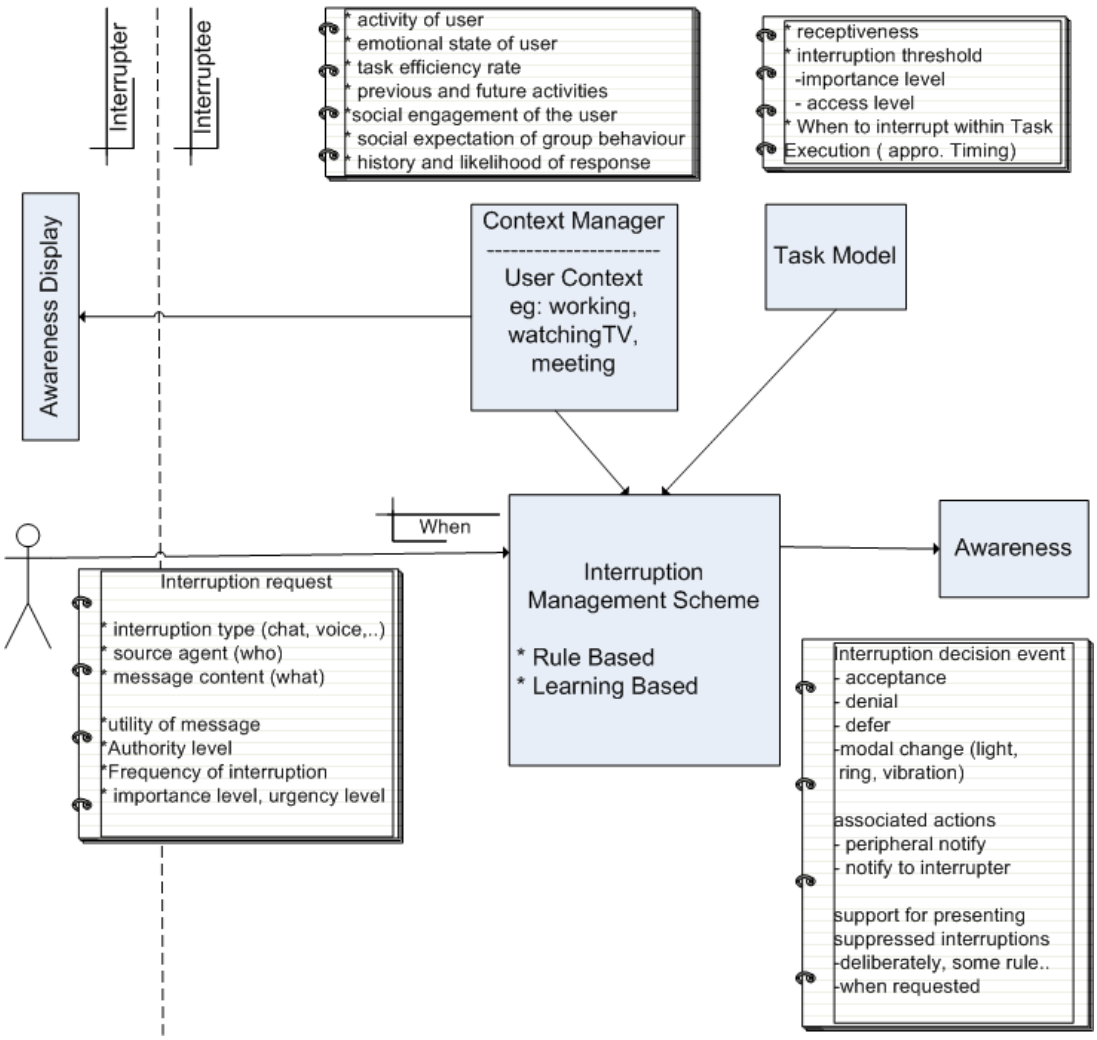


Figure 3.3: Summarization of different elements of Interruption Management

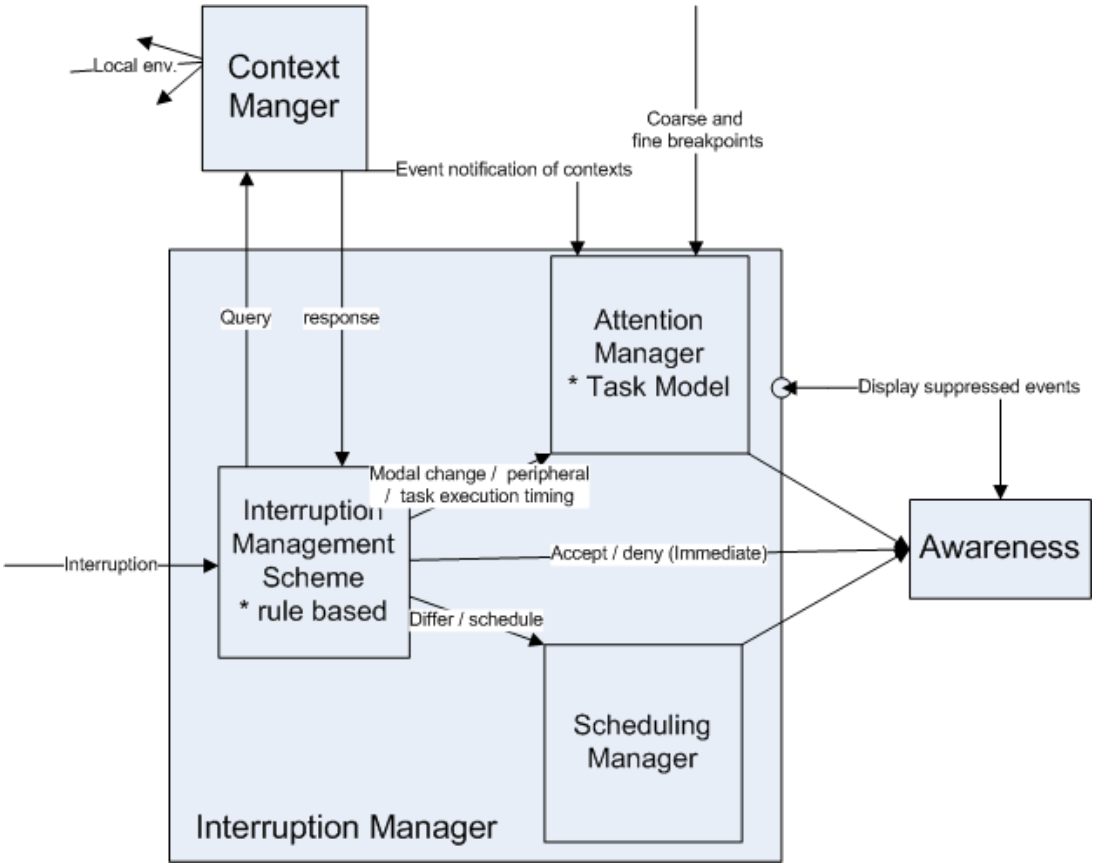


Figure 3.4: Design of interruption manager

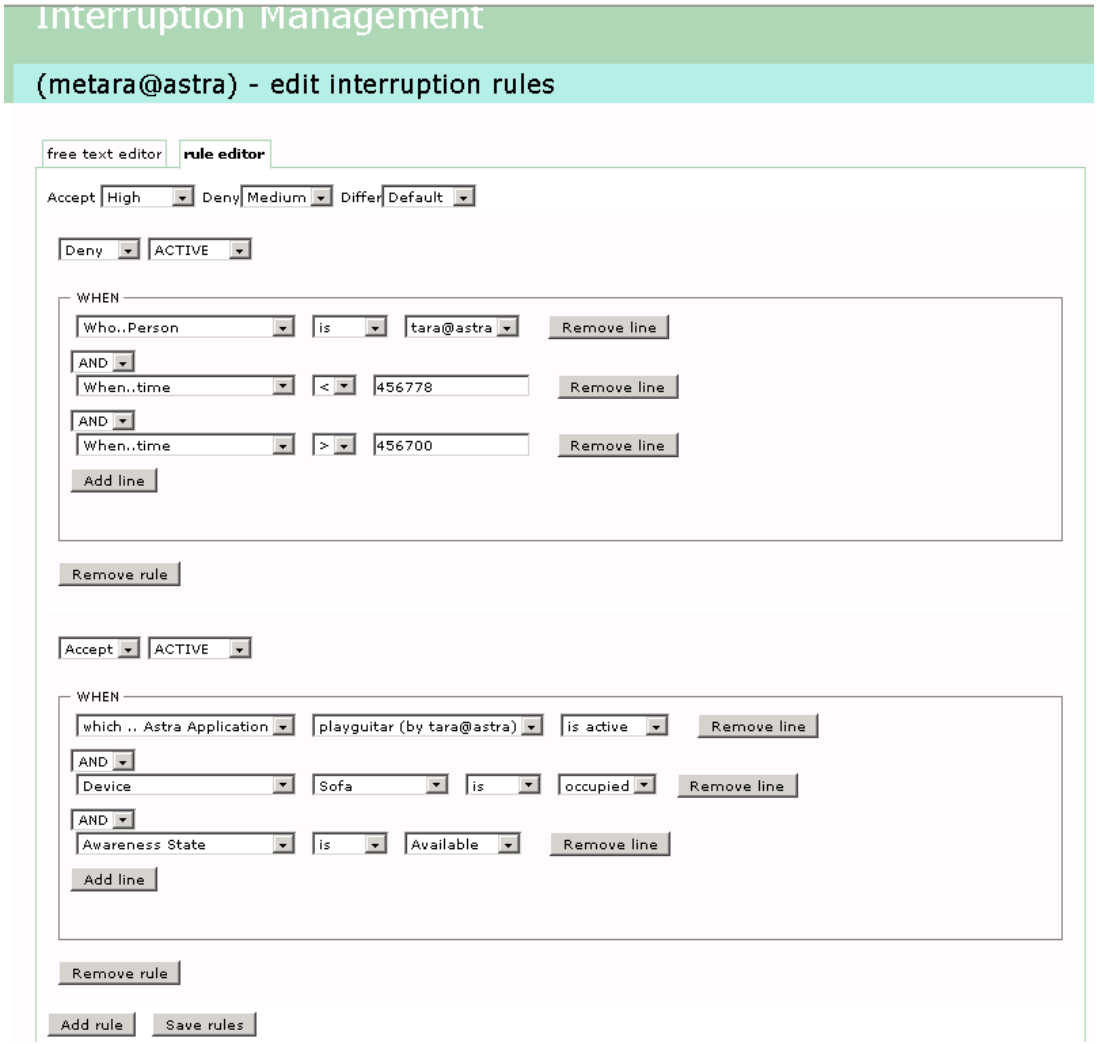


Figure 3.5: Editing rule interface

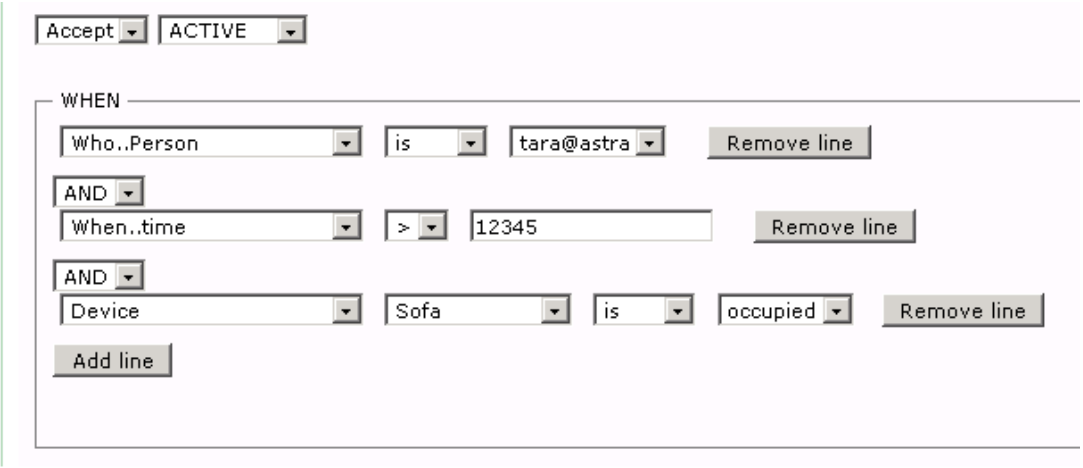


Figure 3.6: Compounded single rule example

Chapter 4

Community Extension

This chapter sets the design and implementation for community support in interruption management.

4.1 Community in Interruption

From previous project [2], Design for the interruption management has been proposed. There are different parameters like who is the interrupter, at what time the interruption is being occurred, which application interrupting, and interruptee's context etc. Moreover the user wants to have control from community perspective. For example, A user may want to avoid colleagues while he is at home. When a person is at meeting, he may want to be interrupted just from a group of people related to the meeting group. So rather than defining the individual person or some time frame, it is equally important to have control with the community. Family,friends, colleagues are some examples of community. This aspect haven't been implemented on the project. So here I have extended this to the community, so that user can decide which community she may want to allow or block in some sort of contexts.

4.2 Community in ASTRA

A user can create community and other users can join and / or leave to the community. The user who have created the community itself can leave and join the community. A user can have different applications that can be published to different community and other users joining to the community can subscribe and focus to the application. Since a user can join in many community and can publish his application to many community and application doesn't bind directly to the community but bind to user. A user from same community can subscribe and focus to the application of other user who have published the application to that community. The user focusing to others' application might be interested to block or allow to particular community. The application information itself just gives who belongs that application and which application from the owner. The community here actually means to the belonging community of the application owner. That simply means community is group of peoples and one people can have belongings

to different group. If the owner is by any way belongs to particular community, then the focusing user wants to either block or allow to all applications from the owner.

This simply puts some points for the design of the interruption management from community extension point of view. The interruptee should be able to define the community that he wants to allow or block. The end user tool should support interface for setting rule to allow or block the community of which the user is member or owner. Since the user can't subscribe and hence focus and interrupted from the community of which he is not member or owner, the membership community is only important to list. Whenever focus application get triggered or in other word, the user get interruption the Node will check different interruption rules and decide either to allow or block to the interruption. EUT will have its rules in Node. And the Node now would require the membership community of the interrupter, so that it can be compared with the interruption rules of community. There should be membership information in rule engine of Node so that it can interpret the rules with sufficient information and decide the result. The membership information itself is part of BackEnd, which need to be queried by the Node. One user join or leave the community in dynamic settings, so the community information should be uptodate on EUT and Node rule engine. The design should be in a way to support scalability, when the user and community expands.

4.3 Scenario

Figure 4.1 depicts three different scenario for interruption rules from community perspective. User sets rule to allow or deny based on community for example family, friends, colleagues.

4.4 Design

High level view of the design implication for the community extension in interruption management is depicted in figure 4.2. The BackEnd has the community related information like member, owner and other details in database in XML document form. Community Manager bundle can access the community related information through the methods and the BackEnd publishes this access service through web service. EUT and Node can use this service to get membership information of user. EUT need to have interface to choose the community to allow or block. The community list to be displayed should be just the membership community of the rule setting user because the focus application couldn't be out of the user's community. In Node, when the focused application get triggered i.e. interruption occurs, the membership communities of the interrupter get updated in rule engine. At this point the rule engine has interruption rules that is related to community and sufficient information about the community of the interrupter, so that it can check its rule to the information available and decide on interruption result. So the implementation details will be divided into EUT, asserting information and rule check.

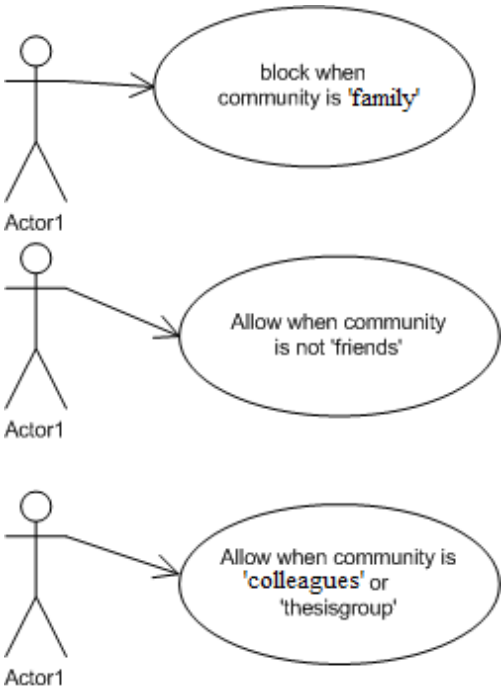


Figure 4.1: Scenario for interruption rules with community

4.4.1 End User Tool

Figure 4.3 presents the interface for the interruption rules for community support. A user can define the incoming message as either equal or not equal to one community and decide to either "block" or "allow". The atomic rule can be made composite rule with anding. Anding multiple atomic rule results complex rule. For example "allow" when the interrupter has membership to one community and not equal to another community. This community list displayed should limit to the membership of the user. Since the user don't get interruption from out of membership community, there is no point of giving choice to select from all community. The membership information is retrieved from the Community Manager web service provided by the BackEnd. This interruption rule is saved in XML document to the Node Database. The rule will be set in Node inference engine, ie CLIPS [18] engine, which is discussed in next subsection.

4.4.2 Asserting membership in rule engine

When there is interruption i.e. focused application get triggered, the Node inference engine will run the interruption rules and check with different parameters of the interrupter like who, which application, when and which community and result allow or deny. Since every information are in inference engine as facts, we also need the membership community of the interrupter before the interruption rule can check its community rule. There could be two design way for this as:

- Complete membership update: Complete membership information could be synced in inference engine as facts in form of (community,member). So that whenever be the inter-

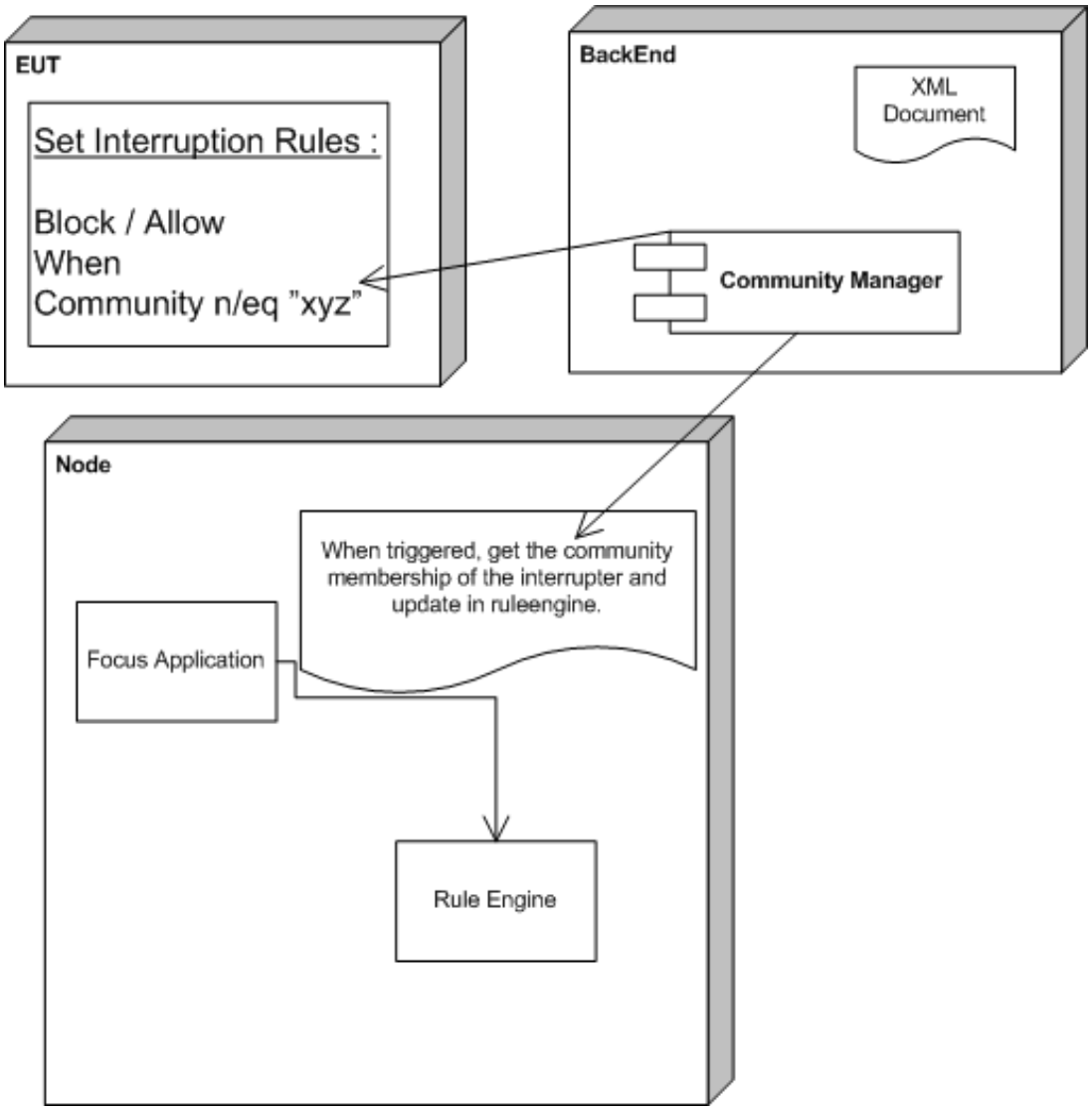


Figure 4.2: High level view of the design

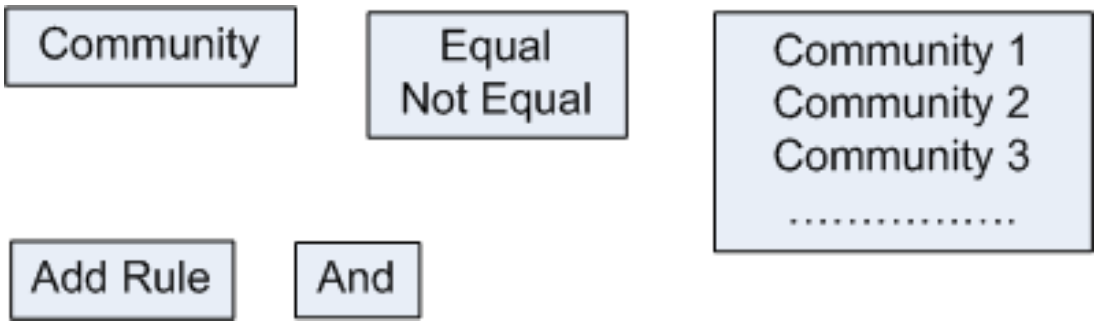


Figure 4.3: End User tool interruption rule interface

ruption, the interruption rules could be checked with the available complete membership information as facts. This is just matter of simply checking if the fact (community, interrupter) is available or not. But this design seems to have quite heavy computation and is not suitable for scalability purpose. If we just think of 100 community with 100 users, then it needs to map every community and every users. But in one node, one user might not be quite related to all this. The rule check need to check to all the facts, which requires high computation. And moreover whenever there join or leave and creation of new community, it need to be updated even in inference engine demanding code addition in many places.

- interrupter's membership only: In this design, the membership information will be updated in inference engine just for the interrupter and this is done while there is interruption. So when the focus application get triggered i.e. interruption occurs, interrupter and interruptee information is already being asserted in inference engine. Now we are also adding or refreshing if already exists the membership information of the interrupter in inference engine. This is quite straight forward and lightweight.

The 2nd approach has been implemented. The membership is asserted as fact in inference engine as: (assert (membership (name tara@astra) (member-of guitar-community)

To get the membership of the interrupter, the Node need to access BackEnd Community Manager through Community Manager Proxy. This assertion need to be done in Awareness Manager bundle, so it need to get service reference of Community Manager API bundle that have been implemented by Community Manager Proxy. The figure 4.4 presents the web services roles example. The figure 4.5 depicts the picture of getting membership information from the BackEnd. The Community Manager from BackEnd provides service ismemberof(user) along with other services. This service returns the membership community associated to the given user. The node has Community Manager Proxy that gets stub from the Backend Community Manager. Awareness Manager Activator can access the service through the Community Manager Proxy after getting the service reference for the Community Manager API. The assertion of the membership facts should be aware of deleting old membership for the same interrupter. This will be described in following section.

In notifyAM_interruption() method, interrupter, interruptee information is being asserted. And now it need to assert community membership information for the interrupter. Inference Engine invokes this method and the service reference need to be passed by Awareness Manager Activator.

4.4.3 Rule check : CLIPS rule

The EUT calls the Node service to set the interruption rule after getting from the user interface. The rule first get saved in database in XML format. while setting in CLIPS format, the rule will be set as: if the rule is: community eq family the clips rule will look like: (exists (membership (name ?who_interrupter) (member-of family)) where the name value will be obtained at run time from the interrupter fact and will looked in facts database if there exists any facts with this membership. if the rule is negative like: community neq home the clips rule will look like:

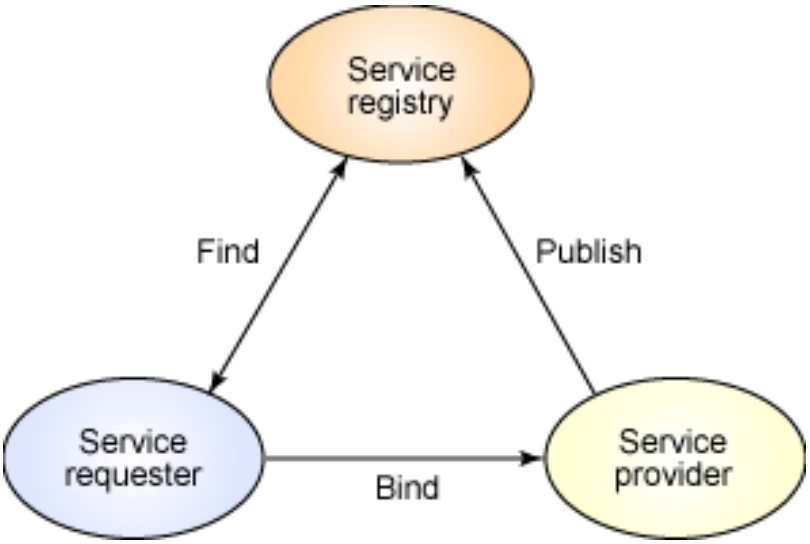


Figure 4.4: Web services roles

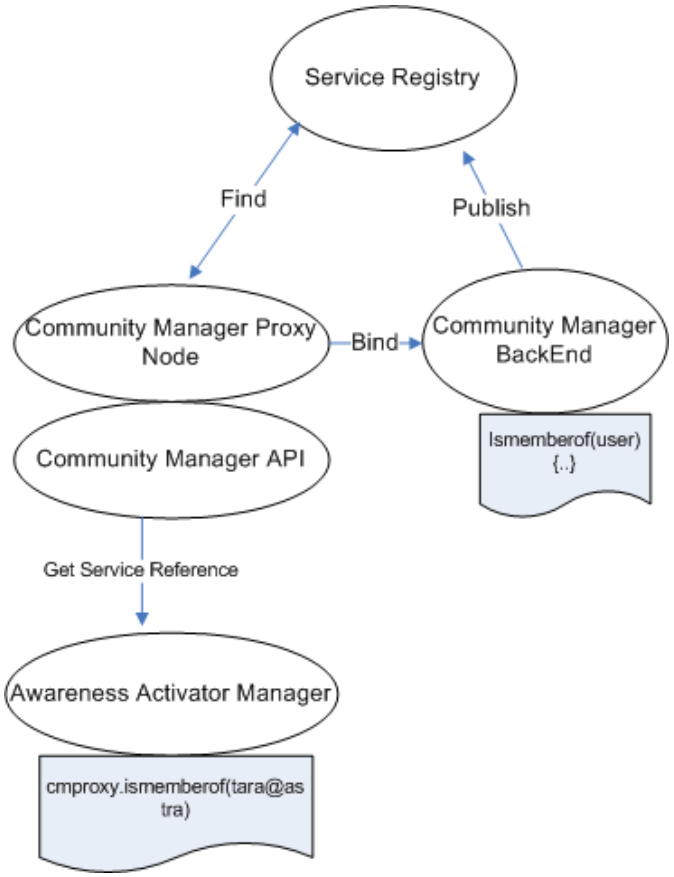


Figure 4.5: Node accessing BackEnd community membership

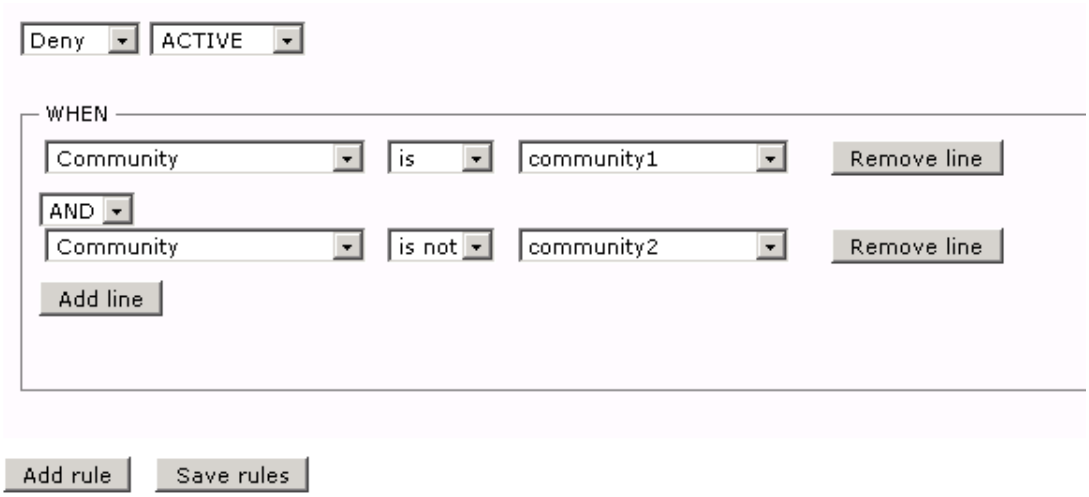


Figure 4.6: Interface showing community part

(!(exists (membership (name ?who_interrupter) (member-of home))) Where if the fact exists then it will be false.

The old membership need to be deleted before adding new facts in rule engine for the same interrupter. Otherwise the membership would be either stale or duplicated. This deletion could be done before asserting membership facts for the interrupter.

4.5 Conclusion

Figure 4.6 displays the interface with community option in interruption rules. The interruption module have been patched so as to support community feature in interruption rule setting and in rule engine.

Chapter 5

Design of tangible interface

This chapter details the concept of the tangible interface and expected outcome and also illustrates different design to achieve the result.

5.1 Motivation

The research focus will be on the design and development of different interaction mechanisms to support the definition of interruption rules as needed in ASTRA. Tangible user interface (TUI) is a user interface in which a person interacts with digital information through the physical environment. Tangible interface will have the interruption rules out of computer desktop, where user will interact to the some physical environment to set the interruption rules.

Research focuses on achieving similar functionality as web interface to interruption rules, but should be simple from user perspective. It focuses more to set rules in easy way rather than managing rules like deleting.

5.2 Introduction

Interruption management model has already been incorporated inside ASTRA system on a previous project entitled "Interruption Management in ASTRA" [2]. It consisted of interruption manager with rule engine support for different rules. As end user tool, user is presented with a web interface where they can define rules. This research aims at extending interruption management to have one more interface with tangible interface. The Overall system in play is depicted in figure 5.1.

5.3 Prototype

Web interface to define the rules look like as shown in figure 5.2.

Since Phidget [19] supports different analog input, digital input and digital output, it could be used to make tangible interface. Slider Phidget, display Phidget etc are easily available in

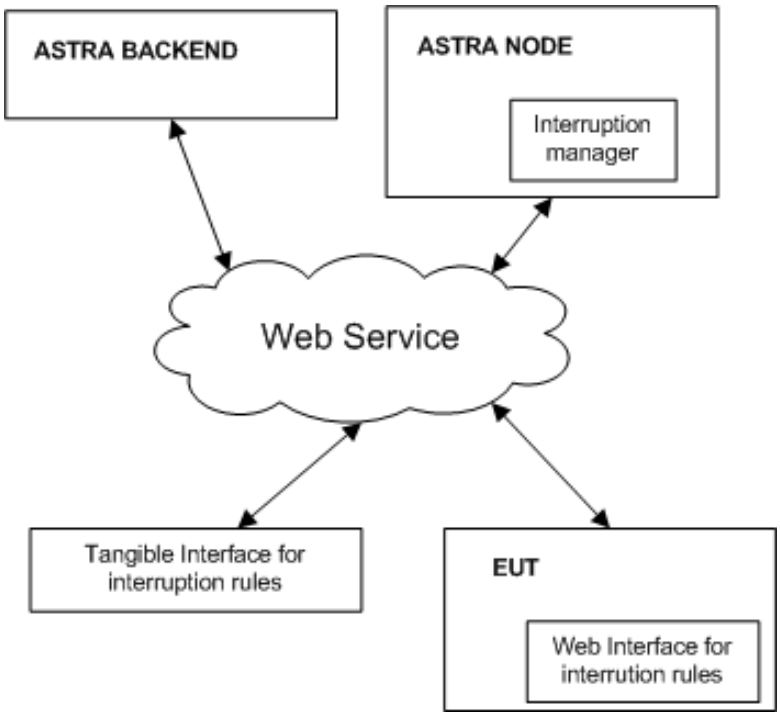


Figure 5.1: High level view

Figure 5.2: web interface to define interruption rules

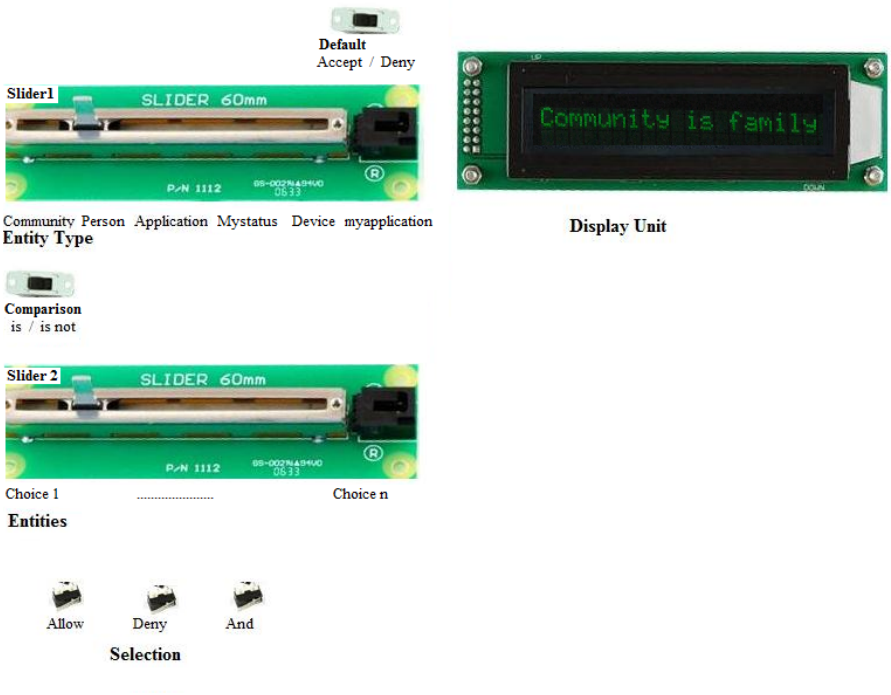


Figure 5.3: Different components combination for tangible interface

market and digital input can also be used to exploit the switch like operation. "Allow", "Deny" and "And" switch buttons results true value on some digital input when pressed. They should be like bell switch, press and release. It should not be true in digital input, unless pressed. "Default" and "Comparison" has one state as default and thus sliding switch could be used for this. So that when changed it would change the state.

The projected tangible interface will use different components as presented in figure 5.3. Sliding Phidget need to be use because the number of choices are not fixed in advance and it is very dynamic list. Moreover different entity type like community, person for eg have different number of choices (entities). If system design expands to add some more entity type in future, for example 'When Time' entity type, it can use slider for that entity type also.

Rules can be composite by anding different atomic rules. Slider1 will select among community, person, mystatus, application, device, myapplication. Slider2 will select particular choice based on slider1 selection. This will be displayed on display unit. Allow or Deny push button will make the rule. Pushing "And" button makes sure another rule added on the current one. Figure 5.4 depicts logical view of this prototype.

5.4 Design Strategy

We are dealing with tangible interface and we may want to have more devices to function the same. For example, we have now small display unit, but later may want to have big screen to display things. We may want to have different sort of interface to select among choices. The

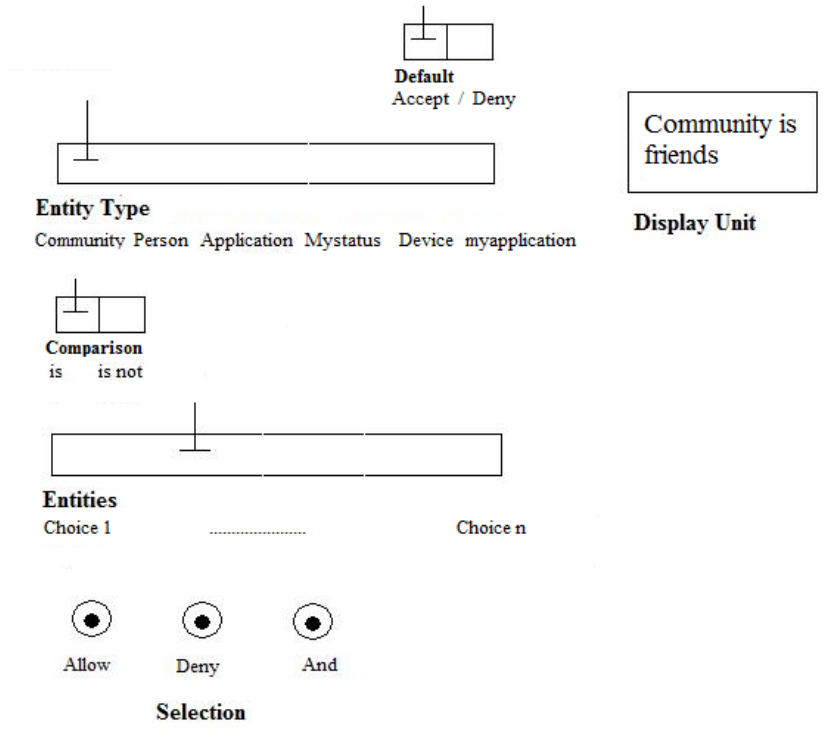


Figure 5.4: Logical view of the prototype

devices could be in different places in room independent to each other. Considering these all, UPnP [20] technology is implemented rather than fixing everything in rigid and monolithic way. In short UPnP suggests,

- Just send data
- Keep implementation private
- Agree on content / format of data

So the each functionalities like slider1, slider2, selection, comparison, default, display could be seen as services. One logical device could interface to one or more physical devices like for e.g. slider1, slider2 and so implement one or more services inside it. The device will notify its capability in network. Control Point will cumulate all the services that it needs to function correctly and ask devices to use some services or devices will notify on the change.

To make use of different types of devices in future, it's necessary to make the device part very simple. The control point just gives the device information like the count of total choices and expect the selected index as choice result. There will be common understanding description for how to be sure all services are available and what capability each service should have. This will be elaborated in upcoming sections.

Beside this since we are developing tangible interface, which itself is end user tool, this needs information from ASTRA System for eg communities, applications. Moreover, after getting different rules from user, it should be set in ASTRA Node for execution and storing in database. It should be done such that user could see and change rules even from web interface. So that it would be sure the tangible interface is working correctly and could be modified as needed. This means both web interface and tangible interface should rely on similar format.

To stick on service oriented architecture (SOA) of ASTRA System, tangible interface is expected not to be as a module within ASTRA Node. This Tangible interface is going to be connected with ASTRA system by web services, so that everything will be in clean code with loosely coupled to ASTRA system. Figure 5.5 depicts how this tangible system will be deployed.

5.5 COTS Analysis UPNP Architecture

UPnP technology defines an architecture for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices, and PCs of all form factors. It is designed to bring easy-to-use, flexible, standards-based connectivity to ad-hoc or unmanaged networks whether in the home, in a small business, public spaces, or attached to the Internet. UPnP technology provides a distributed, open networking architecture that leverages TCP/IP and the Web technologies to enable seamless proximity networking in addition to control and data transfer among networked devices. The UPnP Device Architecture (UDA) is more than just a simple extension of the plug and play peripheral model. It is designed to support zero-configuration, "invisible" networking, and automatic discovery for a breadth of device categories from a wide range of vendors. This means a device can dynamically join a network, obtain an IP address, convey its capabilities, and

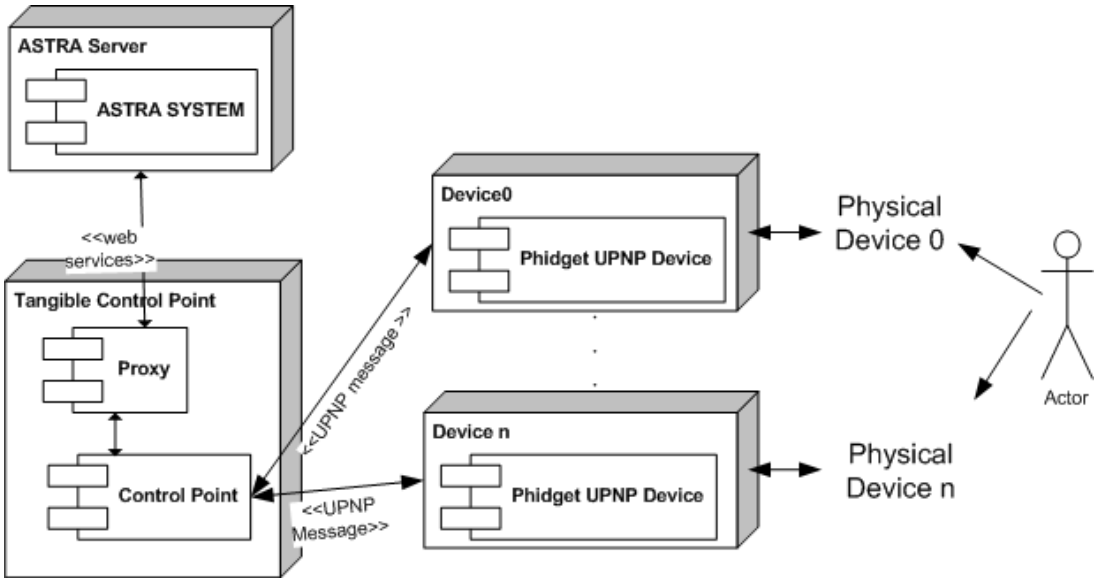


Figure 5.5: Deployment of tangible interface

learn about the presence and capabilities of other devices. Finally, a device can leave a network smoothly and automatically without leaving any unwanted state behind.

Two general classifications of devices are defined by the UPnP architecture: controlled devices (or simply "devices"), and control points. A controlled device functions in the role of a server, responding to requests from control points [21]. The UPnP System Architecture is depicted in figure 5.6.

Figure 5.7 shows the steps of UPnP Networking. Figure 5.8 shows device side of the UPnP Subsystem. Figure 5.9 shows Control Point side of the UPnP Subsystem.

5.5.1 Addressing

The foundation for UPnP networking is IP addressing. Control point and device get address either by a DHCP server or use Auto IP.

5.5.2 Discovery

By this control points finds interesting devices. Given an IP address, Step 1 in UPnP networking is discovery. When a device is added to the network, the UPnP discovery protocol allows that device to advertise its services to control points on the network. Devices refresh advertisements based on lease time and it cancels advertisements when removed. Similarly, when a control point is added to the network, the UPnP discovery protocol allows that control point to search for devices of interest on the network.

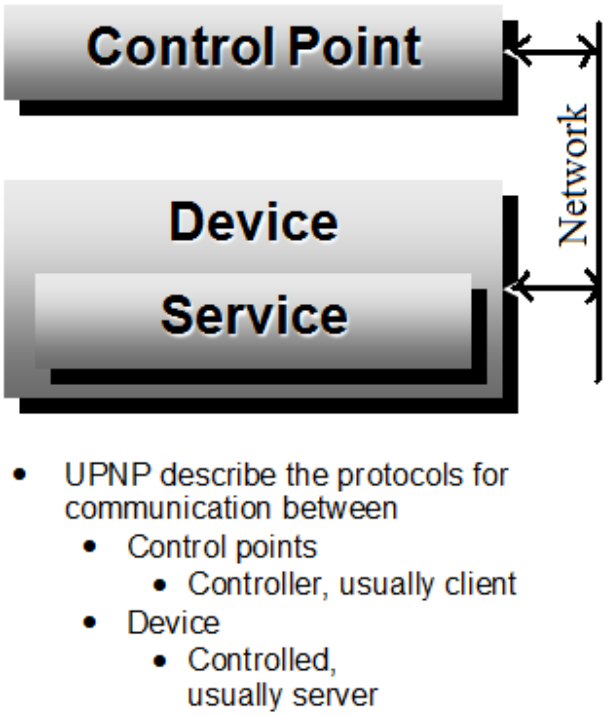


Figure 5.6: Typical Universal Plug and Play System Architecture

0 Control point and device get addresses
1 Control point finds interesting device
2 Control point learns about device capabilities
3 Control point invokes actions on device
4 Control point listens to state changes of device
5 Control point controls device and/or views device status using HTML UI

Figure 5.7: Steps to UPNP Networking

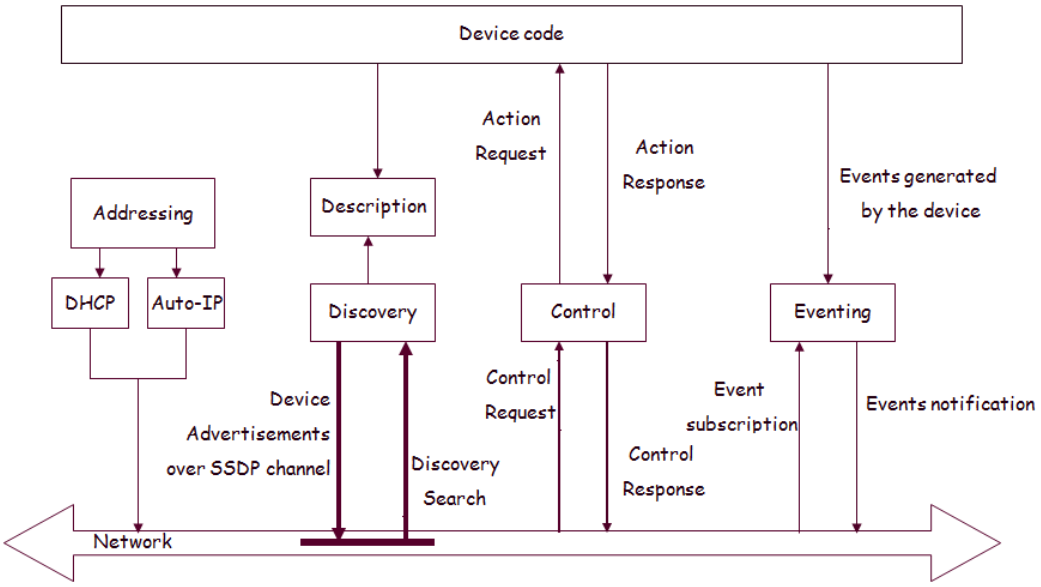


Figure 5.8: Device side of the UPnP Subsystem

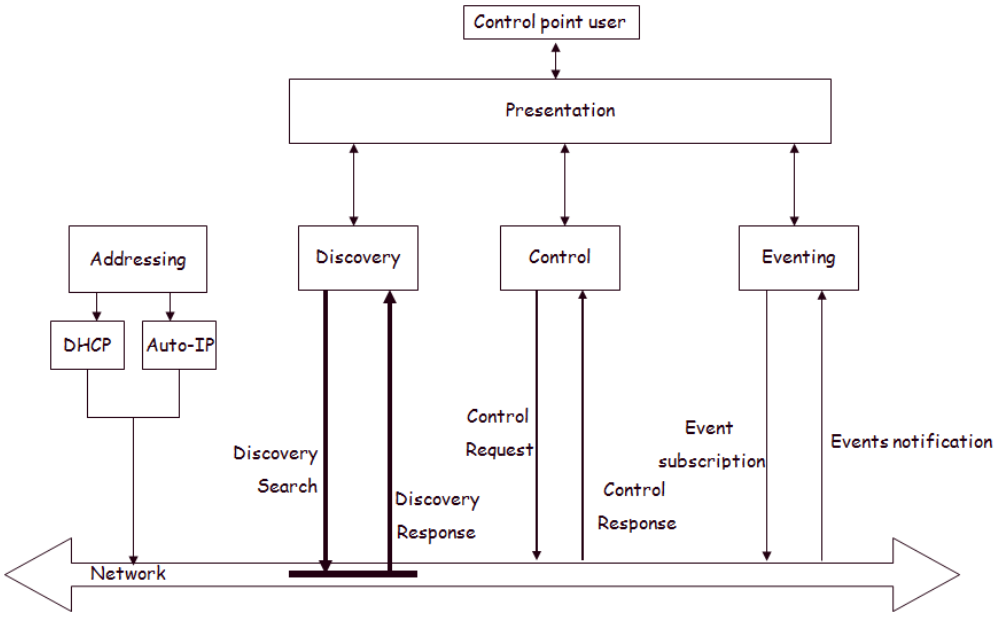


Figure 5.9: Control Point side of the UPnP Subsystem

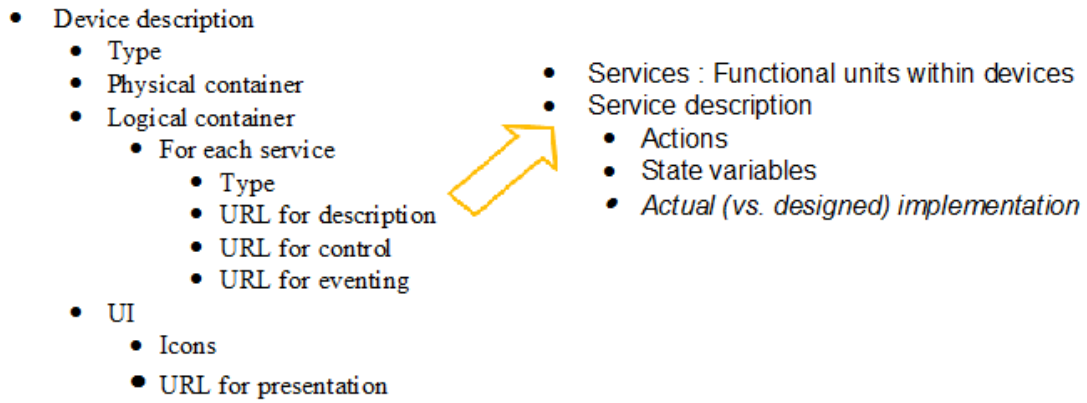


Figure 5.10: Description of device and service

5.5.3 Description

Devices declares and advertises its capabilities with description. Control point learns about device capabilities. After discovery, it gets URL for description and retrieves description. From this description, it gets URL for service description. The figure 5.10 shows how the description of device and services are connected.

5.5.4 Control

Control is Step 3 in UPnP networking. Control comes after addressing (Step 0) where devices get a network address, after discovery (Step 1) where control points find interesting device(s), and after description (Step 2) where control points learn about device capabilities. Control is independent of eventing (Step 4) where control points listen to state changes in device(s). Through control, control points invoke actions on devices and poll for values. Control and eventing are complementary to presentation (Step 5) where control points display a user interface provided by device(s). Given knowledge of a device and its services, a control point can ask those services to invoke actions and receive responses indicating the result of the action. Invoking actions is a kind of remote procedure call; a control point sends the action to the device's service, and when the action has completed (or failed), the service returns any results or errors.

To control a device, a control point invokes an action on the device's service. To do this, a control point sends a suitable control message to the control URL for the service (provided in the controlURL sub element of service element of device description). In response, the service returns any results or errors from the action. The effects of the action, if any, may also be modeled by changes in the variables that describe the run-time state of the service. When these state variables change, events are published to all interested control points. The figure 5.11 shows how action, query occur among control point and device.

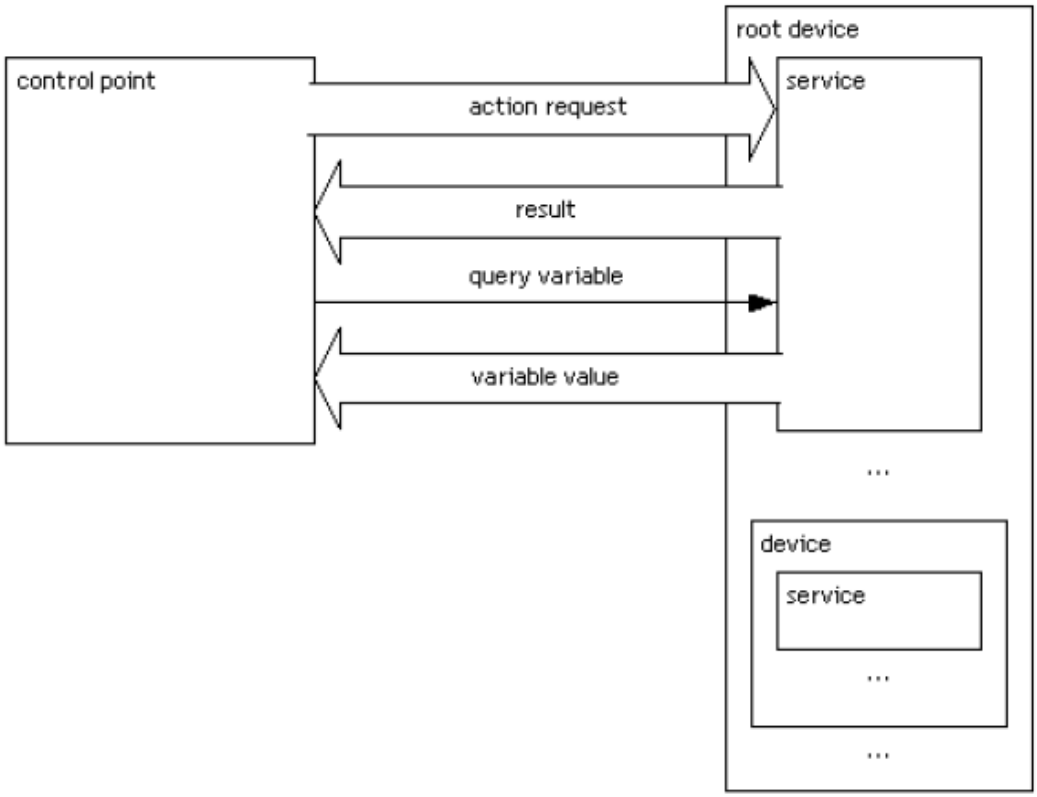


Figure 5.11: Control flow in Control Point and Device

5.5.5 Eventing

Device publishes a list of events for which control points can subscribe and get notifications. Control points subscribe with the devices for the events published. Eventing is Step 4 in UPnP networking. Through eventing, control points listen to state changes in device(s). As the section on Description explains, a UPnP service description includes a list of actions the service responds to and a list of variables that model the state of the service at run time. If one or more of these state variables are evented, then the service publishes updates when these variables change, and a control point may subscribe to receive this information. Throughout this section, publisher refers to the source of the events (typically a device's service), and subscriber refers to the destination of events (typically a control point).

To subscribe to eventing, a subscriber sends a subscription message. If the subscription is accepted, the publisher responds with a duration for the subscription. To keep the subscription active, a subscriber must renew its subscription before the subscription expires. When a subscriber no longer needs eventing from a publisher, the subscriber should cancel its subscription. The publisher notes changes to state variables by sending event messages. Event messages contain the names of one of more state variables and the current value of those variables, expressed in XML. A special initial event message is sent when a subscriber first subscribes; this event message contains the names and values for all evented variables and allows the subscriber to initialize its model of the state of the service. All subscribers are sent all event messages, subscribers receive event messages for all evented variables (not just some), and event messages are sent no matter why the state variable changed (either in response to a requested action or because the state the service is modeling changed). The figure 5.12 describes the subscription and notification flow among control point and devices.

5.5.6 Presentation

Presentation is Step 5 in UPnP networking. Presentation exposes an HTML-based user interface for controlling and/or viewing device status. If a device has a URL for presentation, then the control point can retrieve a page from this URL, load the page into a browser, and depending on the capabilities of the page, allow a user to control the device and/or view device status. The URL for presentation is contained within the presentationURL element in the device description.

5.6 COTS Analysis Phidgets

Phidgets are an easy to use set of building blocks for low cost sensing and control from PC. Using the Universal Serial Bus (USB) as the basis for all Phidgets, the complexity is managed behind this easy to use and robust Application Program Interface (API) library. The bottom level of the API is the C library - phidget21. This is a cross-platform library, which implements the low-level protocols necessary to communicate with the Phidgets, and exports a unified interface to the software programmer. Built upon this low level library are higher level libraries that simplify using Phidgets for many more languages. These higher level libraries contain only glue logic

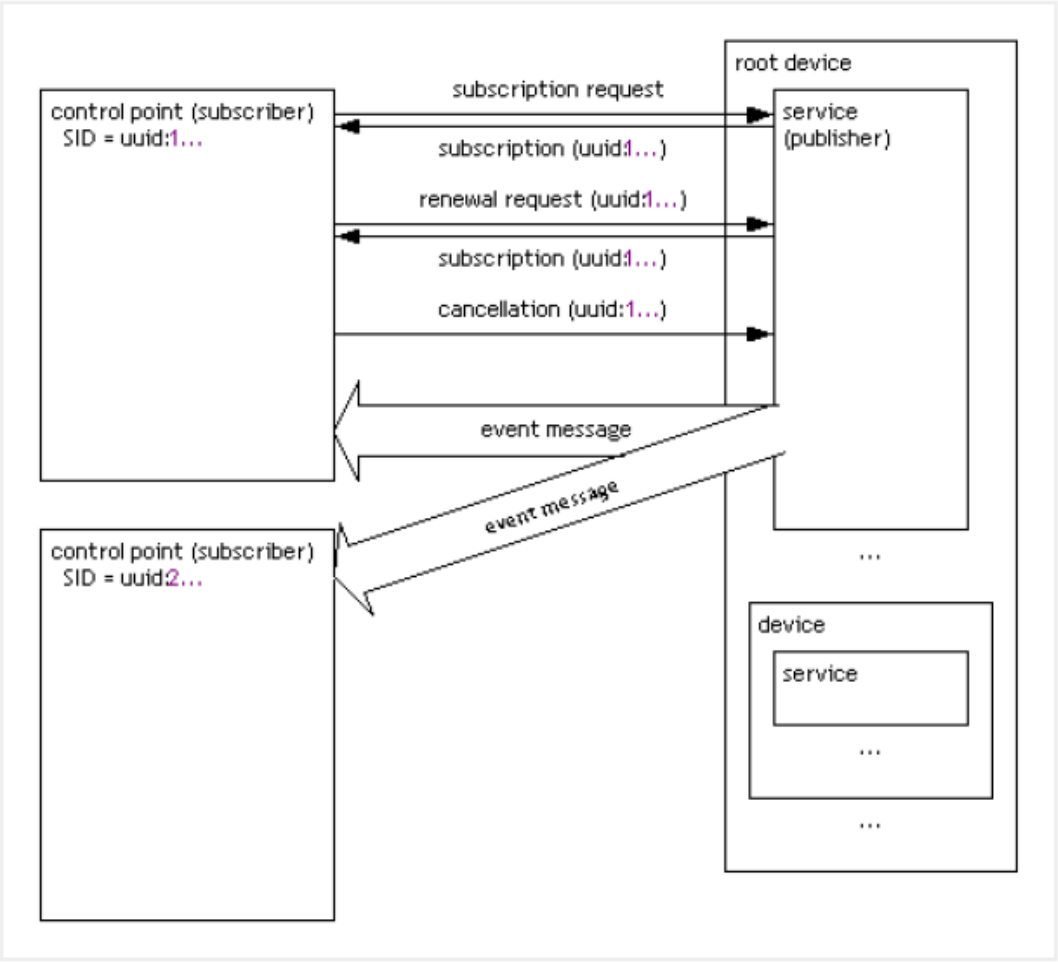


Figure 5.12: Subscription and Notification flow among control point and devices

Figure 5.13: Phidget Kit

for interfacing with the C library, thus making maintenance much easier [22]. The figure 5.13 shows the Phidget kit.

The first step in controlling a Phidget is calling Open() on it. This will signal the library that one would like to use this Phidget and register it for usage. Open will return immediately once called, because it can be called even if the Phidget to be used is not attached to the system. This is known as an asynchronous call. It is important to understand that most calls on a Phidget will fail if they are calls when the Phidget is not attached - in fact the only calls that are allowed on a detached Phidget are Close(), waitForAttachment() and Attached [get]. Once open has been called, there are two options: One can call waitForAttachment(timeout), which will block until either the Phidget is available or until the time out has passed, or one can wait until the Attach event fires. The event based method is recommended, and generally most useful for GUI based applications. The waitForAttachment() method is useful for simple command line applications. If one decide to use events, one need to register the event handlers before calling Open(), or one will miss events. Once the Phidget is attached, the full API can be used on it. Open is also pervasive. This means that once open has been called, it will constantly try to stay attached to a Phidget. Even if the Phidget is unplugged from the computer and then plugged back in, you will simply get a Detach event, and then an Attach event. It is a good idea to handle the Detach event in order to avoid calling the Phidget after it has detached. Phidgets can either be opened with or without using their unique serial number. In the event a serial number is not specified, the first available device will be opened. If there are more than one of the same type of Phidget attached to a computer, there is no way of knowing which of these will be opened. Once a Phidget is

device:
<friendlyName>**irules**</friendlyName>

Services:

<serviceType>urn:schemas-upnp-org:service:**irules-slider**1:1</serviceType>
getSlider1Value slider1Value *(0,1....n)*
setSlider1Resolution slider1Resolution *(0,1...n)*

<serviceType>urn:schemas-upnp-org:service:**irules-slider2**:1</serviceType>
getSlider2Value slider2Value *(0,1....n)*
setSlider2Resolution slider2Resolution *(0,1....n)*

<serviceType>urn:schemas-upnp-org:service:**irules-selection**:1</serviceType>
getSelectionValue selectionValue *(null,accept,deny,and)*

<serviceType>urn:schemas-upnp-org:service:**irules-default**:1</serviceType>
getDefaultValue defaultValue *(null,accept,deny)*

<serviceType>urn:schemas-upnp-org:service:**irules-comparison**:1</serviceType>
getComparisonValue comparisonValue *(null,eq,neq)*

<serviceType>urn:schemas-upnp-org:service:**irules-display**:1</serviceType>
setText text *(texts to display..)*

Figure 5.14: common understanding device and service parameter

opened by an application, it cannot be opened again in another application until closed by the first.

Event Handlers are used to notify your application that a noteworthy event has occurred on a Phidget. By creating a function accepting the proper parameters (an Event Handler), and registering that function with the library, the library is able to call your function whenever events occur. Because your Event Handler can be run at anytime, it is best to register them before calling Open(). Every event will contain a reference to the Phidget that raised the event. This allows properties of the Phidget to be evaluated within the Callback.The Event Type identifies the kind of event being fired, this allows the user to make a choice of how to handle the event.

5.7 Design

It is desired to have some common understanding to have flexibility on implementation of control point and devices. The figure 5.14 shows how the services and the action variables and the parameters are going to be defined.

One device could implement some of these services and other devices as well as other services. No matter which device implements which service until it follows this standard. In the figure 5.14, it's defined which action and action argument should be implemented by each service with expected values shown in italic. The service type is also been defined. The device implementing these service/s must have friendly name 'irules', so that control point just could search or accept for the devices having friendly name irules. Control point will look on the defined service from those devices and use the action, action argument to have action on the service. Control point checks whether it have all services available or not.

Control point can inform irules_slider1 service the number of choices with slider1Resolution argument of the setSlider1Resolution action. Then the device will notify the control point on change of choice from user. The device will notify this with slider1Value that have the selected index. For example if we have Community, Person, Application, mystatus, device then it means we have 5 choices to select on. Then the control point will inform this with slider1Resolution=5. The device will notify the selected value with for eg: slider1Value=3. This means the user has selected Application.

Control Point will update its list for the entity type selected by slider1value. This means it should have fresh list from ASTRA system using proxy. It is necessary because the lists like community,person etc are very dynamic and could have been change in ASTRA system. The number of count of the entities for the selected entity type is passed by control point to the service irules_slider2 by the argument slider2Resolution with the action setSlider2Resolution. The service will notify the selected index with slider2Value argument. If not changed then control point reads current selection with slider2Value argument from getSlider2Value action. The control point asks irules_display service to display text with the setText action.

"irules_comparison" and "irules_default" service notify its value with respective arguments and control point sets accordingly and display it with display service. The default setting need to be updated to ASTRA system by control point.

When irules_selection service notifies about the "allow", "deny" or "and", control point sets on rule. If it is "and", it will make composite rule but caution should be apply not to duplicate the conditions. If it is "allow" or "deny" it should set rule accordingly in XML format similar as used in web interface as depicted in figure 5.15 and sets rule in ASTRA System with proxy. The figure 5.16 shows the initial sequence diagram that will occur when control point get started. It will discover all related services and do some initial action and subscribe to have further notification.

On each service notification, control point will be triggered to do something since it is being implemented as event based. The sequence diagram 5.17,5.18,5.19,5.20,5.21 details the sequence when notification comes from slider1, slider2,selection,comparison,default respectively.

UPnP device should be available when physical device is functioning. Control point should perform broadcast search (msearch), when it starts rather than waiting for device to get alive notification.

Figure 5.15: Compounded XML Rule

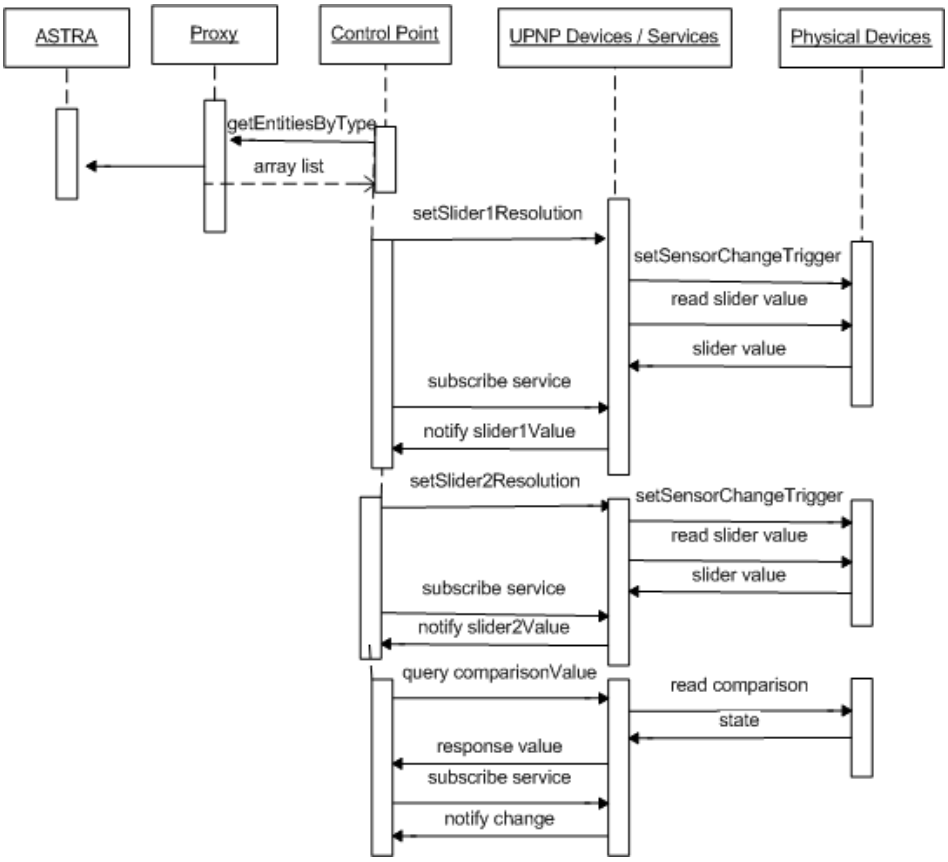


Figure 5.16: Sequence diagram for Control Point initialization

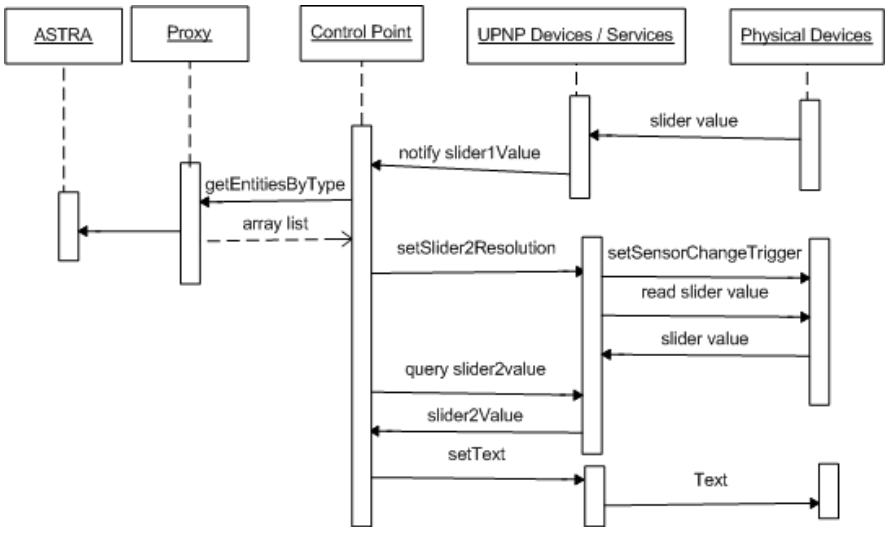


Figure 5.17: Sequence diagram when entity type (slider1) changes

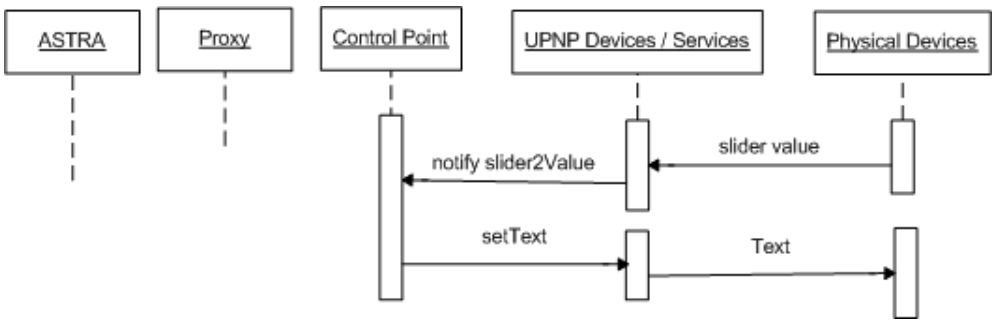


Figure 5.18: Sequence diagram when entities (slider2) changes

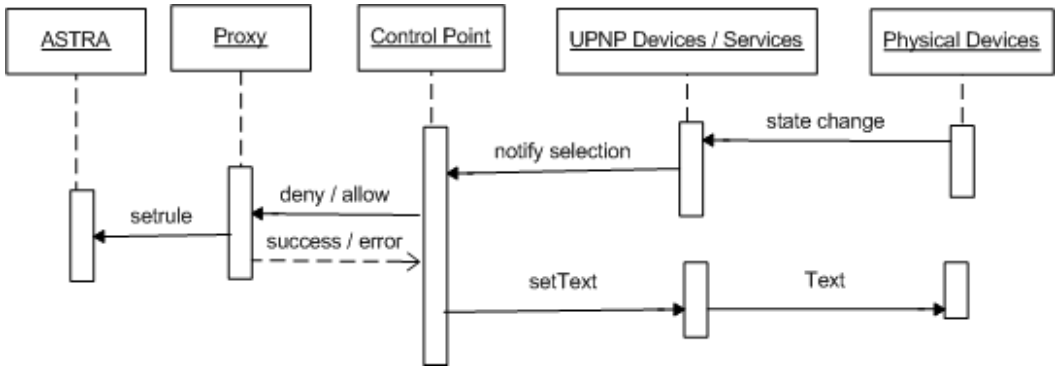


Figure 5.19: sequence diagram when selection (allow, deny, and) pushed

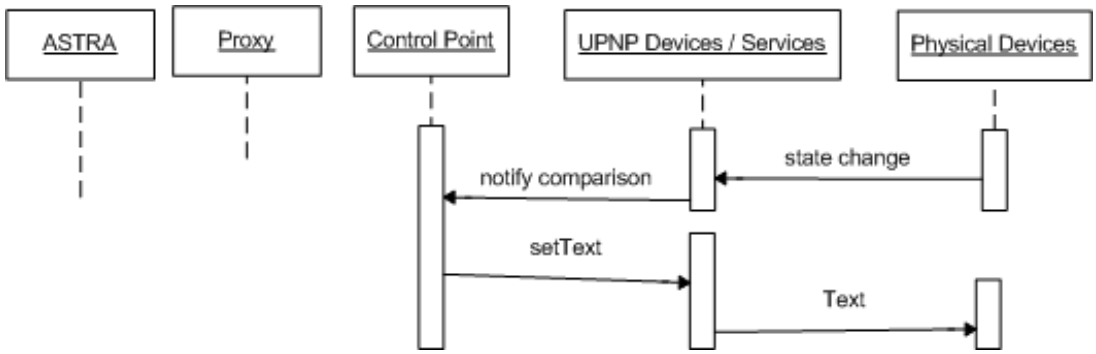


Figure 5.20: sequence diagram when comparison (eq, neq) changed

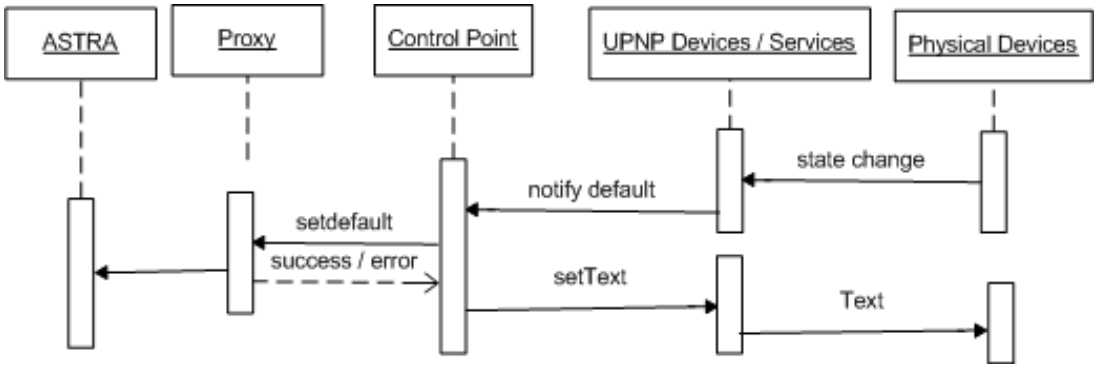


Figure 5.21: sequence diagram when default(accept, deny) changed

Chapter 6

Implementation of Tangible Interface

This chapter details the implementation part of tangible interface.

6.1 Introduction

CyberGarage's clink jar library has been used for UPnP implementation. CyberGarage is a reference implementation of UPnP [23]. UPnP implementation of Device and Control Point ; Phidget interface and proxy to ASTRA will be discussed in following sections.

6.2 Device

slider1, slider2, selection, default, comparison services have been implemented in one UPnP device and that is here named as UPnPdevice0. The figure 6.1 shows the XML device description for UPnPdevice0. It points to its different service description 6.3 6.4 6.5 6.6 6.7.

The class diagram for the Device0 is depicted by figure 6.2. The device is Phidget implementation and when physical device attached, UPnP services will be available, and when physical device get detached, UPnP services would stop. Thus assuring the availability of the services only when there is physical device connection.

Display service has been implemented by another UPnP Device, named UPnPDevice1. The figure 6.8 shows the XML device description. It embodies display service, which is defined as 6.10.

The class diagram for the Device1 is depicted by figure 6.9. The device is phidget implementation and when physical device attached, UPnP services will be available, and when physical device get detached, UPnP services would stop. Thus assuring the availability of the services only when there is physical device connection.

```
<?xml version="1.0" ?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
        <specVersion>
                <major>1</major>
                <minor>0</minor>
        </specVersion>
        <device>
                <deviceType>urn:schemas-upnp-org:device:compositedevice:1</deviceType>
                <friendlyName>irules</friendlyName>
                <manufacturer>NTNU</manufacturer>
                <manufacturerURL>http://idi.ntnu.no</manufacturerURL>
                <modelDescription>ASTRA UPnP Device connecting multiple sliders n push buttons</modelDescription>
                <UDN>uuid:upnp-astra-irules-device0</UDN>

                <serviceList>
                        <service>
                                <serviceType>urn:schemas-upnp-org:service:irules-slider1:1</serviceType>
                                <serviceId>urn:schemas-upnp-org:serviceId:device0-slider1</serviceId>
                                <SCPDURL>/upnpdevice0Services/slider1/description.xml</SCPDURL>
                                <controlURL>/upnpdevice0Services/slider1/control</controlURL>
                                <eventSubURL>/upnpdevice0Services/slider1/eventSub</eventSubURL>
                        </service>
                        <service>
                                <serviceType>urn:schemas-upnp-org:service:irules-slider2:1</serviceType>
                                <serviceId>urn:schemas-upnp-org:serviceId:device0-slider2</serviceId>
                                <SCPDURL>/upnpdevice0Services/slider2/description.xml</SCPDURL>
                                <controlURL>/upnpdevice0Services/slider2/control</controlURL>
                                <eventSubURL>/upnpdevice0Services/slider2/eventSub</eventSubURL>
                        </service>
                        <service>
                                <serviceType>urn:schemas-upnp-org:service:irules-selection:1</serviceType>
                                <serviceId>urn:schemas-upnp-org:serviceId:device0-selection</serviceId>
                                <SCPDURL>/upnpdevice0Services/selection/description.xml</SCPDURL>
                                <controlURL>/upnpdevice0Services/selection/control</controlURL>
                                <eventSubURL>/upnpdevice0Services/selection/eventSub</eventSubURL>
                        </service>
                        <service>
                                <serviceType>urn:schemas-upnp-org:service:irules-default:1</serviceType>
                                <serviceId>urn:schemas-upnp-org:serviceId:device0-default</serviceId>
                                <SCPDURL>/upnpdevice0Services/default/description.xml</SCPDURL>
                                <controlURL>/upnpdevice0Services/default/control</controlURL>
                                <eventSubURL>/upnpdevice0Services/default/eventSub</eventSubURL>
                        </service>
                        <service>
                                <serviceType>urn:schemas-upnp-org:service:irules-comparison:1</serviceType>
                                <serviceId>urn:schemas-upnp-org:serviceId:device0-comparison</serviceId>
                                <SCPDURL>/upnpdevice0Services/comparison/description.xml</SCPDURL>
                                <controlURL>/upnpdevice0Services/comparison/control</controlURL>
                                <eventSubURL>/upnpdevice0Services/comparison/eventSub</eventSubURL>
                        </service>
                </serviceList>
                <presentationURL>http://idi.ntnu.no/</presentationURL>
        </device>
</root>
```

Figure 6.1: device description for UPnPdevice0

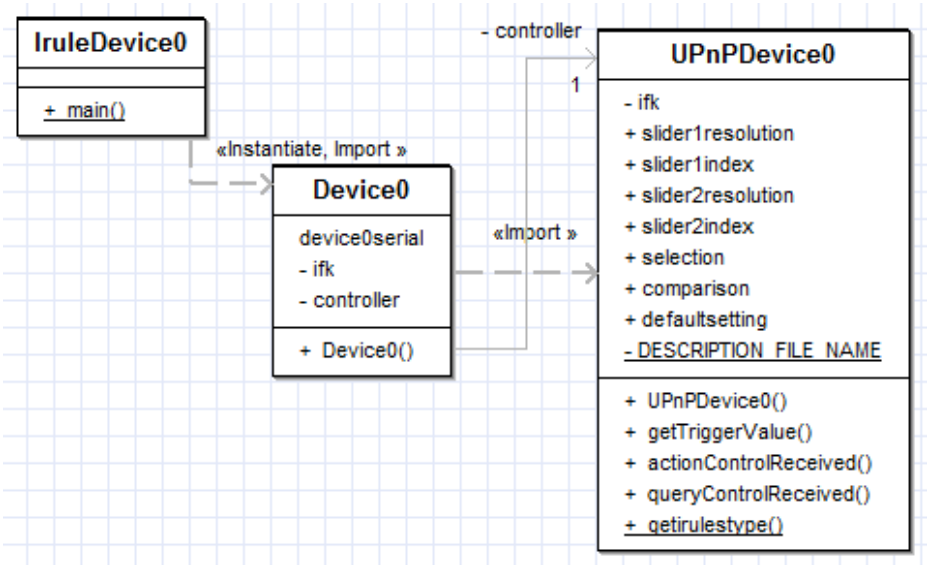


Figure 6.2: Class diagram for device0

6.3 Control Point

When a device is added to the network, it sends a multicast request with method NOTIFY. It will send NOTIFY in periodic time out afterwards. When a control point is added to the network, it will send a multicast request with method M-SEARCH. After implementing NOTIFY, reply to M-SEARCH, control point can discover already existing devices. Otherwise it should wait till the periodic notification.

When the notification are from "irules" devices i.e. device have friendlyname as "irules", then control point will initialize the services and subscribe all the services provided by the device, except display service. Since display service is only output device and do not have any input change to have event notification. When all services are available to control point then only the notification from the service get accounted. If not all services are available but control point already discover display service, then that could be used to display missing services information. The control point works on event trigger. When there is change in the physical interface, like sliding and press on button, then the Device will give event notification to all the control point that has subscribed to it. So this control point, if all services are already available to this, control point decides what to do based on the service the event notification is from. If event notification is from slider1 service, it will refresh the list for the entity type selected by slider1value. It is very essential to update like this, since the ASTRA System is very dynamic and user might have change the focus application, or devices have been updated or awareness state have been changed. By the count of entities of this selected entity type, control point asks slider2 service to set its resolution based on the count. Control point then do query to know the current selected index from slider2 service and display the information through display service. If event notification is from slider2 service, control point asks to display service to set the new display value according to the selection. When notification is from selection service, it acts according to the value either

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0" >
	<specVersion>
		<major>1</major>
		<minor>0</minor>
	</specVersion>
	<actionList>
		<action>
			<name>getSlider1Value</name>
			<argumentList>
				<argument>
					<name>slider1Value</name>
					<relatedStateVariable>slider1Value</relatedStateVariable>
					<direction>out</direction>
				</argument>
			</argumentList>
		</action>
		<action>
			<name>setSlider1Resolution</name>
			<argumentList>
				<argument>
					<name>slider1Resolution</name>
					<relatedStateVariable>slider1Resolution</relatedStateVariable>
					<direction>in</direction>
				</argument>
			</argumentList>
		</action>
	</actionList>
	<serviceStateTable>
		<stateVariable sendEvents="yes">
			<name>slider1Value</name>
			<dataType>int</dataType>
		</stateVariable>
		<stateVariable sendEvents="no">
			<name>slider1Resolution</name>
			<dataType>int</dataType>
		</stateVariable>
	</serviceStateTable>
</scpd>
```

Figure 6.3: slider1 service xml description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0" >
      <specVersion>
            <major>1</major>
            <minor>0</minor>
      </specVersion>
      <actionList>
            <action>
                  <name>getSlider2Value</name>
                  <argumentList>
                        <argument>
                              <name>slider2Value</name>
                              <relatedStateVariable>slider2Value</relatedStateVariable>
                              <direction>out</direction>
                        </argument>
                  </argumentList>
            </action>
            <action>
                  <name>setSlider2Resolution</name>
                  <argumentList>
                        <argument>
                              <name>slider2Resolution</name>
                              <relatedStateVariable>slider2Resolution</relatedStateVariable>
                              <direction>in</direction>
                        </argument>
                  </argumentList>
            </action>
      </actionList>
      <serviceStateTable>
            <stateVariable sendEvents="yes">
                  <name>slider2Value</name>
                  <dataType>int</dataType>
            </stateVariable>
            <stateVariable sendEvents="no">
                  <name>slider2Resolution</name>
                  <dataType>int</dataType>
            </stateVariable>
      </serviceStateTable>
</scpd>
```

Figure 6.4: slider2 service xml description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0" >
      <specVersion>
            <major>1</major>
            <minor>0</minor>
      </specVersion>
      <actionList>
            <action>
                  <name>getComparisonValue</name>
                  <argumentList>
                        <argument>
                              <name>comparisonValue</name>
                              <relatedStateVariable>comparisonValue</relatedStateVariable>
                              <direction>out</direction>
                        </argument>
                  </argumentList>
            </action>
      </actionList>
      <serviceStateTable>
            <stateVariable sendEvents="yes">
                  <name>comparisonValue</name>
                  <dataType>String</dataType>
            </stateVariable>
      </serviceStateTable>
</scpd>
```

Figure 6.5: comparison service xml description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0" >
	<specVersion>
		<major>1</major>
		<minor>0</minor>
	</specVersion>
	<actionList>
		<action>
			<name>getSelectionValue</name>
			<argumentList>
				<argument>
					<name>selectionValue</name>
					<relatedStateVariable>selectionValue</relatedStateVariable>
					<direction>out</direction>
				</argument>
			</argumentList>
		</action>
	</actionList>
	<serviceStateTable>
		<stateVariable sendEvents="yes">
			<name>selectionValue</name>
			<dataType>String</dataType>
		</stateVariable>
	</serviceStateTable>
</scpd>
```

Figure 6.6: selection service xml description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0" >
	<specVersion>
		<major>1</major>
		<minor>0</minor>
	</specVersion>
	<actionList>
		<action>
			<name>getDefaultValue</name>
			<argumentList>
				<argument>
					<name>defaultValue</name>
					<relatedStateVariable>defaultValue</relatedStateVariable>
					<direction>out</direction>
				</argument>
			</argumentList>
		</action>
	</actionList>
	<serviceStateTable>
		<stateVariable sendEvents="yes">
			<name>defaultValue</name>
			<dataType>String</dataType>
		</stateVariable>
	</serviceStateTable>
</scpd>
```

Figure 6.7: default service xml description

```
<?xml version="1.0" ?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
        <specVersion>
                <major>1</major>
                <minor>0</minor>
        </specVersion>
        <device>
                <deviceType>urn:schemas-upnp-org:device:multidevice:1</deviceType>
                <friendlyName>irules</friendlyName>
                <manufacturer>NTNU</manufacturer>
                <manufacturerURL>http://idi.ntnu.no</manufacturerURL>
                <modelDescription>ASTRA UPnP Device connecting display unit</modelDescription>
                <UDN>uuid:upnp-astra-irules-device1</UDN>

                <serviceList>
                        <service>
                                <serviceType>urn:schemas-upnp-org:service:irules-display:1</serviceType>
                                <serviceId>urn:schemas-upnp-org:serviceId:device1-display</serviceId>
                                <SCPDURL>/upnpdevice1Services/display/description.xml</SCPDURL>
                                <controlURL>/upnpdevice1Services/display/control</controlURL>
                                <eventSubURL>/upnpdevice1Services/display/eventSub</eventSubURL>
                        </service>
                  </serviceList>
                <presentationURL>http://idi.ntnu.no/</presentationURL>
        </device>
</root>
```

Figure 6.8: device description for UPnPdevice1

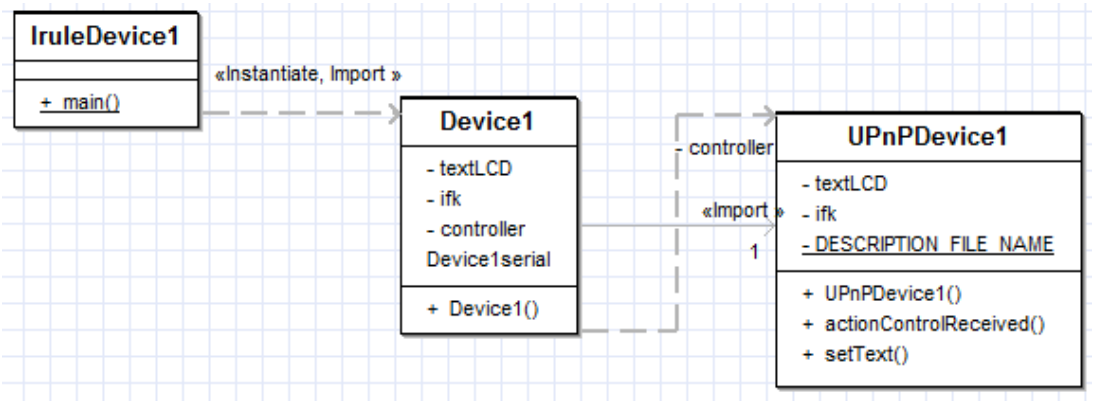


Figure 6.9: Class diagram for device1

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0" >
      <specVersion>
            <major>1</major>
            <minor>0</minor>
      </specVersion>
      <actionList>
            <action>
                  <name>setText</name>
                  <argumentList>
                        <argument>
                              <name>text</name>
                              <relatedStateVariable>text</relatedStateVariable>
                              <direction>in</direction>
                        </argument>
                  </argumentList>
            </action>
      </actionList>
      <serviceStateTable>
            <stateVariable sendEvents="no">
                  <name>text</name>
                  <dataType>String</dataType>
            </stateVariable>
      </serviceStateTable>
</scpd>
```

Figure 6.10: display service xml description

it is 'and', 'accept' , 'deny'. If selection service notification value is 'and' then it will add the current selection to rule but makes sure not to be repeated in this composite rule just to avoid human error for pressing multiple times, or pressing 'and' and after that 'allow' or 'deny'. If the selection value is 'accept' or 'deny' then the current selection is added with caution not to duplicate entry. These rule atoms are converted to XML rule compatible to web interface as we discuss in earlier chapter and save that on ASTRA System through proxy with setRule.

Since the control point and device and so services are loosely coupled and control point keep track of availability or unavailability of devices and so of services. When control points do not get periodic alive notification from device or if it gets byebye notification, control point un set the availability of services and waits again till all services are available before functioning again. Pseudo code of the implementation of the control point is given in figure 6.11.

Class diagram of this is depicted in figure 6.12.

6.4 Phidget

Event handlers are registered before calling Open(). Open will return immediately once called, because it can be called even if the Phidget to be used is not attached to the system. The program waits until the Attach event fires with waitForAttachement(). When ever attach event fires, that means the physical device is attached to it. UPnP device set to start when this event triggered. If after attach, sometime detach event occurs, then UPnP device stops saying bybbye to the control point. This dynamic availability of UPnP device is very important because if there is no physical

```
CtrlPoint extends ControlPoint implements TangibleConstants, NotifyListener, EventListener, SearchResponseListener,DeviceChangeListener
{
    CtrlPoint()
    {
            ....
                    addNotifyListener(this);
                    addSearchResponseListener(this);
                    addEventListener(this);
                    addDeviceChangeListener (this);
    }

    deviceAdded (Device dev)
    {
    if friendlyname of dev is "irules"
            go through all services of the device.
            if slider1 service,
                    setslider1resolution.
            else if slider2 service,
                    setslider2resolution
            else if display..
            else if comparison service
                    query value
            else if default service,
                    query value
            else;

            subscribe to the service.
            flagset for the service.
    }

    deviceRemoved(Device dev)
    {
    if friendlyname of dev is "irules"
            flagunset for all services belonging this device.
    }

   eventnotify(String uuid, long seq, String name, String value)
    {
     if all services are available. ..
            if from slider1, refresh entities, setslider2resolution, query slider2value, display
            if from slider2, display
            if from selection,
                    when value is 'and' add the condition
                    when the value is allow or deny, add the condition and finalize rule and setrule to ASTRA.
            if from comparison, display
            if from default, putuserpreference to ASTRA.

     else if display service available, display error.
     else system error.
    }
}
```

Figure 6.11: Pseudo Code for Control Point

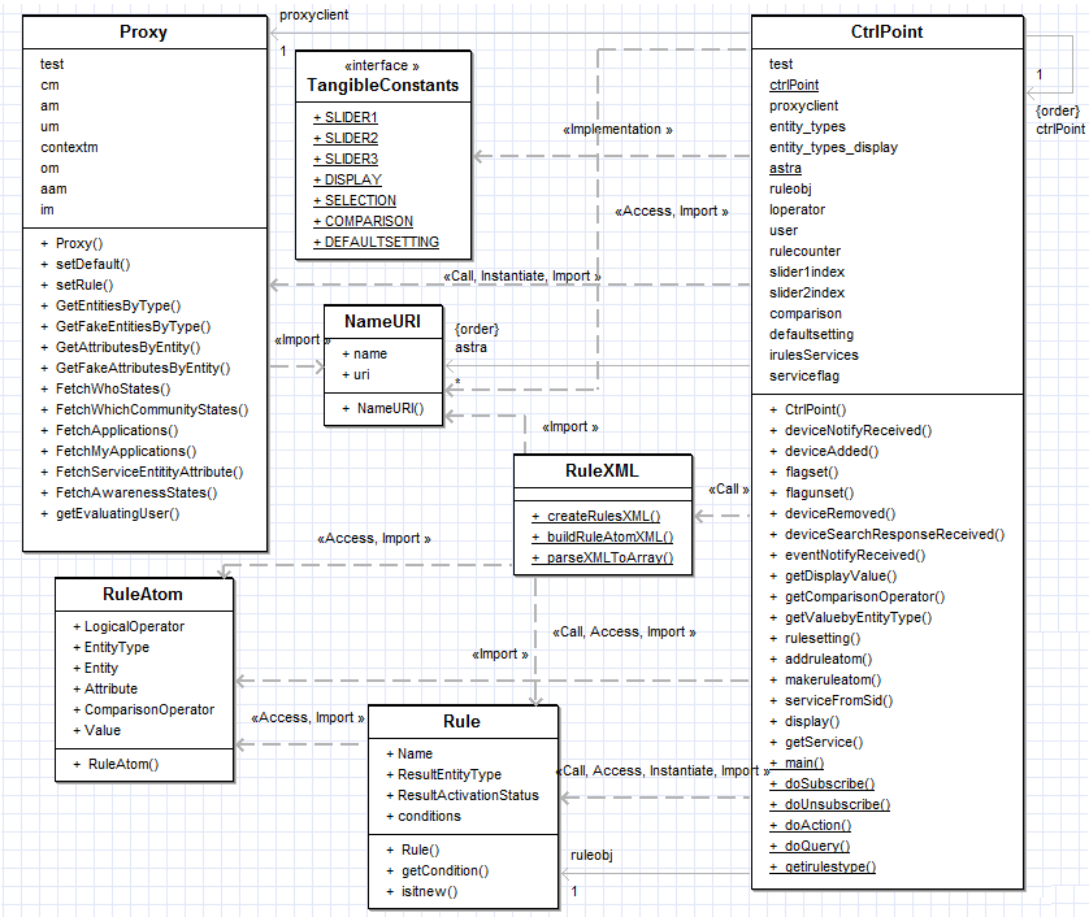


Figure 6.12: Class Diagram of Control Point

device available then there is no meaning of having UPnP (software) device. By this control point get notified of tear off the device.

The change in the Phidget digital input trigger inputchange event and change in analog input can trigger to sensorchange event. On the event, we can set the new values accordingly and UPnP device will notify this. The analog input could be set for how much change should it be notified. For example if we have 10 entities for selection, then we will divide the slider in 10 parts. so that transition among them only have trigger on. If the entities are just 5 then this is divided in 5 parts and will be notified only on that much transition. This is done with setSensorChangeTrigger. The figure 6.13 depicts the pseudo code for interfacing Phidget device0.

6.5 Proxy: to ASTRA System

Proxy connects to ASTRA System by web services. ASTRA System provides different services that we need to populate entities and for setting interruption rules and interruption preferences.

User must be member of community before publishing application to the community. User publishes application to the community. So focusing user do not see published application until she is member of that community. So the application that could arise interruption to the user is only possible from the community where she is member of. isMemberOf service from communitymanager provides this membership list for the particular user. "getCommunitName" from communitymanager gives the user friendly name for particular community id. As interruption will not be from out of application that have not been subscribed and focused, only the people having ownership to this application can interrupt. "listFocusApplications" service from awareness application manager is used to list focus applications and extract the user information and to uniquely represent them. Application part of interruption rules are also need to be extracted from listfocusApplications. Awareness status are listed with getResourceByClass("Awareness",) from ontology manager. Device providers and device sensor services that is device states are extracted with getServiceProviders and getSensorServicesByProvider from context manager. Finally setRule is used from awareness manager to set XML rule and putUserPreference is used from interruption manager to have default setting from user.

As ASTRA publishes WSDL that describes the endpoint / web service, the proxy need to use this WSDL. The figure 6.14 shows the services provided by ASTRA Backend and Node. A WSDL document describes a Web service. A WSDL binding describes how the service is bound to a messaging protocol, particularly the SOAP messaging protocol. A WSDL SOAP binding can be either a Remote Procedure Call (RPC) style binding or a document style binding. A SOAP binding can also have an encoded use or a literal use. This gives you four style/use models: RPC/encoded, RPC/literal, Document/encoded , Document/literal [24].

Figure 6.15 is snapshot of WSDL document of community manager provided by ASTRA Backend. This shows it is based on RPC/encoded. JAX-RPC project [25] develops and evolve the code base for the Reference Implementation of JAX-RPC, the Java APIs for XML based RPC. Axis-1.4 [26] supports JAX-RPC through WSDL2java. It has been used to generate java files from the WSDL XML file so that proxy can use them to get service from existing endpoint. It will create four files for eg: WSName, WSNameService, WSNameServiceLocater, and WS-

```
public class Device0
{
device0serial =xxxx;
private InterfaceKitPhidget ifk;
private UPnPDevice0 controller;

public Device0() {
   ifk = new InterfaceKitPhidget();
         ifk.addAttachListener  //phidget attached
            upnp device (controller) start
         ifk.addDetachListener  //phidget detached.
            upnp device (controller)  stop
         ifk.addInputChangeListener
            update variable as the input state change
         ifk.addOutputChangeListener
            ...
         ifk.addSensorChangeListener
            update slider value as updated
   ifk.open(device0serial);
   ifk.waitForAttachment();
}
```

Figure 6.13: Pseudo code for interfacing phidget device0

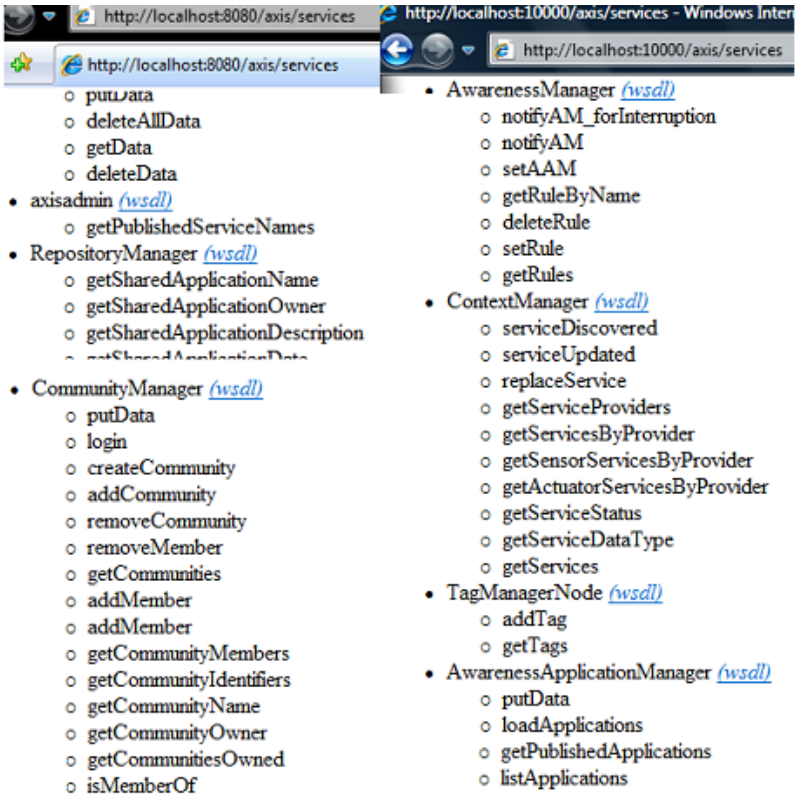


Figure 6.14: Services snapshot provided by ASTRA BackEnd and Node

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace="http://localhost:8080/axis/services/CommunityMan
schemas.xmlsoap.org/wsdl/" xmlns:apachesoap="http://xml.apache.org/xml-soap" x
localhost:8080/axis/services/CommunityManager" xmlns:intf="http://localhost:80
CommunityManager" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xm
schemas.xmlsoap.org/wsdl/" xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soa
www.w3.org/2001/XMLSchema"><wsdl:types><schema targetNamespace="http://localho
CommunityManager" xmlns="http://www.w3.org/2001/XMLSchema"><import namespace="
schemas.xmlsoap.org/soap/encoding/"/><complexType name="ArrayOf_xsd_int"><comp
base="soapenc:Array"><attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:in
complexContent></complexType><complexType name="ArrayOf_xsd_string"><complexCo
base="soapenc:Array"><attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:st
restriction></complexContent></complexType></schema></wsdl:types>
  <message name="loginResponse">
    <part name="loginReturn" type="xsd:boolean"/>
  </message>
....
....
 <message name="isMemberOfResponse">
    <part name="isMemberOfReturn" type="impl:ArrayOf_xsd_string"/>
  </message>
  <message name="getCommunityNameRequest">
    <part name="identifier" type="xsd:string"/>
  </message>
.....
....
 <binding name="CommunityManagerSoapBinding" type="impl:CommunityManager">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap
    <operation name="getData">
      <wsdlsoap:operation soapAction=""/>
      <input name="getDataRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding
impl.cm.astra.ist.eu" use="encoded"/>
      </input>
      <output name="getDataResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding
localhost:8080/axis/services/CommunityManager" use="encoded"/>
      </output>
    </operation>
.....
.....
```

Figure 6.15: Community Manager WSDL / Web Service endpoint

```
WSName, WSNameService   } interfaces
WSNameServiceLocator, WSNameSoapBinding   } classes

//make a service
WSNameService service = new WSNameServiceLocator();
//use the service to get a stub to the service
WSName name = service.getWSName();
//make the actual call..
name.??
```

Figure 6.16: Using service to get a stub

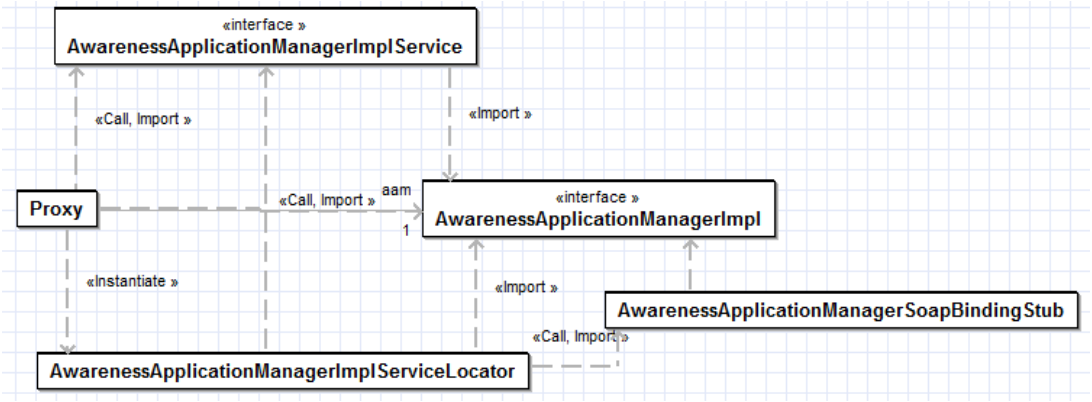


Figure 6.17: Class diagram connecting proxy with awareness application manager

NameSoapBindingStub. First two are interfaces and last two are classes. The figure 6.16 is to show how to create stub, which is local object representing the remote service.

The figure 6.17 depicts the class diagram for how the proxy is connected to awareness application manager.

The figure 6.18 depicts the class diagram for how the proxy is connected to awareness application.

The figure 6.19 depicts the class diagram for how the proxy is connected to community manager.

In the similar way, the proxy is connected with other managers like ontology manager, context manager, interruption manager, user manager.

6.6 circuit diagram

Circuit diagram for device0 is depicted by figure 6.20 and for device1 by figure 6.21.

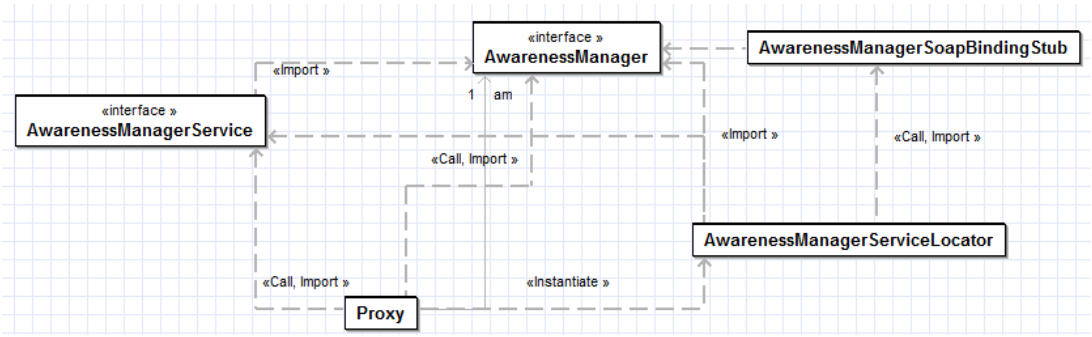


Figure 6.18: class diagram connection proxy and awarness managar

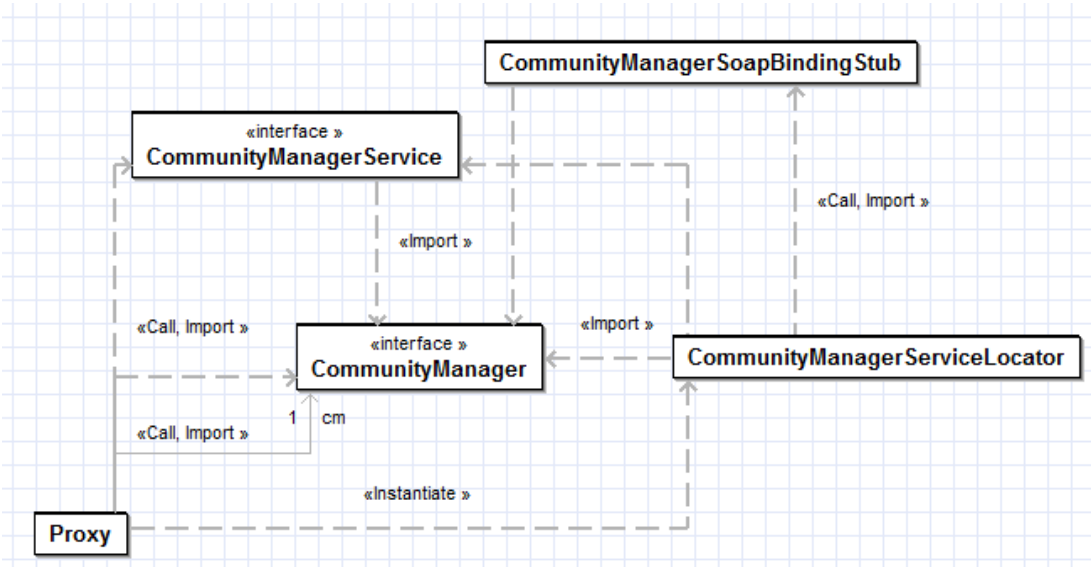


Figure 6.19: Class diagram connecting proxy with community manager

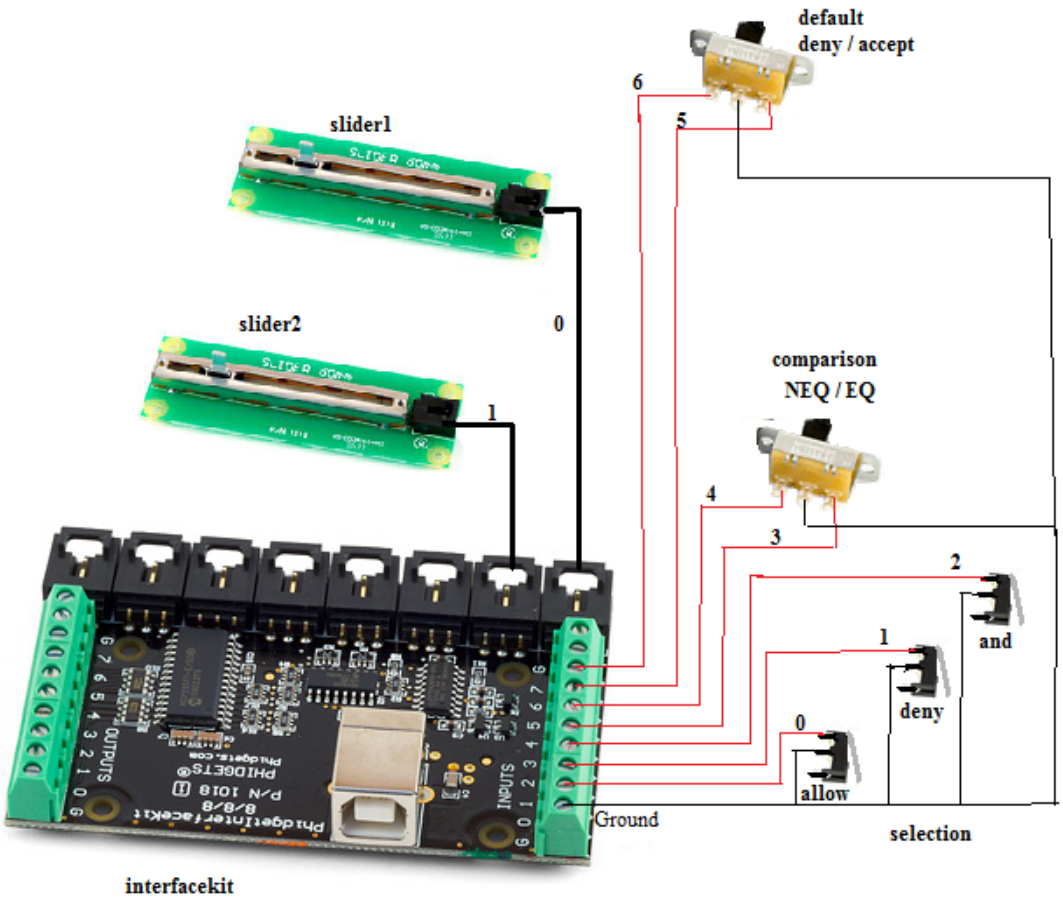


Figure 6.20: Circuit Diagram for Device0

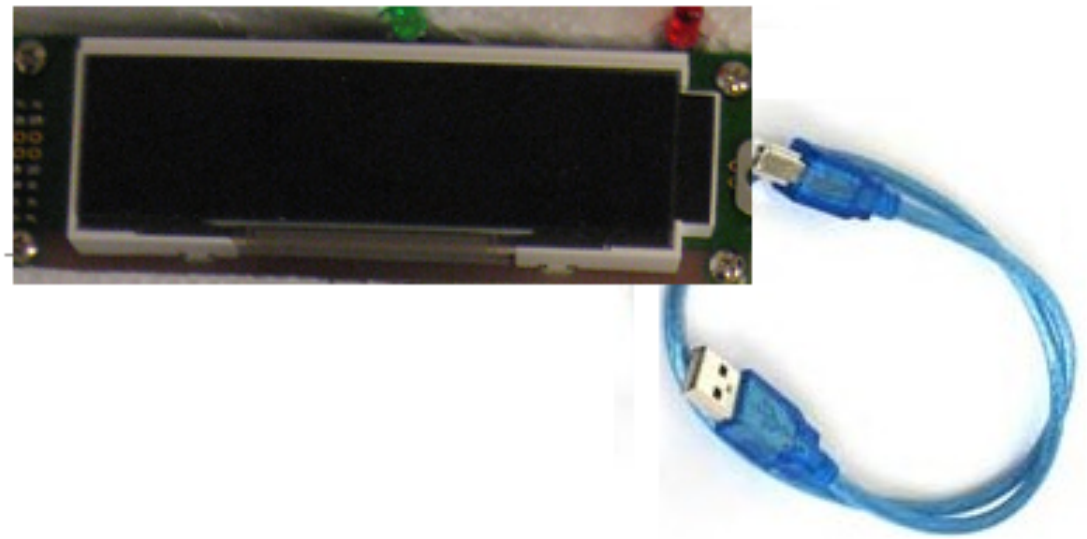


Figure 6.21: Circuit Diagram for Device1

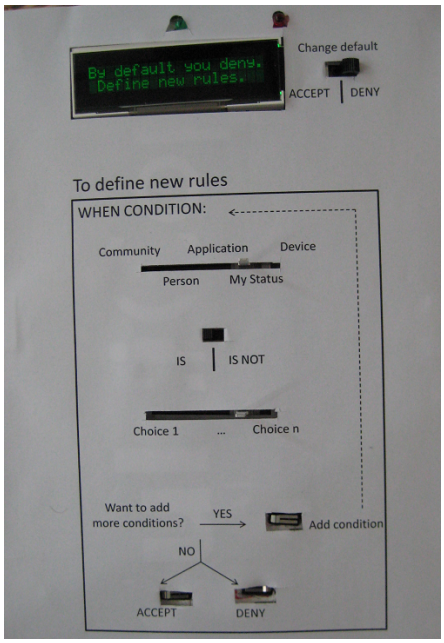


Figure 6.22: Tangible Interface showing default setting

6.7 Final look

This is how finally tangible interface looks like.

6.8 Comparison of tangible and web interface

There is slightly varying functionality in web and tangible interface. Web interface has both "and" and "or" option for making composite rule but tangible interface only have "and" option for making composite rule. Web interface show all rules in single page and can be changed / edited old rules for the user. But in case of tangible interface, we can set the rule but can't view the already set rule and neither can change old rules. But of course, the rules that have been set from tangible interface could be seen and modified from web interface.

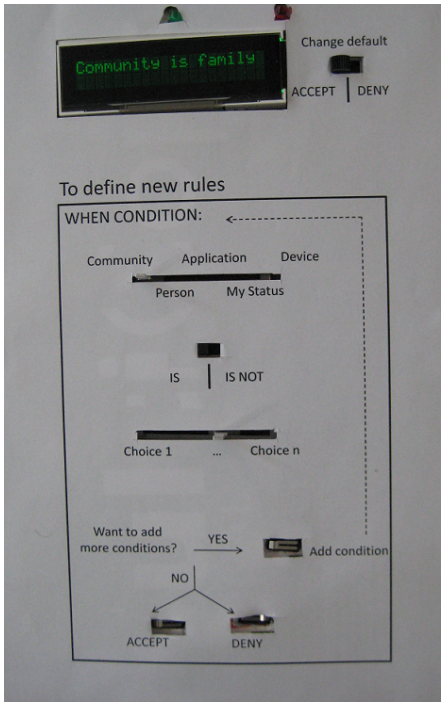


Figure 6.23: Tangible Interface with community selected

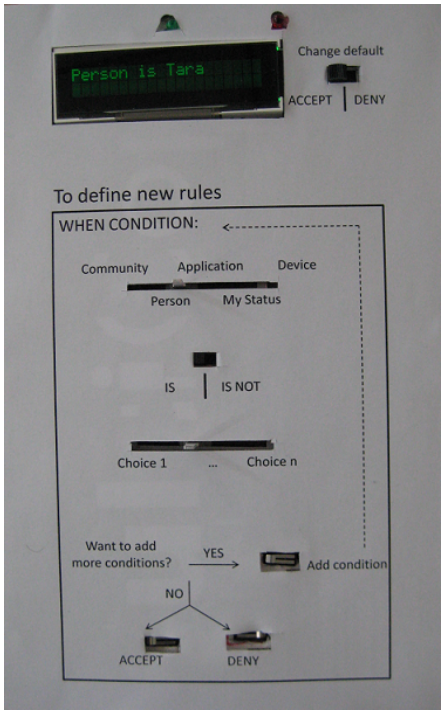


Figure 6.24: Tangible Interface with Person selected

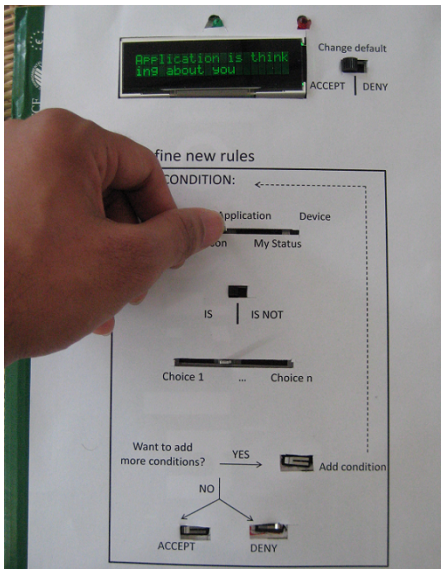


Figure 6.25: Tangible Interface with application selected

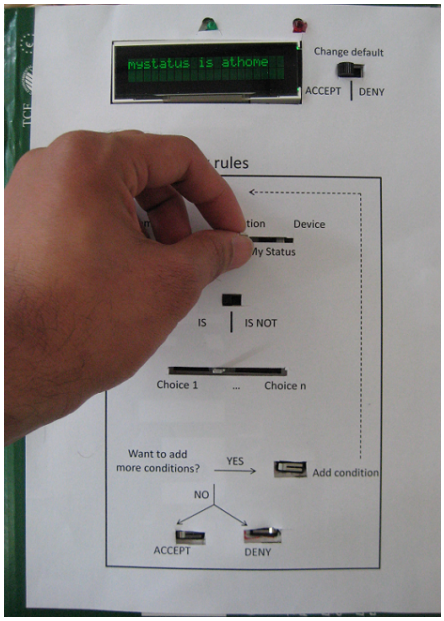


Figure 6.26: Tangible Interface with Awareness state selected

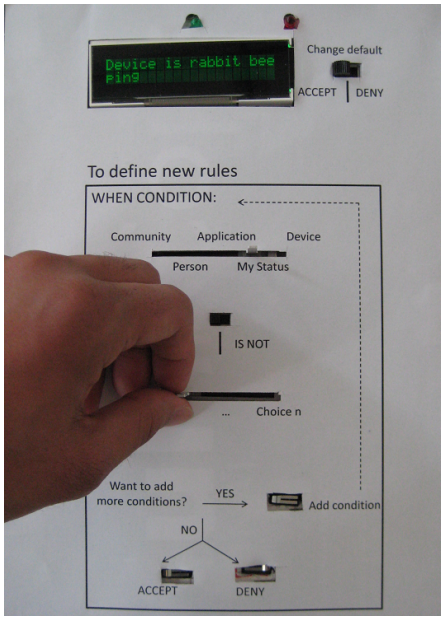


Figure 6.27: Tangible Interface with Device selected

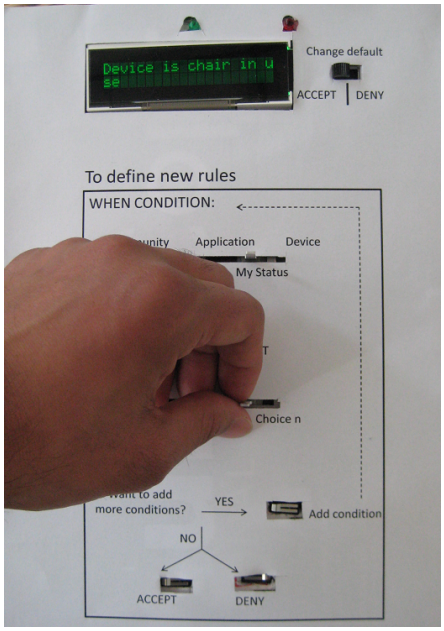


Figure 6.28: Tangible Interface with Device Chair selected

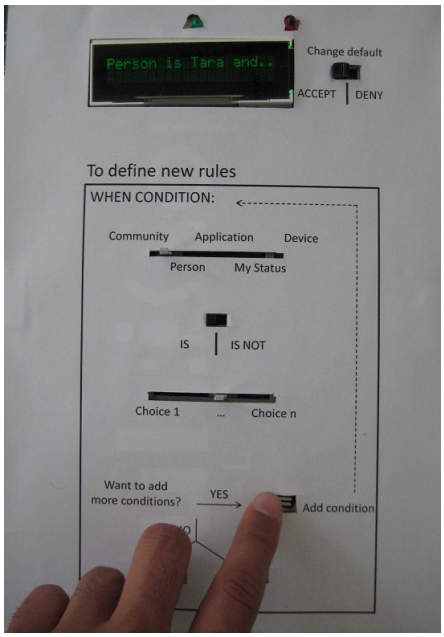


Figure 6.29: Tangible Interface showing making composite rule

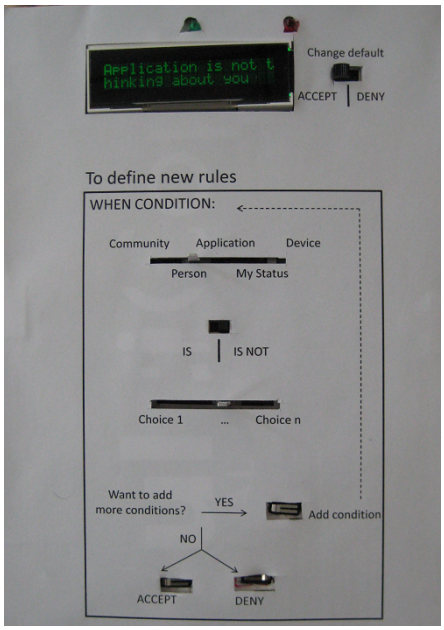


Figure 6.30: Tangible interface showing negation rule

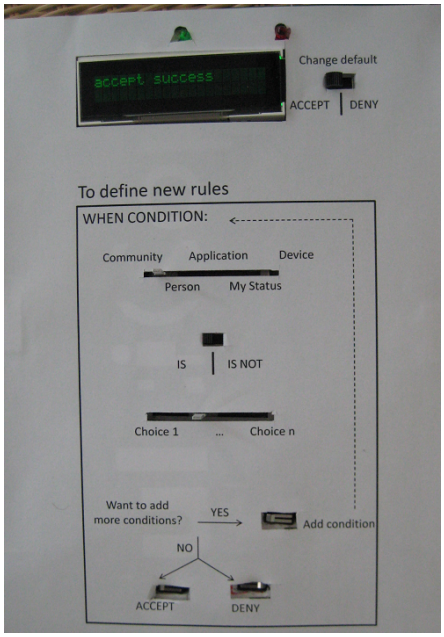


Figure 6.31: Tangible interface after rule accepted

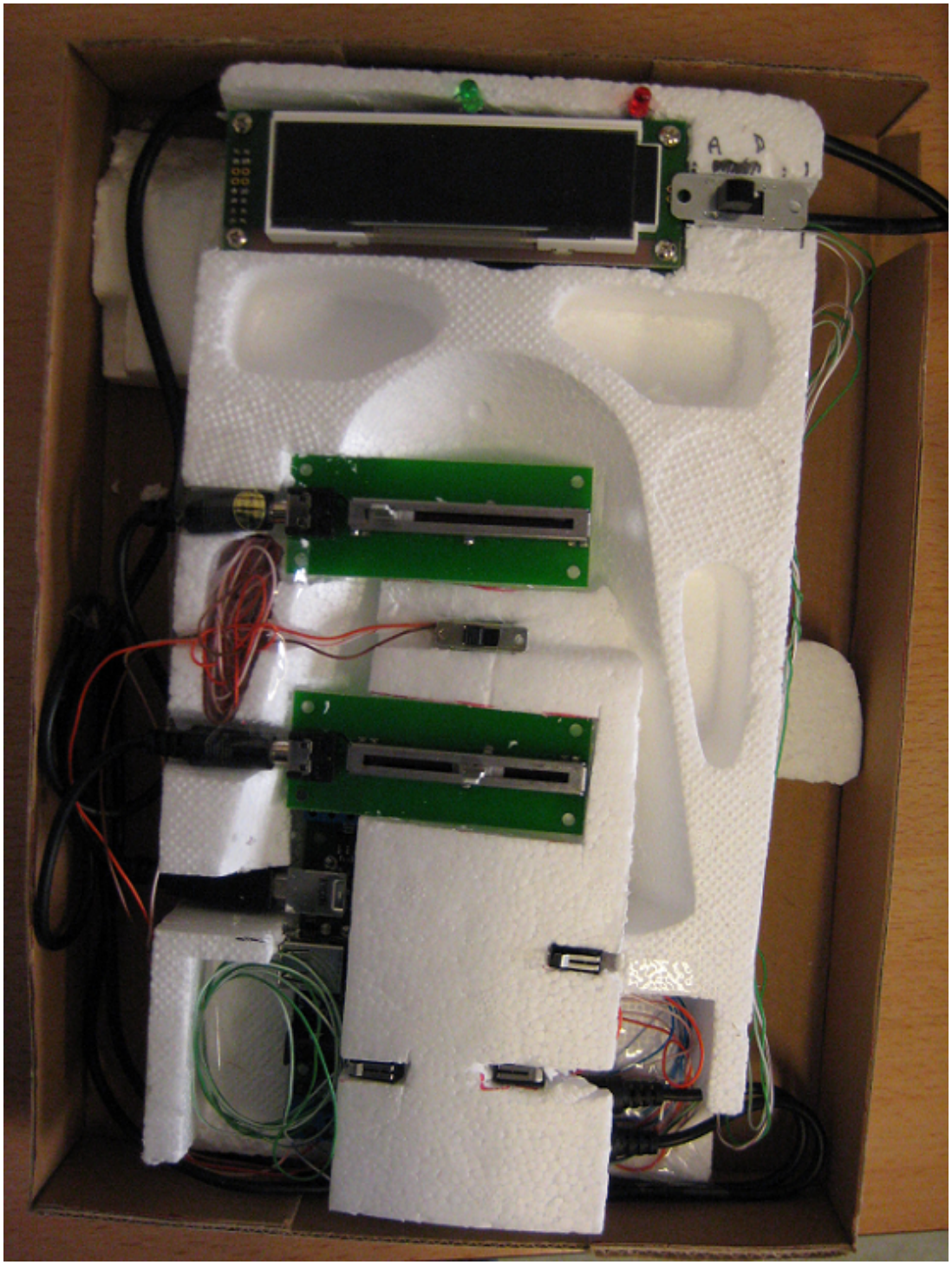


Figure 6.32: Tangible interface with inner component shown

Chapter 7

Evaluation

This chapter details on evaluation design, results and discussion.

7.1 Evaluation Design

The implemented tangible interface need to be evaluated with participants. As tangible interface is similar to the web interface in functionality, they can be compared to have feedback from the participants. The evaluation carried out in audio video recording lab setting, where tangible interface is recorded by camera and also the overall view with participant using tangible interface also recorded to see the perception and mood of the participant. Web interface is also captured with screen capture and the overall view with participant using web interface also get captured to see the mood of participant. Lab setting is depicted in figure 7.1. These settings is so as to record participant's expression on how they feel and on the interfaces to see whats going on to the interface.

It is desired to record participants information so that to know how much they use the technology to be in touch with family and friends. The participants information is collected with the questionare A-1.3.

Participants in pair will be involved in each session of evaluation. As we have 10 participants making involvement of 2 participant in single session, there will be 5 evaluation session. Each evaluation session lasts for 45 minutes. One evaluation session consists of introducing of ASTRA System and role of interruption management (appendix A-1.1); and short power-point demonstration (appendix A-1.2) to introduce the web and tangible interface. Time slot for this section was 15 minutes. Then the participant get the tasklist to do with interfaces. The task have been divided such that participant get involve in both interfaces. The tasks are divided in 3 sections a) Task A b) Task B c)Scenario. One participant starts with one interface and another participant with another interface for the first task. they get 7 minutes time for first task and they switched the interface for 2nd task and continue the same interface for the scenario task. The tasks are made so that to see whether they can make new rule, composite rule with anding, negation and choice on different entities, "accept" or "deny" rule ; to change the default setting for unspecified rules. The task for the participant starting with tangible interface is shown in Appendix A-1.4

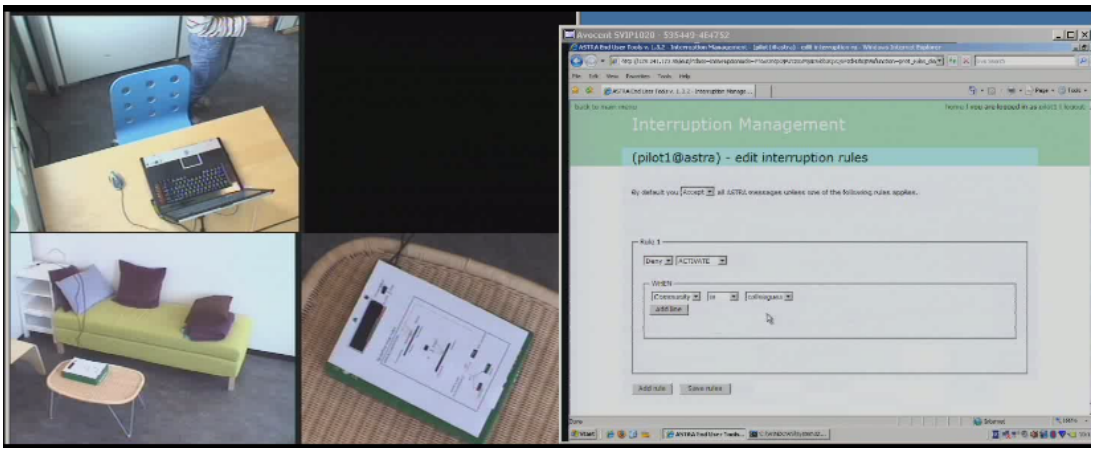


Figure 7.1: Monitoring setting in evaluation lab

and for the participant starting with web interface is shown in Appendix A-1.5. They then did scenario task, which was a bit higher level task extracting rule and rule conditions from scenario. This scenario is to check the participants understanding on real situation. Task A and Task B are more usability of the interfaces and scenario task is more usability of overall rule based system, to see whether it is easy for the user to extract rule from scenario or not. The expected solution is presented on appendix A-1.8. Participants get form to fill on evaluation of the interfaces, where we ask them to write easiest and most difficult task in evaluation, usability of the interface and comparison of the interfaces in terms of intuitive, easy to use, overview, control and fun. It is tried to extract general perception on interruption management and rule based system as seen in appendix A-1.6.

7.2 Results

Figure 7.2 shows participants involvement on evaluation day.

Figure 7.3 is combined result for participant information extracted from the form given to them.

The rule made by participants to the given tasks get extracted from database and analyzed how they are and compared to the expected solution. This log per participant can be seen in appendix A-1.9. Figure 7.4 combines easy and difficult problem written by each participants collected from the task list.

Figure 7.5 compares the hand written answers for some tasks.

Figure 7.6 compares the results of post evaluation.

Figure 7.7 compares the post evaluation results for comparison of web interface and tangible interface. The number means participants id who prefer the interface.

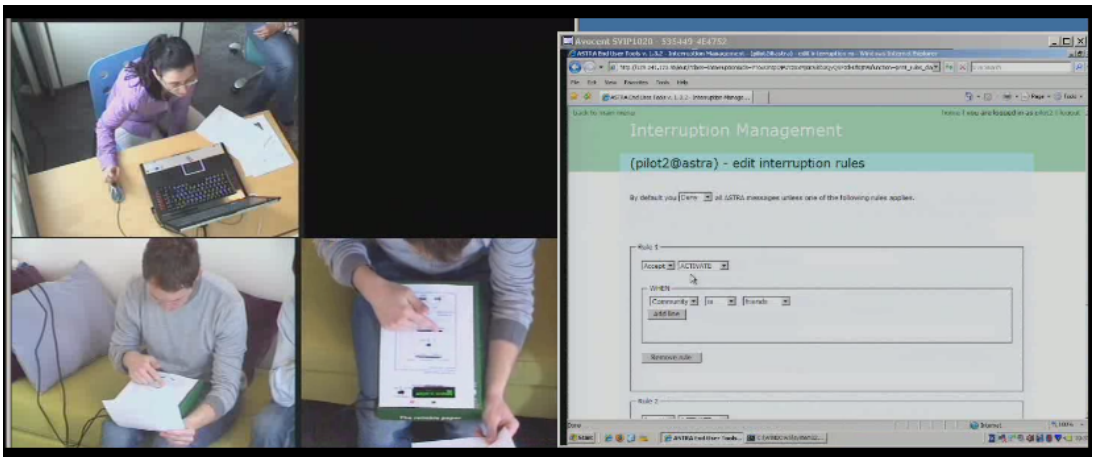


Figure 7.2: Participants evaluating the interfaces

Participants	Questionaire / Participant Information												
	1a	1b	2a	2b	3a	3b	4a	4b	5a	5b	6a	6b	never =1
p1	2	2	2	3	3	3	3	3	1	2	3	1	sometimes=2
p2	2	3	3	1	3	3	3	3	1	2	3	1	often=3
p3	2	3	3	3	3	3	3	3	3	3	3	3	no answer=0
p4	2	0	3	0	3	0	3	0	3	0	2	0	
p5	2	3	3	3	3	3	3	0	2	3	3	2	
p6	3	3	3	0	1	1	3	3	3	3	3	2	
p7	2	2	3	2	3	2	3	3	2	2	3	1	
p8	1	2	2	2	2	3	3	3	2	3	3	1	
p9	2	1	2	2	2	1	1	3	3	1	2	1	
p10	2	2	2	2	2	2	2	3	3	3	3	3	
Question set:													
1) Do you use SMS/MMS to keep in touch with:													
	a) Your home country?												
	b) People in Trondheim?												
2) Do you use social networking software (e.g.facebook, myspace, etc.) to keep in touch with:													
	a) Your home country?												
	b) People in Trondheim?												
3) Do you use instant messaging software (e.g.MSN, ICQ, etc.) to keep in touch with:													
	a) Your home country?												
	b) People in Trondheim?												
4) Do you use Email to keep in touch with:													
	a) Your home country?												
	b) People in Trondheim?												
5) Do you use the telephone to keep in touch with:													
	a) Your home country?												
	b) People in Trondheim?												
6) Do you use IP telephony software (e.g. Skype, volP Stunt, etc) to keep in touch with:													
	a) Your home country?												
	b) People in Trondheim?												

Figure 7.3: Participants information

Participant		***E : The easiest task in the evaluation***
		D : The most difficult task in the evaluation
P1	E	Choose the options by using the slide in the tangible
	D	difference rule vs.rule conditions/ logic operators among rules
P2	E	task list B
	D	To think about my status when I want to change to a new rule. Scenario 3
p3	E	task a) web interface
	D	task b) 5 composite rule on tangible interface
P4	E	Selecting options and just click to add rules
	D	sometimes to add composite rules
p5	E	default configuration and change it
	D	more difficult with more conditions
p6	E	setting and understanding the rules
	D	new rules vs. additional conditions
p7	E	combine conditions
	D	to understand at the first contact
p8	E	deny access from family and colleagues
	D	starting changing parameters in the tangible
p9	E	to add simple rules on the tangible
	D	define complex rules on computer
P10	E	all ok
	D	in tangible lack of feedbacks, especially for complex rules -
		it only says rule saved, but not what rule

Figure 7.4: Easy and difficult problem for individual participant

Participants	startswith	task0	sc1	sc3						
p1	Web	G	G/B	B						
p2	Tangible	B	G/B	B						
p3	Web	G	G/B	B						
p4	Tangible	~~G~~	G/~~G~~	B	**meaning**					
p5	Web	G	~~B/G~~	B	B	Block	The system will block the message			
p6	Tangible	B	G/B	B	G	Get	I will get the message			
p7	Web	~~B~~	G/B	B						
p8	Tangible	B	~~B/G~~	B						
p9	Web	G	~~B/G~~	B						
p10	Tangible	B	G/B	B	**Solution**					
Questions from Task and scenario					**startswith**	**task0**	**sc1**	**sc3**		
task0					**Tangible**	**B**	**G/B**	**B**		
	Guess what?				**Web**	**G**	**G/B**	**B**		
	One of your ASTRA contacts is sending you some awareness information.									
	By looking at your interface, what do you think the system will do? Cross the "right" answer									
	G	B	I have no idea							
Scenario 1										
	You have defined the following rules to manage interruptions									
	•	When the message is not from the community family then accept								
	A friend is sending you a message, what do you think it will happen?									
	G	B	I have no idea							
	Someone of your family is sending you a message, what do you think it will happen?									
	G	B	I have no idea							
Scenario 3										
	14:00 – Someone in your family is trying to contact you.									
	What do you think the system will do?									
	G	B	I have no idea							

Figure 7.5: hand written answers for some tasks

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
p1	3	4	3	4	4	3	4	2	3	3
p2	1	0	3	3	3	0	2	3	3	1
p3	3	1	4	4	4	1	4	4	4	4
p4	3	3	3	3	4	3	2	3	3	4
p5	3	3	4	3	3	2	4	4	3	3
p6	4	4	3	4	4	3	4	3	3	4
p7	3	3	3	4	3	3	4	4	4	3
p8	3	1	4	4	4	3	4	4	4	3
p9	0	3	1	2	1	1	1	1	3	4
p10	3	3	4	4	4	4	4	4	4	4
Evaluation of the interfaces										
Q1 : On the overall, the web interface is easy to use						**0=Disagree strongly**				
Q2 : On the overall, the tangible interface is easy to use						**1=Disagree**				
Q3 : I understand how rules combine						**2=No opinion**				
						3=Agree				
Managing Interruptions						**4=Agree strongly**				
Q4 : The ability to manage interruptions is an important feature of any system to keep in touch:										
Q5 : I would be able to define rules to protect me from unwanted interruptions										
Q6 : It would be easy to understand the behavior of the system, even with a large number of rules										
Q7 : It is important for me to define rules specifying when I want to ACCEPT messages:										
Q8 : It is important for me to define rules specifying when I want to BLOCK messages:										
Q9 : It is important for me to define rules depending on the incoming messages (e.g. the sender):										
Q10 : It is important for me to define rules depending on my context (e.g. my status):										

Figure 7.6: Post evaluation from participants

Mark which of the interfaces you think		
	Web Interface	Tangible interface
is more intuitive	1236810	4579
is easier to use	234810	15679
provides a better overview	1234678910	5
gives user more control	123458910	67
is more fun to use	23	145678910
The numbers are participant number selecting this option.		

Figure 7.7: Comparison of web interface and tangible interface

7.3 Discussion

One participant completely misunderstood the tangible interface. She said she will not do default allow. She will make default deny and make explicit rules as accept one. This lead to different answer than expected, as depicted in Figure 7.8. she had done correctly in web interface, as the default setting was expected "deny" in web interface.

Anding first, later changing to correct solution : Some participants get confused by question whether new rule or new condition for same rule to apply. First most participants who have started from web interface tried to 'and' all rules so as adding conditions on the same rule to make composite rule. When they reach to the task where they are asked to make composite rule, they realize and reverted the previously done task. This may be because they thought first composite situation and later realized. But in tangible, no evident was found like that. While changing after they commit mistake, one participant (P1) change this with "OR". another participant (p4) realize on composite case but didn't change the old one, another participant (p7) even could not notice at all. This is depicted in figure 7.9. Most of the rest users had made composite first with 'and' and later change to new rule. This is also clear from participants comments on most easy and most difficult task as shown in figure 7.4. P1 writes difficult task as "difference rule vs. rule conditions / logic operators among rules". P6 writes as "new rules vs. additional conditions". P4 comment on difficult task as "sometimes to add composite rules" and P5 comment on difficult task as "more difficult with more conditions" clearly states that they are even aware of the growing challenge for expanding conditions.

Lack of feedback: The same slider is used to show different entities. The number of choices are different according to the entity type. Community could have 5 choices, while application could have 15 choices. So for user it is difficult to understand how fast to move the slider to change the choices. Sometimes they slide back n forth to find the choices. It also means lack of feedback or overview of the choices. In web interface of course the user can see all choices at once and select the right one directly. This is supported by participant comments on difficult task as depicted in figure 7.4. P8 writes difficult task as "starting changing parameters in the tangible" and p10 writes difficult task as "in tangible lack of feedbacks, especially for complex rules - it

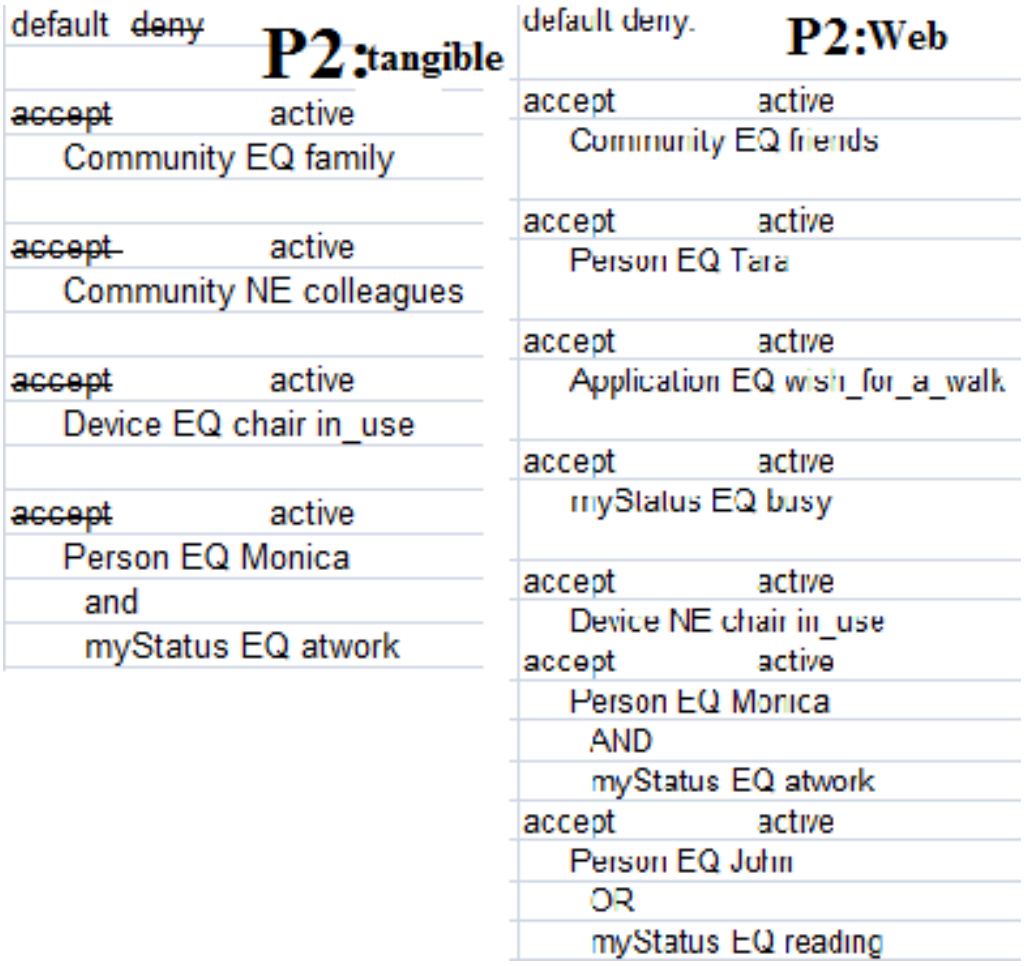


Figure 7.8: Misunderstood the task

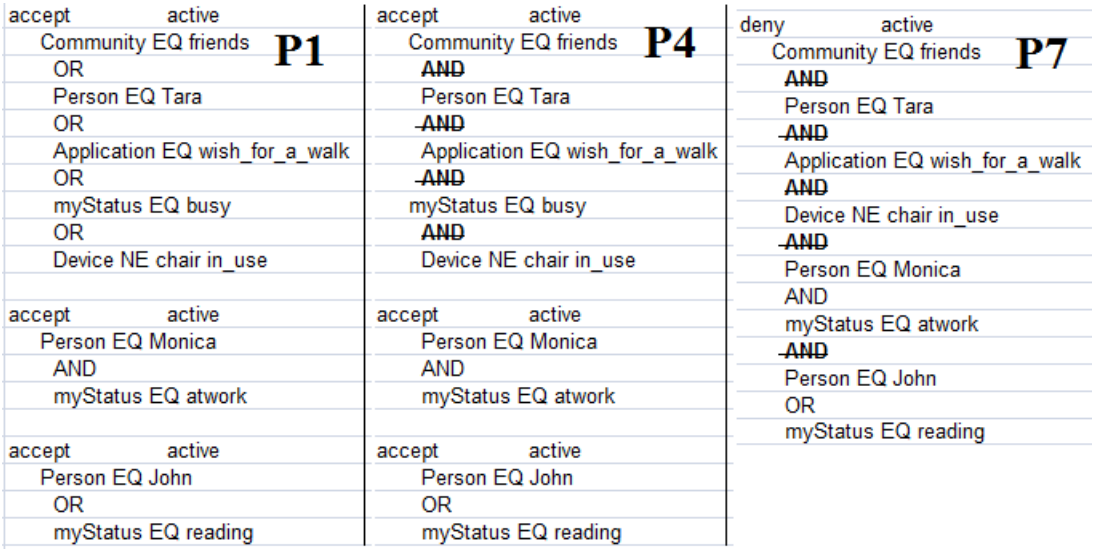


Figure 7.9: Different user understanding

only says rule saved, but not what rule". But some other participants P1, P4 found easy choosing options in tangible. P6 found easy on setting and understanding the rules. P7 found easy on combining conditions. P9 found easy to add simple rules on the tangible. One user mistakenly pushed "and" button but wanted to correct that, but that was not possible in tangible interface. It means they want to have more control as there could be human error and later they realize and want to correct.

Time response and multiple pressing: The tangible interface took some time while rule accepted or denied, this is because it need to write rule to the ASTRA system, and the system was responding only after the success or failure but not showing information like "processing" after the button for "accept" or "deny" pressed. So this caused serious problem on user, they thought either the button is not working or they haven't pressed properly, since they expected fast response. This is clearly seen even in rule. The same rule appeared multiple times with multiple pressing. But this could be even serious while making composite rule and multiple times pressing. The last rule from the composite rule could be made as new rule again. This is depicted in figure 7.10. This poses need of some modification to the system like to show transient state like "processing" while acting on the some buttons not to annoy the users and may be not to take other input while on this state. The system is currently checking just not to duplicate atomic rule inside composite rule. This evaluation study poses more advance rule checking like old rule check in database not to have duplication. This could be problem even in web based interface. Though the user can see all the rules at once in web interface, it has good overview and feedback, but still may need alert mechanism not to push duplicate rule. From system view, it is ok to have duplicate rule since they mean same.

The tasks from all participants are compared, and figure 7.11 tables the participants tasks performance, showing they did right or wrong. We can see two users made the composite rule in scenario 3. one made it one correct. But other didn't end up with composite rule. This is depicted

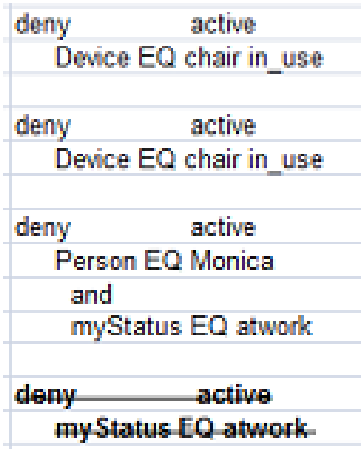


Figure 7.10: Case when buttons are clicked multiple times

in figure 7.12. One user made even richer composite rule with status busy. The participants have successfully composited rules in task1 and task2 with both web and tangible interfaces. But all didn't realized the composition, this means it is not problem with interface usability but there is problem converting the scenario to the rule system. People don't easily grasp every scenario element as rule and forget to mention that. It means technical competency is not sufficient, and there is need of human study to understand how people think.

The post evaluation analysis from the participants as depicted in figure 7.6 and evaluating the task done by participants as depicted in figure 7.11 makes some points clear. Most of participants agree on usability of both interfaces and this is proved by their job on task. Most of them manage to use the interfaces to write proper rules. Most of participants agree on they understand how rules do combine. But the task evaluation don't reflect the same. Only two people get correct answer for the scenario 3 where they are expected to have composite rule. One participant made 1 right answer among 2 needed. It means people are doing things wrong but still think they understand how rules do combine. This could have serious consequences. This means there is need of some sort of feedback to alert people on what actually going on and to show is it the way they expected, just to correct their measure. At the beginning people don't understand how they combine, but when they do , because of lack of feedback, they simply cannot revive in tangible. In web interface, they can think and see other rules while proceeding with new rules and can revert the rules easily. But in tangible interface user can't see earlier rules, so they might think what they have done is correct. When they get opportunity to see it again while doing other rules or if it had overview of whats going on, they will correct the rules, if necessary. The other questions related to managing interruptions sets backing for usefulness of the whole system, whether people prefer the system for their interruption management or not. Participant agree on importance of interruption management in keep in touch systems, and defining rules for interruption management. But they are quite aware of growing challenge for the growing rules and the system behavior would be not so simple. Participants are aware of both use of "accept" and "deny" as default settings and have rules for the opposite. They agree on need of sender information and recipient context information to know either to accept or deny. Since the system

Task	sub task	Expected solution	Participants Starting task on web interface and ending on tangible interface with 2nd task and scenario.					Participants starting Task on tangible interface and ending on web interface with 2nd task and scenario.				
			P1	P3	P5	P7	P9	P2	P4	P6	P8	P10
Task on Web Interface	1	default deny	√	√	√	√	√	√	√	√	√	√
	2	accept active Community EQ friends	√OR	√	√	√and	√	√	√and	√	√	√
	3	accept active Person EQ Tara	√OR	√	√	√and	√	√	√and	√	√	√
	4	accept active Application EQ wish_for_a_walk	√OR	√	√	√and	√	√	√and	×	√	×
	5	accept active myStatus EQ busy	√OR	√	√	-	√	√	√and	-	√	√
	6	accept active Device NE chair in_use	√OR	√	×	√and	√	√	√and	-	-	√
	7	accept active Person EQ Monica AND myStatus EQ atwork	√	√	√	√and	-	√	√	-	-	√
	8	accept active Person EQ John OR myStatus EQ reading	√	√	-	√and	Half part	√	√	-	-	√
Task on Tangible interface	1	default accept	√	√	√	√	√	×	√	√	√	√
	2	deny active Community EQ family	√	√	-	√	√	√acc	√	√	√	√
	3	deny active Community NE colleagues	√	√	√	√	√	√acc	√	√	×	√
	4	deny active Device EQ chair in_use	√	√	√	√	√	√acc	-	×	√	√
	5	deny active Person EQ Monica And myStatus EQ atwork	√	√	√	√	√	√acc	√	×	√	√
Scenario 3	1	deny active Community EQ colleagues AND myStatus EQ athome	NC	⊗	NC	NC	NC	-	√	√*	NC	√
	2	deny active Community EQ family AND myStatus EQ atwork	NC	⊗	NC	NC	NC	NC	NC	√*	NC	√

- Not Available
⊗ Missed because of system error
NC No composite rule
√ Right answer
√OR Use of OR instead of new rule
√and Use of AND instead of new rule
√* Even richer with AND MyStatus busy
√acc accepted instead of deny.

Figure 7.11: Participants task correctness

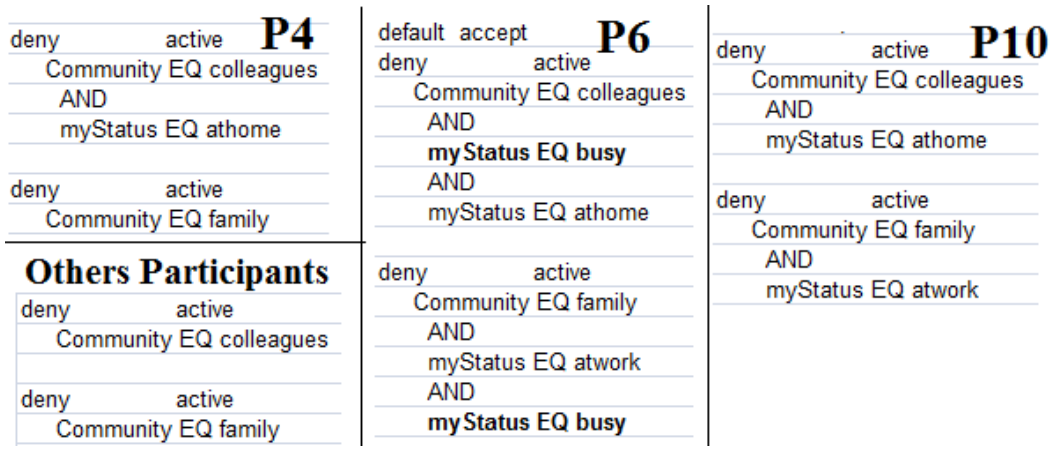


Figure 7.12: Different cases for Composite rule in Scenario 3

have been made in the same context, it gives positive sign for development of such system but equally pose problem for growing number of rules.

The evaluation comparing to the web interface and tangible interface with respect to intuitive, easiness, better overview, control and fun is shown in figure 7.7. This shows tangible interface is definitely fun to use. 5 out of 10 participants vote to tangible interface for easier to use is also good. 4 out of 10 participants vote to tangible interface for intuitive is also good result.

Participants overall understanding of the system and the interface response can be analyzed from the answer presented to the task as compared in the figure 7.5. P7 participant made it wrong in one interface but correct in another, so it is difficult to be in any conclusion for him. Sixty percent of the participants made it correctly.

7.4 Conclusion

The evaluation and discussion clearly gives positive result on usability of the tangible interface and the overall rule based interruption management system. This also paves many rooms for improvements. Technical competency is not only the solution and there is need of human study to catch some points on how they think in real situation. People want to have more control on managing and reviewing the rules, and to correct the old rules. People will later realize and want to revive. So there is quite room for improvement with feedback system, overall view.

Chapter 8

Conclusions

This chapter concludes the research.

8.1 Summary

This research resulted design and development of different interaction mechanisms to support the definition of interruption rules in ASTRA system. Existing Interruption Manager is extended to have community support on interruption rules as well as suitable user interfaces has been developed. A limited evaluation of the proposed solutions get evaluated with users. Creating a tangible interaction device such as this for managing interruption rules turned out to be both possible and even successful. The development gained a lot from state of art study at an early stage. It was robust enough for the experiment. Rule based interruption management also seems feasible but as rule expands it could be hard to track to the user and need to have some support from the system. User evaluation favors the tangible interface design and paves the way for improvement like feedback support, control and overall view.

8.2 Contributions

This thesis contributed on the field of interruption management in pervasive awareness system. This gives proof of concept for rule based interruption management. This resulted to core interruption management module and both web interface and tangible interface as end user tools to define the rules. User evaluation tasks, forms, logs and videos are useful for usability of the interface and overall system. The discussion from the user evaluation after the successful product is hoped to be helpful for future research purpose. And the novel extensible design for the implementation of tangible interface could be useful in similar domain.

8.3 Evaluation of the thesis

Thesis is evaluated as successful as it is achieved as targeted and user evaluation results are also encouraging for usability of the system. User evaluation results pave for the further development and research.

8.4 Future Work

The evaluation and discussion of the system paves many rooms for improvements. As people did correctly to the defined rules but were poor on extracting rules from the scenario, technical competency is not only the solution and there is need of human study to catch some points on how they think in real situation. People want to have more control on managing and reviewing the rules, and to correct the old rules. People will later realize and want to revive. So there is quite room for improvement with feedback system, overall view and more control. Initially, it was thought as to have tangible interface just to make rule but not to have complex managing properties like reviewing and deletion. It was thought like to use tangible interface for making rule and using web interface for modification, deletion and overall view but after evaluation, It is clear that user wants to have more control on tangible interface itself and don't want to have dependency to other interface. Future work could be on extending the system with more control option, feedback support and managing rules options.

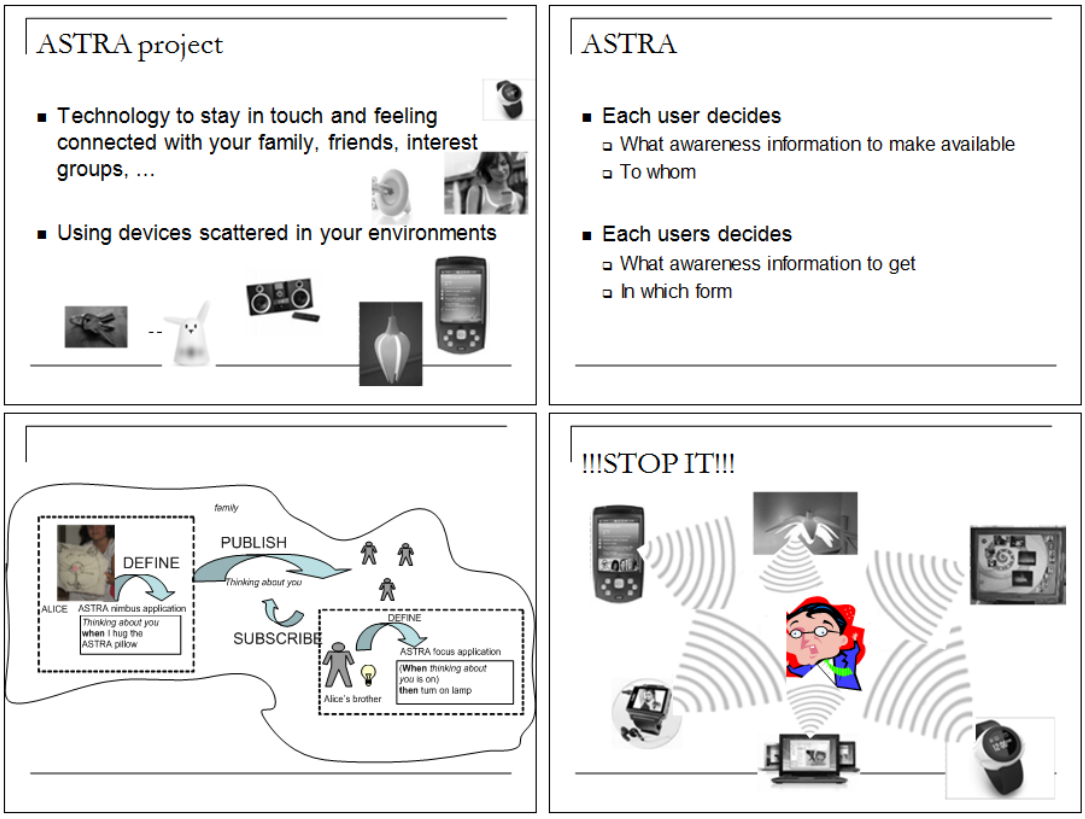
Appendices

A-1 Evaluation Set

Here are the different presentation and demo done to the participants. And the results data from task given to the participants.

A-1.1 ASTRA introduction : Presentation Slides

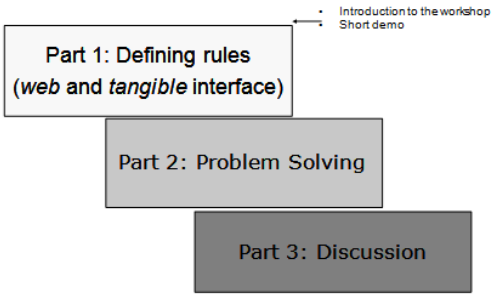
This is slides that have been presented to the participant as introduction to the ASTRA and the background information, before giving task of evaluation to them.



Objective of today's evaluation

- Interruptions in ASTRA are managed by defining rules to allow or block incoming messages
- We ask you to evaluate the two tools that have been developed
  - Web based interface
  - Tangible interface

Organization of the workshop



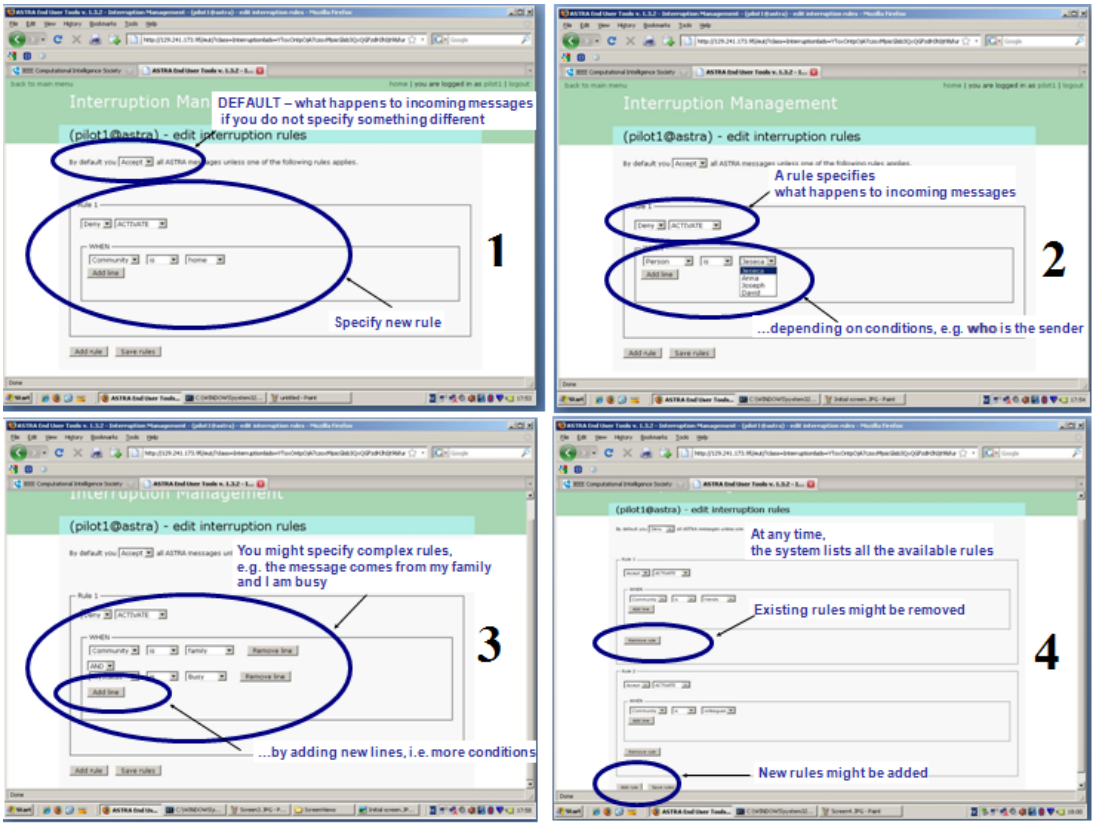
Working rules

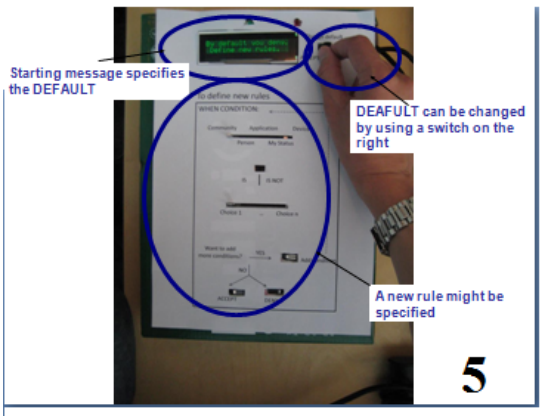
- No wrong answers, just *your answers*
- Think Freely
- Talk Freely
- Have fun!
- What is happening in the room is recorded, but it will be used in anonymous form

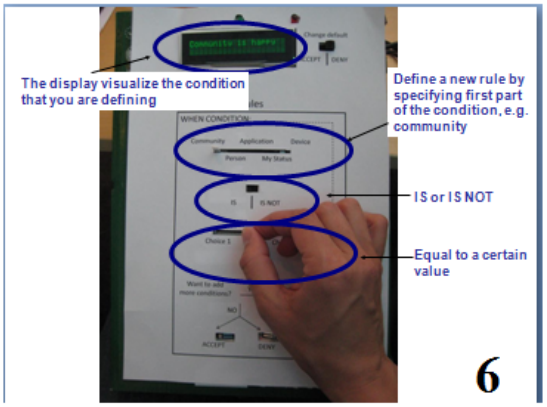
Thank you for your help!

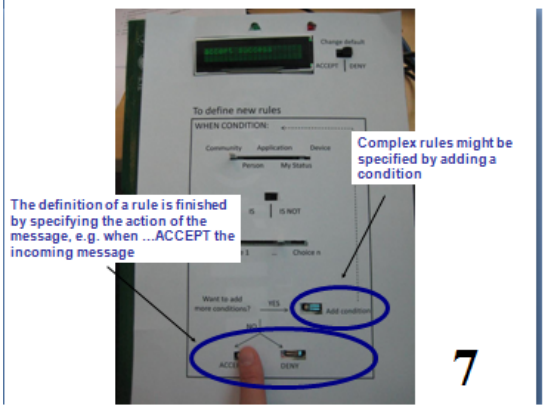
A-1.2 Demo

This is slides that have been presented to the participant as demonstration of the interfaces.




Starting message specifies the DEFAULT
DEAFULT can be changed by using a switch on the right
A new rule might be specified
5


The display visualize the condition that you are defining
Define a new rule by specifying first part of the condition, e.g. community
IS or IS NOT
Equal to a certain value
6


Complex rules might be specified by adding a condition
The definition of a rule is finished by specifying the action of the message, e.g. when ...ACCEPT the incoming message
7

A-1.3 questionnaire-info-participants

This is questionnaire for gaining information about participant.

Participants Information

Evaluation of Interruption Management
Spring 2009

!!!Thanks for you time!!!

Purpose
The purpose of this questionnaire is to get some general information about you. The information that you are asked to fill in is very general. Let us know if you have any concern regarding privacy.

Name: _____
Age: _____ ☐Female ☐Male

Please circle the appropriate options. More than one option can be selected

Do you use **SMS/MMS** to keep in touch with:

Your home country?	☐Not at all	☐ sometimes	☐ very often
People in Trondheim?	☐Not at all	☐ sometimes	☐ very often

Do you use **social networking software** (e.g.facebook, MySpace, etc.) to keep in touch with:

Your home country?	☐Not at all	☐ sometimes	☐ very often
People in Trondheim?	☐Not at all	☐ sometimes	☐ very often

Do you use **instant messaging software** (e.g.MSN, ICQ, etc.) to keep in touch with:

Your home country?	☐Not at all	☐ sometimes	☐ very often
People in Trondheim?	☐Not at all	☐ sometimes	☐ very often

Do you use **Email** to keep in touch with:

Your home country?	☐Not at all	☐ sometimes	☐ very often
People in Trondheim?	☐Not at all	☐ sometimes	☐ very often

Do you use **the telephone** to keep in touch with:

Your home country?	☐Not at all	☐ sometimes	☐ very often
People in Trondheim?	☐Not at all	☐ sometimes	☐ very often

Do you use **IP telephony software** (e.g. Skype, VoIP Stunt, etc) to keep in touch with:

Your home country?	☐Not at all	☐ sometimes	☐ very often
People in Trondheim?	☐Not at all	☐ sometimes	☐ very often

A-1.4 task_list_tangible

This is task given to the participant starting with tangible interface.

Guess what?

One of your ASTRA contacts is sending you some awareness information.

By looking at your interface, what do you think the system will do? *Cross the "right" answer*

I will get the message	The system will block the message	I have no idea

Task list

a) Use your interface to perform the following tasks

1. Change the default setting (so that is set to ACCEPT)
2. When the message is from community *family* then deny [the delivery of the message]
3. When the message is not from community *colleagues* then deny [the delivery of the message]
4. When the device is *chair* *in use* then deny
5. When the message is from *Monica* and I am *at work* then deny

b) Change the tool you are using and perform as many tasks as possible from the following list

1. Change the default setting (so that is set to DENY)
2. When the message is from the community *friends* then accept [the delivery of the message]
3. When the message is from person *Tara* then accept
4. When the message is from the application *wish for a walk* then accept
5. When my status is *busy* then accept
6. When the device *chair* is not *in use* then accept
7. When the message is from *Monica* and I am *at work* then accept
8. When the message is from *John* OR I am *reading* then accept

Scenarios

Scenario 1

You have defined the following rules to manage interruptions

- When the message is not from the community *family* then accept

A friend is sending you a message, what do you think it will happen?

I will get the message	The system will block the message	I have no idea

Someone of your family is sending you a message, what do you think it will happen?

I will get the message	The system will block the message	I have no idea

Scenario 2

You are at home having a nice evening with your closest friend and the ASTRA system is active in the background. You are accepting all the incoming messages.

TASK: You are getting tired of all the beeps and lights. You need to stop all the messages from ASTRA.

Scenario 3

09:00 - you are at home and you are having a busy morning with the carpenters working on your new bathroom. While you are at home, you do not want to receive any message from your colleagues

TASK: Use the interface to define the appropriate rules

12:00 - you are finally at work and you have some pressing deadlines. You do not want to receive any messages from your family

TASK: Use the interface to define the appropriate rules

14:00 – Someone in your family is trying to contact you.

What do you think the system will do?

I will get the message	The system will block the message	I have no idea

A-1.5 task_list_web

This is task given to the participant starting with web interface.

Guess what?

One of your ASTRA contacts is sending you some awareness information.

By looking at your interface, what do you think the system will do? *Cross the "right" answer*

I will get the message	The system will block the message	I have no idea

Task list

a) Use your interface to perform the following tasks

1. Change the default setting (so that it is set to DENY)
2. When the message is from the community *friends* then accept [the delivery of the message]
3. When the message is from person *Tara* then accept
4. When the message is from the application *wish for a walk* then accept
5. When my status is *busy* then accept
6. When the device *chair* is not *in use* then accept
7. When the message is from *Monica* and I am *at work* then accept
8. When the message is from *John* OR I am *reading* then accept

b) Change the tool you are using and perform as many tasks as possible from the following list

1. Change the default setting (so that is set to ACCEPT)
2. When the message is from community *family* then deny [the delivery of the message]
3. When the message is not from community *colleagues* then deny [the delivery of the message]
4. When the device is *chair* *in use* then deny
5. When the message is from *Monica* and I am *at work* then deny

Scenarios

Scenario 1

You have defined the following rules to manage interruptions

- When the message is not from the community *family* then accept

A friend is sending you a message, what do you think it will happen?

I will get the message	The system will block the message	I have no idea

Someone of your family is sending you a message, what do you think it will happen?

I will get the message	The system will block the message	I have no idea

Scenario 2

You are at home having a nice evening with your closest friend and the ASTRA system is active in the background. You are accepting all the incoming messages.

TASK: You are getting tired of all the beeps and lights. You need to stop all the messages from ASTRA.

Scenario 3

09:00 - you are at home and you are having a busy morning with the carpenters working on your new bathroom. While you are at home, you do not want to receive any message from your colleagues

TASK: Use the interface to define the appropriate rules

12:00 - you are finally at work and you have some pressing deadlines. You do not want to receive any messages from your family

TASK: Use the interface to define the appropriate rules

14:00 – Someone in your family is trying to contact you.

What do you think the system will do?

I will get the message	The system will block the message	I have no idea

A-1.6 post_evaluation form

These are post evaluation form, that the participant will fill up after finishing the tasks.

Your ID _____

Evaluation of the interfaces

The easiest task in the evaluation

_____

_____

The most difficult task in the evaluation

_____

_____

On the overall, the web interface is easy to use

Agree strongly	Agree	No opinion	Disagree	Disagree strongly

On the overall, the tangible interface is easy to use

Agree strongly	Agree	No opinion	Disagree	Disagree strongly

I understand how rules combine

Agree strongly	Agree	No opinion	Disagree	Disagree strongly

Mark which of the interfaces you think

	Web Interface	**Tangible interface**
is more intuitive		
is easier to use		
provides a better overview		
gives user more control		
is more fun to use		

Managing Interruptions

The ability to manage interruptions is an important feature of any system to keep in touch:

Agree strongly	Agree	No opinion	Disagree	Disagree strongly

I would be able to define rules to protect me from unwanted interruptions

Agree strongly	Agree	No opinion	Disagree	Disagree strongly

It would be easy to understand the behavior of the system, even with a large number of rules

Agree strongly	Agree	No opinion	Disagree	Disagree strongly

It is important for me to define rules specifying when I want to ACCEPT messages:

Agree strongly	Agree	No opinion	Disagree	Disagree strongly

It is important for me to define rules specifying when I want to BLOCK messages:

Agree strongly	Agree	No opinion	Disagree	Disagree strongly

It is important for me to define rules depending on the incoming messages (e.g. the sender):

Agree strongly	Agree	No opinion	Disagree	Disagree strongly

It is important for me to define rules depending on my context (e.g. my status):

Agree strongly	Agree	No opinion	Disagree	Disagree strongly

A-1.7 filled form for example

These are some forms that have been filled by participants. This is just for one participant.

This is form with information about participant.

Participants Information

Evaluation of Interruption Management
Spring 2009

!!!Thanks for you time!!!

Purpose
The purpose of this questionnaire is to get some general information about you. The information that you are asked to fill in is very general. Let us know if you have any concern regarding privacy.

Name: [redacted]
Age: [redacted] ☐ Female ☒ Male

Please circle the appropriate options. More than one option can be selected

Do you use **SMS/MMS** to keep in touch with:

Your home country?	☐ Not at all	☒ sometimes	☐ very often
People in Trondheim?	☐ Not at all	☒ sometimes	☐ very often

Do you use **social networking software** (e.g.facebook, myspace, etc.) to keep in touch with:

Your home country?	☐ Not at all	☒ sometimes	☐ very often
People in Trondheim?	☐ Not at all	☐ sometimes	☒ very often

Do you use **instant messaging software** (e.g.MSN, ICQ, etc.) to keep in touch with:

Your home country?	☐ Not at all	☐ sometimes	☒ very often
People in Trondheim?	☐ Not at all	☐ sometimes	☒ very often

Do you use **Email** to keep in touch with:

Your home country?	☐ Not at all	☐ sometimes	☒ very often
People in Trondheim?	☐ Not at all	☐ sometimes	☒ very often

Do you use **the telephone** to keep in touch with:

Your home country?	☒ Not at all	☐ sometimes	☐ very often
People in Trondheim?	☐ Not at all	☒ sometimes	☐ very often

Do you use **IP telephony software** (e.g. Skype, voIP Stunt, etc) to keep in touch with:

Your home country?	☐ Not at all	☐ sometimes	☒ very often
People in Trondheim?	☒ Not at all	☐ sometimes	☐ very often

This form is an example of filled post evaluation form.

Guess what?

One of your ASTRA contacts is sending you some awareness information.

By looking at your interface, what do you think the system will do? Cross the "right" answer

It will get the message	The system will block the message	I have no idea
x except if is a colleague		

Task list

a) Use your interface to perform the following tasks

1. Change the default setting (so that it is set to DENY)
2. When the message is from the community *friends* then accept (the delivery of the message)
3. When the message is from person *Tara* then accept
4. When the message is from the application *wish for a walk* then accept
5. When my status is *busy* then accept
6. When the device *chair* is not in use then accept
7. When the message is from *Monica* and I am *at work* then accept
8. When the message is from *John* OR I am *reading* then accept

b) Change the tool you are using and perform as many tasks as possible from the following list

1. Change the default setting: (so that is set to ACCEPT)
2. When the message is from community *family* then deny (the delivery of the
3. When the message is not from community *colleagues* then deny (the delivery of the message)
4. When the device is *chair in use* then deny
5. When the message is from *Monica* and I am *at work* then deny

Scenarios

Scenario 1

You have defined the following rules to manage interruptions

- When the message is not from the community *family* then accept

A friend is sending you a message, what do you think it will happen?

It will get the message	The system will block the message	I have no idea
X ok	* ok!	

Someone of your family is sending you a message, what do you think it will happen?

It will get the message	The system will block the message	I have no idea
	X	

Scenario 2

You are at home having a nice evening with your closest friend and the ASTRA system is active in the background. You are accepting all the incoming messages.

TASK: You are getting tired of all the beeps and lights. You need to stop all the messages from ASTRA. I CHANGE MY DEFAULT STATUS TO DENY

Scenario 3

09:00 - you are at home and you are having a busy morning with the carpenters working on your new bathroom. While you are at home, you do not want to receive any message from your colleagues

TASK: Use the interface to define the appropriate rules

12:00 - you are finally at work and you have some pressing deadlines. You do not want to receive any messages from your family

TASK: Use the interface to define the appropriate rules

14:00 – Someone in your family is trying to contact you.

What do you think the system will do?

It will get the message	The system will block the message	I have no idea
	X	

Evaluation of the interfaces *Your ID* [redacted]

The easiest task in the evaluation

Choose the options using the slide on the tangible interface

The most difficult task in the evaluation

Understand the differences between rules and rule conditions on the web interface. Understand the logic operations and the rules, AND or OR.

On the overall, the web interface is easy to use

Agree strongly	Agree	No opinion	Disagree	Disagree strongly
	X			

On the overall, the tangible interface is easy to use

Agree strongly	Agree	No opinion	Disagree	Disagree strongly
X				

I understand how rules combine

Agree strongly	Agree	No opinion	Disagree	Disagree strongly
	X			

Mark which of the interfaces you think

	Web Interface	Tangible interface
is more intuitive	X	
is easier to use		X
provides a better overview	X	
gives user more control	X	
is more fun to use		X

Managing Interruptions

The ability to manage interruptions is an important feature of any system to keep in touch:

Agree strongly	Agree	No opinion	Disagree	Disagree strongly
X				

I would be able to define rules to protect me from unwanted interruptions

Agree strongly	Agree	No opinion	Disagree	Disagree strongly
X				

It would be easy to understand the behavior of the system, even with a large number of rules

Agree strongly	Agree	No opinion	Disagree	Disagree strongly
	X			

It is important for me to define rules specifying when I want to ACCEPT messages:

Agree strongly	Agree	No opinion	Disagree	Disagree strongly
X				

It is important for me to define rules specifying when I want to BLOCK messages:

Agree strongly	Agree	No opinion	Disagree	Disagree strongly
		X		

It is important for me to define rules depending on the incoming messages (e.g. the sender):

Agree strongly	Agree	No opinion	Disagree	Disagree strongly
	X			

It is important for me to define rules depending on my context (e.g. my status):

Agree strongly	Agree	No opinion	Disagree	Disagree strongly
	X			

extra

A-1.8 Answer Set

This is answer we have expected from the tasks given to the user.

web interface task	Tangible interface task	scenario 1.	scenario 2	scenario 3
default deny.	default accept			
		a) Get		
accept active	deny active		change default to deny.	default accept
Community EQ friends	Community EQ family	b) Block		
accept active	deny active			.deny active
Person EQ Tara	Community NE colleagues			Community EQ colleagues
				AND
accept active	deny active			myStatus EQ athome
Application EQ wish_for_a_walk	Device EQ chair in_use			
accept active	deny active			.deny active
myStatus EQ busy	Person EQ Monica			Community EQ family
	and			AND
accept active	myStatus EQ atwork			myStatus EQ atwork
Device NE chair in_use				
accept active				
Person EQ Monica				. The system will block the message.
AND				
myStatus EQ atwork				
accept active				
Person EQ John				
OR				
myStatus EQ reading				

A-1.9 Log Extraction

These are individual answers from the participants for the provided tasks.

Participant 1

P1	startwith	task1 / Web Interface	task2 / tangible Interface	scenario / tangible interface
	Web interface	default deny	default accept	default accept
		accept active	deny active	deny active
		Community EQ friends	Community EQ family	Community EQ colleagues
		OR		
		Person EQ Tara	deny active	deny active
		OR	Community NE colleagues	Community EQ family
		Application EQ wish_for_a_walk		
		OR	deny active	
		myStatus EQ busy	Device EQ chair in_use	
		OR		
		Device NE chair in_use	deny active	
			Person EQ Monica	
		accept active	and	
		Person EQ Monica	myStatus EQ atwork	
		AND		
		myStatus EQ atwork		
		accept active		
		Person EQ John		
		OR		
		myStatus EQ reading		
Comment		**Ok**. Use of OR.	**OK**	**No composite rule**

Participant 2

P2	startwith	task1 / tangible interface	task2 / web interface	scenario / web interface
	Tangible interface	default ~~deny~~	default deny.	default accept.
		~~accept~~ active	accept active	deny active
		Community EQ family	Community EQ friends	Community EQ family
		~~accept~~ active	accept active	
		Community NE colleagues	Person EQ Tara	
		~~accept~~ active	accept active	
		Device EQ chair in_use	Application EQ wish_for_a_walk	
		~~accept~~ active	accept active	
		Person EQ Monica	myStatus EQ busy	
		and		
		myStatus EQ atwork	accept active	
			Device NE chair in_use	
			accept active	
			Person EQ Monica	
			AND	
			myStatus EQ atwork	
			accept active	
			Person EQ John	
			OR	
			myStatus EQ reading	
Comment		**should be default accept and have other rule denied.**	**OK**	**No composite rule Incomplete**

Participant 3

P3	startwith	task1 / web interface	task2 / tangible interface	scenario / tangible interface
	Web interface	default deny	default accept	
		accept active	deny active	
		Community EQ friends	Community EQ family	
		accept active	deny active	
		Person EQ Tara	Community NE colleagues	
		accept active	deny active	
		Application EQ wish_for_a_walk	Device EQ chair in_use	
		accept active	deny active	
		myStatus EQ busy	Person EQ Monica	
			and	
		accept active	myStatus EQ atwork	
		Device NE chair in_use		
		accept active		
		Person EQ Monica		
		AND		
		myStatus EQ atwork		
		accept active		
		Person EQ John		
		OR		
		myStatus EQ reading		
Comment		**OK**	**OK**	**missed due to system problem**

Participant 4

P4	startwith	task1 / tangible interface	task2 / web interface	scenario / tangible interface
	Tangible Interface	By default you accept		
			default deny	default accept
		deny active		
		Community EQ family	accept active	deny active
			Community EQ friends	Community EQ colleagues
		deny active	**AND**	AND
		Community NE colleagues	Person EQ Tara	myStatus EQ athome
			AND	
		deny active	Application EQ wish_for_a_walk	deny active
		Person EQ Monica	**AND**	Community EQ family
		and	myStatus EQ busy	
		myStatus EQ atwork	**AND**	
			Device NE chair in_use	
			accept active	
			Person EQ Monica	
			AND	
			myStatus EQ atwork	
			accept active	
			Person EQ John	
			OR	
			myStatus EQ reading	
Comment		**OK**	**Use of AND instead of new rule**	**One is ok**
		but missing in between		
		deny active		
		Device EQ chair in_use		

Participant 5

P5	startwith	task1 / web interface	task2 / tangible Interface	scenario / tangible interface
	Web interface			
		default deny	default accept	default accept
		accept active	deny active	deny active
		Community EQ friends	Community NE colleagues	Community EQ colleagues
		accept active	deny active	deny active
		Person EQ Tara	Community NE colleagues	Community EQ colleagues
		accept active	deny active	deny active
		Application EQ wish_for_a_walk	Device EQ chair in_use	Community EQ family
		accept active	deny active	deny active
		myStatus EQ busy	Device EQ chair in_use	Community EQ family
		accept active	deny active	
		Device ~~EQ~~ chair in_use	Person EQ Monica	
			and	
		accept active	myStatus EQ atwork	
		Person EQ Monica		
		AND	~~deny active~~	
		myStatus EQ atwork	~~myStatus EQ atwork~~	
Comment		**no negation for one rule**	**2nd task missing**	**No composite rule**
		missing last task	**last one is because of double pressing**	
			for last rule.	

Participant 6

P6	startwith	task1 / tangible interface	task2 /web Interface	scenario / web interface
	tangible interface			
		default accept	default deny	default accept
		~~deny active~~	accept active	deny active
		~~Community EQ colleagues~~	Community EQ friends	Community EQ colleagues
				AND
		deny active } x 4 times	accept active	**myStatus EQ busy**
		Community EQ family }	Person EQ Tara	AND
				myStatus EQ athome
		deny active } X 2	~~accept active~~	
		Community NE colleagues }	~~Application EQ play_guitar~~	deny active
				Community EQ family
		deny active		AND
		Community NE colleagues		myStatus EQ atwork
		~~and~~		AND
		Device EQ chair in_use		**myStatus EQ busy**
		~~and~~		
		Person EQ Monica		
		and		
		myStatus EQ atwork		
		~~deny active~~		
		~~myStatus EQ atwork~~		
Comments		**mistake, extra rule.**	**Participant first have composited all rule by**	**Ok composite rule**
		rule anded, participant wanted to correct	**anding but later when noticed wrong delete**	**even richer rule with**
		the mistake but wasnot possible from	**line and made it again like this. So run out**	**mystatus busy.**
		tangible interface.	**of time..**	
		last one because of multiple pressing.		

Participant 7

P7	startwith	task1 / web interface	task2 /tangible Interface	scenario / tangible interface
	web interface			
		default accept	default accept.	default deny
		deny active	deny active } X 2	accept active
		Community EQ friends	Community EQ family }	Community EQ colleagues
		~~AND~~		
		Person EQ Tara	deny active } X 2	accept active
		~~AND~~	Community NE colleagues }	Community EQ colleagues
		Application EQ wish_for_a_walk		
		~~AND~~	deny active } X2	accept active
		Device NE chair in_use	Device EQ chair in_use }	Community EQ family
		~~AND~~		
		Person EQ Monica	deny active	accept active
		AND	Person EQ Monica	Community EQ family
		myStatus EQ atwork	and	
		~~AND~~	myStatus EQ atwork	
		Person EQ John		
		OR	~~deny active~~	
		myStatus EQ reading	~~myStatus EQ atwork~~	
Comment		**Use of AND instead of new rule**	**last one is because of double pressin**	**No composite rule**
		missing my_status EQ busy	**for last rule.**	

Participant 8

P8	startwith	task1 / tangible interface	task2 /web Interface	scenario / web interface
	tangible interface			
		default accept	default deny	default accept
		deny active } X 2	accept active	deny active
		Community EQ family }	Community EQ friends	Community EQ colleagues
		deny active } X 2	accept active	deny active
		Community ~~EQ~~ colleagues }	Person EQ Tara	Community EQ family
		deny active } X 2	accept active	
		Device EQ chair in_use }	Application EQ wish_for_a_walk	
		deny active	accept active	
		Person EQ Monica	myStatus EQ busy	
		and		
		myStatus EQ atwork		
		~~deny active~~		
		~~myStatus EQ atwork~~		
Comments		**No Negation on one rule.**	**missing 6,7,8**	**no composite rules**
		last one is because of double pressing		
		for last rule.		

Participant 9

P9	startwith	task1 / web interface	task2 /tangible Interface	scenario / tangible interface
	web interface			
		default deny	default accept	default accept
		accept active Community EQ friends	deny active } X 6 times Community EQ family }	deny active } X 3 times Community EQ colleagues }
		accept active Person EQ Tara	deny active } X 3 times Community NE colleagues }	deny active } X 3 times Community EQ family }
		accept active Application EQ wish_for_a_walk	deny active } X 3 times Device EQ chair in_use }	
		accept active myStatus EQ busy	deny active Person EQ Monica and myStatus EQ atwork	
		accept active Device NE chair in_use		
		accept active Person EQ John	~~deny active } X 2 times myStatus EQ atwork }~~	
Comments		**7 task missing only first part for 8th task.**	**last one is because of double pressing for last rule.**	**No composite rule**

Participant 10

P10	startwith	task1 / tangible interface	task2 /web Interface	scenario / web interface
	tangible interface			
		default accept	default deny	default accept
		deny active } X 3 times Community EQ family }	accept active Community EQ friends	deny active Community EQ colleagues AND myStatus EQ athome
		deny active } X 3 times Community NE colleagues }	accept active Person EQ Tara	
		deny active } X 3 times Device EQ chair in_use }	accept active Application ~~NE~~ wish_for_a_walk	deny active Community EQ family AND myStatus EQ atwork
		deny active Person EQ Monica and myStatus EQ atwork	accept active myStatus EQ busy	
		~~deny active } X 2 myStatus EQ atwork }~~	accept active Device NE chair in_use	
			accept active Person EQ Monica AND myStatus EQ atwork	
			accept active Person EQ John OR myStatus EQ reading	
Comments		**last one is because of double pressing for last rule.**	**wrong negation**	**use of composite rule.**

Bibliography

[1] Joyce Ho and Stephen S. Intille. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. *CHI 2005*, 2005.

[2] Tara Nath Subedi. Interruption management in astra. Technical report, NTNU, 2008.

[3] H. Ishii and B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. *In Proceedings of CHI'97 pp. 234-241*, 1997.

[4] A. Smailagic D. Garlan, D. Siewiorek and P. Steenkiste. Project aura: toward distraction-free pervasive computing. *IEEE Pervasive Computing*, 2002.

[5] Miyata Y. and Norman D. Psychological issues in support of multiple activities. *User-centered System Design: New Perspectives on Human-Computer Interaction*, pages 265–284, 1986.

[6] http://www.dgp.toronto.edu/ml, March 2009.

[7] Chris Beckmann and Anind Dey. Siteview: Tangibly programming active environments with predictive visualization. Technical report, June 2003.

[8] S. Greenberg and C. Fitchett. Phidgets: Easy development of physical interfaces through physical widgets. *In Proceedings of UIST 2001 pp. 209-218*, 2001.

[9] Ana Paiva Rui Prada et al. Towards tangibility in gameplay: Building a tangible affective interface for a computer game. *ICMI*, 2003.

[10] A. Paiva G. Andersson. Designing an affective sympathetic interface to a computer game. *Personal and Ubiquitous Computing*, 2002.

[11] http://www.osgi.org/main/homepage, January 2009.

[12] http://www.knopflerfish.org/, March 2009.

[13] Anders Kofod-Petersen Sobah Abbas Petersen, Jorg Cassens and Monica Divitini. To be or not to be aware: Reducing interruptions in pervasive awareness systems.

[14] S. Benford A. Bullock N. Cook P. Harvey R. Ingram and O. K. Lee. From rooms to cyberspace: models of interaction in large virtual computer spaces. *Interacting with Computers*, 5(2):217–237, 1993.

[15] S. Harrison and P. Dourish. Re-place-ing space: The roles of place and space in collaborative system. *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative work*, pages 67–76, 1996.

[16] T. Rodden. Populating the application: A model of awareness for cooperative applications. *Proceedings of the 1996 ACM conference on Computer Supported Cooperative Work*, pages 87–96, 1996.

[17] G. Metaxas and P. Markopoulos. 'aware of what?' a formal model of awareness systems that extends the focus-nimbus model. *Procceedings of the IFIP conference EHCI 2007*, 2007.

[18] http://clipsrules.sourceforge.net/, March 2009.

[19] http://www.phidgets.com/, April 2009.

[20] http://www.upnp.org, June 2009.

[21] http://www.upnp.org/specs/arch/upnp-arch-devicearchitecture-v1.0.pdf, May 2009.

[22] http://www.phidgets.com/documentation/programming_manual.pdf, April 2009.

[23] http://www.cybergarage.org/, February 2009.

[24] http://www.w3.org/tr/wsdl, March 2009.

[25] https://jax-rpc.dev.java.net/, March 2009.

[26] http://ws.apache.org/axis/, March 2009.