End user programming of awareness systems : addressing cognitive and social challenges for interaction with aware environments

DOI:
10.6100/IR692128

Document status and date:
Published: 01/01/2010

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

End-User Programming of Awareness Systems

Addressing cognitive and social challenges for interaction with aware environments

G. Metaxas

Cover design: Olha Bondarenko & Georgios Metaxas

A catalogue record is available from the Eindhoven University of Technology Library

ISBN: 978-90-386-2382-5

End User Programming of Awareness Systems
Addressing cognitive and social challenges for interaction with aware environments

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op maandag 6 december 2010 om 16.00 uur

door

Georgios Metaxas

geboren te Kozani, Griekenland

Dit proefschrift is goedgekeurd door de promotor:

prof.dr. E.H.L. Aarts

Copromotor:
prof.dr. P. Markopoulos

Contents

Chapter 1

Introduction

1.1 Preface

Awareness can bring important, if subtle, benefits, such as affiliating with others and establishing social ties, understanding ourselves and others by obtaining social information or feedback for our actions, gaining and maintaining status, presenting ourselves positively, supporting self-esteem, protecting ourselves and those we value, attracting and retaining mates, finding and keeping partners (Markopoulos, 2009).

One of the earliest designs supporting awareness appears in the ancient Greek mythology:

"On the two former occasions of the payment of the tribute (to Minos), entertaining no hopes of safety or return, they (the Athenians) sent out the ship with a ***black sail****, as to* ***unavoidable destruction****; but now, Theseus encouraging his father, and speaking greatly of himself, as confident that he should kill the Minotaur, he gave the pilot another sail, which was* ***white****, commanding him, as he returned,* ***if Theseus were safe****, to make use of that…*

…When they were come near the coast of Attica, so great was the joy for the happy success of their voyage, that neither Theseus himself nor the pilot remembered to hang out the sail which should have been the token of their safety to Aegeus, who, in despair at the sight, threw himself headlong from a rock, and perished in the sea" (Plutarch, 17.5).

Poetic expression designed the above communication scheme to be intentionally vulnerable to human factors in order to provide a tragic drive for Aegeus death, and ensure that ever since his name be engraved on the Mediterranean map.

People have been using throughout times their contemporary technological means to construct and maintain awareness, similar to how current technologies are also employed to serve this purpose. In recent years there has been a growing interest in the design of awareness systems, which broadly can be defined as communication systems that support individuals and groups to develop and maintain an awareness of each other, without engaging in direct communication.

Perhaps the most widespread example of such a system, present on the desktops of personal computers, is embedded in instant messaging applications. Users are familiar with typical user interfaces portraying their contact lists populated with information indicating the statuses of their contacts. Typically the extraction of this information is semiautomatic, as some of the indications (e.g. whether the user is online or away) are detected automatically using context sensing (i.e. network connection, mouse activity), while others are manually assigned by users themselves (e.g. 'out for lunch', 'sleeping', 'busy' etc.).

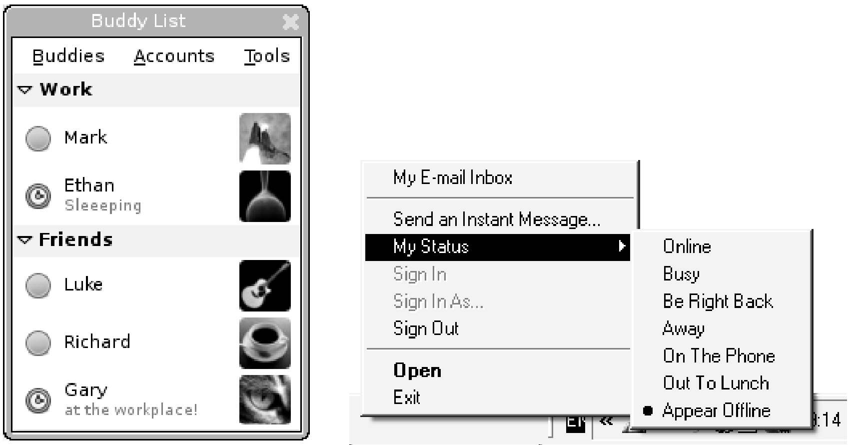


Figure 1. Awareness information is embedded in most Instant Messaging applications. On the left the contact list from the Buddy List IM is displayed, while on the right a popup menu from the Windows Messenger IM is displayed.

Several questions that drive the research presented in this thesis are already illustrated in figure 1. Assuming that the user of the depicted example is the manager of Ethan, then a question that rises is whether Ethan is aware of the fact that his manager knows that he is still asleep. Would it be possible for example that Ethan exposes to his manager that he is stuck on a traffic jam on the way to the office, while at the same time his best friend Mark is informed that he is indeed still sleeping? And if so, would it be possible that Ethan's friend Mark is informed that Ethan is actually lying to their manager so that Mark could avoid revealing Ethan's deception? Furthermore, if we assume that Ethan's status is switched automatically to 'online' the moment he is again using his computer, would it be possible for him to avoid exposing this automatically detected status to his 'want to avoid friends'? Going a step further is it possible that the system automatically detects that Ethan is asleep (as it is unlikely that Ethan will remember every night to set his status to sleeping)? And if so, does the system provide sufficient information to Ethan regarding how it engages in a proactive behaviour such as this? In such a case, how could Ethan alter the system's premises to better match his subjective self-presentation and/or to make better use of the available technological means? What would be the appropriate tools that allow him to evaluate whether the system behaves as expected (given that Ethan can not evaluate this while he is asleep)?

The research presented in this thesis addresses this type of questions, by putting in its focus the social and cognitive challenges that come forward. The thesis argues that the social intelligence in awareness systems is materialized through end-users themselves, and consequently the mechanisms that support them cognitively in the development and maintenance of such systems. In particular the thesis poses three multidisciplinary challenges that have to be met, namely the challenge of appropriate awareness abstractions, the challenge of infrastructure, and the challenge of sustaining intelligibility, accountability, and control.

1.2 Background

Awareness was first studied in the domain of computer supported cooperative work (CSCW), where it was recognized as a central factor for successful collaboration. In this context awareness has been defined as *"an understanding of activities of others that provides a context for your own activities"* (Dourish & Belloti, 1992), and it was argued that awareness enhances user's collaboration and allows them to assign and coordinate work dynamically. When discussing awareness systems, it is frequently assumed that this information is made available for users to observe without interrupting their current activity and without requiring a goal oriented interaction to obtain this information from the system.

In this early period, awareness systems were often discussed as Media Spaces (Baecker, 1993), referring to sustained audio-video links connecting individuals and communities in collaborative environments, and several studies including both implementations and evaluations of Media Spaces were published (e.g., Dourish & Bly, 1992; Fish et al., 1992; Gaver et al., 1992; Gaver et al., 1993; Bly et al., 1993). At the same time the first theoretical works appeared that formalize awareness (Benford & Fahlen, 1993) and implement mechanisms aiming to support people in using their natural social communication skills when working together (Benford et al., 1994).

As the domain matured, theoretical efforts such as the *'focus-nimbus'* model (Rodden, 1996) and the *'event propagation model'* (Fuchs et al., 1995; Sohlenkamp et al., 1997) and design frameworks such as GroupDesk (Fuchs et al., 1995), the workspace awareness framework (Gutwin & Greenberg, 1996), AREA (Fuchs, 1999), and NESSIE (Prinz, 1999) appeared, allowing the incorporation of awareness in computer mediated communication systems.

These developments in the domain of CSCW, along with the evolutions in the domain of context-aware systems and ubiquitous computing and the idea of furnishing the environment of the user with awareness information -or *'populating the periphery of the user's attention'* (Weiser & Brown, 1996)-, emerged to a trend towards addressing awareness in more informal contexts.

In a ground-breaking for the time study, Strong and Gaver (1996) used *"networked devices that are aimed at supporting implicit, personal, and expressive communication, as opposed to the explicit, goal-oriented, and informative communication characterising most CSCW systems"*. Brave and Dahley (1997) presented 'inTouch', a tactile-feedback system that physically link users who are separated by distance. Erickson et al. (1999) presented the Babble prototype –also referred as *'social proxies'*- that used minimalist graphical representation of users and their activities to support the social processes among connected communities. Leichti and Ichikawa (2000) introduce the idea of affective awareness as a general sense of being in touch with one's family and friends.

In a more social context, the concept of interpersonal awareness emerged, which can be considered as an understanding of the activities and status of one's social relations that provides a context for the social interactions with these individuals (Markopoulos et al., 2004).

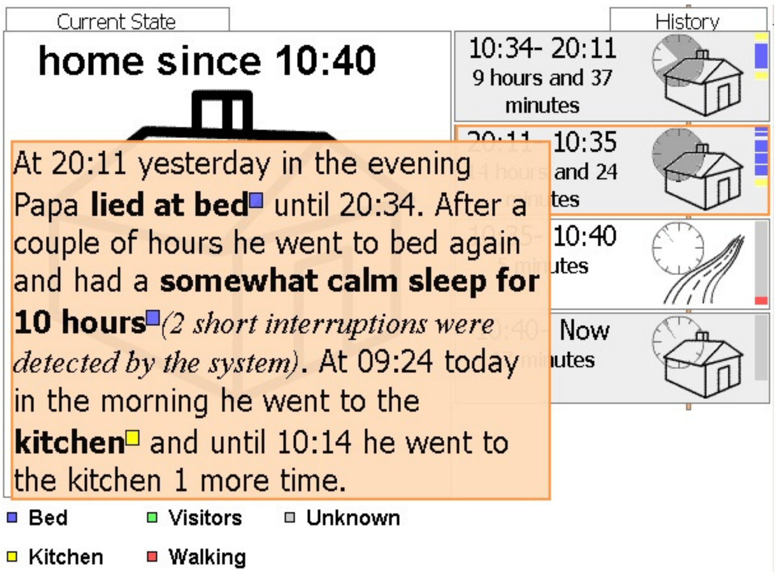


Figure 2. Screenshot of a participant's display from the Diarist system (Metaxas et. al, 2007).

Inline with this definition, works such as the Casablanca project (Hindus et al., 2001) and the Digital Familly Portraits (Mynatt et al., 2001) were of the first to address communication needs in the domestic environment using information extracted through automated context-sensing technologies. The ASTRA prototype (Markopoulos et al., 2004) studied intentional communication for the extended family and demonstrated that such communication can enhance feelings of connectedness and can prompt rather than replace direct communications. PhotoMirror (Markopoulos et al., 2006), another informal lightweight awareness system captures and displays images of trivial daily events and rituals reflecting the commotion and activities of home inhabitants, using sensing technology behind a translucent mirror display. CareNet (Consolvo et al., 2004) focused on "Assisted living" by informing professional care-givers as to medication, nutrition, falls, etc., of elderly patients living alone, an issue further explored with a more realistic deployment with the Diarist system (Metaxas et al., 2007). In figure 2 a display that generates narratives out of a participant's activity from the field study conducted for the Diarist system is shown.

The works cited above represent just a fraction of a growing literature on the topic of awareness systems, which expands to an ever increasing variety of physical and social contexts addressing a diverse range of user needs. Such technological evolutions initiated discussions and criticism regarding the implications of awareness systems, in works such as of Bellotti and Edwards (2001), Erickson et al. (2002), K. Schmidt (2002), Erickson and Kellogg (2000), Chalmers et al.(2003), Hong and Landay (2004), Oulasvirta (2004), Boyle and Greenberg (2005), Oulasvirta et al.(2007), Aarts and de Ruyter (2009), Erickson (2009).

Often the discussion on the nature of awareness pertains to how awareness is achieved, whether it is possible to make a clear separation among systems supporting awareness versus systems and other more purposeful and goal-oriented communication tools, whether awareness can be achieved without focused attention, or what purpose awareness serves.

Different classes of such systems have been identified, distinguishing subdivisions of the general notion of awareness: workspace awareness (Gutwin et al., 1996), social awareness (Tollmar et al., 1996), interpersonal awareness (Greenberg & Rounding, 2001), affective awareness (Liechti & Ichikawa, 2000) etc.

1.3 The challenge of proper awareness abstractions

The research community has grown to realize that drawing subdivisions of this sort does not provide a sound foundation for progress (K. Schmidt, 2002). Rather than attempts to classify awareness systems, it may be more relevant to identify generic mechanisms by which awareness information is collected and made available at different contexts, and the way these mechanisms embed themselves in the social interactions they mediate.

Accordingly, theoretical discussions motivating the design of such systems gravitate towards the phenomena surrounding the social aspects of using awareness. Some recent examples are the ASTRA project (Romero et al., 2007) that examined the affective benefits and costs of using awareness systems, and the work on mobile awareness cues by Oulasvirta et al. (2007) who examined how social inferences can be made through the availability of awareness information.

The general motivation for examining how social practices can be supported by such technology is represented in the work of Erickson et al. (1999; 2002) and Erickson and Kellogg (2002). Erickson introduced the concept of social translucence that encapsulates issues of inter-subjectivity between users of awareness systems. Social translucence distinguishes systems that support one way observation and monitoring of a person or group, with systems that make visible one's ability to observe and thus makes this person accountable for their actions.

By extension, it is clear that awareness brings about accountability which may not always be desirable, compromising one's autonomy (Boyle & Greenberg 2005) and compromising an individual's ability to manage their own privacy borders (Romero & Markopoulos, 2005), (Hong & Landay, 2004), (Palen & Dourish, 2003), (Price et al., 2005), (Iachello et al., 2005a; 2005b), (Ackerman et al., 2001) or even to achieve politeness by means of equivocation, a practice that is very common in our daily face to face communication with each other (Aoki & Woodruff, 2005).

In this thesis, embracing the motivation that awareness systems should allow users to meet social needs and to practice extant social skills, we argue that these practices can be understood in terms of the information sharing that awareness systems support. Focusing on the information sharing mechanisms and practices, we aim to abstract away from architectural issues that have often concerned this domain (e.g., Fitzpatrick et al., 1999) or the principles presenting such information (e.g., Miller & Stasko, 2002;Pousman & Stasko, 2006). While such works are valuable, they do not support the discussion of how awareness influences social interactions, or how concepts borrowed from social psychology relate to system features.

Commenting on the contradictory uses of the term awareness, K. Schmidt (2002) argued that there is the endemic lack of conceptual clarity for the research domain we sketched out above. He remarked that awareness should be described in reference of activities, practices, phenomena, or objects that a person is made aware of. Inline with this argument, sufficient abstractions are needed so that beginning to answer the question 'aware of what?' can lead to reasoning with regard to the social aspects pertaining to awareness systems.

1.4 The challenge of intelligibility, accountability and control

Context Awareness, the ability of computing systems to sense and react dynamically to changes in their physical environment, is a key ingredient of prevalent technological trends, such as ubiquitous computing and ambient intelligence (Aarts, 2003). As sensor technology is increasingly integrated into consumer electronics and mobile computing platforms, but also cars and buildings, context awareness is bound to become more widespread and to be gradually embedded in our every day environments and activities. For users of related systems potential benefits include reduced effort for providing necessary input, automated adaptation of system function to their behaviour, or even autonomous system function on their behalf. On the other hand, harnessing the benefits of context awareness can be problematic for end-users and other affected individuals, who may not always be able to anticipate, understand, or appreciate system function and may feel their own sense of autonomy and their privacy threatened. The increased complexity of such systems, compared to standard desktop applications, raises concerns regarding the ability of people to configure their hardware and software environments and maintain them.

In a classic essay, Bellotti and Edwards (2001) discussed the interaction design challenges that context aware systems present. They argued that such systems should not simply act on behalf of users, but rather should involve users in action outcomes, allowing them to understand, explore, and even define the underlying mechanism governing the behaviour of such systems. According to Belloti and Edwards, two key features for such systems are *intelligibility* and *accountability*:

- Intelligibility: Context-aware systems that seek to act upon what they infer about the context must be able to represent to their users what they know, how they know it, and what they are doing about it.
- Accountability: Context-aware systems must enforce user accountability when, based on their inferences about the social context, they seek to mediate user actions that impact others.

Corollaries from their argument are design principles that emphasize the need to inform users about the capabilities of the system and the model, which a system constructs out of a user's context and intentions. The system has to provide feedback that allows users to tell what the consequences of a change in their context will be, and to defer and control system behaviour (Edwards & Grinter 2001).

1.5 The challenge of infrastructure

We have discussed on one hand the necessity to incorporate the social requirements in the developments of awareness systems, and on the other hand the need to ensure that end-users are able to control and understand the consequences of their interactions with the system.

Supporting awareness has prompted researchers to consider how to integrate the capture of contextual information, its dissemination and display within heterogeneous collections of devices and services comprising Ambient Intelligence or ubiquitous computing environments (Fuchs et al., 1995; Pederson & Sokoler, 1997; Sohlenkamp et al., 1997; Dey et al., 2001; Dey et al., 1998; Hong & Landay, 2004).

Edwards et al. (2010) discuss the infrastructure problem, pointing out that the infrastructure design decisions have several implications on user experience overall:

- Constrained possibilities: Design choices influenced by the infrastructure may preclude entirely certain desirable user experience outcomes.
- Interjected abstractions: Technical abstractions in the interface may appear in the conceptual model exposed to users.
- Unmediated interaction: Users may have to interact directly with the infrastructure to accomplish their goals.

Kindberg and Fox (2002) indicate that context-aware applications are dynamic evolutions rather than static configurations of services and functions, and that their dynamics should be seen as the result of both implicit and explicit interaction with the user. In advancing towards realistic deployments and actual use, devices and services need to be used often in configurations and for purposes that are not foreseen by their designers and developers (Khan et al., 2008; Green, 2003).

Eventually such a dynamic configuration and repurposing of the multitude of devices and applications in an Ambient Intelligence environment requires that they operate collectively, using information and intelligence that is hidden in the interconnection network (Aarts & de Ruyter, 2009). A clear consequence of this position is that interoperability and dynamic aggregations of devices and services are needed; this technical ambition has been pursued consistently by the Ambient Intelligence research field in the past ten years, but its goal has not been met by evolvements up to date.

Apart from ensuring that infrastructure can satisfy such technological requirements and incorporate further developments in an architectural layer, it should at the same time allow social practices and norms to apply and emerge unencumbered, and allow end-users to address the requirements of intelligibility, accountability, and control.

1.6 The role of end-user programming

As the Attention Investment model (Blackwell & Green, 1999; Blackwell, 2002) suggests, users consider four factors before engaging in an action: a) perceived benefits, b) expected pay-off, c) perceived cost, and d) perceived risks. The same factors account for programming[1], since a program can been seen as a benefit while the act of programming as a cost (Repenning, 1993). This costs/benefit balance becomes evermore evident when it comes to end-users themselves, and end-user programming –i.e. *programming to achieve the result of a program primarily for personal, rather public use*[2] (Ko et al., 2010)- as a mechanism to support their needs.

From the perspective of end-user programming, the need for practicing social extant skills along with the challenge of sustainable intelligibility, accountability, and control, requires the definition of appropriate computational abstractions, and supportive interaction tools.

End-user programming emerges thus as a key ingredient for ensuring the acceptability of context aware systems and has accordingly been identified as a key future challenge for research in Ambient Intelligence and its related technological visions of ubiquitous computing and pervasive computing (Aarts & de Ruyter, 2009), particularly motivated by the need to control and manage ad hoc collections of devices and systems which make up the environments in which users live, work,

---

[1] In this thesis we embrace the definitions of Ko et al. (2010) for the terms programming and program as *'the process of planning or writing a program'* and *'a collection of specifications that may take variable inputs, and that can be executed (or interpreted) by a device with computational capabilities'* respectively.

[2] This definition goes beyond the scope of investigations of Nardi (1993) into spreadsheet use in office workplaces, allowing a more open view on the term *'end-user programming'*.

and interact (Davidoff et al. 2006). The visualization and computational representation of context are the quintessential point of interaction between users and system intelligence.

1.7 Approach

The above identified challenges, namely the need for proper awareness abstractions, the need for sustaining intelligibility, accountability, and control, and the infrastructure problem indicate that to advance the field of awareness-systems, a multidisciplinary approach that involves modelling, implementation, design and experimentation in favour of end-user programming has to be pursued. Below we outline the main inductive and deductive steps that were followed to address these challenges, although the work presented here is the result of several iterations.

An approach that proceeds inductively from concrete to abstract and from specific to generic phenomena was used to capture the space of awareness systems in terms of the information exchanged among entities. As a result an abstract model is developed, which is then deductively validated in terms of the socially meaningful communication patterns that it is able to represent. This model was formalized using set-theory notation and is shown to be a sufficient extensional abstraction for awareness systems. As such, one may consider that any awareness system is an instance of the presented model, while the development and maintenance of such systems is a matter of applying the model's principles. Accordingly, end-user programming of awareness systems can be seen as the translation of the formal model and the operations on it directly to executable semantics bonded with some user interaction scheme. However, such an implementation of the model is not at all self evidently forming an appropriate intensional definition, taking in account the broad design space and the implications of infrastructure on the end-user experience.

It is clear that a different approach should be followed in the development of a framework for awareness systems. Hence we first inductively present an architectural framework for context aware applications in general, which takes into account such considerations. Then, we deductively project on this framework the principles identified in the formal model and we show through several examples how the infrastructure can support end-user programming of awareness systems. Our intention is twofold: on one hand we present a direction for future software engineering developments in the field, and therefore focus on the conceptual description of the framework rather than its implementation specifics. On the other hand we present to awareness systems designers a framework in terms of the implications that their design choices might have on effectively supporting crucial requirements on the end-user side.

In terms of the developed framework, an awareness system is a collection of interconnected services that extract, disseminate, and consume awareness-related information. In this respect, end-users are able to program the behaviour of an awareness system by registering and configuring services that perform such tasks.

Thus the key element of this compositional process is the ability to provide appropriate tools and mechanisms addressing cognitive-ergonomics requirements of end-users. This is another main objective of this thesis that is pursued by following a deductive approach to develop and validate such tools and mechanisms.

1.8 Overview of this thesis

This section gives an overview of the remaining chapters of the thesis and summarizes their individual contributions.

Chapter 2 introduces the FN-AAR(Focus-Nimbus, Aspects, Attributes, Resources) model, a formal model for awareness systems and projects on it the social norms and main social requirements of this class of systems as these are derived from literature. Significant effort was put on illustrating its principles and implications in natural language, allowing readers interested in the concepts discussed, but without having affinity to set-theory, to skip the formalizations.

The requirements of intelligibility accountability and control are addressed in depth in Chapter 3 which proposes the means to support end-user development of awareness systems. In the same chapter, the theoretical foundations of these proposed tools are further supported by an experimental study that readers without an interest in the underpinning psychology may skip.

Chapter 4 describes the Amelie framework, a framework suitable for the development of awareness systems, and establishes it as a translation of the FN-AAR model to executable semantics, while at the same time it addresses the requirements of intelligibility, accountability, and control at an architectural layer.

The last chapter evaluates the contribution of the thesis and discusses the emerging relevant issues for the domain of computer mediated communication, ambient intelligence and ubiquitous computing. In the following figure (figure 3) the scheme of this thesis is depicted and next to it we briefly discuss the contributions of each chapter.

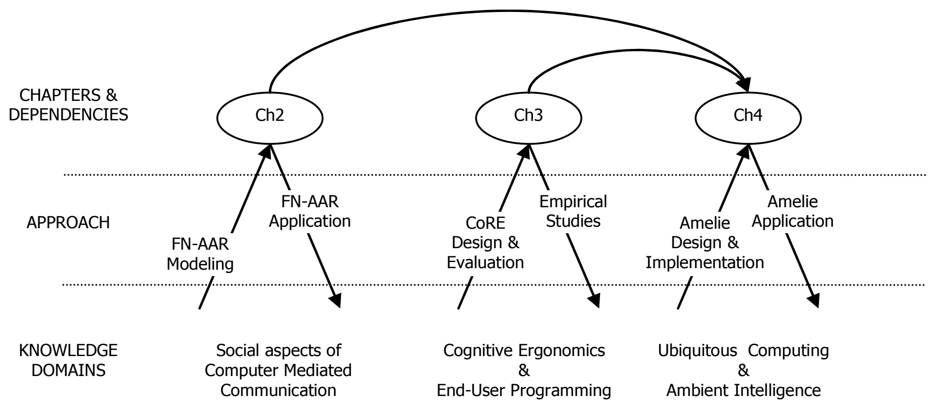


Figure 3. Thesis scheme

Chapter 2: This chapter concerns social aspects of interaction with ambient intelligence applications that support awareness of activities and whereabouts of others. It introduces FN-AAR, an abstract model of such systems, which allows to model and reason about several social salient patterns pertaining to communication. FN-AAR intentionally masks away the underlying mathematical and information propagation concepts in order to leave enough space both for analysing existing applications, and for later developments and implementations. As a formal representation, it brings clarity to the discussion of awareness systems allowing clear relations to be drawn in a way previously not possible, among theoretical concepts that are overlapping and related. The model allows describing and clarifying fine nuances regarding concepts such as social translucence, plausible deniability, and symmetry, lending clarity to earlier theoretical discussions. We argue that building systems that support this conceptual model will allow their users to specify and configure the disclosure and display of information in terms that are meaningful to them and relevant to their concerns.

Chapter 3: In chapter 3, we suggest a set of tools and mechanisms that can uniformly support intelligibility, accountability, and control of context-aware systems. We introduce a context-range notation that allows on one hand context-aware services to describe the range of their outcome, and on the other hand facilitates compositionality and allows services to describe the premises that manage their behaviour. The notation can be translated both to natural language and to a structured editor, and is shown experimentally that it allows non programmers to understand and formulate logical expressions of context of realistic complexity. We argue that by exposing in this respect the -otherwise concealed- system's premises to end-users, they are empowered to answer relevant questions such as "how does the system behave", "why is something happening", "how would the system have behaved in response to a change in context", and "how can the system's behaviour be altered". In the second section of the fourth chapter, the assumptions behind the heuristics that are used for the presentation of context-ranges and their manipulation are further validated in a laboratory experiment. This experiment investigates the role of term-affinity in the spontaneous usage and comprehension of disjunctive and conjunctive normal forms, extending relevant findings of the Mental Models theory.

Chapter 4: This chapter presents Amelie, a framework for the design of awareness systems that adopts the tenets of recombinant computing. Within Amelie, a single recombinant interface is argued to be a sufficient abstraction yet not an overgeneralization of awareness-systems' services, which can engage in compositional structures forming awareness and context-aware applications. The Amelie framework provides the necessary semantics to directly implement systems that ensure socially salient properties, such as symmetry, deception, and social translucency. Furthermore, it addresses the requirements of intelligibility, accountability, and control at an architectural level and allows the dynamic composition and maintenance of applications both through implicit and explicit interaction mechanisms.

Chapter 5: Chapter 5 assesses the contribution of the thesis and discusses the emerging relevant issues for the domain of computer mediated communication, ambient intelligence and ubiquitous computing. The thesis closes with a prologue on the 'future-proof future' concept for awareness systems, ambient intelligence and ubiquitous computing.

Chapter 2

Abstractions of Awareness

In this chapter we introduce FN-AAR, an abstract model of awareness systems, which allows one to model and reason about several social salient patterns pertaining to mediated communication. First the context for the model presented here is given in section 2.1, while related work on models of awareness systems is reviewed in section 2.2. The concepts of the model are introduced (section 2.3), and then the formal description of the model is shaped (section 2.4). In the remaining sections (2.5 to 2.8) we go on to discuss in terms of the model several concepts such as deception, translucency and symmetry.

2.1 Introduction

In the previous chapter it was pointed out how the great majority of awareness systems concepts proposed in related literature, cluster around some basic themes; some of the most common themes are, communicating to someone that you are thinking about him/her, conveying simple presence information at a particular location, sustained audio video links between places, serendipitous discovery of information about others, supporting flexibility and the conjoint creation of meaning between participants, etc.

On the other hand, theoretical discussions motivating the design of such systems gravitate towards the phenomena surrounding the social aspects of using awareness. For example, Erickson et al. (2002) have introduced the concept of social translucence that encapsulates issues of inter-subjectivity between users of awareness systems. Other issues relate to privacy of people and ways in which they might manage their accessibility to others (Price et al., 2005; Hong & Landay 2004; Iachello et al. 2004).

These two trends point to the need for a clear conceptualization of awareness systems that lends some clarity to the description of relevant phenomena. More specifically, such a conceptualization should abstract away from detailed aspects of form and application context, to describe the communication aspects of awareness systems in terms relevant for discussing social interactions between users.

K. Schmidt (2002) discussed the endemic lack of conceptual clarity for the research domain we sketched out above. Noting the contradictory uses of the term awareness, he argued that dichotomies between attention and peripheral awareness, active and passive awareness, explicit and tacit, etc., are misleading. Rather he argued that awareness should be described in reference of activities, practices, phenomena, or objects that a person is made aware of.

In line with this argument, the remainder of this chapter presents an abstract model of awareness systems that incorporates related concepts and supports reasoning regarding social aspects of using awareness systems.

The model introduced in this chapter is a development of the focus and nimbus model originally introduced by Benford and Fahlen (1993), to model awareness in the domain of virtual reality. Relying on a set-theoretic[3] description, our model provides a conceptual tool for reasoning about this class of systems based on the notions of *aspects*, *attributes*, *resources* and *observable items* in order to expose the communicational aspects of awareness-systems. To distinguish our model from the original focus and nimbus model, it shall be referred to below as 'FN-AAR'. The FN-AAR model is abstract in that it does not detail how information is collected, stored or disseminated, or by which mechanisms users interact with it. This abstraction allows for very generic descriptions that are generalizable across contexts and system architectures.

2.2 Related Works

One of the early-introduced influential models was the *'event propagation model'*, introduced by Fuchs et al. (1995), and Sohlenkamp et al. (1997). This model identifies three basic information processing functions that an awareness system has to support: capturing information regarding a particular individual, group or location, disseminating it, and displaying it to the intended receivers. The event propagation model (figure 1) proposes the representation of the environment as a semantic network where awareness about changes and activities in the system is supported by the generation and distribution of events in the semantic network. The propagation of events from a source to a sink is filtered by individual-outgoing-filters, such as privacy filters, at the source (event-generation) side, and individual-incoming filters (interest filters), at the event-consumption (sink) side.

[3] The notation used follows classical Zermelo–Fraenkel set theory, incorporating some elements from the Z notation (Spivey 1992)

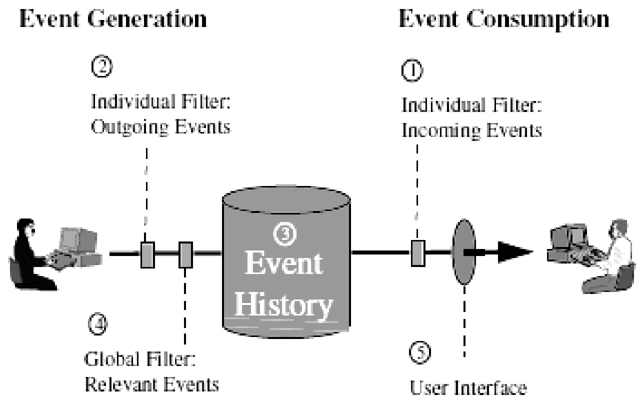


Figure 1. The event propagation model; figure adapted from Sohlenkamp et al.(1997)

Fuchs's model can be useful as an abstract reference model for the implementation of awareness systems, but does not allow further reasoning regarding the nature of information captured and how it is transformed through each of the functional components it identifies.

In the same direction with Fuchs, and inspired from biology, Simone and Bandini (2002) proposed the reaction-diffusion metaphor that aimed to make "*awareness mechanisms fully visible and accessible to the involved actors for the purpose of adaptability*". The model is based on the notions of *space*, and *fields*. *Space* is populated by entities, and it is used to evaluate when entities come in contact and to express how fields propagate in the space. *Fields* are the means by which awareness information is brought in and propagated in the space, and influences the entities able to perceive it. Mechanisms governing the *emission* and *reception* of fields provide the capability of modulating *awareness* on the side of the emitter as well as on that of the receiver.

The most influential mathematical conception of awareness that abstracts away from the information-flow aspects and focuses on the communicational aspects of awareness is the *focus-nimbus* model. Benford and Fahlen (1993), and Benford et al. (1993; 1995), introduced the notions of Nimbus and Focus in a spatial model of group interaction, in order to address mutual levels of awareness within a virtual environment.

- Focus represents a sub-space within which a person focuses her attention. The more an object is within your focus the more aware you are of it.
- Nimbus on the other hand represents a sub-space across which a person makes their activity available to others. The more an object is within your nimbus, the more aware it is of you.

Based on these notions Benford et al. define a "*measure of awareness*" as a functional composition of *Focus* and *Nimbus* quantifiers; this measure answers the question: "*In a given room, how aware is entity i of entity j via medium k?*"; i.e.

$$\textit{Level of Awareness}: \ A_{kij}(f_{ik}, n_{jk}) : \mathbb{R}^2 \rightarrow \mathbb{R}$$

This function evaluates to a measure of awareness of a given entity *i* to another *j* based on values of the focus of entity *i* (f_{ik}) and the nimbus of entity *j* (n_{jk}) at *k*.

Rodden (1996) rendered the focus-nimbus model in set-theory terms extending its application to a wider range of cooperative applications, beyond the boundaries of spatial applications. This model's principal aim is to allow reasoning about the potential awareness among users, in terms of reflecting on the 'likelihood' of actions by one user being noticed by another. Rodden abstracts away from the spatial approach by linking users to the presence space by nimbus and focus functions; i.e. functions that relate users to objects that characterize their nimbi, and foci. By estimating the awareness overlap for two users one can evaluate the strength of awareness between two users, either from a continuous or a discrete point of view. Such estimation depends on the existence of metric functions for focus and nimbus that are considered application specific and subject to empirical investigation (Rodden, 1996). In figure 2, we can see some of the different modes of awareness that can pertain among two users, when we consider a discrete representation of awareness according to Rodden's focus-nimbus model.

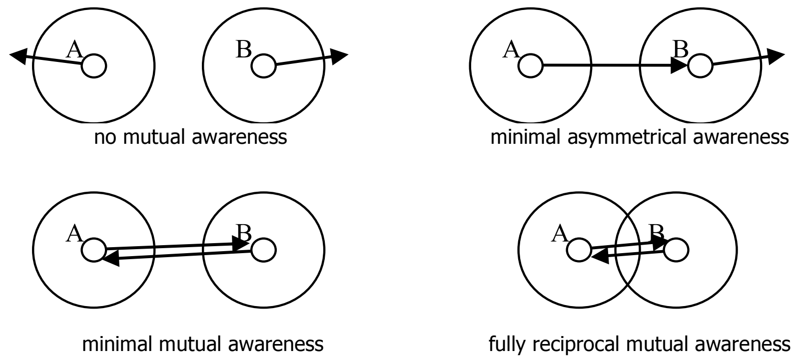


Figure 2. Some of the discrete awareness modes (4, out of 16 arrangements)

The focus-nimbus model has provided the conceptual foundation for several applications; Fernado et al. (2004) constructed a first-order logic representation of focus and nimbus enabling the definition of higher level operations for controlling multimedia streams between communicators using higher level operations such as mute, hide, etc. The service-oriented group awareness model, (Ji et al., 2006), is a recent implementation oriented model, focusing on web services that can support group-awareness. These renditions of Rodden's model are application specific and are not appropriate for supporting a general model of awareness systems and for reasoning on user relevant aspects such as, privacy, translucence, etc.

Privacy and awareness represent flip sides of the same coin. Noting the duality of these needs Boyle and Greenberg (2005) applied the concepts of *attention*, *fidelity*, and *identity* in order to define privacy needs in the ubicomp domain. They proposed the following characterizations for privacy needs:

- Solitude: control over one's interpersonal interactions, specifically one's attention for interaction.
- Confidentiality: control over other's access to information about oneself, specifically the fidelity of such accesses.

- Autonomy: control over the observable manifestations of the self, such as action, appearance, impression, and identity.

Boyle and Greenberg go on to project their tripartite conception of privacy on Rodden's focus-nimbus model for awareness. *Foci* correspond roughly to attention, so solitude can be thought of as focus regulation. *Nimbi* correspond to embodiments, socially constructed personas, and to one's relationships with information and artifacts in the environment. Nimbus regulation therefore roughly corresponds to confidentiality and autonomy. *Awareness*, which is defined as a functional composition of focus and nimbus, is analogous to the dialectic negotiation of privacy boundaries.

The present chapter continues where Boyle and Greenberg left this discussion, trying to give formal semantics to such a conception of privacy and awareness. The model we introduce in this chapter is based on Rodden's abstract version of the focus-nimbus model. We show how this model can provide a sound basis for mathematically describing the design space of awareness systems, in terms of the content exchanged, elementary user behaviours pertaining to sharing information about themselves or perceiving information about others. The sections that follow shall introduce the FN-AAR model and demonstrate how some principles for the protection of user privacy can be expressed succinctly, lending clarity and conciseness to the discussion of awareness systems and their design.

2.3 Model overview

Where the original focus-nimbus model describes *how* aware two entities are of each-other in a particular *space,* the FN-AAR model describes *what* are the entities aware of regarding each-other in a particular *situation.* The model we propose is an extension of the focus-nimbus model, populated with the notions of *entities*, *aspects, attributes, resources* and *observable items.* These notions are introduced below with the help of the following scenario:

"John, Anna, and their young daughter Doty use an awareness system to share with each other their status and daily activities. Among others, John configured the system to let Anna know how busy he is (i.e. his availability) by using a simple plug-in at his computer. The plug-in makes the assumption that while John is using his computer he is either using his mouse or his keyboard and at the same time the more windows are open at John's computer the busier he is. Anna is using an 'aware-watch'; this gadget normally displays the time, but when she pushes a small button it shows John's availability by highlighting a corresponding icon."

Entities are representations of actors, communities, and agents (possibly artificial) within an awareness-system. The actors of the above scenario (i.e. *John* and *Anna*) are represented in an awareness system with the corresponding entities. The family above can be thought of as a community, and their house could be seen as an agent.

Aspects are any characteristics that refer to an entity's state. An aspect is actually the complement to the incomplete-statement "*I want to be aware of your …*" . In our scenario "*Anna wants to be aware of John's availability*"; thus the word "*availability*" is an aspect, i.e. a characteristic of *John*'s state that may be shared with *Anna*. The notion of aspect is broad and loose enough to encompass terms like "location", "activities since yesterday", "aspirations", or even "focus", and "nimbus".

Attributes are the place-holders for the information exchanged between *Entities*. An attribute can be thought of as a potential answer to the request "*Tell me something about your 'X' aspect".* In our scenario an answer to *Anna's* request *"John tell me something about your location"* could be "*My location is home*"; thus the statement *"My location is home"* is an attribute, binding the value "*home*" to the aspect "*location*".

In any situation an entity makes its state available to other entities using one or more attributes. To reflect the fact that awareness is dynamic, we populate one's ***nimbus*** with *attribute-providers*; i.e. functions that return those attributes that one makes available to other entities in a specific situation. In the scenario above the "*plug-in*" that detects *John's availability* can be seen as an *attribute-provider*, which returns attributes about the *availability* of *John* depending on the situation(i.e. the number of open windows), and makes them available to *Anna*.

A ***resource*** is a binding of an *aspect* to a way of rendering (displaying) one or more attributes about this aspect. In any situation an entity might employ one or more resources to express its 'interest' about certain aspects of other entities. Roughly speaking a *resource* is a statement such as *"I shall render the attributes of you that you provide to me about your 'X' aspect by … ".* In our example, "*Anna plans to render the attributes of John tht he provides to her about his availability by highlighting an appropriate icon on her 'aware-watch'*".

Like one's *nimbus*, ***focus*** is also dynamic. In the example above, *Anna* assigns her watch to display *John's availability* when she presses a small button. In our model, ***focus*** is populated with *resource-providers*; i.e. functions that return one's *resources* that are engaged to display information about other entities in a specific situation. *Anna's 'aware-watch'* can be seen as a *resource-provider* that depending on the situation (i.e. the *show-john's-availability* button is pressed) returns a resource which renders *John's availability*.

An ***observable item*** is the result of displaying the product of rendering one or more *attributes* about an *aspect* using a specific *resource.* Roughly speaking an *observable item* contains the answer to the question *"How are these attributes displayed to you?".* In our scenario a possible answer to the question *"How is 'availability-busy' displayed to you?"* could be *"by highlighting the busy icon on my aware-watch".*

Conforming to the original focus-nimbus model, the negotiation of the reciprocal *foci* and *nimbi* of two entities in a given situation (i.e. the corresponding 'produced' *attributes* and *resources*) is a function which returns the *observable-items* that are displayed to the two entities about each other's states, effectively characterizing their reciprocal awareness.

In the above scenario, *John* indicates his *availability* to *Anna* using the *plug-in*. This plug-in is an *Attribute-Provider* in *John's* nimbus that returns (in any situation) an attribute about *John's availability*, which is made available to *Anna*. On the other hand, *Anna* can check *John's availability* by pressing a small button on her *'aware-watch'*. System-wise we can consider that *Anna's Focus* is populated by a *resource-provider* that returns a *resource* for rendering *John's availability* whenever this small button is pressed. This resource claims to render *John's availability,* by highlighting an appropriate icon on her *'aware-watch'* display.

Needless to say, neither the *availability-plug-in* nor the *aware-watch* implies necessarily John's *availability* (the plug-in is obviously imprecise) or that Anna is indeed aware of it. However, we can imagine that *Anna* can choose whether to focus on John's availability, or even to '*assign*' her *aware-watch* to another person. So, *Anna* becomes aware of *John's availability*, by manipulating her *focus.* Similarly, we can imagine that *John* can choose not to let *Anna* know his *availability*, thus *John* lets *Anna* become aware of his situation by manipulating his *nimbus*.

2.4 Model formalization

2.4.1 Attributes, Attribute Providers & Nimbus

In the focus-nimbus model, nimbus represents a sub-space across which an entity makes its state available to others in order to quantify the level of awareness among two entities. Here instead, we wish to address the question *"what is an entity x aware-of regarding an entity y"*; for that, apparently we have to address the question *"what is an entity y exposing to an entity x"*.

To commence we consider that in any situation an entity's state (as it is exposed to other entities) holds information about a wide range of aspects. We use the scheme "*Attribute"* to describe a piece of information (*"value"*) about an aspect (*"aspect")*.

Attribute	
aspect	: *Aspect*;
value	: *Data*;

For convenience, we use the idiom *(a: v)* to denote *'⟨aspect⤳a,value⤳v⟩'*. i.e. the idiom *(a: v)* denotes an attribute about aspect *a* with value *v.*

As it was mentioned in the overview above, in terms of the FN-AAR, attributes are the place-holders for the information exchanged between *Entities*, in a context that changes dynamically. To reflect on that, an entity's nimbus is populated with attribute providers; i.e. functions that, when applied to a situation, return an attribute and the set of entities that this attribute is made available to. Hence, an attribute provider may return different attributes exposed to different entities depending on the situation:

$$AttributeProvider : RealSituation \rightarrow (Attribute \times \mathbb{F}\ Entity)$$

For an instance of *AttributeProvider p* we use p^r to denote *first p(r)* and $p^r.e$ to denote *second p(r)*; i.e. p^r denotes the attribute that *p* returns at situation *r*, and $p^r.e$ denotes the set of entities that p^r is made available to.

In this section and elsewhere *RealSituation* is an abstraction that we use to encapsulate the dynamic nature of the universe to which awareness refers. The model itself is neutral regarding the notion of reality; the model and the user-related properties in the following sections do not make any assumptions about what is "real".

Nevertheless, for each entity *i* we assume that $nimbus_i$ includes all the entity's *i* attribute providers, hence defining implicitly the attributes of entity *i*:

$$\forall\ i{:}Entity;\ nimbus_i : \mathbb{F}\ AttributeProvider$$

Given $nimbus_i$, we can define a function n_{ij} such that when applied to a real situation it returns all the attributes of entity *i* that are exposed to entity *j*:

$$\begin{gathered}\forall\ \ i,j{:}Entity;\ \ n_{ij}\ : RealSituation \rightarrow \mathbb{F}\ Attribute\ \ | \\ \forall\ r{:}RealSituation;\ \ n_{ij}(r)= \\ \{a{:}\ Attribute|\ (\exists\ p{:}AttributeProvider;\ p \in nimbus_i \bullet\ (a{=}p^r)\wedge(\ j \in p^r.e\ ))\}\end{gathered}$$

Figure 3 shows three attribute providers of entity *i* (*p1, p2, p3*), and their corresponding attributes in a situation *r* (i.e. *a1, a2, a3*). Attribute provider *p2*, makes attribute *a2* available to entity j; *p1* makes *a1* available to entities *j*, and *k*; *p3* makes *a3* available to entity *k*. Consequently the nimbus of entity *i* to *j* at this situation is $n^r_{ij}=\{a1,a2\}$ and the nimbus of entity *i* to *k* at this situation is $n^r_{ik}=\{a1,a3\}$.

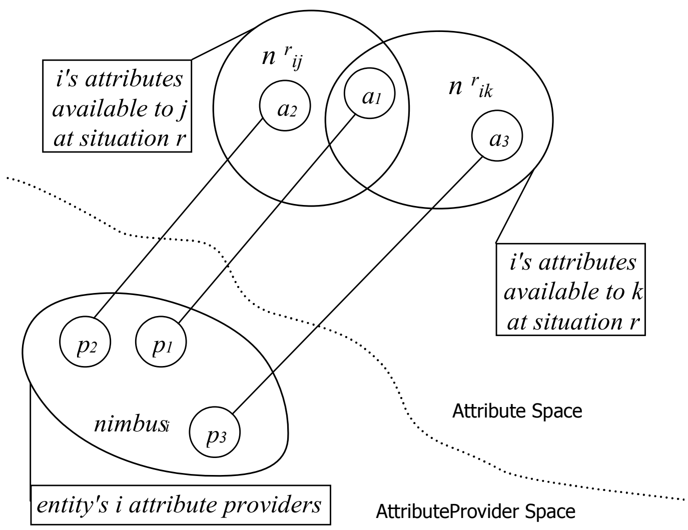


Figure 3. The nimbus of an entity i to entities j and k respectively.

We can reflect on the nimbus of *John* to *Anna* in the scenario introduced earlier: *John* lets *Anna* know his availability by configuring the availability-detector plug-in at his computer. In terms of the system, in any situation *r*, *John* makes available to *Anna* an attribute a ($a \in n^{r}_{John,Anna}$) about his "*availability*". Following the model, *John's* nimbus contains an attribute-provider that depending on the situation returns the aforementioned attribute occupied by a value that corresponds to an estimation of his availability.

$$p1: AttributeProvider;\ p1 \in nimbus_{John} \mid \forall\ r{:}RealSituation;$$
$$(p1^{r}.aspect = availability) \wedge$$
$$(p1^{r}.value \in \{(availability{:}available), (availability{:}busy)\}) \wedge$$
$$(p1^{r}.e = \{Anna\})$$

Thus, p*1* is an attribute provider in *John's* nimbus, which when applied in a situation *r,* returns an attribute ($p1^{r}.aspect$: $p1^{r}.value$) and an entity-set $p1^{r}.e$ that includes *Anna*. The attribute's aspect is '*availability*' and its value is either '*available*' or '*busy*'. We can wrap up John's nimbus ($nimbus_{John}$), and its exposed attributes to Anna ($n^{r}_{John,\ Anna}$) as follows:

$$nimbus_{John} = \{p1\}$$
$$\forall\ r{:}RealSituation;\ n^{r}_{John,\ Anna} = \{p1^{r}\};$$

2.4.2 Resources, Resource Providers & Focus

The definition of nimbus in the previous section addressed the question *"what is an entity exposing to other entities in a given situation"* by defining an entity's nimbus in terms of its attributes that it makes available to other entities. The question *"What is an entity aware of regarding other entities?",* however, is two-fold; not only do we need to know what is available for observation to an entity, but we also need to know "*what is this entity inquiring from other entities*", and more particularly how the entity 'plans' to transform (render) the acquired attributes of others to observable items.

To address the latter question we elaborate on focus, by introducing the notions of resources and resource-providers. In coordination with the original focus-nimbus model where focus represents a sub-space within which an entity focuses its attention, we assume that system-wise an entity has a set of resources to inquire and represent the available information of other entities. Below we introduce the scheme *Resource* to define an aspect of interest and a function that transforms the corresponding attributes to an observable item.

Resource	
aspect	: *Aspect*;
render	: $\mathbb{F}$ *Attribute* $\rightarrow$ *ObservableItem*;

One's resources may change depending on the situation; to incorporate this in the model we define a function-type *ResourceProvider*, that when applied to a real situation returns a resource and an entity that it is assigned to. Hence, a single resource provider may return different resources assigned to different entities depending on the situation:

$$ResourceProvider\colon RealSituation \rightarrow (Resource \times Entity)$$

For a *ResourceProvider* instance p we use p^r to denote *first p(r)* and $p^r.e$ to denote *second p(r)*; i.e. p^r denotes the resource that p returns at the situation r, and $p^r.e$ denotes the entity that p^r is assigned to. We populate the focus-space with resource-providers assuming that for each entity i, $focus_i$ includes the set of entity's i resource providers.

$$\forall\ i{:}Entity; focus_i : \mathbb{F}\ ResourceProvider$$

Given $focus_i$ we can define f_{ij} to return only those resources of i that focus on entity j, characterizing, in terms of resources, entity's i focus on entity j in a situation r:

$$\forall\ i,j{:}Entity; f_{ij} : RealSituation \rightarrow \mathbb{F}\ Resource \mid$$
$$\forall\ r{:}RealSituation; f_{ij}(r) =$$
$$\{c\colon Resource \mid (\exists\ p{:}ResourceProvider; p \in focus_i \bullet (c = p^r) \wedge (j = p^r.e))\}$$

In figure 4 we sketch out on the bottom-right three resource providers of entity's i focus (i.e. *p1 p2 p3*), and their corresponding resources in a situation r (i.e. *r1,r2,r3*). The resource-provider *p1* assigns the resource *r1* to display information from entity j; *p2* assigns *r2* to j; *p3* assigns *r3* to k. Consequently the focus of entity i on j at this situation is $f^r_{ij}=\{r1,r2\}$ and the focus of entity i on k at this situation is $f^r_{ik}=\{r3\}$.

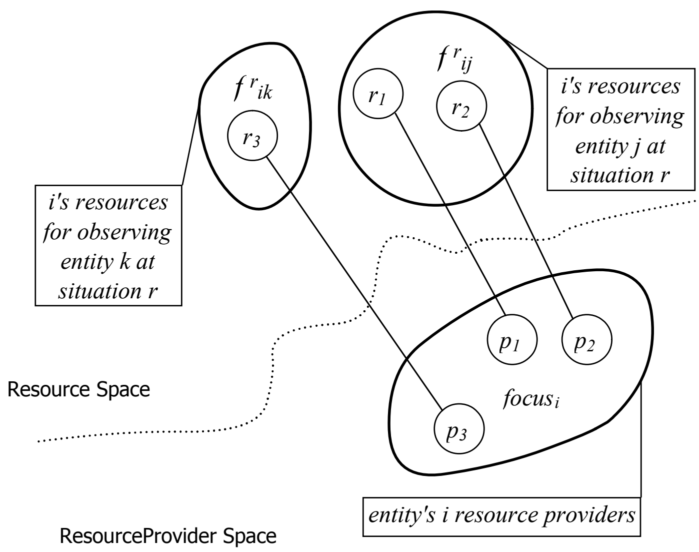


Figure 4. Focus of an entity i on the entities j and k respectively.

Going back to the scenario we introduced earlier, we can elaborate *Anna's* focus on *John*. *Anna* can check *John's availability* by pressing a small button on her *'aware-watch'*. System-wise we can consider that *Anna's Focus* is populated by a *resource-provider* that returns a *resource* for rendering *John's availability* whenever this small-button is pressed. This resource claims to render *John's availability* by highlighting an appropriate icon on her *'aware-watch'* display:

$$\begin{gathered} p2\text{: } ResourceProvider;\ p2 \in focus_{Anna} \mid \forall\ r\text{:}RealSituation; \\ (\ buttonpressed(r) \wedge (p2^{r}.aspect = availability) \wedge \\ (\forall s\text{:}\mathbb{F}\ Attribute;\ \ p2^{r}.render(s) = \\ if\ \ (\exists\ p\text{:}Attribute;\ p \in s \mid p.aspect = availability\ \wedge\ p.value = available)\ then \\ AvailableIconHighlight\ \ else\ \ BusyIconHighlight) \wedge (p2^{r}.e = John)) \vee \\ (\neg\ buttonpressed(r) \wedge p2^{r} = \varnothing) \end{gathered}$$

Thus *p2* is a *ResourceProvider* that when the button at *Anna's* aware-watch is pressed, *p2* returns a resource, which when provided with an attribute about *availability,* renders it by highlighting a corresponding icon (i.e. Available Icon or Busy Icon); *p2*.e denotes that the returned resource should be assigned to *John.* Consequently, *p2* is a resource provider in *Anna's* focus which when applied to a real situation *r*, returns a resource that can render *John's* availability.

Nevertheless one may question how *Anna* for example could *"focus at the same time both on John's and Doty's location"*. The trivial solution would be to assign two different resources, one for *Anna*, and one for *Doty.* Of course the question would be more subtle if it implied that an aggregation of *John's* and *Doty's* location (whatever that means) should be displayed. In this case we would model the pair of *John* and *Doty* as a community, let us say '*family*' that exposes the aggregated location of its members to *Anna*; sequentially *Anna's* focus would be redirected on the *'family'*. In later sections we will examine communities, and how community-properties can be modelled by means of the FN-AAR.

2.4.3 Observable Items and Focus/Nimbus Negotiation

Imagine a situation where *"A highlighted icon on Anna's aware-watch is flashing indicating that John is at the office."*

In the situation above, the flashing-icon is an *Observable Item*[4] that indicates to *Anna* whether *John* is at the office or not. It should be stressed here that by the term observable we do not imply that *Anna* is seeing the lamp or even whether she perceives it as an indication for *John's* location. We only stipulate that the lamp is available for observation, and that it is possible (in principle) for *Anna* to perceive. The icon may be highlighted regardless of whether she is looking at it or not. In an abstract view, what the entities system-wise are actually aware-of, are observable-

[4] The term observable does not imply a modality; information could be presented in any perceivable manner (auditory, visual, tactile, etc...).

items rather than the attributes themselves; whether an observable-item displays an attribute of some entity or not can be assessed with an explicit association such as the following[5]:

$$\begin{gathered}\_displays\_of\_ : ObservableItem \times\ Attribute \times Entity| \\ \forall\ o\colon ObservableItem;\ a\colon Attribute;\ e\colon Entity; \\ o\ displays\ a\ of\ e \bullet (o,a,e) \in \_displays\_of\_\end{gathered}$$

Nevertheless, building on the above example, the model asserts that in any situation there is a set of observable items that a given entity can observe. The set of observable items that are available to an entity characterize its awareness of other entities' situation and activities. In the context of an awareness system we consider that an entity *i* becomes aware about the state of an entity *j* through an awareness-characteristic function a_{ij} which under a given situation *r* returns the set of observable by entity *i* items that present information regarding entity *j*:

$$\begin{gathered}\forall\ i,j\colon Entity;\ a_{ij}\colon RealSituation \to \mathbb{F}\ ObservableItem| \\ r\colon RealSituation;\ o\colon ObservableItem; \\ o \in a_{ij}(r) \bullet \exists\ u\colon Attribute|\ o\ displays\ u\ of\ j\end{gathered}$$

In the aforementioned scenario we can state that

$$\begin{gathered}HighlightedOfficeIcon\colon ObservableItem\ | \\ HighlightedOfficeIcon\ displays\ (location{:}office)\ of\ John\ \wedge \\ \exists\ r\colon Realsituation\ |\ HighlightedOfficeIcon \in a^{r}{}_{Anna,\,John}\end{gathered}$$

i.e. in some situation *r, Anna* is aware of an observable-item (*HighlightedOfficeIcon*) referring to *John,* which indicates that his location is at the office. Note that it would be more appropriate to say 'potential awareness', since we have no information about *Anna's* physical (inherent) focus. For brevity, we use instead the term "*awareness*" and we imply a corresponding interpretation of statements such as "*Anna is aware of John's location*".

However, the definition of a_{ij} above is weak, as it does not specify the relation between what is available about *j* and how this is presented to *i*. Our interest is to describe a_{ij} more strongly, in coordination with the original focus-nimbus model, as a functional composition of nimbus and focus.

Figure 5 shows the attributes that an entity "*j*" makes available to an entity "*i*" at a situation "*r*" (i.e. *a1, a2, a3*) through n^{r}_{ji}. On the top-left we see their projection (*A*) on the *Aspect Space* i.e. the aspects they refer to. For example the attribute a_1 contains information about aspect *Y*, so its projection on the *aspect space* is *Y*. We

[5] Notice that this explicit association does not imply that a displayed attribute is of a particular entity, but rather that is being 'perceived' as such. Nevertheless, the exact definition of such an association is out of scope of this chapter, as it is rather specific to system implementation.

notice also the resources that i assigns for observing j at r (i.e. *r1,r2*) through f^r_{ij} and the resource projection (*B*) on the *Aspect Space;* i.e. the aspects that the resources claim to (i.e. are set to) render. For example, the resource r_2 claims to render the aspect *X*, so its projection on the aspect space is *X*. The intersection $A \cap B$, represents the aspects that i wants to observe about j, and j is making available to i at the situation r. Consequently, the set of items that i can observe about j (a^r_{ij}), are the result of rendering those attributes of n^r_{ij} that project on $A \cap B$ *(i.e. a2,*and *a3),* using those corresponding resources of f^r_{ij} that project on $A \cap B$ *(i.e. r1);* therefore (see bottom of figure 5) a^r_{ij} includes the observable item *o1=r2.render({a2,a3}).*

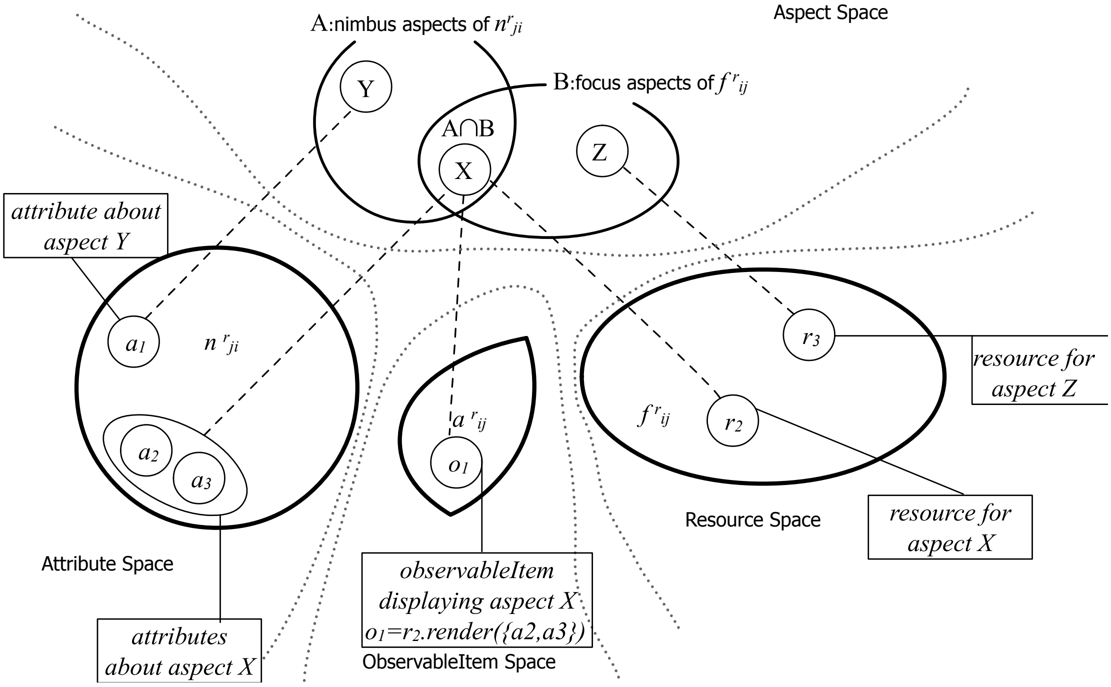


Figure 5. Illustration of focus-nimbus negotiation between an entity i and some entity j.

The negotiation of the reciprocal foci, and nimbi between two entities can be generalized as follows:

$$a_{ij} : RealSituation \nrightarrow \mathbb{F}\ ObservableItem|$$
$$\forall\ r: RealSituation;\ i,j:Entity;\ o: ObservableItem; \bullet\ o \in a_{ij}\ (r) \Leftrightarrow$$
$$(\exists\ c: Resource;\ c \in f^r{}_{ij}|\ o= c.render(\{u: Attribute;\ u \in n^r{}_{ji}|\ u.aspect=c.aspect\}))$$

i.e. at some situation r, an entity's i awareness of an entity j is characterized by the set of observable items that are the product of applying the rendering functions of the focus-resources of the entity i on entity j that match the aspects of the attributes that j exposes to i.

The above definition, does not take into account whether the observable items in a^r_{ij} are indeed displaying attributes of j, or whether any observable, by entity *i,* items are indirectly displaying attributes of j. We can elaborate on this formally by introducing a function that returns exactly those observable, by entity *i,* items that display, directly or indirectly, attributes of j:

$$a^{*}ij : RealSituation \nrightarrow \mathbb{F}\ ObservableItem \mid$$
$$\forall\ r: RealSituation;\ i,j{:}Entity;\ o: ObservableItem; \bullet$$
$$o \in\ a^{*}ij\ (r) \Longleftrightarrow \exists\ u{:}Attribute;\ k{:}Entity \mid o \in a_{ik}(r) \wedge o\ displays\ u\ of\ j$$

2.4.4 Ontology extensions

The FN-AAR model, as described so far, allows more than one attribute about the same aspect to be exposed from the nimbus of a single entity; for example one's state may include an attribute about "*location*" with value "*home*"*(location:home)*, and another attribute also about "*location*" with the value "*kitchen*" *(location: kitchen)*. Notice also that the model does not forbid one's state to include even contradictory attributes (thus allowing for imperfect technology or intentional misinformation by the user). For example *John* could make available to *Anna* an attribute about his *location* with value *home (location: home)*, and to his mother a contracting attribute *(location: away).* To elaborate, we can populate the attribute space with a relationship that denotes contradicting attributes:

$$\_contradicting\_ : Attribute \leftrightarrow Attribute;$$
$$\forall\ a,b{:}\ Attribute;\ a\ contradicting\ b \Longleftrightarrow (a,b) \in \_contradicting\_$$

Nevertheless, one may agree that an attribute(*a1*) about the aspect "*activity*" with value "*sleeping*" implies an attribute(*a2*) about the aspect "*location*" with value "*bed*", and the latter may imply an attribute(*a3*) about "*location*" with value "*home*" and so on. The exact ontological relationships and whether the ontology can be global, or application-specific, entity-specific or moreover situation-specific is out of the scope of this chapter. However given an implication relationship between attributes:

$$\_implies\_ : Attribute \times\ Attribute$$

We can define a function that returns all possible attributes that are implied from a single attribute:

$$impliedAttributes : Attribute \rightarrow \mathbb{F}\ \ Attribute\ ;$$
$$\forall\ a{:}Attribute;\ impliedAttributes(a)=\{u{:}Attribute \mid\ (a,u) \in \_implies\_^{*}\ \}$$
$$where\ \_implies\_^{*}\ is\ the\ reflexive\ transitive\ closure\ of\ \_implies\_$$

More generally we can take into account implications from attribute tuples, triads, quads, or from any set of attributes; we assume that the "*impliedAttributes*" function is extended to return all attributes implied from a set of attributes:

$$impliedAttributes : \mathbb{F}\ Attribute \rightarrow \mathbb{F}\ Attribute\ ;$$

The exact definition of this extensive function is out of scope; given its existence however, we can define n^{*r}_{xy} to return all implied attributes of n^{r}_{xy}:

$$\forall\ r{:}\ RealSituation;\ n^{*\,r}_{ij} = \{a{:}Attribute \mid a \in impliedAttributes\ (n^{r}_{ij})\}$$

Apparently such ontological associations could be incorporated in the focus-nimbus negotiation function to amplify the model's application. More importantly however, in following sections the key role of the ontological model on the attribute-space will become more evident as we explore notions such as deception, translucency, symmetry etc…

2.4.5 Closing the gap

In the model presented above, we addressed the question *"what are the entities aware of regarding each other in a particular situation"*. On the other hand, in the model definition we discuss notions such as observable items without regard to whether real world entities (such as actors) actually do perceive them and therefore are physically (inherently) aware of them. An interesting question would be whether we can connect the notion of observable-items, and the awareness-characteristic function a_{ij}, with the quantitative notion of modelling awareness with the original focus/nimbus model; i.e. to answer the question *"how aware(physically) is a physical entity (e.g. an actor) of an observable-item"*.

For that we can consider that each *observableItem* has an inherent/physical nimbus, and each *entity(actor)* has an inherent focus. The lamp in the simple example introduced in section *2.2* has an inherent nimbus that is defined by its physical position, its brightness etc. Likewise, *John* (as an actor) has an inherent (physical) focus that is defined also by his position, his posture, his eye-gaze etc. Apparently the composition of an entity's inherent focus with an observable item's inherent nimbus defines how aware the entity is of the observable item it-self. If we assume that a system has sufficient resources/capabilities to apply Rodden's focus-nimbus model in the *Entity-ObservableItem* relationship (i.e. we can define the focus/nimbus composition), then we can reason in detail about the information (observable-items) that one is physically aware-of.

Therefore we may think of a function n^{+} that associates an *ObservalbeItem* with its inherent nimbus in any situation, a function f^{+} that associates an *Entity* with its inherent focus in any situation, and an awareness quantifier function a^{+} :

$$n^{+}{:}\ RealSituation\ \times ObservableItem \rightarrow InherentNimbus;$$
$$f^{+}{:}\ RealSituation \times Entity \rightarrow InherentFocus;$$
$$a^{+}\ \ InherentFocus \times InherentNimbus \rightarrow InherentAwareness$$

For an *entity x*, and an *observableItem u*, the expression $a^{+}(f^{+}(r,x),\ n^{+}(r,u))$ quantifies the question *"How aware is entity x of observable item u at situation r"*. Assuming a predefined threshold h we can state that x is aware of u at situation r when its inherent awareness $a^{+}\ (f^{+}(r,x),n^{+}(r,u))$ is greater than the predefined threshold.

The feasibility of computing functions n^{+}, f^{+}, and a^{+} can be disputed, as they refer essentially to human perception and cognition. Yet, coming back to our simple scenario, we observe that the information extracted from *John's* plug-in, i.e.

whether John is using his computer, can also be used to approximate *John's* physical focus on any observable items that are rendered on the computer's display or on the computer's physical surroundings. Besides, it could be that *John* is using, among others, a small light indicator, attached to the USB port of his computer, which displays some information about *Anna*. The fact that the small light indicator is attached to *John's* computer intrinsically defines its physical nimbus, i.e. that the more one is near the light indicator, the more she is aware of it. Consequently, in a situation when *John's* activity plug-in detects that he is using his computer the system could infer that he is aware of the light indicator attached to the computer's USB port.

Generalizing, one's physical focus may be approximated with varying degrees of success by knowing whether they are present in front of the computer, or even further, monitoring their head pose or even their eye-gaze. In other words, an entity's nimbus can be used to approximate/define its inherent focus allowing reasonable approximations of n^+, f^+, and a^+.

It is important to notice here, that although we can model an entity being physically aware of an observable item, we can not assume that the entity is also cognitively aware of the presented information, since we do not model the cognitive processes of awareness (e.g., *Anna's aware-watch* may display *John's availability*, *Anna* may be physically aware of the displayed information, but still at the same time *Anna* may be unaware of *John's availability*). As noted, user cognition is outside the scope of the model presented here; such issues can be addressed by models such as the formalization of *performative interaction* (Dix et al., 2005) that enables for example the distinction among directly or indirectly perceived phenomena, or the model of Modica and Rustichini (1994) who formalize awareness by examining it in contrast with the notions of *unawareness, certainty*, and *uncertainty.*

2.5 Plausible deniability

The term plausible deniability has been often used,(e.g., Price et al., 2005; Aoki & Woodruff, 2005) to describe how users of communication systems may rely on ambiguity in order to have a plausible excuse for avoiding communication or interaction with a third party.

Price et al. (2005) explore the social need for plausible deniability in ubicomp systems and in relation with one's location and identity. As they point out, many systems depart from social norms that are otherwise present in face-to-face interactions (where a person can easily see whether he/she is being observed by others); their classification involves five types of user controlled *"noise"* to protect location privacy (*Anonymizing*, *Hashing*, *Cloaking, Blurring,* and *Lying)*.

In a similar line, Lederer et al. (2003) report that people decide to disclose information about their activities and location based on the identity of the requester and the situation in which it happens. Consolvo et al. (2005) introduce several requirements for location-aware applications. Among these they mention the need to support denial (e.g. the ability not to disclose any information), and deception (e.g. the ability to deceive in the response). In their studies, blurring (i.e. the ability to disclose less specific information) was encountered less frequently. Summarizing, we can identify three basic deceptive patterns:

- Deception/Lying: intentionally false information
- Denial/Cloaking: no information disclosure
- Blurring/Evasion: revealing part of the information

2.6 Modelling plausible deniability

In this section we will present and analyse several communication aspects that can emerge within an awareness system using the model described above. We will use as a starting point a slightly more elaborate scenario that involves a few actors and an awareness system that they use as means of lightweight communication:

"John is using an awareness system to communicate with his family, their daily activities. John is sharing his location with his wife Anna and his mother, so that they have a feeling about him. On the other hand, at John's office, he is using a digital-frame that helps him stay aware of his family's situation. Moreover John is able to see on the same display information about his colleagues activities and his own tasks at hand, helping him be more efficient at work…"

2.6.1 Blurring

In the above context we can imagine that among other things *"John is making available his location to Anna"*. Let us project this statement on the model: in terms of the system, a situation exists when (the entity that corresponds to) *John* makes available to (the entity that corresponds to) *Anna* some attributes about *John's* "*location*". The exact value of the attribute(s) about John's location can vary both in detail and accuracy; for example it could be *(location: home)*, *(location: away), (location: car)*, *(location: university-campus),* or *(location: auditorium)* and so on.

Likewise, we can imagine that *John* is also exposing some attributes about his location to his *mother*. However, in contrast to *Anna*, *John* is revealing less details to his *mother*; for example at a certain situation *r*, *John's* nimbus to *Anna* contains the attribute *(location: auditorium)* while his nimbus to his *mother* contains the attribute *(location: university-campus)*. This selective presentation of information about oneself can be for the purposes of self-presentation, politeness or simply privacy protection. In this case where information is presented at a diminished level of detail we talk of *'blurring'*. It is interesting to see how such patterns can be modelled.

While *"blurring"* typically refers to image processing, Price et al. (2005) describe "*blurring*" as the ability to decrease the precision of one's location. In a wider context we can replace "*location*" with any aspect of one's nimbus. To account with the term "*decrease*" we define "*blurring*" in comparison to a reference entity. Hence we consider that an entity is *blurring* information about an aspect to another entity, when the first is revealing *less information about this aspect* to the latter than a reference entity. The most typical example is when a video image from another person is blurred so that the identity of the individuals shown is not conveyed; see for example the use of filters by Hudson and Smith (1996).

Before proceeding to a formal definition let's consider the phrase "*less information about an aspect*". To evaluate the expression "*less information"* we may consider that if an attribute-set *A* is a subset of an attribute-set *B*, then the set *A* contains less information than the set *B.* For example a set that includes an attribute about *location* with value *home (location: home)* contains less information than the set *{(location: home), (location: bedroom)}* since the first set is a subset of the latter.

In our example however *John's* attributes about his location that are exposed to *Anna* is the set *{(location: auditorium)}* while to his mother is the set *{(location: university-campus)}*; apparently despite the second set being intuitively less informative that the first, it is not directly a sub-set of it. Nevertheless, in our scenario domain we can assume that *(location: auditorium)* implies the attribute (*location: university-campus),* hence deductively it can expand to the set *{(location: auditorium), {(location: university-campus)}* which is in turn a superset of *{(location: university-campus)},* thus allowing us to evaluate whether *John* is blurring his location to his mother.

On the other hand, a similar blurring effect can be achieved following quite a different approach. Let us consider that in some ontological association the attribute *(location: auditorium)* is contradictory to the attribute *(location: cafeteria)* while at the same time both imply the attribute *(location: university-campus)*. In this case, *John* could expose to his mother the attribute set *{(location: auditorium), (location: cafeteria)}*; this set would now expand to *{(location: auditorium), (location: cafeteria), ¬(location: auditorium), ¬(location: cafeteria), (location: university-campus)*}. Whether we assume that the interpretation of the expanded set is based on conjunction (i.e. that all the attributes in it are presented as valid) or that the interpretation of the expanded set is based on disjunction (i.e. that any of the attributes in it are presented as valid), we could say that it contains valid information only about the set *{(location: university-campus)}* which contains 'less information' than the set *{(location: auditorium), (location: university-campus)}.* In the first case (conjunctive interpretation) because the contradicting attributes lead to invalid statements, while in the second (disjunctive interpretation), and most plausible, the contradicting attributes lead to an arbitrary choice[6].

The above insights can be generalized to allow us to assess whether one set of attributes contains 'less information' than another, by first expanding the two sets with their respective implicative attributes, and then by removing their respective contradicting attributes before finally comparing whether the transformed first set

[6] Note the non monotonicity of evasion as a function of the exposed information. This is a similar but orthogonal phenomenon, to the non monotonicity of privacy and importance of information discussed by Dix (1990).

is a subset of the second. In this respect, taking in account a simple attribute-ontology like the one described earlier, we could now state that indeed both the attribute-sets *{(location: university-campus)}* and *{(location: auditorium), (location: cafeteria)}* contain less information than the set *{(location: auditorium)}*.

Generalizing the aforementioned observations we can formally define *'blurring'* by incorporating the ontological relationships of attributes, with the introduction first of a function *attributesAbout*, which when applied to a set of attributes and an aspect, it returns only those attributes that concern the specified aspect accounting for the term "*information about an aspect"*.:

$$attributesAbout : \mathbb{F}\ Attribute \times Aspect \rightarrow \mathbb{F}\ Attribute |$$
$$\forall\ s\colon \mathbb{F}\ Attribute;\ a\colon Aspect;$$
$$attributesAbout(s,a) = \{u{:}Attribute;\ u \in s |\ u.aspect{=}a\}$$

$$\_isBlurring\_to\_ : RealSituation \rightarrow \mathbb{P}(Entity \times Aspect \times Entity)\ |$$
$$\forall\ \ x, y\colon Entity;\ a{:}Aspect;\ r\colon RealSituation\ \bullet$$
$$x\ isBlurring\ a\ to\ y\ (r) \Longleftrightarrow (x,a,y) \in \_isBlurring\_to\_(r)\ \Longleftrightarrow$$
$$\exists\ z\colon Entity\ |\ \ attributesAbout(n^{*\,r}{}_{xy}, a) \subset\ attributesAbout(n^{*\,r}{}_{xz}, a)$$

i.e. an entity *x* is blurring information about an aspect *a* to *y*, when all the attributes about *a* that are made available to *y* (explicitly or by implication as denoted by '$n^{*\,r}{}_{xy}$'), are a subset of the attributes about *a* that are made available to an entity *z* (explicitly or by implication). Note that the reference entity *z* can be any entity including *x* itself.

Returning to the above scenario, as far as *John* is concerned, perhaps it is not as beneficial to reassure him that his location is presented 'blurry' to his *mother* compared to *Anna*, but probably it would be more beneficial that he can rest reassured that while he is exposing information about his location to his mother, she is not able to infer that he is at the office. We could generalize such statements in terms of the model as follows:

$$\_isBlurringAttribute\_to\_ : RealSituation \rightarrow \mathbb{P}(Entity \times Attribute \times Entity)\ |$$
$$\forall\ x, y\colon Entity;\ a\colon Attribute;\ r\colon RealSituation\ \bullet$$
$$x\ isBlurringAttribute\ a\ to\ y\ (r) \Longleftrightarrow (x,a,y) \in \_\ isBlurringAttribute\_to\_(r)\ \Longleftrightarrow$$
$$(a \notin\ n^{*r}{}_{xy}\ ) \wedge$$
$$(\exists u\colon Attribute;\ z\colon Entity;\ u.aspect{=}a.aspect;\ z \neq y\ |\ (u \in n^{*r}{}_{xy})\ \wedge\ (a \in\ n^{*r}{}_{xz}\ ))$$

i.e. an entity *x* is blurring the information of an attribute *a* to an entity *y*, when despite there exist attributes about *a*'s aspect that are exposed to *y* (explicitly or by implication), the attribute *a* itself is neither explicitly nor by implication exposed to *y* as opposed to some other entity *z*.

In the following sections and similarly to the definition of blurring we show how one can model other deceptive patterns, such as *'deception'* and *'denial'*.

2.6.2 Deception

Deception (lying) can be thought as giving intentionally false information about an aspect. We consider that an entity is lying when it is exposing to some other entity contradicting information about an aspect compared to its own information.

For example, consider an entity "*a*" that makes available to itself an attribute *(location: home)* whereas it makes available to entity "*b*" an attribute *(location: away)*. Given that *(location: home)* is contradictory to *(location: away)* we can state that "*a*" is lying to "*b*" about its *location*.

Bearing in mind a simple ontology like the one we described earlier, if entity "*a*" would make *(activity:sleeping)* available to itself, then the predicate *"a is lying to b about its location*" would still hold since in the context of the specific ontology, the attribute *(activity: sleeping)* implies *(location: home)* which contradicts to the attribute *(location: away).* Following the above we can formalize deception/lying:

$$\_isLyingTo\_About\_ : RealSituation \rightarrow \mathbb{P}\ (Entity \times Entity \times Aspect)|$$
$$\forall\ r{:}RealSituation;\ x, y\ Entity;\ a{:}aspect;$$
$$x\ isLyingTo\ y\ About\ a\ (r) \Longleftrightarrow\ (x, y, a) \in\ \_isLyingTo\_About\_(r) \Longleftrightarrow$$
$$\exists\ u, v{:}Attribute \mid u \in n^{*r}{}_{xy} \wedge v \in n^{*r}{}_{xx}\ \wedge u.aspect{=}a \wedge u\ contradicting\ v$$

i.e., *x* is lying to *y* about an aspect *a*, when there is at least one attribute about *a* that is made available to *y* (explicitly or by implication), such that it contradicts with an attribute that *x* makes available to him/her-self (explicitly or by implication).

2.6.3 Denial

Denial (*Cloaking)* is the ability to hide one's location or identity (Price et al., 2005). More generally, cloaking can concern any aspect of one's nimbus. Hence we consider cloaking as the ability to conceal any attributes about an aspect of an entity from another entity.

For example, consider an entity '*a'* that makes no attributes available to some entity '*b'* about its *location*, whereas it makes available an attribute *(location: home)* to an other entity 'c'. We can say in this example that '*a'* is hiding its *location* from *'b'.*

Taking into account ontological associations among attributes, as we did earlier, we could say that even if only an attribute *(activity:sleeping)* would be exposed to entity "*c*", the predicate "*a* is hiding its *location* from *b"* would still hold since in the context of the specific ontology, *(activity: sleeping)* implies several attributes about *location* such as *(location: bedroom)* and *(location: home).* In general we can introduce a formal definition for cloaking as follows:

$$\_isHiding\_From\_ : RealSituation \rightarrow \mathbb{P}(Entity \times Aspect \times Entity) |$$
$$\forall\ r{:}RealSituation;\ x, y\ Entity;\ a{:}aspect;$$
$$x\ isHiding\ a\ from\ y\ (r) \Longleftrightarrow (x, a, y) \in \ \_isHiding\_From\_(r) \Longleftrightarrow$$
$$\exists\ z{:}\ Entity\ |\ \ (\exists\ u{:}Attribute\ ;\ u \in\ n^{*r}_{xz} \wedge u.aspect{=}a) \wedge$$
$$\neg(\exists\ v{:}Attribute;\ v \in\ n^{*r}_{xy} \wedge v.aspect{=}a)$$

i.e., *x* is hiding an aspect *a* from *y*, when there are no attributes about *a* that are made available to *y* either explicitly or by implication, and at the same time there is at least one attribute about *a* that *x* makes available to an other entity *z*. Note that *z* can be any entity including *x* it-self.

To demonstrate the application of the aforementioned formalizations of deceptive patterns in a graphical representation, let us consider the following example:

"Anna and John have been using for some time an awareness system, to let each other know about their activities and situation. As Anna is spending a lot of time in the house moving from one room to another, she thought that she should not overwhelm John with constant updates of detailed information about her location; therefore she configured the system to inform him that she is either home (location: home), or away (location: away)."

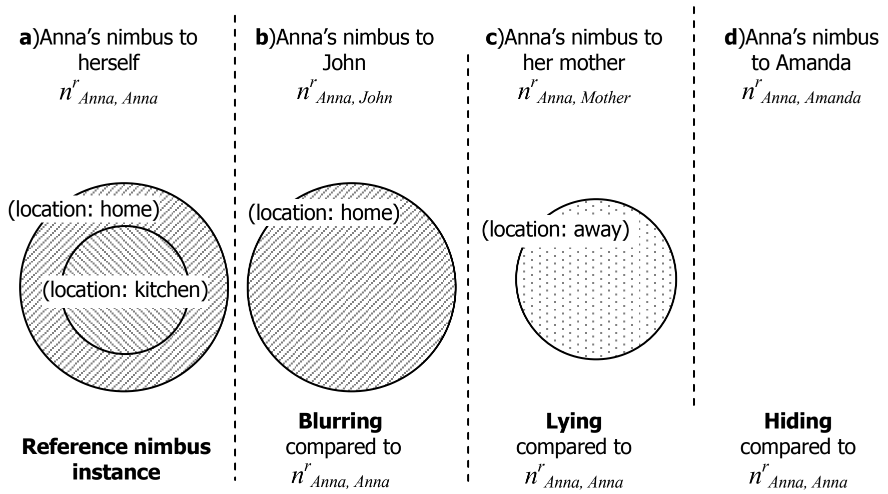


Figure 6. Anna's nimbus-instance emerging various deception patterns

The above example implies that if Anna were in the kitchen (figure 6.a), the system would expose to John just an attribute *(location: home)*; in such a situation the system apparently would be blurring her location to John (figure 6.b). In formal terms we can write:

$$(\exists\ z{:}Entity;\ z{=}Anna\ |\ attributesAbout(n^{*}_{Anna,John}\ ,\ location) =$$
$$\{(location{:}home)\}\ \subset \{(location{:}home),(location{:}kitchen)\} =$$
$$attributesAbout(n^{*r}_{Anna,z}\ ,\ location)) \Longleftrightarrow$$
$$Anna\ isBlurring\ \ location\ to\ John$$

Continuing the example above let us consider the following scenario:

"Recently Anna convinced her mother to install a similar system so that they can keep in touch easier. After a while Anna realized that her mother was calling her exactly when she would come back home. So Anna configured the system to allow her some time when she is coming home, before exposing her new location to her mother."

In a situation that Anna is just back at home the system apparently is reporting her location as away, 'lying' to her mother about Anna's location (figure 6.c). We could write formally:

$$\begin{gathered}Anna\ isLyingTo\ mother\ about\ location \Longleftrightarrow \\ \exists\ u, v{:}Attribute; u=(location{:}home); v=(location{:}away) \mid \\ u \in n^{*r}{}_{Anna,mother} \wedge v \in n^{*r}{}_{Anna,Anna} \wedge u.aspect=location \wedge u\ contradicting\ v\end{gathered}$$

At last let us consider the following scenario

"Anna's neighbour Amanda has also installed a similar system. Amanda is a curious person, and wants to know everything; so she assigned a 'where-about-display' to keep an eye on Anna's location. Obviously this doesn't do much because Anna is not exposing any attributes about her location to the neighbours."

In this situation, the system is hiding Anna's location from Amanda (figure 6.a). Formally we could write:

$$\begin{gathered}Anna\ isHiding\ \ location\ \ from\ Amanda \Longleftrightarrow \\ \exists\ z{:}\ Entity; z=Anna\ \mid \\ (\exists\ u{:}Attribute\ ; u=(location{:}home);\ u \in\ n^{*r}{}_{Anna,Anna} \wedge u.aspect=location) \wedge \\ \neg(\exists\ v{:}Attribute; v \in\ n^{*r}{}_{Anna,Amanda} \wedge v.aspect=location)\end{gathered}$$

2.7 Social translucency

Erickson et al. (1999) examine the notion of social translucency and socially translucent systems; i.e. "*systems which provide perceptually based social cues which afford awareness and accountability*". They state the need to make socially salient information visible in communication applications. In this context, the social norms that influence people's behaviour towards each other are brought to discussion.

We can summarize some of these norms in statements like:

- *"Because I know your situation, I adjust my behaviour accordingly".*
- *"Because I know you know my situation, I adjust my behaviour accordingly".*
- *"Because I know that you know that I know your situation, I adjust my behaviour accordingly".*

To reflect on the above statements let us consider that *John* and *Anna* share their mood for walking using a rudimentary system: When one of them feels like walking (s)he flicks a switch and a lamp lights up at the other side indicating his/her wish.

Imagine that *John* wants to go for a walk, and *Anna* becomes aware of his wish. *Anna* knowing the situation of *John* could respond to it, e.g., by calling him to arrange going for a walk together; or better, we can state that "*because **Anna knows** John's situation, she adjusts her behaviour accordingly".*

Now, let us think that the system provides an additional feedback on *John's* site that lets him know that *Anna's* lamp is enabled, and assigned to display his (*John's*) mood. So *John* knows (assumes) that *Anna* knows (or could know) his situation (if *Anna* is near the lamp); therefore *John* waits for a couple of minutes for a reaction from *Anna* before going for a walk alone. In contrast, if *John* would see that *Anna's* lamp is disabled he could leave for a walk directly. In other words, *"because **John knows that Anna knows** his situation, he adjusts his behaviour accordingly".*

Finally suppose that *Anna* knows that the system informs *John* whether she is using the lamp, as mentioned earlier. So *Anna* may think that it is impolite not to respond to *John.* Actually, although she is not keen to go for a walk, she decides to join him, i.e. *"because **Anna knows that John knows that she knows** his situation, she adjusts her behaviour accordingly".*

2.8 Modelling social translucency

2.8.1 Internal translucency

The first statement, i.e. *"because I know your situation, I adjust my behaviour accordingly",* is already captured in our model as described up to this point. Indeed the essence of any awareness system is to allow entities to adjust their behaviour based on the knowledge of the situation of others.

However a non trivial statement that is not directly addressed by the fundamental definitions of our model is the following: *"because I know **my** situation, I adjust **my** behaviour accordingly".* Consider for example that *Anna* is in her living room on a Sunday evening. If she were aware that the bright lighting of the room allows passers-by to gaze at her, she could potentially avoid socially embarrassing situations. In situations like this, which unfold in the physical world, our prior knowledge and experiences enable us to be more or less aware of our "*nimbus*". However, this capacity is hampered when it comes to our presentation through networked applications. Therefore, one of the properties we might wish to apply in a mediated environment is that of "*Internal translucency*".

We can summarize *internal translucency* in the statement "*I am aware of my nimbus*". Thus, I am aware of the information that I am making available to you. This statement involves both "*I focus on my nimbus*" and "*I **can** be aware of my nimbus".* The first ("*I focus on my nimbus*") signifies that I am focusing on the information that I am making available to you. The second ("*I **can** be aware of my nimbus"*) signifies that the information about me that is available to you is also available to me. This may sound redundant, but in the context of an awareness system it is not necessarily the case, since there may be (privacy threatening) situations where an entity is unable to be aware of its nimbus.

The statement "*I can be aware of my nimbus (to you)"* is equivalent to the statement *"I expose to myself my nimbus to you"* or, in terms of the model:

$$\_\ canBeAwareOfItsNimbusTo\_ : RealSituation \rightarrow (Entity \times\ Entity)|$$
$$\forall\ r{:}RealSituation;\, x, y{:}Entity;\bullet\ \ x\ canBeAwareOfItsNimbusTo\ y \Longleftrightarrow$$
$$(x, y) \in\ \_canBeAwareOfItsNimbusTo\_\ (r) \Longleftrightarrow$$
$$\exists\ u{:}\ Attribute;\, u \in n^{r}_{x,x}\ |\ u.aspect{=}\text{'}nimbus\ to\ y\text{'} \wedge u.value = n^{r}_{y,z}$$

Hence in the above definition we consider that an entity *x* exposes to itself its nimbus to an entity *y,* when there exists an attribute in *x's* nimbus to *x* about the aspect "*nimbus to y*" that has as value *x's* nimbus to *y (i.e.* $n^{r}_{x,y}$*).* Note how this definition considers that the hole instance of *x*'s nimbus to *y* is exposed to *x* itself, i.e. all the attributes that the entity *x* is exposing to *y* are made available to *x* as an explicit value of an attribute about the aspect '*nimbus to y*'. One could also modify this to account for different levels of detail; for example, to expose only the set of aspects that are included in *x's* nimbus to *y*, or even to expose the actual function-body of the relevant attribute provider(s) as a value of the attribute about the '*nimbus to y'*.

Besides, the statement "*I focus on my nimbus to you"* is equivalent to the statement *"there exists at least a resource in my self-oriented focus that renders the attributes about by nimbus to you'"*:

$$\_isFocusingOnItsNimbusTo\_ : RealSituation \rightarrow (Entity \leftrightarrow Entity)\ |$$
$$\forall\ r{:}RealSituation;\, x, y{:}Entity;\bullet\ \ x\ isFocusingOnItsNimbusTo\ y \Longleftrightarrow$$
$$(x{,}y) \in \_\ isFocusingOnItsNimbusTo\ \_(r) \Longleftrightarrow$$
$$\exists\ v{:}Resource;\, v \in f^{r}_{x,x}\,|\, v.aspect =\ \text{'}nimbus\ to\ y\text{'}$$

One could consider that *x* is aware of its nimbus to *y, w*hen both of the aforementioned statements are satisfied. However, we can not assume a priori that a focus-resource presents its corresponding aspect successfully (e.g., due to poor design an attribute is mal-presented). Yet, earlier we introduced the relationship "*displays*" that relates an observable item to the attribute(s) it presents successfully; using *'displays'* we can clarify the statement "*I am aware of my nimbus to you",* by taking in account whether *"the observable items that I can see indeed display my nimbus to you"*:

$$\_\ isAwareOfItsNimbusTo\_ : RealSituation \rightarrow (Entity \leftrightarrow Entity)\ |$$
$$\forall\ r{:}RealSituation;\, x, y{:}Entity;\bullet\ \ x\ isAwareOfItsNimbusTo\ y \Longleftrightarrow$$
$$(x, y) \in \_isAwareOfItsNimbusTo\_(r) \Longleftrightarrow$$
$$(\exists\ o{:}ObservableItem;\, o \in a^{r}_{x,x}\,|\, o\ displays\ (\text{'}nimbus\ to\ y\text{'}{:}\ n^{r}_{x,y})\ of\ x)$$

Thus, we consider that an entity *x* is aware of its nimbus to an entity *y* when there is an observable-item of which the entity *x* is aware of, such that it displays the entity's *x* nimbus to *y*.

To demonstrate the potential of the above formalizations in a real system, let us consider the scenario introduced in an earlier section, where John uses a plug-in counting the number of open windows on his desktop computer, as an indication of his availability to Anna. We can elaborate this scenario as follows:

"… This of course quite often leads to misinforming Anna. Therefore John added on his computer an indication of his activity as it is detected by the system, allowing him to manually change it when he disagrees with the system's assessment."

Up to this point we can see that John, by displaying on his computer his extracted availability as it is exposed to Anna, has engaged a strategy in which he is aware of his nimbus. We can also see how the system could benefit by detecting John's strategy and enhance its abilities:

"The plug-in is able to detect that John is now aware of his nimbus. So it makes the assumption that if John is not approving of the extracted value for his availability he will change it manually. Therefore the plug-in increases its confidence on the extracted attributes (e.g., instead of displaying "probably-busy" it displays just "busy").

We see therefore how both users and systems can mutually adapt to each other's behaviour to enhance the conjoint performance of the system.

2.8.2 External translucency

As a starting point for the concept of external translucency, we use Erickson's statement *"because I know that you know my situation, I adjust my behaviour accordingly"*. We can broaden this statement to *"because I know that you know* ***mine or someone else's situation****, I adjust my behaviour accordingly"*.

Imagine for example that *Anna* and *John* are using an awareness system to keep an eye on their daughter *Doty*. *Anna,* apart from periodically checking *Doty*'s activities, makes available to *John* her focus. *John* can therefore focus on *Anna's* focus to check if she is focusing on *Doty;* hence he can decide if he also needs to check (focus) on their daughter. In a broader sense, because *John* knows that *Anna* knows *Doty's* situation, he adjusts his behaviour (in this case his focus on *Doty*) accordingly.

Based on the aforementioned insights we summarize external translucency in the statement *"I am aware of your focus"*. Thus, *"I am aware of what you are focusing on me (and possibly other entities)"*. This statement involves both *"I focus on your focus"* and *"I can be aware of your focus"*. The first (*"I focus on your focus"*) signifies that some of my focus resources are assigned to display your focus. The second *("I can be aware of your focus"*) signifies that your focus (e.g., the focus resources that you assigned to render information that I or others make available to you) is made available to me. Hence you allow me to observe how you are observing me (or other entities).

In more detail, the statement "*I can be aware of your focus on me (or someone else)"* is equivalent to the statement *"you expose to me your focus on me (or someone else)"*, or that there exists an attribute which indicates your focus on me included in your nimbus to me (i.e. an attribute about the aspect *"focus on me/someone else"*):

$$\_exposesTo\_ItsFocusOn\_ : RealSituation \rightarrow \mathbb{F}(Entity \times Entity \times Entity) |$$
$$\forall\ r{:}RealSituation;\ x,\ y,\ z{:}\ Entity; \bullet\quad y\ exposesTo\ x\ ItsFocusOn\ z \Longleftrightarrow$$
$$(y,x,z) \in \_exposesTo\_ItsFocusOn\_\ (r) \Longleftrightarrow$$
$$\exists\ u{:}\ Attribute;\ u \in n^{r}_{y,x}\ |\ u.aspect = \text{'focus on } z\text{'} \wedge u.value = f^{r}_{y,z}$$

Hence in the above definition we consider that an entity *y* exposes to an entity *x* its focus on an entity *z,* when there exists an attribute in *y's* nimbus to *x* about the aspect "*focus on z*", that has as value *y's* focus on *z (i.e.* $f^{r}_{y,z}$*).* Note that this definition considers that the whole instance of *y's* focus on *z* is exposed to *x,* i.e. all the resources that entity *y* has assigned for observing *z* are made available to *x*. One, however, could easily modify the above definition for different levels of detail, for example, expose only the set (or a subset) of aspects that are included in *y's* focus on *z,* or even the actual body of the render functions in each resource of *y*'s focus on *z* etc.

The statement *"I focus on your focus"* can be formalized by claiming the existence of a resource in my focus that renders your exposed attribute(s) about your focus on me (or other entities):

$$\_isFocusingOnTheFocusOf\_On\_{:}RealSituation \rightarrow \mathbb{F}(Entity \times Entity \times Entity) |$$
$$\forall\ r{:}RealSituation;\ x, y, z{:}Entity; \bullet$$
$$x\ isFocusingOnTheFocusOf\ y\ On\ z \Longleftrightarrow$$
$$(x, y, z) \in \_isFocusingOnTheFocusOf\_On\_(r) \Longleftrightarrow$$
$$\exists\ v{:}Resource;\ v \in f^{r}_{x,y}\ |\ v.aspect = \text{'focus on } z\text{'}$$

We can formalize the statement *"I am aware of your focus on me"* similarly to the case of internal translucency:

$$\_isAwareOfTheFocusOf\_On\_\ {:}RealSituation \rightarrow \mathbb{F}(Entity \times Entity \times Entity) |$$
$$\forall\ r{:}RealSituation;\ x, y, z{:}Entity; \bullet$$
$$x\ isAwareOfTheFocusOf\ y\ On\ z \Longleftrightarrow$$
$$(x, y, z) \in \_isAwareOfTheFocusOf\_On\_(r) \Longleftrightarrow$$
$$\exists\ o{:}ObservableItem;\ o \in a^{r}_{x,y}\ |\ o\ displays(\text{'focus on } z\text{'}{:}\ f^{r}_{y,z})\ of\ y$$

Hence we consider that an entity *x* is aware of an entity's *y* focus on *z*, when there exists an observable item (that *x* is aware of) that displays *y's* focus on *z.*

To demonstrate the potential of implementing the above in a real system let us build up the scenario we introduced earlier in this section.

"John is quite satisfied with the modifications he made. Now he can always check whether the system is correct and change his availability if he disagrees. The only problem is that the icon that displays his own availability to him, takes too much space on his desktop. John asked Anna to expose her focus to him, so that he can tell when she is interested in his availability. Now John's plug-in is able to detect that Anna exposes her focus on him, therefore it only has to display to John his availability to Anna when she is indeed focusing on him."

Using similar notation, a wide range of relevant statements can be formalized, such as *'I am aware of your nimbus to everybody in a particular situation'*, or *'I can be aware of the entities you are focusing on regarding an aspect'*, or *'I am aware of what exactly you are aware of me'*, or even *'I am aware of how the information I'm exposing to you will be displayed to you'*, and so on.

2.9 Symmetry

Symmetry is a concept with intuitive appeal in the domain of groupware and, more generally, the research area concerned with technologies designed to mediate the interactions between individuals, whether for work or for social purposes. Researchers have put forward and demonstrated in research prototypes several concepts that are assumed to represent desirable properties for such technological media.

For example, Stefik et al. (1987) described the notion of WYSIWIS (What You See Is What I See) as a property desirable for software tools to support group meetings; their definition of this property was that every meeting participant should be able to view exactly the same information about the meeting and also to see where everyone is pointing. There can be several variations to this theme. Broadly speaking, the emphasis of this work as well as of related efforts that aim to support collaboration pertains to the equal access to shared content and the awareness of the interactions of group members upon this content.

Elaborating on the notion of awareness regarding interactions by others, Erickson et al. (2002; 1999), Erickson and Kellogg (2000) introduced the notion of social translucence, a property of the physical world that enables the application of social norms relating to the accountability of individuals. This accountability, they argue, is a result of the visibility of each other's interaction and of a common ground that is created between individuals as they become aware of each other's understanding of a situation. The concept of social translucence is very akin to, but different to symmetry. Social translucence in the sense discussed by Erickson et al. does not in itself require or entail symmetry; moreover, it is a weaker requirement than WYSIWIS with regards to minimizing inequalities between participants.

Social translucence is an evocative property when discussing media spaces and awareness systems. It means that an individual cannot observe while unobserved, and is thus subject to social norms regarding the observation of another; see Friedman et al. (2006) for a survey on the differing attitudes people hold regarding the roles of observer and observed. Social translucence can be seen as the telling difference between a system supporting interpersonal awareness and a monitoring system.

The possibility of seeing without being seen, to hear without being heard has been noted early on by several researchers in Media Spaces, e.g. Root (1988), as a breakdown of the symmetry that characterizes interactions in the physical world (Gaver, 1992). Heath and Luff (1991) discussed the notion of 'disembodiment', the effect of separating one's ability to act and to perceive a mediated world.

Researchers at the time attempted to repair this asymmetry by adding cues to warn individuals when they were being observed. For example, RAVE (England et al., 1996), a classic media space prototype provided visual cues to the observed to alert them when recording was taking place, this with the intention to compensate for the disembodiment of the observer-observed relationship.

Bellotti and Sellen (1993) proposed a design framework for personal privacy in ubicomp environments; they argued in favour of symmetric communication to overcome privacy concerns. Symmetry was understood as the concurrent exchange of the same information in both directions between two individuals (e.g., both are observers and observed).

The concept of symmetry appears to have a direct interpretation and an intuitive appeal in the context of media spaces. In the context of systems where information exchanged might be collected through context sensing or through user input, this concept is not as self-evident. Typical usage of such systems is asynchronous as they decouple the production of information from its consumption for the purposes of awareness. This decoupling can accentuate potential asymmetries. Also, the information captured and disseminated may be of the same nature but may carry radically different significance in context for the particular individuals that are connected through technologies.

For example, during the design and evaluation of the DIARIST system (Metaxas et al., 2007) supporting awareness of daily life activity between an elderly person living alone and his/her adult children, it became clear that while both parties were interested to have awareness of each other, they wished to be aware of very different types of information.

Even a concept such as social translucency is not universally applicable and desirable in this domain. More often than not, users of awareness systems will need means to equivocate rather than be accountable to the many others that they can connect to; 'leaving room for ambiguity' (see Aoki and Woodruff, 1995). A clear scenario where translucency is not desirable is the following.

"Anna, an elderly person living alone can indicate to her neighbouring friend John that she would like to go for a walk. John has the ability to do the same (symmetry) but both would rather maintain plausible deniability in cases of declining an invitation."

Thus, for both the observed and the observer the act of observing the display should not be observable to the other party. For both sides in this scenario it is important not to hurt the feelings of the other and by the same token not to lose face.

In other occasions, social translucency is not at all sufficient and a stronger form of reciprocity is required. An elderly man living alone may be happy to know when his daughter's family are consulting their awareness display for information about his whereabouts and activities (thus having a social translucent system), but would be more interested in being aware of the whereabouts and activities of his daughter and her children (thus having a symmetrical system). Perhaps even, the information he wishes to know is not of the same type as what they know of him but carries equal value and represents a corresponding level of openness and sharing as he is exhibiting. For example, while his children may be interested to know if he is sleeping and eating well, he is probably more interested in knowing if his children are stressed at work, or when a next visit is planned. It appears that the discussion on symmetry and related issues requires a more nuanced understanding of awareness and awareness systems that focuses on the information-sharing between users and even the inferences users can draw from information shared, and the value those inferences present to users.

The tension between efforts of the research field to mitigate the asymmetrical nature of such technologies has been revealed by a rich and growing collection of empirical studies and is discussed in depth by Voida et al. (2008) who propose a classification of asymmetries that relate to the use of media spaces and awareness systems. These are:

- Media asymmetry, when one is observing through one modality but cannot be observed through this modality as well.
- Fidelity asymmetry, when the level of detail and frequency of information of the observers differs to that of the observed, e.g., when giving independent controls for mechanisms such as blurring (Hudson & Smith, 1996).
- Participation asymmetry, when different individuals may have different levels of participation in the community, potentially being excluded from activities that others can join or having differential level of access.
- Engagement asymmetry, when individuals may be active at different levels and in ways that are inherently asymmetrical, e.g. preferring the role of consumer or producer of information.
- Benefit asymmetry. As in all types of groupware, the way benefits are distributed may not match the way costs are distributed (Grudin, 1994).
- Place Asymmetry. A system may connect places in which different norms apply and where different affordances are perceived by users, e.g. a private office to a mobile user in the train.

This classification and the related discussion by Voida et al. (2008), even when symmetry can be achieved as a property of the medium, it is a very frail one: a broad range of contextual factors outside the control of system designers can lead to asymmetrical mediated social interactions. This should come as no surprise; it appears to be in the nature of humans and groups to create structures of power, paths of communication and inequalities of participation in almost any social situation; see for example Klein (1956). Mediated settings are no exception. Sensitizing designers of awareness systems to potential inequities and how to mitigate asymmetry or its negative effects can be a key to design successful awareness systems and generally groupware.

2.10 Modelling symmetry

The following sections attempt to take a closer look at what symmetry could mean in awareness systems. Considering the above classification of potential asymmetries, we note how these are emergent properties that characterize the relation between individuals, their context and the technology at hand. At a first glance the last three categories are almost out of the control of designers. However, the notion of media, fidelity and participation (a)symmetries relate directly to the nature of the information presented, the participation structures encouraged by a system design and the way users can or cannot control the fidelity of the information they share with each other. For all the above it becomes necessary to examine more closely how information sharing takes place in awareness systems, and what is the vocabulary based on which researchers communicate and reason about the otherwise vague concepts surrounding symmetry. Let us first examine an earlier attempt to model such notions based on information flows between communicating individuals.

Jiag et al. (2002) considered information sharing between users in ubiquitous computing environments. Inline with Belloti and Sellen (1993), they put forward the principle of minimum asymmetry as a key objective for minimizing the imbalance between the persons about whom data is being collected, and the systems and people that collect and use that data. As they point out, several privacy concerns are raised in communication systems due the lack of symmetry; the source of asymmetry comes from the fact that one entity knows a great deal of information about another entity, while the latter knows little about the first. Therefore, they propose that a privacy-aware system should minimize the asymmetry of information flow by:

- *Decreasing the flow of information* from data owners to data collectors/users
- *Increasing the flow of information* from data collectors/users back to data owners

This principle is demonstrated through the Approximate Information Flow (AIF) model that supports varying degrees of asymmetry within ubicomp systems. With this model they point out three main ways that can help minimize asymmetry: *prevention*, *avoidance*, and *detection*. The AIF model introduces the notion of privacy zones within information spaces to support varying degrees of information asymmetry, and hence privacy, in ubicomp systems.

A mapping near to the notions of the AIF model was introduced by Hong and Landay (2002) who developed "confab", an architecture framework that allows users to augment their private information with information flow policies that are applied during communication enforcing privacy regulation.

While perhaps relevant in the case of document access and sharing in a ubicomp environment, e.g., for collaboration purposes, the AIF model and the interactions it assumes do not capture the type of sensitivities that arise in the context of mediated social interaction. To illustrate this, let us examine a more extreme scenario, which may be unusual for the research community to discuss but brings to life actual sensitivities of people about making others aware and being aware. Such scenarios are played out regularly in applications used for social communication and social networking. In real life, asymmetries underlying these acts can have high emotional and social costs as we hope to illustrate with the scenario below.

"John and Anna are two colleagues and both are engaged in a relationship with someone else. Both are registered as friends in an online social network. John is, however, interested in having an affair with Anna as well, so he wants to know Anna's emotions towards him."

In the example above and in terms of the AIF model John is acquiring from Anna information about her feelings towards him. So the primary privacy concern of the system would be to protect Anna's privacy. Paraphrasing this to the confab architecture, we could expect a popup message on Anna's side with the prompt *"John wants to know whether you are interested in an affair with him"*. As one can observe, however, the principle of minimum asymmetry should instead apply primarily to John, as his request for Anna's information is a privacy threatening situation for him. Consequently, the application of the principle of minimum asymmetry in the above situation should happen on John's request rather than on Anna's response. This points out that we can not have a de facto separation of users in those who own information and those who collect information; in the example above John's request implies by itself a great deal of private information about him. The latter is not predicted and, to our knowledge, can not be applied within AIF and confab, or if it would, the result would be no information exchange overall.

We attribute the above counter-privacy threat exactly on the coupling of the principle of minimal asymmetry with the notion of social translucency. It is social translucency, and the fact that Anna knows what John wants to know that makes John (undesirably) accountable for the information he is inquiring of her. Both in terms of computing and in real life we can not disregard from privacy regulation and symmetry the information carried by a request. Hence, it is important to regulate the communicational symmetry prior to information acquisition requests and information exposure responses in order to take in account the privacy threats that they both entail.

Another profound issue regarding the principle of minimum asymmetry and the regulation of privacy that we want to point out is the need to abstract away from information flow and focus on the information in itself. Even in the trivial example of location sharing, the application of the principle of minimum asymmetry can not be applied successfully without allowing a richer vocabulary for symmetry. The so called concept of "privacy zone" by itself is not sufficient to cover some of the related symmetry issues we want to address here, and that affects directly the quality of mediated communication. In this section we try to demonstrate how some deeper symmetry related notions can be addressed through the use of the FN-AAR model.

2.10.1 Media (a)symmetry

Media asymmetry implies that individuals share information through different modalities, as when one is observing through one modality (e.g., video) but is observed through another (e.g., audio). Conversely, we may conclude that media symmetry is apparent when individuals share reciprocal information through similar media.

The FN-AAR model incorporates the notion of "*Observable Item*" in order to describe the binding of information to rendering contexts. In the trivial case where we are not concerned about which aspects are being shared, we could form two sets with the modalities being used for observation among two individuals and conclude whether there is a media symmetry or not in their relationship. In a more thorough examination we may take into account that the information being exchanged among entities reflects a variety of aspects. This gives us the ability to examine the observable items among two entities (individuals) and compare them for their respective modalities for some aspect that is being shared. We may for example filter all the '*activity*'-related observable items of two individuals observed about one another, and compare the modalities where the information is presented to account for media (a)symmetry regarding the aspect "*activity*".

As introduced by the model earlier, the awareness characteristic function a^r_{xy} yields the observable items that the entity x is able to observe at some situation r relevant to information that the entity y is exposing to x. Let us first define the function '*relevantObservableItems*' which yields the observable items of a set O that display attributes about a specific aspect a.

$$\begin{gathered} relevantObservableItems\colon\ (\mathbb{F}\ ObservableItem \times Aspect) \rightarrow \mathbb{F}\ ObservableItem \mid \\ x \in\ relevantObservableItems(O,a)\ \Longleftrightarrow \\ (x \in\ O)\ \wedge\ (\ \exists\ u\colon Attribute;\ (u.aspect{=}a) \wedge x \in (x\ displays\ u)) \end{gathered}$$

Considering that each observable item has a property *medium* characterizing its contextual modality, we may define the function '*mediaOf*' to yield the involved mediums in a set of observable items:

$$\begin{gathered} mediaOf\colon\ \mathbb{F}\ ObservableItem \rightarrow \mathbb{F}\ Medium \mid \\ m \in mediaOf(O) \Longleftrightarrow\quad \exists\ o\colon ObservableItem \mid o \in\ O \wedge o.medium{=}m \end{gathered}$$

With the help of the above functions it becomes straightforward to examine the media symmetry among two entities x, and y for some aspect s by comparing the mediums of the relevant observable items:

$$\begin{gathered} \_\ displayMediaSymmetrical\_\colon RealSituation \times Aspect \rightarrow \mathbb{F}(Entity \times\ Entity) \mid \\ \forall\ r\colon RealSituation;\ x, y\colon Entity;\ s\colon Aspect \bullet\ \ x\ displayMediaSymmetrical\ y\ \Longleftrightarrow \\ (x,y) \in \_displayMediaSymmetrical\_(r, s) \Longleftrightarrow \\ mediaOf(relevantObservableItems(a^r_{xy}\ , s)) = \\ mediaOf(relevantObservableItems(a^r_{yx}, s)) \end{gathered}$$

As the above formalism implies, we consider that two entities x, and y are engaged in media-symmetry at some situation r regarding an aspect s if and only if the media that they used for observing each other are of the same type.

Consider, for example, a situation where a camera at Anna's side is exposing a live video stream to John and displayed on his TV. At the same time, a live video stream exposed by John to Anna for the same purpose, is displayed on Anna's TV. Following the abovementioned definition, one could state that the two parties are engaged in media symmetry.

Now let us consider a situation where John is not able to display the live video stream from Anna because he is on the go. It could be for example that as John is on the go he is using his phone to render the acquired media stream, yet his phone is only able to display some still images from the stream. In this case, and following again the abovementioned definition one could state that the two parties are not engaged in media-symmetry.

Nevertheless, media symmetry can be attributed to the source of information as well. Let us consider a different situation, where Anna's activities are exposed to John as simple phrases, and displayed on his TV as a video stream (perhaps an intelligent service renders the acquired activities to very realistic animations). At the same time, a live video stream exposed by John to Anna, is displayed on Anna's TV as in the previous example. Following our previous definition we would conclude that the two parties are engaged in display media symmetry. Yet, in this case what appears to be of a higher importance is the media of the exposed attributes rather than the media of the observable items. We can elaborate on this mode of media symmetry in a similar way like in the abovementioned *display-media-symmetry,* by introducing a relationship '*source-media-symmetry*' that captures whether the exposed attributes encode their information using similar media:

$$\_\ sourceMediaSymmetrical\_ : RealSituation \times Aspect \rightarrow \mathbb{F}\ (Entity \times Entity) \mid$$
$$\forall\ r{:}RealSituation;\ x, y{:}Entity;\ s{:}Aspect \bullet\ x\ sourceMediaSymmetrical\ y \iff$$
$$(x,y) \in \_\ sourceMediaSymmetrical\_(r, s) \iff$$
$$mediaOf(relevantAttributes(n^{r}_{xy}, s)) = mediaOf(relevantAttributes(n^{r}_{yx}, s))$$

2.10.2 Fidelity (a)symmetry

Quoting the classification of Voida et al. (2008) *"the different amount of detail provided in media spaces creates an asymmetry of fidelity"*, fidelity asymmetry relies on the fact that either by choice (e.g., personal preferences) or by means (e.g., image quality) the level of detail and frequency of information of the observers differs to that of the observed.

Looking closer, one can doubt how trivial it is to directly compare both the detail and the frequency of exchanged information, since such comparison is highly sensitive to the context. One, for example, may consider that the attribute *(age: 30)* is of the same detail to the attribute *(age: 29)*, based on the fact that both are of numerical values; however the first could be used implicitly to denote 'around 30' whereas the second can not. Furthermore, in both cases it could be that even a frequency of updating such age information every second can not be perceived as 'higher' than updating it every month. Now consider the detail of the attributes *(location: home)* and *(location: 40° 36' N, 22° 58 E)*. Depending on the context it could be that either of the two is of higher fidelity. Then, let us take a look at the attributes *(location: work)* and *(location: home)*. One could assume that in this case the attributes are of the same detail; if we imagine however an underlying ontology that implies from *(location: work)* the attribute *(availability: busy)*, whereas the attribute *(location: home)* does not imply any attributes about *availability*, the ontology could change our view regarding the fidelity symmetry of the two attributes.

Nevertheless, reviewing the above and the definition of the term fidelity symmetry, and inline with Hudson and Smith (1996), we notice that the consequence of fidelity asymmetry on the communication channel in terms of awareness would be identical to the consequence of deception; e.g. the difference of frequency results in "information cloaking", while difference of detail results in "information blurring". Hence, in a broader sense we can think of fidelity symmetry as a situation where individuals apply to their reciprocal relations similar deceptive patterns.

Whether talking about intentional or not deceptive patterns, the possibility to model such deceptive patterns using the FN-AAR model has been presented in an earlier section. There we propose that an entity x is blurring information about an aspect a to some entity y, when all the attributes about the aspect a that are made available to y (explicitly or by implication), are a subset of the attributes about a that are made available to some third entity z (explicitly or by implication). Similarly, x is hiding an aspect a from y, when there are no attributes about a that are made available to y either explicitly or by implication, and at the same time there is at least one attribute about a that x makes available to an other entity z.

Returning to these definitions we may account for fidelity symmetry by comparing the involved deceptive patterns:

$$
\begin{gathered}
\_\mathit{fidelitySymmetrical}\_ : \mathit{RealSituation} \times \mathit{Aspect} \rightarrow \mathbb{F}\ (\mathit{Entity} \times \ \mathit{Entity}) \mid \\
\forall\ r{:}\mathit{RealSituation};\, x, y{:}\mathit{Entity};s{:}\mathit{Aspect} \bullet\ \ x\ \mathit{fidelitySymmetrical}\ y \iff \\
(x{,}y) \in \_\mathit{fidelitySymmetrical}\_(r{,}s) \iff \\
(x\ \mathit{isHiding}\ s\ \mathit{from}\ y\ (r) = y\ \ \mathit{isHiding}\ s\ \mathit{from}\ x\ (r)\ ) \wedge \\
(x\ \mathit{isBlurring}\ s\ \mathit{to}\ y\ (r) = y\ \mathit{isBlurring}\ s\ \mathit{to}\ \ x\ (r))
\end{gathered}
$$

As the above formalism implies, we propose that two entities x, and y are engaged in fidelity-symmetry at some situation r regarding an aspect s if and only if the involved entities apply to each other the same deceptive patterns regarding this aspect.

2.10.3 Participation asymmetry

Different individuals may have different levels of participation in the community potentially being excluded in activities that others can join or having different level of access. The FN-AAR model does not make any explicit separation among entities whether these are individuals, communities, or even agents. In this respect, apart from the reciprocal communication of individuals participating in a community one could model a community itself as an entity, implying that the latter exposes and inquires information to and from its participants (for example, forums typically expose the number of subscribed members, or a list with those who are active in a discussion).

This FN-AAR feature enables the assessment of symmetry within a community at various levels. For example a community can be characterized by participation symmetry when it is exposing the same information to all of its members, and inquires information from its members using the same resources. Furthermore, two community-member entities can be characterized by participation symmetry when they both expose-to and inquire-from the community, attributes about the same aspects. We can assess the participation symmetry within a community and summarize the above statements in terms of the FN-AAR model as follows:

$$\begin{gathered}
symmetricalCommunityOf : RealSituation \rightarrow \mathbb{F}\,(Entity \times \mathbb{F}Entity) \mid \\
\forall r{:}\ RealSituation;\ c{:}\ Entity;\ m{:}\ \mathbb{F}Entity \bullet\ c\ symmetricalCommunityOf\ m \iff \\
(c,m) \in _symmetricalCommunityOf_(r) \iff \\
\forall x,y{:}Entity;\ x \in m;\ y \in m \bullet \\
(n^{r}_{cx} = n^{r}_{cy}) \wedge (f^{r}_{cx} = f^{r}_{cy}) \wedge \\
(\forall a{:}\ Attribute;\ a \in n^{r}_{xc} \bullet \exists b{:}\ Attribute;\ b \in n^{r}_{yc} \mid a.aspect{=}b.aspect) \wedge \\
(\forall u{:}\ Resource;\ u \in f^{r}_{xc} \bullet \exists v{:}\ Resource;\ b \in f^{r}_{yc} \mid u.aspect{=}v.aspect)
\end{gathered}$$

Accordingly, at some situation *r* a community *c* with members *m,* is engaged in participation symmetry if and only if it applies the same focus and nimbus to any two entities *x*, and *y* that belong in *m*, while at the same time *x* and y expose to and also inquire from the community *c* attributes about the same aspects[7].

2.10.4 Place Asymmetry

An awareness system may connect places in which different norms apply and where different affordances are perceived by users, e.g., a private office to a mobile user in the train. The norms pertaining to such different contexts impose asymmetry in the communication. In a broader sense, the term place may refer to any specific context; in its most abstract form it could refer to any situation involving some entities, and consequently a social norm could be perceived as a constraint that is connected to a situation defining a set of aspects that should (or should not) be exposed or focused among entities.

One could compare the contextual norms deriving from different places to assess their symmetry, but furthermore and more importantly, having such contextual norms available regarding the respective foci and nimbi of two entities, we can examine their reciprocal compliance to the imposed norms to figure out whether the entities themselves are engaged or not in a place asymmetrical relation. If the entities engage in a place-symmetrical relationship we can expect that their respective foci and nimbi will conform to their respective norms.

[7] In section 2.11, where several properties of communities are discussed, we expand on the topic of symmetry to capture different modalities of participation symmetry within communities

We may model such contextual constraints that should apply in the relation of two entities using the FN-AAR model in a straightforward way:

$$\_mustExpose\_To\_ : RealSituation \times Entity \rightarrow \mathbb{F}\ (Entity \times Aspect) \mid$$
$$\forall\ r{:}RealSituation;\ x{,}y{:}Entity;\ a{:}Aspect \bullet$$
$$x\ mustExpose\ a\ To\ y \Longleftrightarrow (y{,}a) \in \_mustExpose\_To\_(x{,}r) \Rightarrow$$
$$\exists\ u{:}Attribute;\ u \in n^{r}_{xy} \bullet u.aspect{=}a$$

The '*mustExposeTo*' relation as defined above implies that in any situation for any entity there is a set of tuples that describe what kinds of information should be exposed to any other entity. Having implemented such functions in an awareness system, we may compare at any situation the constraints that apply to the participating entities in order to assess their contextual (place) symmetry.

$$\forall\ x{,}y{:}Entity;\ r{:}\ RealSituation;$$
$$A{=}\_mustExpose\_To\_(r{,}x),\ B{=}\ \_mustExpose\_To\_(r{,}y) \bullet$$
$$x\ symmetrical\ to\ y\ at\ r \Rightarrow$$
$$(\forall\ a{:}Aspect;\ (y{,}a) \in A\ \exists\ u{:}Entity \times Aspect \bullet u \in B\ \wedge\ u{=}(x{,}a))\wedge$$
$$(\forall\ a{:}Aspect;\ (x{,}a) \in B\ \exists\ u{:}Entity \times Aspect \bullet u \in A\ \wedge\ u{=}(y{,}a))$$

As the above formalism implies, we propose that if two entities *x,* and *y* are engaged in a place-symmetry relation at some situation *r*, then they should reciprocally conform to the constraints that apply at this situation.

In a comparable way one could easily introduce both affirmative and negative constraints, applying both to the foci and the nimbi of the involved entities. What is of interest, and not obvious above, is that the coupling of "norms" to "places" is presented indirectly through a context definition that is defined by the tuple situation and entity. This view abstracts away from the underlying mechanisms for the triggering of the contextual constraints, which typically would be provided through agents or communities. For example, a community providing automatic pictures to its members, may acquire from them their location, hence by coupling the individuals to cameras at various places, the community can generate the defining norms for exposing to them the appropriate content.

2.10.5 Engagement asymmetry

Individuals may be active on different levels and in ways that are inherently asymmetrical, preferring e.g. the role of consumer or producer of information. In terms of the FN-AAR model it is quite straightforward to assess the role of an entity as a consumer or producer of information, for example, just by looking at the entity's focus and nimbus. We can consider the nimbi of two entities to assess their engagement symmetry by comparing the kinds of information they expose to each other; or consider their reciprocal foci to assess their engagement symmetry by comparing the kinds of information they inquire from one another:

$$\_\ nimbusSymmetrical\_: RealSituation \rightarrow \mathbb{F}\ (Entity \times\ Entity) \mid$$
$$\forall\ r{:}RealSituation;\ x,\ y{:}Entity;\ \bullet$$
$$x\ nimbusSymmetrical\ y\ \Longleftrightarrow$$
$$(x,y) \in \_\ nimbusSymmetrical\_(r) \Longleftrightarrow$$
$$attributeAspects\ (n^{r}{}_{xy}) = attributeAspects\ (n^{r}{}_{yx})$$

where

$$attributeAspects : \mathbb{F}\ Attribute \rightarrow \mathbb{F}\ Aspect \mid$$
$$\forall\ s{:}\ \mathbb{F}\ Attribute\ ;$$
$$attributeAspects\ (s) = \{a{:}Aspect \mid (\forall u{:}Attribute;\ u \in\ s \bullet\ \ a = u.aspect)\}$$

Renditions such as the above can cover to a vast extent the principle of minimum asymmetry discussed also earlier in this section addressing it in various layers. Furthermore, in a broader sense, where the engagement symmetry is subjectively perceived by the participants of a media space, we could consider that individuals express their subjective notion of engagement symmetry by using constraints that should be applied on their relationships with others. Thus returning to the earlier presented scenario where John needs to express his lust to Anna without threatening his privacy, it could be that an "engagement symmetry constraint" prevents the sensitive information to be exposed to Anna unless she exposes in a similar manner the same feelings to him (in fact a lot of dating sites are built on such mechanisms).

2.10.6 Benefit asymmetry

Benefit asymmetry (and by the same token cost asymmetry) implicitly assume the existence of some evaluation of costs and benefits that an awareness system brings to its users. This is an emergent and experiential property of such systems that is person and context dependent, and may be evaluated empirically (Markopoulos et al., 2004).

Apart from such empirical evaluations one can consider the degree to which users are enabled to achieve awareness through the use of the system. We can simply consider that two individuals are engaged in a benefit-asymmetrical situation when the first is aware of the information of interest about the second whereas the latter is not.

In terms of the FN-AAR model we could say that an entity '*x*' "benefits" from an awareness system regarding some aspect '*a*' and a focused entity '*y*' when '*x*' is focusing on the aspect *a* of *y* and the latter is exposing some information about this aspect to the first. Since it is not possible – unless we turn to some external quantifier function – to quantify a priori the benefits rising by the observed information, we have no reason to pursue a direct comparison of the reciprocal awareness benefit; however, we can claim that benefit symmetry should not raise any situations where information about a focused aspect is not being exposed by the observed entity:

$$\_\ benefitSymmetrical\_ : RealSituation \rightarrow \mathbb{F}\ (Entity \times\ Entity)\ |$$
$$\forall\ r{:}RealSituation;\ x,\ y{:}Entity;\ \bullet$$
$$x\ benefitSymmetrical\ y\ \iff$$
$$(x,y) \in \_\ benefitSymmetrical\_(r) \Rightarrow$$
$$(\forall\ u{:}\ Resource\ |\ u \in f^{r}{}_{xy}\ \ \exists\ a{:}\ Attribute,\ a \in n^{r}{}_{yx} \bullet a.aspect = u.aspect) \wedge$$
$$(\forall\ v{:}\ Resource\ |\ v \in f^{r}{}_{yx}\ \ \exists\ b{:}\ Attribute,\ b \in n^{r}{}_{xy} \bullet b.aspect = v.aspect)$$

Thus, formally speaking, benefit symmetry among two entities *x, and y* implies that the entities should expose to each other all inquired aspects of information.

2.11 Modelling properties of communities

In the introduction of the model we indicated that entities may be actors, agents or communities. The usage of communities was pointed out in the definition of focus; in that section we made clear that we can use communities as the target of observation in order to express focus on an aggregation of entities; to model, for example, that *John* is interested in knowing if anyone of *Anna* or *Doty*(i.e. his family) is at home, we could consider a community '*family*' that exposes to *John* an attribute about the aspect '*relatives at home*', and consequently that *John* is focusing on this aspect of '*family*'.

To explore the notion of communities and their relevant properties, imagine the following scenario:

"John is registered in a 'walkingClub'. The members of the walkingClub need to expose their 'mood' to the club, and in response they can check how many of the club's members feel like walking"

In the above scenario if we consider the '*walkingClub*' we can observe some particularities that distinguish it from other entities such as actors, and its members. The '*walkingClub*' acts as a sink that collects information from its members (i.e. their '*mood*'), and as a source (broadcaster) that exposes to its members some common attributes (i.e. the '*number of people that feel like walking*') that are connected to the acquired information. The aforesaid insights lead us to consider an entity *c* as a community with members '*m*', when each entity in the member set '*m*' provides some attributes to the community, and has access to some common community information that is an aggregation of the provided attributes:

$$\_communityOf\_ : Entity \times \mathbb{F}\ Entity|$$
$$\forall\ c{:}Entity;\ m{:}\mathbb{F}\ Entity;\ \bullet\ \ c\ community\_of\ m\ \iff\ (c,m) \in \_communityOf\_ \iff$$
$$\forall r{:}RealSituation;\ e{:}\ Entity;\ e \in m \bullet\ (n^{r}{}_{xc} \neq \emptyset) \wedge$$
$$(\exists\ a{:}\ Attribute;\ aggregation{:}\ \ Entity \times \mathbb{F}\ Entity\ \ \times RealSituation \rightarrow\ Attribute\ |$$
$$(a = aggregation(c,m,r)) \wedge (\forall x{:}\ Entity;\ \ x \in m\ \ \bullet a\ \ \in\ \ n^{r}{}_{cx}))$$

The exact function of the aggregation function in the above definition is implementation specific. In our running 'walking-club' scenario, for example, the aggregator returns the number of entities that expose to the community one's wish for a walk:

$$
\begin{gathered}
walkingMoodAggregator: Entity \times \mathbb{F}\ Entity\ \times RealSituation \rightarrow Attribute \mid \\
\forall r: RealSituation; c{:}Entity; m: \mathbb{F}Entity; a{:}Attribute \bullet \\
a = walkingMoodAggregator\ (c, m, r) \iff \\
(a.aspect=\text{'}\ number\ of\ people\ that\ feel\ like\ walking\text{'}) \wedge \\
(a.value=countOf(\{e{:}Entity; e \in m \mid \exists\ u{:}Attribute \bullet \\
(u \in n^{r}{}_{ec}) \wedge (u.aspect=\text{'}mood\text{'}) \wedge (u.value=\text{'}\ feel\ like\ walking\text{'})\ \})
\end{gathered}
$$

Utilizing this, each member of the walking-club community provides to the community an attribute about the aspect *mood*, and the community exposes to its members in response the number of members that wish to go for a walk:

$$
\begin{gathered}
walkingClub{:}Entity;\ walkingClubMembers{:}\mathbb{F}\ Entity \mid \\
\forall\ \ r{:}ReaSituation;\ member{:}Entity; member \in walkingClubMembers \bullet \\
(\exists\ a: Attribute; (a \in n^{r}{}_{member,\, walkingClub}) \wedge (a.aspect=\ \text{'}mood\text{'})) \wedge \\
(\exists\ p{:}AttributeProvider; p \in nimbus_{\ walkingClub} \mid (member \in\ p^{r}.e) \wedge \\
(p^{r} = walkingMoodAggregator(walkingClub, walkingClubMembers, r))
\end{gathered}
$$

2.11.1 Anonymous communities

In the above example it could happen that the '*walkingMoodAggregator*' returns instead an attribute about the *'names of people that feel like walking'* with value the set of those community members who expose to the community that they are in such a mood. As one can observe, in this case, as a community member, e.g. *Anna*, becomes aware of an observable item displaying the community's attribute, e.g. *('names of people that feel like walking': {Tom,...})*, this observable item could also reveal to her implicitly the attribute *('mood': 'feel like walking')* of *Tom*; i.e.:

$$
\begin{gathered}
\exists\ o: ObservableItem; o \in a^{r}{}_{Anna,\ walkingClub} \mid \\
(o\ displays(\ \text{'}names\ of\ people\ that\ feel\ like\ walking\text{'}: \{Tom\})\ of\ walkingClub) \wedge \\
(o\ displays\ \ (\text{'}mood\text{'}:\ \ \text{'}feel\ like\ walking\text{'})\ of\ Tom)
\end{gathered}
$$

It becomes apparent, that scenarios such as the above, could jeopardize one's anonymity, as entities could potentially indirectly expose their and become aware of others' situation. We can classify such cases formally by introducing a relevant property for anonymous communities:

$$
\begin{gathered}
\_\ isAnonymous : RealSituation \rightarrow \mathbb{F}\ Entity \mid \\
\forall\ r{:}RealSituation; c{:}Entity; m: \mathbb{F}Entity\ ; c\ community\_of\ m\ \ \bullet \\
c\ isPartiallyAnonymous\ \iff c \in \_isPartiallyAnonymous(r) \iff \\
\neg(\exists\ i{,}j: Entity; o: ObservableItem; u{:}Attribute \mid (o \in a^{r}{}_{i,c}) \wedge (o\ displays\ u\ of\ j))
\end{gathered}
$$

Thus, we classify a community *c* as anonymous if and only if no two entities *i*, and *j* exist, such that an observable -by entity *i*- item about the community *c* displays to *i* any attribute of *j*, and hence exposes the identity of *j* to *i*.

2.11.2 Symmetrical communities

We may further explore the concept of participation symmetry that was examined in the previous section, by continuing the discussion from the walking-club example above. It could be that the community exposes to some of its members the identities of those who wish to go for a walk, apart from only the number of them. To identify these cases we introduce some notions relevant to symmetry within communities.

1. Each member of the community provides attributes about the same aspects to the community:

$$\begin{gathered}
_\ isNimbusSymmetrical: RealSituation \rightarrow \mathbb{F}\ Entity \mid \\
\forall\ r{:}RealSituation;\ c{:}Entity \mid \exists\ m{:}\mathbb{F}\ Entity\ ;\ c\ community_of\ m \bullet \\
c\ isNimbusSymmetrical \Leftrightarrow c \in _\ isNimbusSymmetrical\ (r) \Leftrightarrow \\
(\forall\ x,y{:}Entity \mid x,y \in m \bullet attributeAspects(n^{r}{}_{x\,c}) = attributeAspects(n^{r}{}_{y\,c}))
\end{gathered}$$

i.e. an entity *c* is considered a nimbus-symmetrical community, when while being a community of members *m,* any entity member pair *x, y* exposes to the community a set of attributes about the same aspects.

2. Each member of the community is focusing to same community aspects:

$$\begin{gathered}
_\ isFocusSymmetrical: RealSituation \rightarrow \mathbb{F}\ Entity \mid \\
\forall\ r{:}RealSituation;\ c{:}Entity \mid \exists\ m{:}\mathbb{F}\ Entity\ ;\ c\ community_of\ m \bullet \\
c\ isFocusSymmetrical \Leftrightarrow c \in _\ isFocusSymmetrical\ (r) \Leftrightarrow \\
(\forall\ x,y{:}Entity \mid x,y \in m \bullet\ resourceAspects(f^{r}{}_{c\,x}) = resourceAspects(fr\ c\ y))
\end{gathered}$$

i.e. an entity *c* is considered a focus-symmetrical community, when while being a community of members *m,* any entity member pair *x, y* assigns a set of resources about the same aspects for observing the community.

3. Each member of the community is aware of the same community aspects:

$$\begin{gathered}
_\ isAwareSymmetrical: RealSituation \rightarrow \mathbb{F}\ Entity \mid \\
\forall\ r{:}RealSituation;\ c{:}Entity \mid \exists\ m{:}\mathbb{F}\ Entity\ ;\ c\ community_of\ m \bullet \\
c\ isAwareSymmetrical \Leftrightarrow c \in _\ isAwareSymmetrical\ (r) \Leftrightarrow \\
(\forall\ x,y{:}Entity \mid x,y \in m \bullet displayedAspects\ (a^{r}{}_{x\,c}) = displayedAspects\ (a^{r}{}_{y\,c}))
\end{gathered}$$

i.e. an entity *c* is considered an aware-symmetrical community, when while being a community of members *m,* any entity member pair *x, y* is aware regarding the community of a set of observable items about the same aspects.

2.11.3 Protected communities

Nevertheless it may be that the '*walkingClub*' community above exposes the number of its members that wish to go for a walk, also to entities outside the community (i.e. entities that do not expose their '*mood*' to the community). We can identify these cases formally by introducing a definition of protected communities.

We consider a protected community one that does not expose any information to non-members, and consequently we can identify situations like the aforementioned by examining a community in accordance to the following definition:

$$\_\ isProtectedCommunity: RealSituation \rightarrow \mathbb{F}\ Entity \mid$$
$$\forall\ r{:}RealSituation;\ c{:}Entity \mid \exists\ m{:}\mathbb{F}\ Entity\ ;\ c\ community\_of\ m \bullet$$
$$c\ isProtectedCommunity \Leftrightarrow\ c \in \_\ isProtectedCommunity\ (r) \Leftrightarrow$$
$$\forall\ e{:}\ Entity;\ (e \notin m)\ \mid n^{r}{}_{c\,e} = \varnothing$$

Thus, we consider that a community '*c*' is protected if and only if the nimbus of the community to any entity '*e*' that is not a member of '*c*', is an empty set.

2.12 Conclusions

We have introduced FN-AAR, a formal model of awareness systems, based on the focus-nimbus model of Benford and Fahlen et al. (1993) and Rodden (1996). Whereas the original focus and nimbus model describes *how aware* is an entity *i* of some entity *j* in a particular *space,* our model describes *of what* is an entity *i aware of* regarding some entity *j* in a particular *situation*.

The FN-AAR model abstracts away from modelling the propagation of awareness information and information flow modelling as in Simone and Bandini (2002) and Fuchs et al. (1995). It advances the focus-nimbus model in that it is explicit about the object of awareness: i.e. the relationship of the information an entity can potentially provide about itself, to the information actually being observed by another entity. This is necessary for modelling the social aspects of awareness systems as shown above.

FN-AAR provides an abstract yet sufficient vocabulary for describing with clarity notions that characterize social aspects of awareness achieved through mediated communication. This allows designers to discuss and build systems with well defined a priori properties, and, perhaps more importantly, serves as a solid basis for improving the quality of empirical studies by providing the necessary foundation for controlling the design variables.

We have demonstrated that the model allows the formal expression of abstract concepts such as focus, nimbus, awareness, but also socially oriented behaviours that have previously been discussed at an informal level. More specifically, the model allowed a clear definition of deception related behaviours (blurring, lying), intentionality of information sharing, social translucency and symmetry (distinguishing various flavours of these general concepts), and finally describing different community models.

We acknowledge that there may be plenty of alternative definitions that approach concepts such as the aforementioned providing both more depth and width in the exploration presented here. Nevertheless this is exactly why FN-AAR can be seen both as a conceptual tool and also as an analytical one that the research community can use as a foundation for building the next generation of awareness systems.

The model was rendered in simple set-notation to ensure simplicity and abstraction. Our exposition of FN-AAR has shown that simple set-theory is sufficient to model the very high level concepts discussed. At earlier stages of this work we considered using higher level logic; that seemed appropriate given that some awareness information is exchanged synchronously, other asynchronously, and timing in sending and reception is central to the way people share information, represent themselves socially and interact with others. However, with the FN-AAR model we have abstracted away from issues of sequencing, timing and propagation of information, assuming at any moment that there is a 'situation' that is perceived differently by the various actors modelled: these different perspectives of each actor could be modelled effectively by relating the aspects, attributes and resources involved in sharing and presenting awareness information.

To conclude, the FN-AAR model provides a domain specific abstraction that can facilitate the task of creating awareness systems focusing the design and implementation effort not on the interaction with the technology itself (as input or output) but on the more crucial social interaction among connected individuals or groups.

Chapter 3

Sustaining Intelligibility, Accountability, and Control

In this chapter we suggest interactive mechanisms and supportive tools that engage end-users in the composition of context-aware applications. The context of the chapter's goals and associated research work are presented in sections 3.1 to 3.4. A notation that facilitates service compositionality is presented in section 3.5. Then we show how this notation can be translated both to natural language and to a structured context-range editor, and we discuss how intelligibility, accountability and control are supported in terms of corollary requirements (section 3.6). In the same section an experimental evaluation of the editor is presented, while the editor's underlying assumptions are further validated and generalized through a study presented in section 3.7.

3.1 Introduction

Consider a very simple and widespread manifestation of context awareness familiar to users of instant messaging systems. After some time of inaction, these applications typically declare that their user is away. Usually this behaviour is implemented by detecting mouse and keyboard inactivity for some set amount of time. Putting aside whether users may experience this feature as smart and convenient or as erratic and annoying, most important for this chapter is whether users understand the premises for this system inference, whether they can anticipate this behaviour, and eventually, whether they can control it.

Current instant messaging applications might let users disable this feature or modify its parameters for a shorter or longer delay, or even let them set this status indication manually. Such an interface layer though only touches on the surface of the problem, and raises several questions:

- Considering a situation where the user's status is shown as 'away' can the user know why this is so? For example, how can the user know whether this indication should be attributed to the computer's inactivity or to some other parameter setting by the user?
- Assuming the user does not agree with how the system reasons to set the status to 'away', are there supporting tools to allow her to extend the underlying rules? For example, most laptop computers today also have a camera attached, that can easily detect if there is anybody in front of it facing towards the screen; to extend the messenger application with such a computer vision functionality is currently out of reach for end users and this level of system control is, in general, opaque for end-users.
- Are there supporting mechanisms that allow users to examine the system's behaviour under some imaginary situation in the future, hence leaving them a wider space for trial and error before they actually modify the system's behaviour?
- Finally, does the system expose either its internal context-sensing capabilities (i.e. the mouse, and keyboard inactivity period) or its effective outcome (i.e. whether the user is 'away' or not) so that these become accessible to other applications, e.g., an energy saving application, or a social networking application. And in such a case, are there sufficient mechanisms to do so in an efficient and comprehensible way?

Current state of the art does not address such questions convincingly. Finding acceptable solutions is a key challenge for ensuring acceptability by end users, especially for applications that rely more extensively on context awareness than the instant messaging example above. This chapter examines how users of context aware applications can be empowered to view, comprehend, and modify relevant parts of the software.

3.2 Intelligibility, Accountability and Control

Bellotti and Edwards (2001) argue that context aware systems should not simply act on behalf of users but, rather, should involve users in action outcomes, allowing them to understand, explore, and even define the underlying mechanism governing the behaviour of such systems.

Empirical evaluations of context aware systems reported since (e.g., Cheverst et al., 2005; Metaxas et al., 2007), support the views of Bellotti and Edwards and so far there has been no evidence to suggest the contrary. Especially where context-awareness is embedded in communication applications, e.g., by providing one with automated updates regarding the whereabouts and activities of friends or family, or even of colleagues at work, there is converging evidence regarding the need of people to control the capture and disclosure of information about themselves, see e.g. (Neustaedter 2003).

Such general requirements are not trivial to address in practice as there exists a trade-off between user control and the effort users need to expend to exercise that control (Abowd & Mynatt, 2000; Barkhuus & Dey, 2003; Markopoulos, 2005). Appropriate interfaces are needed through which users can inspect workings of context aware systems, modify them, and understand how a system interprets and acts upon their actions and adapts to their context. The notion of *seamful design* (Chalmers et al., 2003) has attracted substantial attention by researchers in the field of ubiquitous computing, as it explores the tension confronting interaction designers between hiding irrelevant to users mechanics of system operation and allowing them to gain insight into what may appear as erroneous operation, or even to capitalize on the idiosyncrasies of context sensing technology, using them as a design resource rather than as a constraint (Gaver et al., 2003).

Beyond seamful design, addressing the principles relating to intelligibility and accountability amounts to supporting end-users to control a system's operation, by specifying the contextual range under which the system should manifest desired behaviours. Conversely, such functionality can further support users to understand system operation, by allowing them to specify queries regarding system behaviour, explore the 'seams' of the system, and control its operation.

End-user programming emerges thus as a key ingredient for ensuring the acceptability of context aware systems and has accordingly been identified as a key future challenge for research in ambient intelligence and its related technological visions of ubiquitous computing and pervasive computing (Aarts & de Ruyter 2009), particularly motivated by the need to control and manage ad hoc collections of devices and systems which make up the environments in which users live, work, and interact. The visualization and computational representation of context are the quintessential point of interaction between users and system intelligence.

Approached from the perspective of end-user programming, the challenges of intelligibility, accountability, and control, require the definition of appropriate computational abstractions.

Accordingly, in this chapter we target situations in which open and extensible collections of distributed system services should be composed and their behaviour coordinated through simple programmatic controls accessible to the non-professional developer end-user. In the following section, we review some research efforts that have made inroads in modelling context and providing programmatic controls for it. We then go on to introduce a parsimonious and generic notation that allows heterogeneous and numerous services used by context-aware applications to define and expose their contextual behaviour at the user interface. We show how their idiosyncratic semantics can be employed and translated to natural human-comprehensible text and manipulated using tools that pay particular attention to human cognitive abilities and limitations, allowing intuitive and comprehensible formulation of logical expressions. Finally, we show how this approach can uniformly support the intelligibility and control of context-sensing applications.

3.3 Related work

Research that has focused on supporting end users to intentionally modify and control how computational representations of context are constructed and interpreted, typically targets the composition of available system components and the tailoring of their function for a specific context. In fewer occasions, user intervention is seen as a way to improve system performance. Some of the key works in this area are discussed below.

CAMP (Truong et al., 2004) is a system that, based on elicitation of user requirements for smart home environments and relevant context-aware applications, enables end-user programming based on a magnetic poetry metaphor; in a graphical interface users are allowed to place freely predefined 'magnetic tags' (keywords) related to a smart home environment in order to create rules defining the behaviour of the system. The iCAP system (Dey et al., 2006) enables visual-design of context-aware applications by providing a simplified interface for defining situations under which certain actions should be performed. In the study of Humble et al. (2003) a jigsaw puzzle metaphor was used to represent information flow and reasoning while assembling end-user applications. E-Gadgets (Mavrommati et al., 2004), explored a component composition model where end-users were asked to specify data flows between components: a laboratory evaluation showed that users could understand and compose simple component configurations. To inspect and modify the logic of the system though, users were exposed to relatively involved programmatic concepts, like typing and polymorphism, in a manner no different than conventional programming.

The *aCAPpella* (Dey et al., 2004) system illustrated how a programming by demonstration approach can enable end-users to compose interactively heterogeneous context recognition functions to create recognizers of specific activities/events relevant to them. Users could select from available streams of data, those which are relevant for sensing a particular activity or event; also, they could indicate to the system which time intervals are associated with a particular activity of interest. Test cases and limited user testing illustrated how recognition performance is improved thanks to putting the user in the loop. Focusing on achieving acceptable levels of recognition performance, this research did not aim to address intelligibility and accountability and the evaluation did not touch upon these issues either.

Dey and Newberger (2009) introduced *Situation* components that provide easy programmatic access to services and devices for implementing context sensitive behaviours using the Context Toolkit, an early and well known infrastructure for programming context sensitive system behaviour (Salber et al., 1999). They show how their 'Situation' component can be instrumental to allow the construction of user interfaces suitable for end-users to access that infrastructure. To this point though they have only addressed the challenges of intelligibility and control at the level of designers and developers, rather than end-users, by providing them with components and tools that expose the application logic, and support debugging.

While these works are very instructive and have booked very valuable results, they do not yet provide sufficient and general solutions for end-user programming of context aware systems. Some lack a vocabulary sufficiently extensible and flexible to allow their application outside a very specific domain (e.g., *home media space* (Neustaedter & Greenberg, 2003), *AutoHan* (Blackwell & Hague, 2001)) or otherwise they do not provide sufficient intelligibility and control due to their inherent constraints. For example, pictorial notations (Humble et al., 2003; Dey et al., 2006; Jones et al., 1999) are apparently a popular choice although they do not as such provide sufficient expressiveness regarding system behaviour and intelligence. Their graphical nature is quite often assumed to render them automatically more comprehensible to end-users despite the fact that earlier research has shown (Petre, 1995; Green & Petre, 1992) that text and graphics are not necessarily an equivalent exchange, and that textual and visual representations for software differ in effectiveness and comprehension (Green & Petre, 1992; Green et al., 1991). Programming by demonstration (as for example discussed above), is often seen as a more accessible way of programming, but in itself is not particularly suited for making system behaviour and intelligence intelligible; at least there does not appear to be an obvious way in which programming by example offers some advantages for intelligibility by end users.

Returning to the concept of exposing the seams of the system, we can distinguish the approaches to end-user control of context awareness systems by the degree to which they expose end-users to the underlying system architecture and the role they foresee for their end-users, a programmer of system behaviour versus a system architect. For example, *aCAPella* (Dey et al., 2004) lets the user inspect and directly control the recognition technology used, but provides less control of how this information will be used. The *e-Gadgets* editor (Mavrommati et al., 2004) offers two levels of granularity for the composition of system components: a device and a service level, versus one that exposes users to the typing of data flows, and the detailed handling of system events. It appears that intelligibility and accountability for different users and different contexts call for a compositional and scalable approach that will allow users to "open up the box" at different levels and treat lower level entities and their higher level combinations, in a uniform way.

Explicit explanations of system function are a plausible way to support intelligibility. Lim et al. (2009) evaluated five different types of explanations (i.e. *What did the system do? Why did the system do(or not) X ? What would the system do if Y happens? How can I get the system to do Z, given the current context?*). They found that in all cases, explanations generated by the system regarding its context sensitive behaviour increased user's comprehension and feelings of trust towards the system. Explanation functionality has however not been implemented in many context aware systems reported to date. Notable exceptions are the Intelligent Office System by Cheverst et al (2005) and the *DIARIST* system (Metaxas et al., 2007); in both cases realistic in situ evaluations were carried out but these were of limited scale and their focus was more generally on the acceptability of a context aware application as a whole rather than on evaluating the degree to which system intelligibility had been achieved.

3.4 Our approach

Often overlooked in the context of initial research evaluations that make up the related research literature is the need for sustainable intelligibility that will allow initially unforeseen services and system components to be used within the context-aware logic that a user describes. Sustainability can be achieved if existing infrastructure, system-behaviour specified by users, and newly admitted system services, all are described in a uniform approach. From a software perspective, this boils down to the compositionality of the abstractions used. *Compositionality* means that composing system entities within any particular representation framework provided will produce itself an entity which can be treated in exactly the same way as its operands. This requirement is typically addressed in formal systems and mathematical notations, e.g., logic, and has been argued to be a critical requirement for the software engineering of user interface software (Markopoulos et al., 1997). Based on the arguments above, we also believe it is necessary to ensure the feasibility and sustainability of intelligibility of context-aware systems by end users.

The above insights set the field for an inquiry into appropriate interactive mechanisms for end-users to specify, comprehend, and modify context awareness. We follow the *What You See Is What You Meant* (WYSIWYM) approach (Power et al., 1998), which allows end-users to construct queries by directly performing editing operations on the underlying semantic representation in the form of natural language that is produced by the system. WYSIWYM has been applied with success in cases where a well defined ontology and vocabulary are available (e.g., Evans et al., 2008). What is unique in our approach is that we provide a uniform way for an open set of heterogeneous services to contribute to a compositional model of application development by exposing their underlying rationale, their possible outcome, and the 'seams' for their current state. This semantic information is exposed using the WYSIWYM approach, enabling end-users to maintain control over the system's behaviour. We introduce a context-range notation for handling context representation using an XML-based scheme. Based on this executable representation we introduce an editor of logical expressions relating context ranges. This editor builds on the findings of Byrne and Johnson-Laird (1992) regarding the spontaneous usage of propositional connectives, on the findings of Pane and Mayers (2000) regarding the perceived associativity of Boolean operators, and in our own heuristics to make the presentation and manipulation of logical expressions more intuitive for end users who are not assumed to be fluent in Boolean logic.

Despite that the domain of context-aware systems is still fragmented and lacking a common ground that allows compositionality and interoperability among existing context-aware services, promising concepts such as the recombinant computing approach (Edwards et al., 2002) discussed in the chapters 1 and 4, allow us to envision a future where the research community will be able to embrace a common semantic vocabulary. A proposal for such a vocabulary is presented here and employed to allow a translation of contextual semantics to natural language, and eventually to support intelligibility, accountability, and control.

3.5 Context Range Semantics (CRS)

To accommodate the overwhelming population of heterogeneous services, the need to manipulate rich vocabularies and the requirement for compositionality, we developed a notation and a corresponding simple range mark-up language that allows any context-aware service to define both its premises (i.e. the range under which it performs some task), as well as to expose its effective outcome on the context itself, i.e. the range of its output. The evaluation of a context-range against a specific context is further supported by the proposed notation with prefixes that describe the parts of the context that were matched by the range or that failed to match the range. Below we present briefly the semantics of the operators that services can use to describe their premises and their effective outcomes.

The *"all"* operator accepts any number of operands, and when used by a service to define its premises, it denotes that all its operands should be matched in the test context. The same operator, when used to define the effective behaviour outcome of a service, denotes that all its operands appear in the resulting context. For example, the evaluation of the expression "*all ({X, Y})*" versus the context "*{X, Y, Z}*", yields "*match (all ({match (X), match (Y)}))*", meaning that the expression is validated because both of its operands *X*, and *Y* are matched in the test context. In contrast, the evaluation of the same expression versus the context "*{X, Z}*" would yield *"fail (all ({match (X), fail (Y)}))"*, denoting that the expression is rejected because despite *X* was matched, it failed to match also *Y* in the target context. Furthermore, the expression "*all ({X, Y})*", appearing as a definition of the effective outcome of a service, denotes that both of its operands *X*, and *Y* appear in the service's outcome.

The *"any"* operator when used to define reasoning denotes that any number of its operands should be matched in the test context. When it is used to expose the outcome of a service it denotes that any number of its operands may appear in the resulting context. For example the evaluation of the expression "*any (all ({X, Y}), Z)*" versus the context "*{X, Y, W}*" yields *"match (any (match (all ({match (X), match (Y)}))))"*, meaning that the expression is validated because the operand "*all{X, Y}*" was matched in the test context "*{X, Y, W}*".

The *"one"* operator is employed to denote that exactly one of its operands should be matched in its test context; correspondingly it may be used to describe that the resulting context of a service contains exactly one of its operands. For example the evaluation of the expression "*one ({X, Y})*" versus the context "*{X, Y, Z}*" yields *"fail (one ({match (X), match (Y)}))"* meaning that the expression is rejected because it failed to matched exactly one of its operands in the target context.

The *"opt"* operator accepts one operand, and denotes that it may influence only positively the result of evaluation; i.e. if its operand is not matched in the test context, it will not reject the expression. When used to expose the outcome of a service it denotes that its operand may appear in the resulting context. Hence the evaluation, for example, of the expression *"all ({X, opt (Y)})"* versus "*{X,Y}*" yields *"match (all ({match (X), match (Y)}))"*, and the same expression versus "*{X, Z}*" also yields *"match (all ({match (X)}))"* because the lack of *Y* doesn't influence negatively the result.

The *"ext"* operator denotes that the evaluation of its test context should be performed by an externally defined evaluating function. This enables the services to extend their reasoning capabilities beyond the limits of the other operands. For example, the expression *"ext ('earlier', {X, Y})"* could denote that the evaluation should pass to the external operator "earlier", that as its name suggests is likely to be a temporal logic operator, while the expression *'ext('at-least',2,{X,Y,Z})'* could denote, through the externally defined operator *'at-least'*, that at least two of the operands *X, Y* and *Z* appear in the outcome.

All the operators can be prefixed with a negation operator *'neg'*. When used to define a context under which a service performs an operation, the negation operator denotes that its operand should not be matched in the test context, rather than that its operand should be falsified. For example the expression *'neg (sensor-high-activity)'* denotes that the test-context should not contain the term *'sensor-high-activity'*. Notice that this doesn't imply necessarily that the sensor should report low activity. The latter could be stated using "affirmative negation" with the term *'all (sensor-low-activity, neg (sensor-high-activity))'*, or simply as '*sensor-low-activity*', given that the sensor exposes its contextual-range outcome as 'one *(sensor-high-activity, sensor-low-activity)'*.

The aforementioned operators, are context independent, and can be translated to an XML schema that may be applied on any form of XML defined context. Consider for example the following XML snippet:

```
<book>
   <range:one>
      <title>Odyssey</title>
      <title>Iliad</title>
   </range:one>
</book>
```

In the above extract the context of the *'range: one'* is a *'<book>'* tag denoting, when used to define a service's reasoning, that within that context exactly one of the terms *'<title>Odyssey</title>'* and *'<title>Iliad</title>'* should be matched; the same snippet when used to define a service's possible output would denote that the service would return a *'<book>'* tag containing either *'<title>Odyssey</title>"* or *'<title>Iliad</title>'*.

In the rest of this chapter, we will consider the notion of *'attribute'* for defining a specific context, an attribute being a binding of an aspect with some value (as defined in chapter 2 for the FN-AAR model), and a specific context being a set of attributes. Below, for example, we see the extracts from two services returning their possible outcomes, one that senses whether the phone is ringing, and the second that senses whether the door-bell is ringing.

```
<range:opt>
   <attribute>
      <aspect>phone</aspect>
      <value>ringing</value>
   </attribute>
</range:opt>
<range:opt>
   <attribute>
      <aspect>door bell</aspect>
      <value>ringing</value>
   </attribute>
</range:opt>
```

The definitions of the possible outputs of the above services could be transformed, with respect to their range semantics, in the composition of the underlying rationale (premises) of a third service. In the following snippet we see such a transformation that could reflect the reasoning of a service that mutes the volume of the stereo system when the phone or the door-bell is ringing.

```
<range:any>
   <attribute>
      <aspect>phone</aspect>
      <value>ringing</value>
   </attribute>
   <attribute>
      <aspect>door bell</aspect>
      <value>ringing</value>
   </attribute>
</range:any>
```

The evaluation of the latter premise against a known context, such as the last known outcome of the phone, and door-bell services, could yield for example that the context-range premise was matched because the door-bell is ringing despite that there was no indication whether the phone is ringing as well; this is illustrated in the following XML extract:

```
<range:match>
   <range:any>
      <range:fail>
         <attribute>
            <aspect>phone</aspect>
            <value>ringing</value>
         </attribute>
      </range:fail>
      <range:match>
         <attribute>
            <aspect>door bell</aspect>
            <value>ringing</value>
         </attribute>
      </range:match>
   </range:any>
</range:match>
```

This last snippet would be sufficient as to be transferred as the carrier of seams describing the behaviour of the sound controller service under the undergoing situation. The sound controller could populate the context with the following attribute that defines both the state of the sound-volume and the system's reasoning behind its decision:

```
<attribute>
   <aspect>sound volume</aspect>
   <value>muted</value>
   <seams>
      <range:match>
         <range:any>…</range:any>
      </range:match>
   </seams>
</attribute>
```

Overall the sound-controller service could by itself also describe its possible influence on the context using yet another context-range such as the one in the following snippet, which declares that the sound controller is returning both an attribute about sound-volume expressed by enumeration, and a second attribute where the sound-volume is expressed using an externally defined range operator type (e.g., decibel).

```
<range:all>
   <attribute>
      <aspect>sound volume</aspect>
      <range:one>
         <value>muted</value>
         <value>low</value>
         <value>normal</value>
         <value>high</value>
      </range:one>
   </attribute>
   <attribute>
      <aspect>sound volume</aspect>
      <value><range:ext type="decibel"/></value>
   </attribute>
</range:all>
```

As one can observe, the same notation was used in all the above steps, despite the fact that each is representing a different view of the system's intelligence. In fact, the notation described above is rich enough to allow a wide range of heterogeneous services to express their reasoning, describe their behaviour, and communicate their underlying mechanisms contributing towards a compositional approach for developing context-aware applications. In the following sections we examine appropriate interactive mechanisms for end-users to control this process through a context-range direct editing environment in natural language.

3.6 Contextual Range Editor (CoRE)

The composition of syntactically correct and semantically meaningful expressions can be a challenging cognitive task, especially for non-expert programmers; to do so programmers need to resolve well-formedness constraints before they can get to the point of ensuring that the statement written, expresses the intended meaning. As the interaction framework of Dix et al. (1998) suggests, the user has to pass a significant articulatory gap travelling the distance for mapping her tasks to the input language of the system, and also an observation gap that has to be travelled to map the system output to her tasks. With our work we intend to a) support recognition rather than recall allowing the articulatory gap to be smaller, and b) to support natural and easily comprehensible descriptions of contexts as output to reduce the observation gap.

More specifically in our context-range direct-editing approach, we address this issue by presenting the user with a semantically equivalent aggregation of all the context-ranges of available services[8] corresponding to their possible outcomes. This aggregate range is populated by interrogating all services registered with the system and related to a particular task. The collected semantic information, expressed in the notation we introduced earlier, is sufficient to allow the editor to apply regrouping and simplification, and to transform the contextual-range from the involved services to interactive textual representations that are easier to comprehend by the user. Users are presented with a group of prefabricated expressions (representing the contextual range of the effective output of the involved services), in natural language, that are adorned with configurable operators, and operands that allow end-users to modify the underlying context-range in order to produce the desired result, by collapsing the undesired terms, changing the operands that connect them, and editing their specific parameters in a unified view (see figure 1.).

To ensure that the aggregated service range is easier to comprehend, well known results from mental models theory (Johnson-Laird, 1983) have been applied to guide the grouping and composition of terms by the editor. This theory suggests that people tend to think in terms of disjunctive normal form (DNF) when presented with a set of contingencies (Byrne & Johnson-Laird, 1992). This means that contingencies are easier to think of as a disjunction of conjunctions of simple terms. Accordingly, the editor is able to avoid compromising expressiveness by allowing users to describe a range of contexts using one or more alternative contingencies accompanied by any number of exceptions that may apply on each of them.

However, if applied blindly, the automatic population and grouping of terms can degenerate to modelling requirements on context, equivalent to an exhaustive truth table (due to the arbitrary number of information sources), which would make the logical expression too long and tedious to read. Furthermore it could even be made unnecessarily long and complex for the human reader by including contingencies known to be invalid due to incompatible logic statements. For this reason, it is interesting to present the user with the model of context that is valid in a particular contingency in order to simplify the presentation of terms and their manipulation.

Formative evaluations of the editor described in the following section, conducted during its iterative development, revealed that users are more likely to deviate from the general tendency to think in DNF when they encounter terms that are of a high affinity. This was subsequently confirmed through a laboratory experiment, presented in section 3.7. In such cases, the higher the affinity of terms, the more

[8] How services are made available is outside the scope of this chapter; one can consider that there is an appropriate middleware such as Amelie presented in the following chapter.

likely is that the user will think in terms of conjunctive normal form (CNF). For example, it is rather possible that one thinks and comprehends the CNF phrase *"I am eating and (I am holding a fork or I am holding a knife)"* rather than its equivalent DNF *"(I am eating and holding a fork), or (I am eating and holding a knife)"*. Furthermore, it is quite unlikely that a person would even consider a statement implying *"(I am eating and holding a fork), or I am holding a knife"* as intuitive.

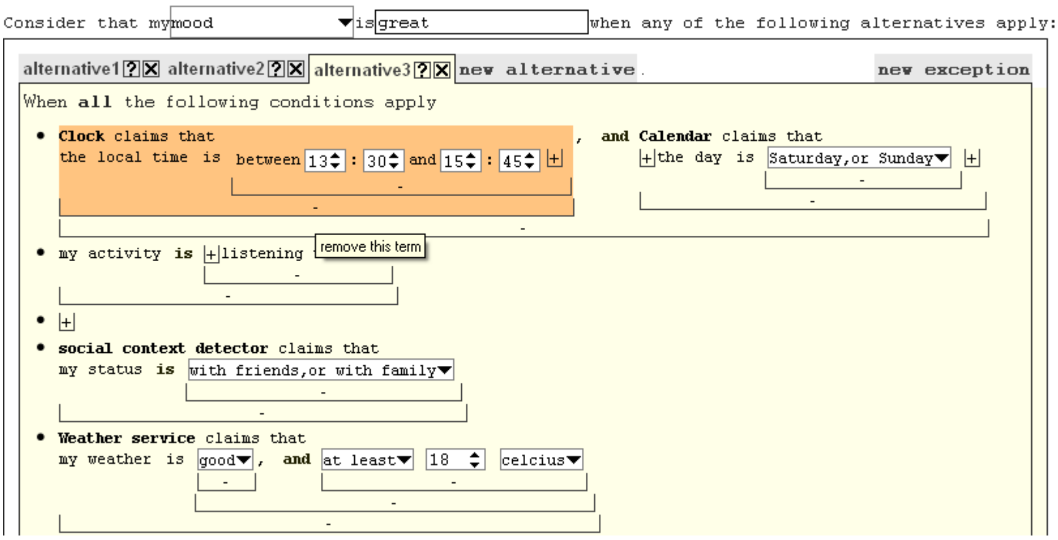


Figure1. The editor employed by a user to define the context-range under which she is in good mood; for that the user collapsed the undesired terms, and altered the term specific parameters

Therefore, to further facilitate non-expert programmers, several heuristics were applied in the context range editor by which terms are regrouped within each contingency, based on their affinity to each other.

3.6.1 Term-regrouping based on affinity heuristics

The editor uses the semantics of the ranges returned by its services not only to define transformations of the operators, but also in order to reorganize the visual representations in natural language for easier reading, without compromising the overall semantics and expressiveness. Below, we present the affinity defining regrouping heuristics and mechanisms that are applied throughout the contextual-ranges of the involved services.

Common Aspect Range

Let's consider for example that the user has installed two services; the first one detects whether the user's activity is working and the second one detects whether the user is listening to music. The first service's range could be described as *"opt (Attribute ('activity', 'working'))"* meaning that the service may optionally return an attribute about *activity* with value *working*. Similarly, the range of the second service could be described as *"opt (Attribute ('activity', listening to music))"*.

We would like the two ranges to be grouped together in logical expressions, since both of them refer to the same aspect (i.e. *activity*). The semantic information that the services disclose, is a sufficient basis for the editor to recognize their affinity and allows it to combine them in a common range *"any (Attribute 'activity', 'working'), Attribute ('activity', 'is listening to music'))"*. In the current version of the editor optional operators are considered by default to not be mutually exclusive, hence they can be combined using the *any* operator. The attentive reader may remark that it would be safer -with regards to correctness of the produced expression- to consider by default that the merged operators are mutually-exclusive; logical terms that in one's conception are mutually-exclusive (hence not related with conjunction) could be inclusive for someone else (hence could also be related with conjunction). In some cases the underlying service premises and in others an underlying personalized ontology would allow the editor to identify such mutually exclusive cases in a manner appropriate to each particular user. However, here we consider the non-ideal situation where the editor has no access to such ontology; thus the editor favours expressiveness and considers the joined terms as mutually inclusive.

Common Value Range

In the same direction, we can think of a service called *service1* that detects whether a fridge is open, and a second one called *service2*, that detects whether a cupboard is open. The first service's range could be described as *"Attribute ('fridge', one ('open', 'closed'))"* meaning that the service returns an attribute about the fridge with value either open or closed. Similarly the range of the second service could be described as *"Attribute ('cupboard', one ('open',' closed'))"*. In this case the editor could use the value ranges of the two attributes as a final way to identify similarities in order to combine the two services' ranges. In fact, the editor is able to recognize cases similar to the above, and combines them in the same range group if it can not apply any other way of regrouping.

Preferred group

Let's consider another example, where the user has installed a service, *service1,* which detects the activity sensed through some notional 'smart' table, and a second service, *service2,* which detects the number of chairs occupied in the dining room. In this case the editor could group the above services together, since they are both relevant to the "dining room". To enable this kind of regrouping we allow the services to declare in their ranges their "preferred" group. So, in the aforementioned case, the user could have labelled during the installation of the two services that the involved components are located in the dining room. The services could benefit from this, and when queried, return not only their range of values but also their preferred "group".

Explicit grouping

Obviously there may be cases that the above forms of regrouping are not desired. Designers may have evidence that the affinity of terms within a context-range can be more precisely captured 'by hand' rather than by applying the heuristics discussed above. We can easily imagine a service that detects various kinds of attributes from a specific context that the designer would rather keep grouped together. For this purpose, the "*group*" operator may be used, which is semantically equivalent to the "*all*" operator, with the difference that it can not be merged, broken apart, or altered, and forces an independent 'bulleted-list' with the textual representation of its contents. Actually, the editor finalizes the regrouping process, by generating groups for all the corresponding service ranges except those that were already fixed.

3.6.2 Translation to Natural Language and Interactive Elements

The above heuristics are applied to automate the simplification and regrouping of the contextual-ranges within each contingency resulting in an intuitive environment that, while maintaining formal expressiveness, does not compromise the users' ability to use this expressiveness and comprehend context aware descriptions.

In order to populate the graphical interface that allows the manipulation of expressions, the editor transforms the acquired output ranges from the involved services to a list of prefabricated expressions that the user is able to edit in order to construct a desired expression. In this transformation process, it is essential to construct the interface following the semantics of the context ranges so that, throughout the interaction, syntactic and logical integrity is ensured by allowing the user to make only "meaningful" modifications. This is achieved by taking in account the implications that a manipulation could have on a range.

Consider for example the contextual range *"any (Attribute ('activity', 'working'), Attribute ('activity', 'is listening to music'))"*. The above range can be transformed to its textual representation *"my activity is working* ***or*** *my activity is listening to music"*, and adorned – in the case of editing - with interactive elements that enable its manipulation (see figures 1 & 2). In this case the user is able to change the *"or"* operator *to "and"* and the other way around, since the range semantics yield that the two attributes are not mutually exclusive (i.e. according to the semantics of the above range, one may be *working* and *listening to the music* at the same time). Moreover the editor allows the user to collapse or expand operands in order to alter the semantics of the expression. So in the above example the user is able to collapse any of the two operands ("*my activity is working*", and "*my activity is listening to music"*) and of course to change the operator either to "*or"* or to *"and*".

Consequently the paraphrase expression and its underlying semantic may be modified with a single click to any of the following alternatives (or completely discarded):

- my activity is working **or** my activity is listening to music
- my activity is working **and** my activity is listening to music
- my activity is working
- my activity is listening to music

In the last two expressions, the terms "*working*" and "*listening to music*" may also be collapsed yielding the expression *"any information about my activity is known"* which is equivalent to the range *'any(Attribute('activity'))'.*

In a different example, consider the range *“Attribute (‘weather’, one (‘good’, ’fair’, ’terrible’))”*. This range is transformed by the editor to the textual-representation: *“the weather is good,* *fair* ***or*** *terrible”*. The operator *“one”* as mentioned earlier denotes that its operands are mutually exclusive; hence in this case the user is not allowed to change the *“or”* operator to *“and”* because this would lead to an expression that cannot be matched (since the mutual exclusive *‘one’* implies that the weather can not be good and terrible simultaneously). The original expression, in this case, can be modified, for example, to *“my weather is good or terrible”*, or to “*my weather is fair”* but not to “*my weather is good* ***and*** *terrible”*.

Contrary to a typical programming activity, the programmer is offered an all encompassing representation of the context ranges, which she can trim down by collapsing terms, and selecting from the operators to combine terms that make sense given the choices made so far during editing.

Special attention is paid to support users in handling negations in order to ensure comprehension and expressiveness. Services report in their semantics whether the absence of information implies an affirmative negation (note how this contrasts formal systems and logical programming environments that typically commit to one of the two choices independent of context). For example, a service detecting home presence could declare that the absence of detection signifies that the user is not at home or it could otherwise denote that the system does not have sufficient information to judge presence at all. When services support affirmative negation, they may explicitly provide the negative and affirmative forms of their range of values (in the example above ‘away’ could be denoted as the affirmative negation of ‘at home’, instead of ‘not at home’).

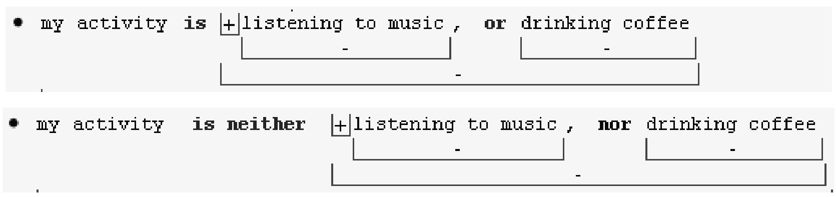


Figure 2. Clicking on the verb “*is*” changes the affirmative ‘*is listening to music or drinking coffee*’ at the top to the negative form ‘*is neither listening to music nor drinking coffee*’ in the bottom and vice versa.

When the system groups terms from such services, the user is able to express negation within the group, by clicking on the preceding verb; in order to minimize ambiguity when negation is applied over disjunction (i.e. *not (X or Y)*), the terms in their negative form are preceded with the preposition ‘*neither’* and connected with ‘*nor’* (i.e. *neither X nor Y*) (see figure 2 for an example).

Alternatively, negations over contingencies can be declared as an exceptional context-range that may apply to any of the alternatives (see figure 3). This feature not only guarantees uncompromised formal expressiveness, but also follows the dynamics of real life usage. Since we can expect that within a context-aware application users would typically notice exceptional situations that they would like to exclude from the typical system's behaviour on the fly, it is rather straightforward to apply them on the existing system behaviour as they are encountered during day-by-day use. In this respect, every time a user adds an "alternative" she effectively extends the matching context for some specific system behaviour, whereas whenever she adds an "exception", the context is being restricted.

Apart from the predefined range operators, services can extend and populate the editor with customized range editing and checking. For this, a service may use the operator *"ext"*, which instructs the editor to embed an external range editor/evaluator. During the transformation of the semantics to editable phrases, the editor embeds the external range-type editor in the flow of the textual expressions produced, whereas during evaluation the external range-type is requested to either match or reject its corresponding sub-context.

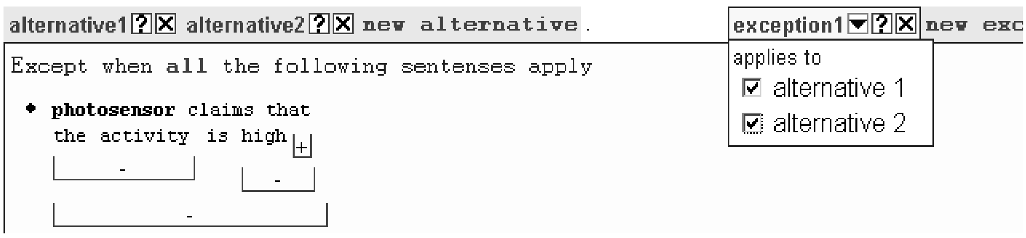


Figure 3. The editor allows the user to declare exceptions from the typical behaviour. Here the system is instructed to invalidate two alternatives when a photo sensor detects high activity.

To demonstrate the use of the *"ext"* operator consider the *desktop-activity-monitor* service. This is a service that we have designed to extract, among other information, the state of the mouse of the user's computer. In the initial design, the service, when queried, would return the range *"Attribute ('mouse', one ('moving', 'idle'))";* i.e. the service would return an attribute about the user's mouse with the value either *moving* or *idle*. Some experimentation with the service however indicated that it would be useful to be able to include temporal information in the returned data, so that the user can construct statements such as *"my mouse is idle for 2 minutes at least"*. Therefore, we implemented an external range definition, for the custom range-type "state-duration" and altered the returned range semantics to *"Attribute ('mouse', one (ext ('state-duration', 'moving'), ext ('state-duration', 'idle')))"*.

The new semantics allow the user (see figures 6 & 8) to create expressions such as *"my mouse is moving for less than 5 minutes, or idle", or "my mouse is idle between 10 and 30 minutes"*.

3.6.3 Supporting Intelligibility, Accountability and Control

Intelligibility, accountability and control can be examined in terms of their consequent design principles such as those identified by Edwards and Grinter (2001), and Lim et al. (2009). Such principles emphasize the need to inform users about the capabilities of the system and the model, which a system constructs of user's context and intentions, to provide feedback that allows users to tell what will be the consequences of a change in their context, and to defer and control system behaviour.

The contextual range semantics and the ability to transform them into both natural language and editable expressions support the aforementioned design principles by answering four main questions: (1) How does the system behave? (2) Why is something happening? (3) What will happen in a different situation? and (4) How can the system's behaviour be modified? We consider these below in turn.

How does the system behave?

Edwards and Grinter (2001) point out that systems relying on inference will always be susceptible to unreliability, and thus users have to have models of how such systems arrive at conclusions. In line with this observation, the question 'how does the system behave?' emerges as a key underpinning to intelligibility and accountability. Addressing this question becomes more important and compound as systems evolve and form compositional structures and complex premises.

In the previous sections we have shown how services can define both their outcome and the context under which they perform a behaviour using the context range semantics, and how such semantics can be uniformly transformed to natural language and interactive elements. This mechanism can be employed to allow the system to accurately describe its underlying logic.

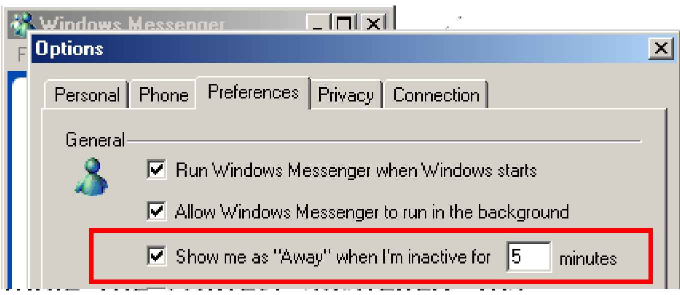


Figure 4. Screenshot from the preferences of a widespread messaging application; the user is informed that the system declares her as away when she is inactive for 5 minutes

Consider for example the context awareness manifestation that we discussed in the introduction of this chapter. Instant messaging applications typically detect and declare whether the user is away, based on the inactivity of the user's mouse and keyboard. This behaviour, however, is not self-evident because the user has to find out the underlying logic herself. In figure 4 a screenshot from a popular messaging application illustrates this exact ambiguity, as it is neither clear who has access to this information nor what the term 'inactive' might exactly mean.

Status Detection And Sharing

- **away**

  This service claims that your **status** is **away** and shares this information with all the entities of your contact list marked as **friend** when the `*desktop activity service*` claims that

  - your **mouse** is **idle** for 5 minutes at least , and your **keyboard** is **idle** for 4 minutes at least

Figure 5. The system gives precise feedback describing in natural language its behaviour

In contrast, using our notation and editor, the exact premises that guide this behaviour can be described and translated precisely to natural language allowing the user to understand, predict and anticipate the behaviour of the system in accordance to her interaction with the environment (see figure 5).

Why is something happening?

Residual to the requirements of intelligibility and accountability is the ability to provide feedback to end-users that allows them to understand why is something happening (or not). Consequently, as applications perform tasks and exhibit (or not) certain behaviours by evaluating their context, it is essential that the result of this evaluation should embed the decision path that yields the result itself. This is exactly what our prototypical tools support since the evaluation of a range versus a context not only returns a logical result but actually returns the semantics describing the matching or rejected context as shown in section 3.5.

The premise defining the detection of the user's status in the above example is described by the range *"all (Attribute ('mouse', 'idle'), Attribute ('keyboard', 'idle')"*. The evaluation of the above expression versus a situation where the mouse is idle while the keyboard is in use (i.e. *Attribute ('mouse', 'idle'), Attribute ('keyboard', 'in use'))* would yield a failure and also return the semantics describing why the rejection occurred. In the depicted example (figure 6), the resulting semantics (i.e. *"**fail** (all (**match** (Attribute ('mouse', 'idle')), **fail** (Attribute ('keyboard', 'idle'))))"*) indicate that the premise is falsified because the system failed to match the conjunctive term "*idle for 4 minutes at least*" in the current context. In figure 6 we see how the user can examine the current behaviour of the same service that we used in the previous example. The user is able to see that currently the system can not infer whether she is away; by clicking on the link under "explain" the user is able to examine exactly why.

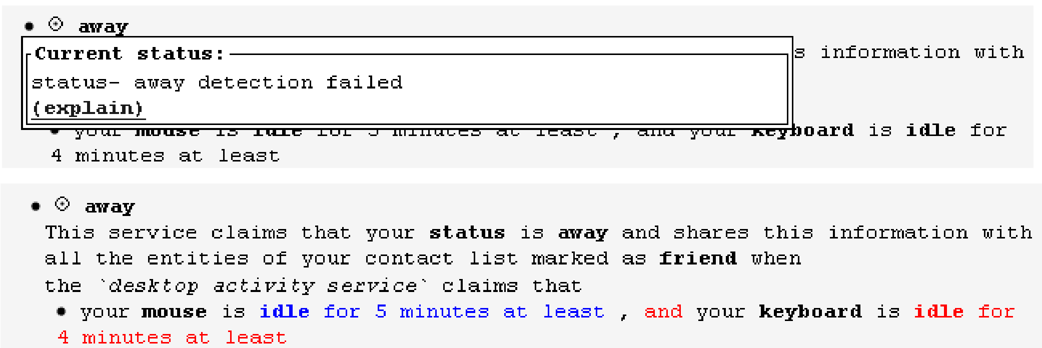


Figure 6. Clicking on the 'explain' anchor (top) reveals the seams (bottom) that trigger the specific behaviour. Blue annotations denote terms that are matched in the context, while red terms are the critical terms whose failure to be matched in the context results in the falsification of the expression.

The notation allows the system to highlight exactly the terms of the expression that are falsifying, or otherwise verify its outcome in a specific context. Broadly speaking, end-users are able not only to inspect the "variables" (terms) within expressions, but actually to get direct feedback for those that result in the verification (or not) of the expressions' evaluation.

How can I change the system's behaviour?

We have seen how the system allows users to realize the unreliability and imprecision of its inferences, and exposes its internals rather than masking them away. Essential to intelligibility and accountability is, furthermore, the ability of end-users to defer the system's behaviour.

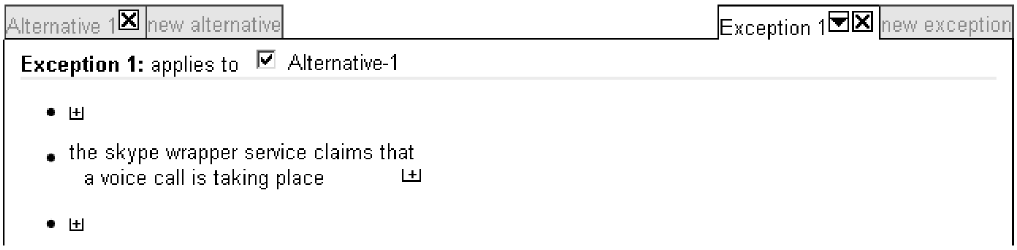


Figure 7. The user instructs the system to exclude a situation when she is in active conversation

In our running example the user could identify situations where the inactivity of her input devices does not imply that her status is away or she could even realize that different premises should be used depending on her audience. For example the user might desire to instruct the system to exclude very short mouse activity, or to exclude a situation when a program such as a movie-player or a web-conference is running on her computer. Since the range of factors influencing the user's self perception of being away is unpredictable, the system should provide sufficient flexibility to accommodate them. The level of control that state of the art applications provide to end-users is insufficient for addressing such cases where a compositional approach is clearly required.

In figure 7 the user has modified the behaviour of the system to better match her perception of being away. For that the editor is employed to exclude from its premises the situation when she is in an active voice call using Skype.

The structure of the editor, using alternatives and exceptions, allows end-users to expand and restrict the context that triggers specific system behaviours on a progressive day-by-day basis as users experience the system's outcomes.

What will happen if I do this?

Using the range semantic information, we are able not only to construct logical expressions concerning contextual-ranges but also define specific information contexts resembling imaginary yet plausible situations. For this we follow the same heuristics as we do while constructing the context-range editing environment, making sure however that the interactive elements enforce contingencies (see figure 8 top) and that a specific situation is defined instead of a context-range.

Returning to our running example, the user is able to query the outcome of the service that detects her status. As in the case of feedback about why something is happening or not, the system evaluates its premises against the imaginary situation populating this time the range editor with the colour coded evaluation semantics (see figure 8 bottom). This mechanism is likely to allow the users to iteratively explore the impact of their interaction with the context-sensing environment, and support them in modifying the system's behaviour according to their needs. In the example above, a user could verify that a short movement of her mouse could cause the system to stop declaring her as away.

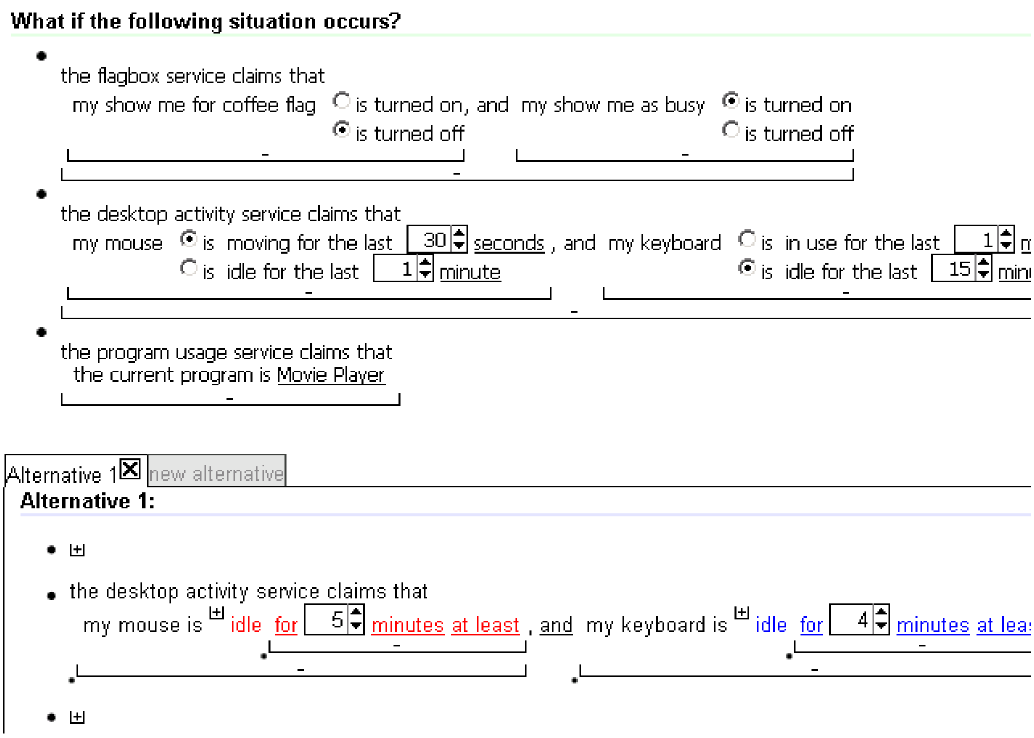


Figure 8. Enabling the user to test the system's behaviour under a specific context. On top the editor is employed to allow the user define an imaginary situation, while on bottom the premises of the service that declares the user as away are evaluated against this situation.

3.6.4 Experimental Evaluation

Several informal formative evaluations were conducted throughout the iterative development of the editor. Here we present a formal user test that was conducted in order to evaluate the features or the editor in terms of errors made and time needed to complete some representative tasks. The selection of tasks was such so that with a small set of tasks, several assumptions underlying the design of the editor could be evaluated. Participants were instructed to compose several logical expressions using the interface of the editor. The available terms that the participants could manipulate in order to complete the tasks were selected to cover a variety of affinity groups and terminal symbols.

The first group of terms consisted of date and time services (see figure 9), the second group contained several activities (reading the news, listening to the music, and drinking coffee) that notionally could have been produced by a diversity of heterogeneous services but still had been grouped together in the editor's interface because of the fact that they refer to the same aspect (i.e. activity). The third group consisted of attributes that could be perceived as mutually exclusive (e.g., home vs. office), all referring to locations (i.e. home, office, downtown). The last group contained a service that would describe one's social context (alone, with friends, with family, with colleagues).

Below we consider in turn the selected set of tasks in terms of the expressions that the users were instructed to construct.

- Task 1: *"You are at home listening to music, or you are downtown with some friends"*. With this task we wanted to examine the basic underlying concept of the editor, which was the structured editing of DNF expressions. We expected participants to respond by using an alternative for defining the contingency *'home and listening to music'*, and a second alternative for the contingency *'downtown with friends'*.
- Task 2: *"You are at home listening to music or reading the news"*. Based on the principle of grouping together terms with high affinity discussed earlier, we expected that participants would consider that the above statement was a conjunction of the phrase "*at home*" with the phrase "*listening to music or reading the news*" rather than a disjunction of the phrases "*at home listening to music*" and "*reading the news*".
- Task 3: *"You are with some colleagues or friends but not at the office"*. The participants could successfully complete this task either by writing a single contingency that is using a negation operator on the term *"at the office"*, or by writing an exception that could be applied on the phrase "*with some colleagues or friends*".
- Task 4: *"It is Sunday evening (but not dinner time) and you are home alone reading the news and listening to the music"*. This task was used to evaluate the editor's ability to capture concisely complicated expressions that require the use of negation in order to be completed successfully. In contrast with task 3, task 4 required participants to exclude from a contingency a subset in such a way that it could only be done by using an exception.
- Task 5: *"It's a Saturday afternoon, and you are drinking coffee with your friends either at your place or downtown"*. Task 5 was selected as a complement to tasks 1 and 2, allowing a wide space for possible solutions. In task 5, however, we used an implicit disjunction between two mutually exclusive terms ("you are at your place" and "you are downtown") expecting that, because of this strong affinity and the editor's ability to group these terms apart, the participants would complete the task using a single contingency in CNF rather than two contingencies in DNF.
- Task 6: *"You are reading the news but you are neither drinking coffee nor listening to music"*. This task was intentionally selected to be quite complicated by containing a negation on a disjunction of two out of three terms belonging to the same affinity group; the interface allowed only a unique solution for translating the above expression flawlessly, because the involved terms were placed in a flat affinity group and chosen not to support local negations (i.e. participants could not create within a single contingency-sheet the paraphrase *'reading the news and neither drinking coffee nor listening to music'*).

Procedure

Initially, each participant was shown a short video demonstration (approx. 3 minutes) of the editor's features, followed by a couple of example use cases. Then they had to complete three trial tasks. At the end of each trial task, and before proceeding to the next one, they could review a possible solution that was shown on a side pane. Then, participants would proceed to the main tasks. For each of the tasks, they were prompted with the task specification in English, (see figure 9) which then they were asked to express as a logical expression using the editor. The completion time for each task was recorded and then the participants would proceed to the next task. In the end of the session participants completed a short questionnaire with demographic data, and some open questions regarding their overall impressions of the editor. The terms, and the prompts that were used for the demonstration and the trial tasks were selected to differ from the terms used in the real tasks to avoid priming and confusing the participants.

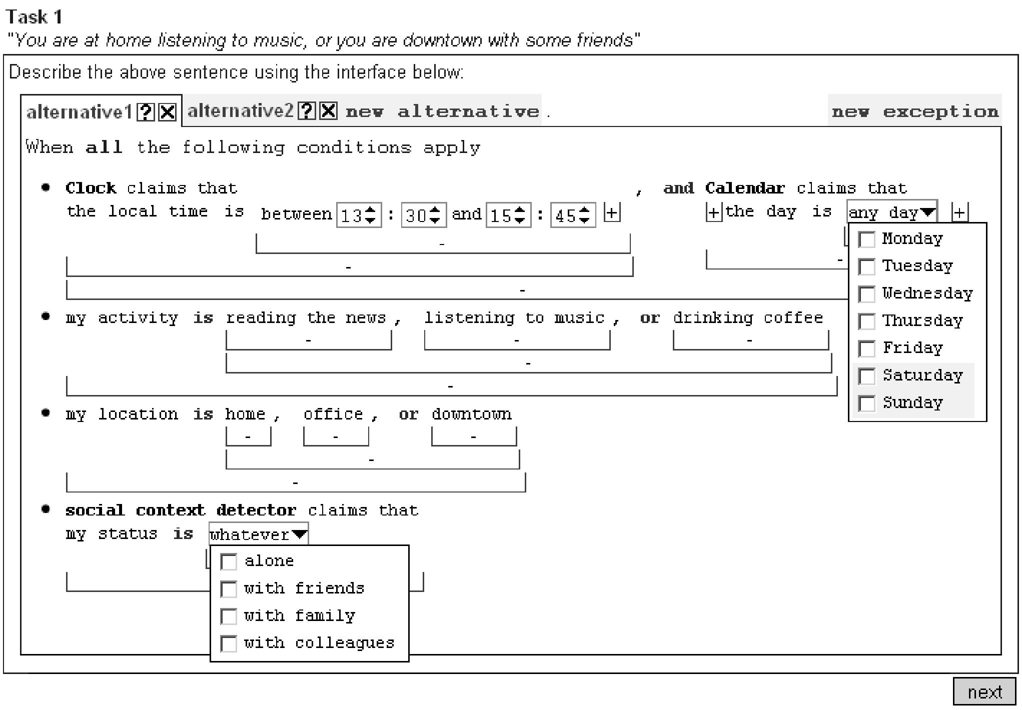


Figure 9. The setup of the editor for one of the tasks (task1)

Participants

Thirteen students (7 female, 6 male) from various departments of our University (ages 19-25, non native English speakers) participated in the evaluation study. Participants were screened to have none or very small programming experience and no training in formal logic. They received a small money compensation for their participation in the experiment, which lasted approximately 30 minutes.

Table1.
Summary of tasks and participants' responses

Task & response times(seconds)	Response variations	Frequency	Correct	Incorrect
"You are at home listening to music, or you are downtown with some friends", (mean=77.9; sd=11.3)	(home **and** listening) **or** (downtown **and** friends)	12	12	
	(home **and** listening **and** alone) **or** (downtown **and** friends)	1		1
"You are at home listening to music or reading the news",(mean=45.2; sd=17.3)	home **and** (listening **or** reading)	12	13	
	(home **and** listening) **or** (home **and** reading)	1		
"You are with some colleagues or friends but not at the office", (mean=53.1; sd=19.7)	(colleagues **or** friends) **and not** (office)	8	11	
	(colleagues **or** friends) **except** (office)	2		
	(colleagues **and not** office) **or** (friends **and not** office)	1		
	(colleagues **or** friends) **except** **(…)**	1		2
	(colleagues **or** friends)**and** (home **or** downtown)	1		
" It is Sunday evening (but not dinner time) and you are home alone reading the news and listening to the music ", (mean=131.4; sd=14.7)	(Sunday **and** evening **and** home **and** alone **and** reading **and** listening) **except** (dinner time)	11	11	
	Sunday **and** "evening" **and** home **and** alone **and** reading **and** listening	2		2
" It's a Saturday afternoon, and you are drinking coffee with your friends either at your place or downtown", (mean=53.1; sd=17.8)	Saturday **and** afternoon **and** drinking **and** friends **and** (at my place **or** downtown)	13	13	
"You are reading the news but you are neither drinking coffee nor listening to the music", (mean=71.7; sd=20.2)	reading **except** (drinking **or** listening)	7	7	
	reading **except** (drinking **and** listening)	1		6
	Reading	2		
	not (reading **or** drinking **or** listening)	3		

Results

Overall participants had no difficulties in correctly completing the tasks 1 to 5, while in task 6 the smallest number of correct responses was given (only seven of the participants completed the task without any errors). On average, male participants completed 5 tasks successfully while females completed 5.14 tasks successfully. In table 1 the summary of responses along with the response times per task is shown, while in figure 10 the frequency of correct responses per participant is shown. Ten participants made at most one mistake; five of them gave exactly one error response, while five gave no error responses at all. One participant responded successfully only in half (3/6) of the tasks successfully, and two participants responded correctly in 4 tasks.

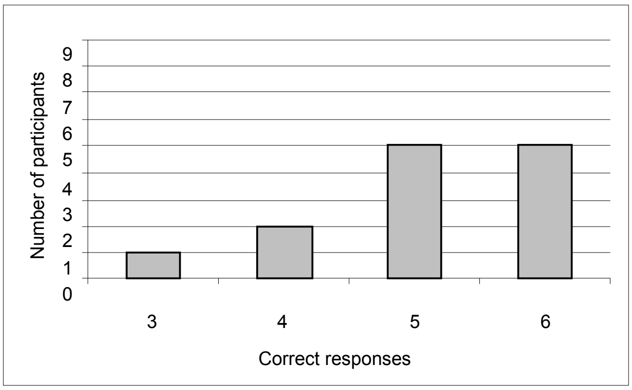


Figure 10. Frequency of correct responses

Discussion

The laboratory evaluation of the CORE editor shows that users are able to construct and manipulate realistic expressions to describe different contexts with only a minimal introduction to the system. To the best of our knowledge, there is no prior report in literature of the performance of non programmers in composing logical expressions of similar realism. Early studies like those relating to the IOS system (Cheverst et al., 2005) involved only expert computer scientists as users who were also involved in its development and reported no formal evaluation. Closer to our study, in the evaluation of iCAP (Dey et al., 2006) the experimental tasks required the creation of simple expressions composing conjunctions and hardly any negations or disjunctions. While we cannot directly compare with the results of Dey et al. as they were asking participants to map contexts to actions, we note that the time it took their participants to formulate much simpler logical expressions was of a much larger scale. Jones et al. (1999), using a purely graphical interface for boolean query specification, report much higher number of incorrect responses in expressions of similar complexity to the ones used here.

The structured editing approach presented here addresses to a large extent the issues identified by Pane, and Myers (2000). As they point out, people interpret the meaning of *'and'* either as conjunction or as disjunction depending on context. Therefore, at a higher level, within each contingency sheet we use the qualifier '*all the following conditions apply*', while we use the term '*alternative*' to make explicit that each sheet represents a contingency within a disjunction. The range-semantics and the exploitation of affinity further resolves this issue; for example when the semantics imply that there is an exclusive disjunction among terms, the user cannot misinterpret and erroneously alter the disjunctive operator *'or'* to *'and'*. Further supporting our approach, we note that in the examples given by Pane and Myers to identify this context-dependent interpretation of *'and'* as a disjunction, the terms used are typically mutually exclusive.

In the same work Pane and Myers remark that the perceived precedence of *'not'* differs when it is applied on disjunctions or conjunctions. People tend to perceive that *'not'* has lower precedence than *'or'*, while it has higher precedence than *'and'*. Therefore the expression '*not a or b*' is –erroneously from the perspective of Boolean logic- usually understood as *'not (a or b)'*; on the other hand it is difficult to interpret correctly the expression *'not (a and b)'* because the higher precedence of *'not'* over *'and'* may pose a blind (regarding parenthesis) interpretation as *'not (a) and b'*. As they suggest, this last ambiguity is bypassed in English by using instead the word *'unless'* and placing it in the end of the phrase followed by the conjunction. In order to compensate for these issues, CORE allows users to express negations over conjunctions using *'exceptions'* as a direct mapping of the word *'unless'*. Similarly, negations over disjunctions (i.e. *'not (a or b)'*) are displayed as *'neither a nor b'*, while at the same time negations on single terms are pushed to the end of phrases; hence expressions such as *'not(a) or b'* are displayed as *'b or not a'* in order to avoid the perceived precedence of *'or'* over *'not'*. Furthermore, the context-range notation promotes the use of 'affirmative' negation (i.e. the usage of explicit terms referring to the negative form of a term), eliminating the ambiguities that 'not' raises.

Overall, the concept of using the affinity of terms, and the prior knowledge of contextual range in order to regroup and generate an interactive editor that attempts a close mapping to the cognitive capabilities of users has not been reported elsewhere. This makes it quite difficult to attempt a direct comparison of our editor and notation against prior art, regarding performance in terms of effectiveness and efficiency. However, the results presented are clearly encouraging regarding the effectiveness of the editor and confirm the underlying rationale introduced in earlier sections.

Judging from the small number of incorrect responses we may conclude that the insights for grouping the terms based on their affinity within each alternative contingency were beneficial. The success of the semantic translation to natural language is further amplified by taking into account that none of the participants was a native English speaker.

Of course, the experimental setup and the results presented concern users' comprehension and the effectiveness of the editor and notation regarding their expressiveness, but can provide limited evidence regarding the actual support and benefit for the users' real contexts. In real life, end-user programming requires much more than the ability of users to comprehend and manipulate expressions about contexts or rules about context sensitive behaviour. It requires also that users have the motivation to do so, seeing enough benefits in their investment of effort to program the system. This is partly an evaluation of the cognitive costs and benefits of such an activity (Blackwell, 2002), but also relates to the existence of the appropriate context socially or organizationally, where the relevant investment in effort is justified (Mehandjiev et al., 2004). Future research could aim at setting up a field study that involves users of existing social networking tools, equipped with a set of preconfigured services that can autonomously collect information from their environments and make inferences about the users' situation and publish it in their existing social network. Combining surveys of user attitudes and observational data collected over a sustained period of use we will be able to provide a more realistic assessment of the perceived benefits and costs for the end-users.

3.7 The impact of context on the naturalness of logical expressions

In this section we attempt to establish the role of context in writing and comprehending logical programs.

Consider two logical expressions:

1) *John is walking in the forest and he is drinking a soft drink, or he is listening to the music and watching around.*
2) *John is walking in the forest or he is drinking a soft drink, and he is listening to the music or watching around.*

The first is a logical expression in Disjunctive-Normal-Form (DNF), while the second is a logical expression in Conjunctive-Normal-Form (CNF). What makes one think of the first as natural, while the second is almost impossible to comprehend? According to the Mental Models theory (Johnson-Laird, 1983; 1994) the human mind constructs reality, conceives alternatives to it, and searches out the consequences of assumptions, in terms of representations in short term memory that Johnson-Laird calls mental models. In this transformation process a number of alternative mental models each representing a possibility is generated. Crucially for this theory, mental models represent what can be true according to the premises, but not what is false (Johnson-Laird, 2001). Compared say to a truth table or other computer based representations, this mechanism allows for parsimony because reasoners do not allocate cognitive resources with what is false; it does though also give rise to a variety of fallacies in deductive reasoning that cognitive scientists have studied extensively over the years. Ormerod (2000) argued that reasoners construct the minimal set of models needed to infer a conclusion, while in a similar direction it has been observed that reasoners often draw conclusions based on a single model of the premises (Sloutsky & Goldvarg, 1999). Such findings explain why reasoners while constructing multiple models, are likely to fail to envisage a model (Sloutsky, & Johnson-Laird, 1999). Accordingly, Mental Models theory accounts for the general tendency of people to produce logical expressions in Disjunctive-Normal-Form (DNF), when presented with a set of contingencies, such as a truth table (Byrne & Johnson-Laird, 1992); in the example above the DNF expression requires 2 models as opposed to the CNF expression that requires 4.

One could benefit from this fundamental tendency and apply it to the design of a logical expression editor, expecting that users will tend to find a DNF based structured editor as an intuitive translation of their indented concepts.

On the other hand, the theory of Johnson-Laird is agnostic to a big extent regarding the intended meaning of the terms. Phrases like *"it is raining"*, *"it is windy"* and *"it is Monday"* are not all alike. People can differentiate between concepts that are similar, more like each other, or referring to different things. In the aforementioned phrases one would find naturally more akin the phrases *"it is raining"* and *"it is windy"* when compared to the phrase *"it is Monday"* but not when compared to the phrase *"it is cloudy"*. This notion we describe as "affinity" of the terms related in the logical expression. Affinity is implicit knowledge that people bring into the task of describing logical expressions; it is not derived from the terms themselves but rather reflects knowledge of a particular user or knowledge some users might assume to be shared (we do not need to draw this distinction here). This type of knowledge is typically not addressed in programming environments and has not yet been addressed in mental models theory. The notion of affinity is a key factor in the design of the CoRE editor, and it is used to group terms with high affinity within a contingency. Putting aside whether the heuristics that we used in CoRE are proper models for the affinity between terms, in order to validate the approach taken in the design of the editor it is important to validate in a controlled setup the role that affinity plays in the understanding of logical expresssions, as this could better instruct the design of structured logical editors that minimize both the articulatory and the observation gaps (Dix et al., 1998).

Our starting point is the general tendency people have for using DNF constructs to form logical expressions as argued by the original theory of Johnson–Laird. Then, we attempt to identify the impact of affinity of terms over this tendency and how affinity may lead people to favour CNF. Our findings, relating to what expressions seem more natural and are better understandable by people, are central to the design of the CoRE editor; based on these findings we group terms in disjunctive-groups within contingencies.

For example, instead of the previous –rather difficult to comprehend- CNF expression (2) consider the following one with the same form, but where the terms are slightly changed:

3) John is drinking coffee or tea, and he is listening to music or watching TV.

This CNF expression, typically, seams quite easy to comprehend, despite the form of the expression is identical to the previous (2). In this section we argue that what makes this last expression so different from (2) is exactly the affinity of its disjunctive terms (i.e. *drinking coffee/drinking tea,* and *listening to music/watching TV*) that allows for an easy translation to mental models. (One could perhaps distinguish between perceived affinity and affinity to emphasize that only when users perceive a certain affinity will this phenomenon take place. In the following we shall not maintain this qualification for brevity, but the distinction of what is perceived and what can be known to designers a priori has been noted.

Given three logical terms *A, B* and *C* one could use –in the context of some problem- both the DNF expression *'(A and B) or C'* or its CNF equivalent *'(A or C)* *and (B or C)'*. Similarly, in the context of another problem, one could use the CNF expression *'(A or B) and C'* or its DNF equivalent *'(A and C) or (B and C)'*.

In an editing environment such as the CoRE editor, DNF is supported using alternative contingencies that appear in separated tab-sheets. On the other hand similar terms within alternatives are put together in -presumably- disjunctive groups based on their affinity, hence allowing CNF expressions such as *'(A or B) and C'* to be edited within a contingency. This ability is expected to be beneficial not only because the user can avoid the –rather lengthy- equivalent DNF *'(A and C) or (B and C)'*, but more importantly because we expect that there are situations where the user may be 'blind' regarding the latter (i.e. they will not be able to conceptualize it).

3.7.1 Experiment 1

The aim of this experiment was to investigate the importance of affinity in the selection between alternative syntactical forms. In terms of logic-formalization, DNF and CNF are of equivalent expressiveness and can both be used to represent any Boolean expression.

More specifically, we tried to identify whether the affinity of terms can make participants perceive the CNF formulation *"A and (B or C)"* as more natural than its semantically equivalent DNF *"(A and B) or (A and C)"* and vice versa when any of the above equivalent forms are the solutions to a given problem. Similarly we tried to identify whether the affinity of terms makes one think of the DNF *"A or (B and C)"* as more natural rather than its semantic equivalent CNF *"(A or B) and (A or C)"* and vice versa. Our prediction was that when the participants would be prompted to select the more natural of two identical expressions, the first in DNF and the second in its CNF equivalent, the affinity of terms would play a central role in their selection.

Let us look at different affinity relations between such terms. First we shall consider the case where two terms have close affinity and the third one does not. Then we shall consider cases where the two similar terms can be assumed (here with some common sense) to be mutually exclusive. Then we consider the case where all three terms are equally similar or mutually exclusive.

Consider for example the following set of phrases:

A. Joe is holding a pen
B. Joe is holding a pencil
C. Joe is listening to music

In the above set, a strong affinity among the two statements (*"Joe is holding a pen"*, *"Joe is holding a pencil")* is imposed in relation to the third statement ("*Joe is listening to music*").

One can combine the abovementioned phrases in a CNF statement *"(A or B) and C", i.e.:*

*E1cnf: (Joe is holding a pen **or** a pencil) **and** he is listening to music*

Based on the distributivity property of conjunction over disjunction, its DNF equivalent *"(A and C) or (B and C)"* could also be used, i.e.:

*E1dnf: (Joe is holding a pen **and** he is listening to music) **or** (Joe is holding a pencil **and** he is listening to music)*

In a different context one can also combine the abovementioned phrases in a CNF statement *"(A or C) and B", i.e.:*

*E2cnf: (Joe is holding a pen **or** he is listening to music) **and** he is holding a pencil*

Based on the distributivity property of conjunction over disjunction, its DNF equivalent *"(A and C) or (B and C)"* could also be used, i.e.:

*E2dnf: (Joe is holding a pen **and** he is holding a pencil) **or** (Joe is holding a pencil **and** he is listening to music)*

We were expecting that this exact affinity would impose participants to perceive as natural the statement *E1cnf* (a statement that can be directly edited within a single contingency of the CoRE editor) compared to *E2cnf* (a statement that can not be directly edited in a contingency of CoRE). Assuming then that the affinity does not influence how natural the CNF and DNF equivalents are perceived, our null hypothesis is that the proportion of CNF statements in the first case (i.e. *E1cnf* vs *E1dnf*) should be equal to the proportion of CNF statements in the second case (i.e. *E2cnf* vs *E2dnf*).

Consider next, the following set of phrases:

A. Joe is lying on the sofa
B. Joe is sitting on the chair
C. Joe is listening to music

In this set, the imposed mutual exclusion among the terms *"Joe is lying on the sofa"* and *"Joe is sitting on the chair"* implies such a strong affinity among them compared to the third (i.e. "*Joe is listening to music"*), that our expectation is that the CNF paraphrase "(*Joe is lying on the sofa **or** sitting on the chair) **and** (he is listening to music)* " should be considered more natural compared to its equivalent DNF "(*Joe is lying on the sofa **and** he is listening to music)* ***or*** "(*Joe is sitting on the chair **and** he is listening to music)"*.

Contrary to the above, and when considering phrases of similar affinity such as the phrases below we were expecting as predicted by the Mental Models theory, that the DNF expression *"(A and C) or (B and C)"* would still be perceived as natural despite being longer than its CNF equivalent *"(A or B) and C"*.

A. Joe is holding a red pen
B. Joe is holding a green pen
C. Joe is holding a blue pen

Furthermore, and in accordance to the Mental Models theory, we expected that, independently of the terms' affinity, 'short' DNF expressions of the form *"(A and B)* ***or*** *C"* should be perceived as more natural compared to their equivalent, but lengthy, CNF expressions of the form *"(A or C) and (B or C)"*.

In total 6 different affinity modes have been examined; they are summarized in table 2 together with a shorthand notation to represent each case. Note also the special shorthand introduced for mutually exclusive terms.

Table 2.
Affinity modes and example phrases[9].

Affinity	Term A	Term B	Term C
AB	Joe is holding a pen	Joe is holding a pencil	Joe is listening to music
ABc	Joe is holding a pen	Joe is holding a pencil	Joe is holding a sheet of paper
Ab	Joe is holding a pen	Joe is holding a sheet of paper	Joe is listening to music
mxAB	Joe is lying on the sofa	Joe is sitting on the chair	Joe is listening to music
ABC	Joe is holding a red pen	Joe is holding a green pen	Joe is holding a blue pen
Abc	Joe is drinking coffee	Joe is listening to music	Joe is reading the news

Design

The experiment had a within-subjects design. Each participant was presented with the same set of semantically equivalent pairs of sentences each combining three terms. Then the participant was asked to choose for each pair the more natural one. Assuming that the affinity of terms does not play a role in the selection of CNF over DNF, one could expect that given that the terms *A* and *B* are of a high affinity (compared to the third term *C*) then the proportions of CNF and DNF selections should be the same regardless of whether the term *C* appears as conjunctive or disjunctive. The order of the terms *B* and *C* was swapped where applicable, resulting in two additional pairs for the affinity modes *AB*, *ab*, and *ABc*. Overall 17 pairs of expressions were generated (tables 3 and 4) to check the effect of the affinity on what is considered more natural by the participants. To control for fatigue effects, a scheme was applied for randomizing the order in which pairs were presented.

[9] In modes *AB*, *ABc*, and *ab* the affinity of the terms *A* and *B* is considered higher compared to the third. In the mode *mxAB* the terms *A* and *B* are implicitly mutually exclusive forming a rather strong affinity compared to the third. In the mode *ABC* the three terms are considered to be equally similar. The same holds for the mode *abc* based however on the fact that there is no apparent affinity among any two of the terms compared to the third.

Table 3
List of CNF (compact syntax) and DNF (expanded syntax) equivalents in the case of conjunction over disjunction[10]

Affinity mode	Order of terms	Conjunction over disjunction CNF compact (A or B) and C	Conjunction over disjunction DNF expanded (A and C) or (B and C)
AB	ABx	He is holding a pen or a pencil, and he is listening to music	He is holding a pen and he is listening to music, or he is holding a pencil and he is listening to music
	AxB	He is holding a pencil or he is listening to music, and he is holding a pen	He is holding a pencil and a pen, or he is listening to music and he is holding a pen
ABc	ABc	He is holding a pen or a pencil, and he is holding a sheet of paper	He is holding a pen and a sheet of paper, or he is holding a pencil and a sheet of paper
	AcB	He is holding a pen or a sheet of paper, and he is listening to music	He is holding a pen and he is listening to music, or he is holding a sheet of paper and he is listening to music
ab	abx	He is holding a pencil or a sheet of paper, and he is holding a pen	He is holding a pencil and a pen, or he is holding a sheet of paper and a pen
	axb	He is holding a pen or he is listening to music, and he is holding a sheet of paper	He is holding a pen and a sheet of paper, or he is listening to music and he is holding a sheet of paper
mxAB	mxAB	He is lying on a sofa or he is sitting on a chair, and he is listening to music	He is lying on a sofa and he is listening to music, or he is sitting on a chair and he is listening to music
ABC	ABC	He is holding a red pen or a blue pen, and he is holding a green pen	He is holding a red pen and a green pen, or he is holding a blue pen and a green pen
abc	abc	He is drinking coffee or he is listening to music, and he is reading a newspaper	He is drinking coffee and he is reading a newspaper, or he is listening to music and he is reading a newspaper

Participants:

Thirty eight students from the university campus participated in the experiment (Ages 19 – 24, no or very small programming experience and training in formal logic). They were paid 6 Euros for participating in the experiment which lasted approximately 30 minutes. To avoid biasing the participants they were asked posteriori about their background and training; as a result 2 participants were omitted from the analysis.

[10] Grayed cells account for the expressions that we were expecting to be perceived as less natural.

Table 4
List of DNF (compact syntax) and CNF (expanded syntax) equivalents
in the case of disjunction over conjunction

Affinity mode	Order of terms	Disjunction over conjunction	
		DNF compact (A and B) or C	CNF expanded (A or C) and (B or C)
AB	ABx	He is holding a pen and a pencil, or he is listening to music	He is holding a pen or he is listening to music, and he is holding a pencil or he is listening to music
	AxB	He is holding a pencil and he is listening to music, or he is holding a pen	He is holding a pencil or a pen, and he is listening to music or he is holding a pen
ABc	ABc	He is holding a pen and a pencil, or he is holding a sheet of paper	He is holding a pen or a sheet of paper, and he is holding a pencil or a sheet of paper
	AcB	He is holding a pen and a sheet of paper, or he is holding a pencil	He is holding a pen or a pencil, and he is holding a sheet of paper or a pencil
Ab	abx	He is holding a pen and a pencil, or he is listening to music	He is holding a pen or he is listening to music, and he is holding a pencil or he is listening to music
	axc	He is holding a pencil and he is listening to music, or he is holding a pen	He is holding a pencil or a pen, and he is listening to music or he is holding a pen
mxAB	mxAB	*Excluded because the mutual exclusiveness of A and B yields invalid the conjunction of A and B.	
ABC	ABC	He is holding a blue pen and a red pen, or he is holding a green pen	He is holding a blue pen or a green pen, and he is holding a red pen or a green pen
Abc	abc	He is drinking coffee and he is listening to music, or he is reading a newspaper	He is drinking coffee or he is reading a newspaper, and he is listening to music or he is reading a newspaper

Procedure and materials:

The experiment took place in our premises at the Psychological Laboratory which is partitioned in sound-isolated individual booths for each participant, allowing experimental sessions with more than one participant simultaneously. Each booth is equipped with a PC connected to the internet, which we used to collect the data and administer the overall process.

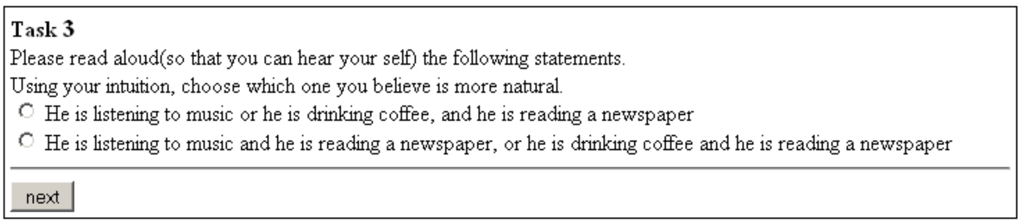


Figure 10. A screenshot from a prompt to the participant during the second experiment

Each participant was presented (see figure 10) with the 17 pairs of expressions in a random order and was asked to select from each pair the statement that according to her intuition seemed more natural. The participant would be presented with the first pair, then asked to read aloud the two statements and choose the one that seemed more natural. To avoid overlooking any of the statements or participants choosing for their convenience to rush through items and score hastily, the participants could only advance to the next pair after a predefined time (60 seconds) had elapsed.

Results

In tables 5 and 6 the participants' responses are presented for each test case. Table 5 summarizes the responses of participants for the 9 pairs where the CNF construct was in compact form (i.e. conjunction over disjunction: (x or y) and z), whereas table 6 summarizes the responses of participants for the 8 pairs where the DNF construct was in compact form (i.e. disjunction over conjunction: (x and y) or z). Below we discuss in turn the results for the case of Conjunction over Disjunction and for Disjunction over Conjunction.

Table 5
Subject responses for conjunction over disjunction in each affinity mode.

Conjunction over disjunction - CNF compact, DNF expanded		
Affinity Mode	DNF constructs	CNF constructs
ABx	2	33
AxB	18	18
ABc	10	26
AcB	20	14
abx	3	32
axb	17	18
mxAB	7	28
ABC	20	14
abc	21	15

Conjunction over Disjunction:

Consider three logical terms *A*, *B*, and *X* where *A,* and *B* are two terms with high affinity compared to *X*. We may combine these terms using conjunction over disjunction in order to form the expression *"(A or B) and X"*. This compact CNF expression can also be written in its DNF equivalent *"(A and X) or (B and X)"* using the distributivity property of conjunction over disjunction. By altering the order of the three terms we may also produce the CNF expression *"(A or X) and B"* and similarly its DNF equivalent *"(A and B) or (X and B)"*.

If the affinity of terms does not play a role in what form (CNF versus DNF) is perceived as more natural, then we should expect that the proportion of frequencies of preference of CNF over DNF would be the same irrespectively of the permutation of the terms *B* and *X*. The z-test for proportions of frequencies rejects the above hypothesis, and indicates that participants consider significantly more natural the CNF than its equivalent DNF in the fist permutation (i.e. *"(A or B) and X"* versus *"(A and X) or (B and X)"*) compared to the second (i.e. *"(A or X) and B"* versus *"(A and B) or (X and B)")*, (*z=3.884, a<0.05*). A logical explanation of this phenomenon is that during the process of transforming the presented expression to mental models, an implicit mechanism allows this translation to occur more naturally when the two disjoint terms are of high affinity (i.e. *A*, and *B*) as opposed to when the two disjoint terms are not (i.e. *A*, and *X*).

Indeed, the same holds also in the other similar degrees of affinity that we have tested: more specifically, when the affinity mode is *ABc,* the z-test for proportions of frequencies indicates that the subjects consider the CNF significantly more natural than its equivalent DNF in the 1st permutation (i.e. *ABc*) compared to the 2nd (i.e. *AcB*), (*z=2.381, a<0.05*), and similarly when the affinity mode is *ab* the z-test for proportions of frequencies indicates that the subjects consider significantly more natural the CNF than its equivalent DNF in the 1st permutation (i.e. *abx*) compared to the 2nd (i.e. *axb*), (*z=3.439, a<0.05*).

In the special affinity mode, where the terms *A* and *B* are perceived as mutually exclusive, the outcome is even stronger. The exact binomial sign test indicates that the subjects consider the CNF significantly more natural than its equivalent DNF when the involved terms are perceived as mutually exclusive, 28 of 35 subjects finding the CNF more natural compared to its equivalent DNF, *a<0.05*. Based on the same insight as in the previous cases, the mutual exclusiveness of the terms *A* and *B* in this case apparently makes the disjunction of *A* and *B* strong enough to allow a direct mapping of the CNF form *"(A or B) and X"* to a mental model.

Based on the same insight we would expect that when there is no clear affinity between two of the three terms compared to the third (*ABC*, and *abc* affinity modes), the mechanism that makes CNF perceived as natural should fade away. Consider three logical terms *A*, *B*, and *C* that have the same degree of affinity compared to each other. In this case we may compare the CNF expression *"(A or B) and C"* with its DNF equivalent *"(A and C) or (B and C)"* (e.g. *'He is holding a red pen or a blue pen, and he is holding a green pen'* versus *'He is holding a red pen and a green pen, or he is holding a blue pen and a green pen'*). While the majority of subjects (20 of 34) indeed consider the DNF more natural than its equivalent CNF, the exact binomial sign test does not allow us to consider the difference (*a>0.05*) significant. The same also holds when the three terms have no apparent affinity (affinity mode *abc*), 21 of 36 subjects finding the DNF more natural than the CNF, yet the exact binomial sign test does not allow us to consider significant the difference (*a>0.05*).

The results suggest that, as the affinity between disjunctive terms becomes more apparent compared to the third, participants tend to regard the CNF statements more natural. On the other hand, when the participants can not identify a clear affinity between two terms compared to the third (mode *abc*), as well as when the three terms are perceived by the participants as belonging to the same affinity group (mode *ABC*), they tend to consider the DNF statement more natural than its CNF.

Disjunction over Conjunction:

Consider three logical terms *A*, *B*, and *X* where *A* and *B* are two terms with high affinity compared to *X*. We may combine these terms using disjunction over conjunction in order to form the compact expression *"(A and B) or X"*. This DNF expression can also be written in its CNF equivalent *"(A or X) and (B or X)"* using the distributivity property of disjunction over conjunction. By altering the order of the three terms we may also produce the DNF expression *"(A and X) or B"* and similarly its CNF equivalent *"(A and B) or (X and B)"*.

Table 6
Subject responses for disjunction over conjunction in each affinity mode.

Disjunction over conjunction – DNF compact, CNF expanded		
Affinity Mode	DNF constructs	CNF constructs
ABx	31	5
AxB	34	2
ABc	32	4
AcB	27	9
abx	32	4
axb	28	7
ABC	33	3
abc	32	4

In contrast with the case of conjunction over disjunction, we find that in the above cases the affinity itself does not play any role in the preference of CNF over DNF; in reality and in compliance with the general prediction of Mental Models theory, the preference of DNF prevails consistently, irrespective of the permutation and the degree of affinity. This should come as no surprise; not only are the DNF constructs in this case shorter than their CNF equivalents (hence occupying less mental models), but also their conjunctive terms are in most cases of high affinity, making them quite easy to translate into mental models.

The exact binomial sign test indicates that the subjects consider the compact DNF significantly more natural than its equivalent expanded CNF in the case of disjunction over conjunction, when two of the terms have high affinity irrespective of the order of terms in the prompted expression, 31 of 36 subjects finding the DNF more natural in the 1st case (*ABx*), $a<0.05$, and 33 of 36 subjects finding the DNF more natural in the 2nd (*AxB*), $a<0.05$.

The same results hold also in lower degrees of affinity *ABc*, and *ab*. When the affinity mode is *ABc* (cases *ABc* and *AcB*) the exact binomial sign test indicates that the subjects consider the DNF significantly more natural than its equivalent CNF in the case of disjunction over conjunction when two of the terms have a clear affinity irrespectively of the order of terms in the prompted expression, 32 of 36 subjects finding the DNF more natural in the 1st case (*ABc*), $a<0.05$, and 27 of 36 subjects finding the DNF more natural in the 2nd (*AcB*), $a<0.05$.

When the affinity mode is *ab* (cases *abx* and *axb*), the exact binomial sign test indicates that the subjects consider the DNF significantly more natural than its equivalent CNF in the case of disjunction over conjunction when two of the terms have high affinity, irrespectively of the order of terms in the prompted expression, 32 of 36 subjects finding more natural the DNF in the 1st case (*abx*), $a < 0.05$, and 28 of 35 subjects finding more natural the DNF in the 2nd (*axb*), $a<0.05$.

Unsurprisingly this is also the case when the three terms are of the same affinity when compared to each other: considering three logical terms *A*, *B*, and *C* that have the same degree of affinity compared to each other, we may compare the DNF expression *"(A and B) or C"* with its CNF equivalent *"(A or C) and (B or C)"*. The exact binomial sign test in the case of high affinity among all terms (*ABC*) indicates that the subjects consider the DNF significantly more natural than its equivalent CNF, with 33 of 36 subjects finding the DNF more natural in the case *ABC*, $a < 0.05$. The same holds also when the three terms have no apparent affinity (affinity mode *abc*), with 32 of 36 subjects finding the DNF more natural ($a<0.05$).

3.7.2 Experiment 2

The previous experiment addresses whether the grouping of terms based on affinity, as was done in CoRE editor for example, can help reduce the observation gap (i.e. the ability of users to map interactive elements to mental models). A different setup could serve in addressing whether the articulatory gap can be reduced as well (i.e. the ability of users to map their mental models to interactive elements).

In terms of CoRE and in the case, for example, of the affinity mode *AB* while the user-interface allows users to directly map the expression *"(A or B) and X"*, it does not do the same with the expression *"(A or X) and B"*; the latter has to be first transformed by the user to its DNF equivalent *"(A and B) or (X and B)"*. Similarly, when three terms of the same affinity (i.e. affinity modes *ABC*, and *abc*) are involved in a group, the interface -as it avoids parenthesization- does not allow to directly map the expression *"(A or B) and C"* and again the user needs to first do the transformation to its DNF equivalent *"(A and C) or (B and C)"*.

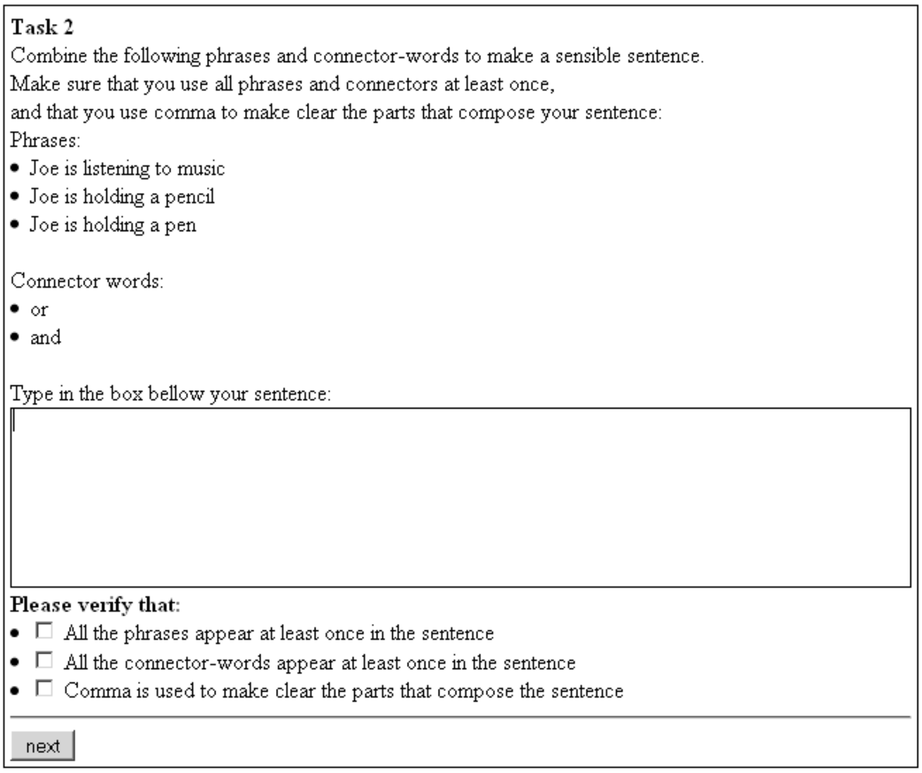
Participants were presented with a set of three phrases (terms), in random order, and were instructed to combine them using the connectives "and", and "or" in order to construct a sensible sentence. Our goal was to elicit spontaneous logical constructs that we could classify either as Disjunctive Normal Form (DNF) or as Conjunctive Normal Form (CNF). According to mental models theory we were expecting that participants would tend to combine the phrases using DNF. However, like with the previous study, we were expecting that by manipulating the affinity among the presented terms we would elicit more CNF paraphrases when two of the terms were of high affinity compared to the third. Of our interest here, was to assess to which extent a structured editing approach that facilitates regrouping of terms based on affinity would encounter situations that can not be directly mapped on such an interactive structure.

Design

In this experiment a within subject design was chosen; all participants were exposed to all six of the affinity modes and were instructed to combine their respective terms into sensible sentences using both of the connectives 'and' and 'or'. To counterbalance for fatigue effects, a scheme was applied randomizing the order of presentation of each affinity mode.

Procedure and materials

The experiment took place in our premises at the Psychological Laboratory which is separated in sound-isolated individual booths for each participant, allowing experimental sessions with more than one participant simultaneously. Each booth is equipped with a PC connected to the internet, which we used to collect the data and administer the overall process.


Task 2
Combine the following phrases and connector-words to make a sensible sentence.
Make sure that you use all phrases and connectors at least once,
and that you use comma to make clear the parts that compose your sentence:
Phrases:
- Joe is listening to music
- Joe is holding a pencil
- Joe is holding a pen

Connector words:
- or
- and

Type in the box bellow your sentence:

Please verify that:
- ☐ All the phrases appear at least once in the sentence
- ☐ All the connector-words appear at least once in the sentence
- ☐ Comma is used to make clear the parts that compose the sentence

next

Figure 11. Prompting a participant to type in a paraphrase using three terms and the connectives 'or' and 'and'

Participants were presented (see figure 11) with each of the six affinity modes in random order and were instructed to generate a sensible sentence using the provided terms and the connectives "*and*", and "*or*". Before proceeding to the next set, participants were requested to confirm that they used all the terms, both of the connectives, and confirm that a comma was used when necessary, to identify the parts that compose their sentence.

Participants

Thirty eight students from the university campus participated in the experiment (Ages 19 – 24, non native English speakers, no or limited experience with formal logic and programming). They were paid 6 Euros for participating in the experiment which lasted approximately 30 minutes. To avoid biasing the participants they were asked posteriori about their background and training; as a result 2 participants were omitted from the analysis.

Results

After eliciting spontaneous logical constructs from the users two independent observers classified them as either DNF-compact, DNF-expanded, CNF-compact, CNF-expanded and also in terms of the ability of directly mapping them on a user interface that a priori groups the involved terms based on their affinity. Table 7 summarizes the number of elicited constructs per affinity mode.

Table 7
Number of constructs per affinity mode and syntactical form generated from the participants. In parentheses is the number of constructs that could not be directly mapped on a structured editor such as CoRE.

	DNF constructs		CNF constructs				
Affinity Mode	Compact (x and y) or z	Expanded (x and y) or (z and y)	Compact (x or y) and z	Expanded (x or y) and (z or y)	Supported	Unsupported	Total
AB	9(0)	3(0)	21(0)	0(0)	33	0	33
ABc	11(0)	5(0)	18(0)	0(0)	34	0	34
Ab	16(0)	2(0)	16(0)	0(0)	34	0	34
mxAB	11(0)	6(0)	17(0)	0(0)	34	0	34
ABC	14(0)	8(0)	9(9)	0(0)	22	9	31
Abc	18(0)	7(0)	8(8)	0(0)	25	8	33

Quite unsurprisingly none of the elicited constructs were of the Expanded-CNF. The only cases of unsupported constructs were encountered when participants were asked to elicit sentences using terms of equal affinity; 9 out of 31 of the elicited constructs in the *ABC* affinity mode, and 8 out of 33 of the elicited constructs in the *abc* affinity mode were of this unsupported kind (e.g. *"Joe is holding a red pen, and a green or red pen"*, *"He is drinking coffee or he is listening to music, and he is reading a newspaper"*). Yet, as the exact binomial test suggests, in each one of the affinity modes the number of unsupported spontaneous constructs is significantly lower than the number of supported constructs ($a<0.05$).

3.7.3 Discussion

Overall the participants tend to prefer the DNF and consider it more natural compared to its CNF equivalent. This tendency is rather apparent when the proposed DNF constructs are shorter than their CNF equivalent constructs. However the picture changes dramatically when the proposed DNF constructs are longer than their CNF equivalents. In these cases the participants seem to find the CNF *"(x or y) and z"* quite natural, exactly when they perceive the terms *x* and *y* as highly akin compared to the term *z*. This finding indicates that the overall tendency for expressing logic in disjunctive normal form may be manipulated by introducing terms that have a contextual affinity. Additionally, it highlights why a purely DNF logical expression editor is not always an intuitive representation.

For example consider the following sentence *"I am usually with friends when it is Saturday or it is Sunday and I am downtown"*. The above sentence is in CNF and the high affinity of the terms *"it is Saturday"* and *"it is Sunday"*, due to their mutual exclusiveness, makes the appeal of CNF so strong that it is quite unlikely for one to think of its DNF equivalent *"I am usually with friends when it is Saturday and I am downtown, or it is Sunday and I am downtown"* as more. Now let's replace the term *"it is Saturday"* with the term *"I am having coffee"*, assuming that for most of the readers there is no inherent affinity between exactly two of the three terms *"I am having coffee"*, *"it is Saturday"*, and *"I am downtown"*. According to our findings the reader should find the sentence *"I am usually with friends when I am having coffee and I am downtown, or It is Saturday and I am downtown"* quite natural and straightforward, whereas the sentence *"I am usually with friends when I am having coffee, or It is Saturday and I am downtown"* should be less clear and understandable.

These last examples and their support from our experiments point out that in a purely DNF based logical expression editor it would be rather difficult for users to express such statements as the aforementioned. However, assuming an identifiable affinity of the involved terms we would be able to allow the appearance of "disjunctive terms" in a DNF construct *"(a and b and …) or (c and d and …) or …"* by allowing the replacement of the terms (*a, b…etc.)* with either disjunctions or conjunctions of terms which share a high affinity level compared to the rest; therefore, allowing a layered DNF construct such as *"((a1 and/or a2) and (b1 and/or b2) and …) or ((c1 and/or c2) and (d1 and/or d2)…)"*. Returning back to the design of the CoRE editor this is exactly what we try to achieve by applying the heuristics for regrouping ranges of terms based on their affinity. Each alternative resembles the outer disjunction of the above DNF construct, while within each alternative a contingency is presented involving terms that are grouped forming the inner conjunctions which in turn are composed as either disjunctions or conjunctions of terms of high affinity.

Overall, the above experiments have added to the existing body of knowledge for Mental Models theory, by documenting and nuancing the role of affinity. It has also validated the core design rationale of the CoRE editor regarding the use of context information in guiding the design of the logical terms presented in the structured editor.

3.8 Application

The notation described in this chapter, and the interactive tools that support editing, inspecting and evaluating of the system's reasoning can be applied in any domain where there is an underlying mechanism that can describe the effective outcome of its components (e.g. services) on the context itself.

With respect to awareness systems, the expression editor described here allows end-users to define, control, and inspect their behaviour. The prototypical editor has been developed to compose logical expressions based on a model of context that is automatically populated using the Amelie framework (chapter 4). As Amelie adopts the tenets of recombinant computing it inherently allows the combination of heterogeneous services and components into awareness applications. Amelie services are designed to expose their contextual-range using the notation described here, hence allowing all the concepts described in this chapter to be transferred directly to the domain of awareness systems. Several prototypical services on this framework, addressing various relevant abstract programming tasks, make use of the editor described here, not only by defining the underlying reasoning and behaviour of the system, but also by allowing users to inspect the insights of the system's status and predict its behaviour in a future potential situation. Below we present briefly some indicative programming tasks that are supported by such prototypical services.

For example, a prototypical service enables users to combine the effective output from other services using logical premises in order to define a new behavior. Hence, one could assert that her activity is watching TV when an electrical current sensor attached to her TV senses that the TV is on and a sensor attached to her sofa senses that someone is sitting on it.

In a different task, the user may need to keep track of events related to her information, allowing her to coordinate her social interactions (see Khan et al., 2009). Having for example instantaneous awareness information is not always enough and people need to extend it with expressions that define when someone was first or last seen somewhere. To support this need, an abstract service allows end-users to extract the traces from specific services when an event occurs such as when certain conditions apply for the first or last time during a specified period of time. The user for example could instruct such a service to keep the trace of her activity the last time that she is at the office, or for example one could instruct such a service to track the date of her last visit to Paris.

A wide range of services is expected to pertain to the extraction or rendering of information, rather than to provide also the means for regulating the user's privacy. Tasks that enable users to expose or inquire information from others are needed and implemented as services that also use the CoRE editor allowing the user to have direct control on their privacy regulation over time. For example, a user who whishes to expose extracted information from one or more services to other entities (e.g. users) can employ a task that allows her to select under which conditions the system should expose the extracted information to certain entities.

In such a case even the group of entities could be defined to dynamically adjust to the context. For example the user could instruct the system to maintain a group of all her contacts marked as friends who are currently working, say a group called "busy-friends". A prototypical service that implements the above statement returns an attribute that contains the list of entities that the user has marked as friends whenever the specified conditions are met (i.e. they expose to her that their activity is working). Utilizing this task the user could use this attribute in some other service, to fine tune her privacy control, defining for example that she wants to expose only to her busy-friends whether she is also working.

Overall, the tools presented in this chapter can facilitate end-user programming, whether the programming tasks are domain specific such as those described above or even general programming tasks. In most of the domains that end-user programming can be useful, we can expect mechanisms that rely on conventions, or linguistic models, which can provide similar ways for defining affinity; thus allowing us to expect that the same intuitive expression editing tools, can be applied to a wide range of application domains.

3.9 Conclusions

We have proposed a notation and an editor for defining, inspecting, modifying and comprehending context aware system behaviours. Both have wide applicability as they are independent of any application domain semantics. We argue for their particular relevance for solving the challenges of intelligibility, accountability, and control by end-users of context aware systems. They are beneficial for several reasons:

- our approach is compositional, allowing the system behaviour to be discussed and disclosed in a uniform fashion on a variety of abstraction levels that may be relevant for different applications, context and user groups.
- we exploit well known cognitive principles and design heuristics relating to the affinity of logical terms to largely simplify logical expressions and make them more directly understandable to non programmers. Where traditional programming and logic formalisms rely on formal-associativity and parentheses to disambiguate related terms, the editor goes a long way in making such expressions clear and understandable by end users directly.

We have shown how the semantic information describing the range of outputs from a very low layer of fundamental services can be translated to more abstract concepts and services through direct manipulation of natural language in the form of a structured editor. Compositionality was enabled by the ability to describe both the premises of services and their effective outcome using a uniform notation that allows designers and end-users to create and control context-aware applications. In this layering process we have shown how each level of system intelligence and reasoning can be unpacked exposing its underlying level of behaviour and internal coherence, while enabling users not only to modify it but also use it in building higher levels of abstraction.

The transformation of context-ranges by the tools presented here, maps the users' cognition of the semantic structure on the visual layout of textual notation. Through both realistic and complex evaluation tasks, we have shown that the editor is effective and intuitive for end-users, resulting in very few logical flaws. The experiments that we have conducted validate the core design rationale of the CoRE editor, while at the same time extend the existing body of knowledge for Mental Models theory, by documenting and nuancing the role of affinity.

While a range of other tools support end-users in defining the system's behaviour, this chapter has introduced for the first time a uniform mechanism that can be applied through a context aware system, that allows the same semantic information to be used to give in natural language answers not just to the question "how can the system's behaviour be altered" but also and more importantly to the questions "how does the system behave", "why is something (not) happening at the moment", and "how would the system behave in response to a change in context".

Chapter 4

Amelie: a recombinant computing framework for ambient awareness

This chapter introduces Amelie, a recombinant computing framework supporting the implementation of awareness systems. Related research into software architectures is reviewed (sections 4.1 and 4.2) motivating a recombinant computing approach. Amelie is then introduced (section 4.3) as a recombinant approach that addresses infrastructure related challenges. We follow showing, in section 4.4, how the concepts of the FN-AAR model are implemented on the Amelie framework. Finally (section 4.5) we go on to explore the framework's impact through several examples and case studies.

4.1 Introduction

In earlier chapters, the importance of supporting awareness through networked technologies and some of the challenges relating to their design were discussed. Supporting awareness has prompted researchers to consider how to integrate the capture of awareness information, its dissemination, and display within heterogeneous collections of devices and services comprising ambient intelligence computing environments.

For example, as a precursor to the research presented in this thesis, the DIARIST system was created and deployed in order to address the well known scenario of a lone elderly relative living independently (Metaxas et al., 2007). The DIARIST system comprised of a collection of sensors and sensor motes that were used to monitor activity within a household drawing inferences such as, whether the elder is at home or not, 'getting up in the night' and sleeping patterns, the presence of visitors, cooking activity, etc. The detailed logs of related events were processed to provide almost real time updates to the connected party, historical overview for the last 24 hours, and a more detailed narrative that would synthesize a succinct but meaningful summary of the activity of the elder as detected by the sensor network.

Existing implementations of awareness systems of this latter kind, which we call *ambient awareness systems,* have so far been of limited scale. Along with DIARIST, other influential ambient awareness systems such as the Casablanca project (Hindus et al., 2001) and CareNet (Consolvo et al., 2004), also present point solutions focused on the implementation of a fixed scenario and so do not provide sufficient flexibility for addressing emerging needs and behaviours during actual deployment and use. In advancing towards realistic deployments and actual use, devices and services need to be used often in configurations and for purposes that are not foreseen by their designers and developers (Khan et al., 2008). Eventually such a dynamic configuration and repurposing of the multitude of devices and applications in an Ambient Intelligence environment requires that they operate collectively, using information and intelligence that is hidden in the interconnection network (Aarts & de Ruyter, 2009). A clear consequence of this position is that interoperability and dynamic aggregations of devices and services are needed, a technical ambition that has been pursued consistently by the ambient intelligence research field in the past ten years.

This chapter describes Amelie, an application development framework designed to meet this challenge a) by dealing with the problem of interoperability and, b) by dealing with social aspects of awareness that are inadequately addressed by generic software architectures for ubiquitous computing and ambient intelligence. Amelie is based on the theoretical foundations of the recombinant computing concept (Edwards et al., 2001) and the FN-AAR model of awareness introduced earlier in chapter 2.

The chapter sections can be mapped on the "4+1" architectural views by Kruchten (1995) who distinguishes between logical view, process view, development view, physical view and use case view. Logical view is portrayed in section 4.3.1 describing the framework's service methods, and sections 4.4.1 to 4.4.3 describing the mapping of the FN-AAR concepts onto Amelie. Process-view and development-view aspects are discussed in sections 4.3.2 (Profile-Processor), 4.3.3 (service registration), and 4.4.5 (Awareness-Manager). Nevertheless, throughout the chapter a multitude of scenarios are used to provide insights to the use-case view of the Amelie framework.

4.2 Software Architectures for Awareness Systems

An early and influential model of awareness systems was the *'event propagation model'* (EPM) (Sohlenkamp et al., 1997). EPM identifies three basic processing steps that comprise the essential functionality of awareness systems: capturing information regarding a particular individual, group, or location, disseminating it, and displaying it to intended receivers. GroupDesk (Fuchs et al., 1995), the initial prototype implementation of the EPM, is an application that allows users to stay informed about events, that happen currently or that have happened in the past in the surroundings of their actual position. Despite that this architectural model was not originally intended for the implementation of ambient intelligence applications, it has been adopted as a reference framework for generic awareness-system infrastructures such as Nessie (Prinz, 1999).

Whereas GroupDesk focused more closely on supporting collaboration, the AROMA project (Pederson & Sokoler, 1997) is another remarkable early attempt, aiming at peripheral awareness of remotely located colleagues; AROMA evolved its own object-oriented architecture to abstract sensors and the information interpretation, reflecting the idea that the system should enable viewing of the same raw data at different levels of abstraction.

The Context Toolkit (Dey et al., 2001) is a development of CyberDesk (Dey, 1998), an earlier framework for self-integrating software, where integration is driven by the user's context, which contained many of the mechanisms that were transferred in the Context Toolkit. The Context Toolkit provides designers with abstractions such as widgets, interpreters, aggregators, services and discoverers as well as a distributed infrastructure (Dey et al., 1999) to help application designers build context-aware services and applications.

Confab (Hong & Landay, 2004) is a prototypical toolkit for facilitating the development of privacy-sensitive ubiquitous computing applications; building on the *Approximate Information Flows model (*Jiang et al., 2002*)*. Confab affords basic support for building ubiquitous computing applications, providing features to allow the easy implementation of a spectrum of trust and privacy levels.

A canonical example of context awareness is a mobile tour guide (Bederson, 1995); a mobile tour-guide is an application that runs on a handheld computer, or other device, and enhances a visitor's experience providing location indexed information or information that is adapted to the user's activity and presumed interests. This scenario has been revisited regularly in ambient intelligence and ubiquitous computing literature (e.g., Abowd et al., 1997; Bederson, 1995; Davies et al., 1998; Feiner et al., 1997; Fels et al., 1998; Dey et al., 2001). These works emphasize different aspects: the information visualization (e.g., Abowd et al., 1997; Bellotti et al., 2002), or the software architecture used (e.g., Dey et al., 2001). In all these cases the core of the user interaction has remained constant, relying on location based information retrieval and auto-discovery of services. It is instructive to look closer at a detailed analysis of a scenario by (Dey et al., 2001) that illustrates location based information retrieval and auto-discovery of services in terms of their context toolkit architecture:

'*A Location Widget is built around the centralized location server. Each Room Aggregator (one for each room) subscribes to the Location Widget to be notified when a user enters and leaves its corresponding room. The Room Aggregators also subscribe to the Exhibit Widgets in their respective rooms so they know which of the exhibits are available at any given time. Each Exhibit Widget contains information about an exhibit on the tour. The application subscribes to the Location Widget for location updates for its user. When it receives an update, it updates its map display and polls the*

Room Aggregator representing the user's current location for a list of the available exhibits, if any. It displays the exhibit information to the user and also subscribes to the Room Aggregator to be notified of any changes to the exhibit list.'

This blow by blow description that typifies the research works cited above in this section, illustrates the transparent and relevant information retrieval that is achieved thanks to context awareness. A publish-subscribe model is followed, where sensing components publish information and functional core artefacts also; an interactive application obtains notifications of sensed information maps, polls related information sources and presents it to users who enjoy the benefits of automatically matching information sources to context.

The 'success' of this scenario where users are apparently facilitated in their activity depends upon a crucial interaction step that is brushed over in these descriptions. In real life, museum and exhibition visitors indicate and commit their need to find out information about exhibits, the moment they get hold of the device whereby it can rather safely be assumed for what purpose they do so. In the more general case, intentions need to be inferred from sensed information, and this is where adaptation can go very wrong. As Edwards et al. (2001) argue, it is humans who give semantic meaning to devices, so we note here how this implicit step of committing to a particular purpose and use assigns semantics to the system. This may appear exaggerated at fist sight, but let us examine how a slight modification of this scenario, where picking up an available single-purpose device is not involved:

'When hotel visitors feel lonely they are shown on their TV whether anybody else in the hotel is in the same mood[11]*'.*

Taking a user centred perspective we focus on the interaction tasks required by users and the social implications of their actions. The function described can be expanded in a number of ways, depending on what level of system initiative is supported.

1. John turns on his computer gaining access to the internet automatically. On starting his browser he is presented with the hotel welcome page where he is prompted to join the *HotelClub* community. The page prompts John to add *HotelClub* in his contact list, expose his mood to *HotelClub*, and acquire from the *HotelClub* the 'people that feel lonely'. The above explicit steps are an extra overhead and an unfamiliar task for John (a secondary task that is a by-product of having to operate the technology rather than an essential step to meet John's goal). Worse, they are totally irrelevant at the moment he checks his e-mail.

2. When John enters his room, and establishes his internet connection, the system exposes his computer's IP address to an auto-discovery server, which establishes a connection with the hotel community matching his computer's IP addresses with

[11] In correspondence with the museum example, we could say they move through their 'emotional space'

the router's IP address. The *HotelClub* community inquires about his mood; transparently to John his computer provides this information. Twenty minutes later, while he is chatting online with his girlfriend he mentions that he is lonely; he then notices that the TV suddenly shows a banner: *'John why don't you go for a drink at the lobby with Anna, Tom, and Linda? They are also lonely!'.* Bemused, John wonders how to disable this automated service or whether he should just disconnect his computer physically from the internet.

3. John enters his room and starts browsing on the internet, and gazing at the TV. He notices the hotel's webpage printed on a small card on his desk. He navigates to the home page of the hotel where he finds out that the hotel invites him to join the *HotelClub* community. A banner prompts that if he feels lonely, and likes going for a chat with someone, he can get informed about who else of the visitors also feels lonely. Feeling safe that nothing will be done without his authorization he clicks on the banner and a familiar page from his familiar awareness service provider appears. The authorization page, explains to him that by granting authorization to the *HotelClub*, an application will be activated: by turning on a switch on his desk - informing the *HotelClub* that he would rather have a chat at the lobby- the TV will be displaying on a flying banner whether anyone else is in the same mood.

Let us examine the different degrees of initiative illustrated in these variations of the scenario. In the first case the hotel's Wi-Fi router automatically discovers John's computer and accepts the connection, allowing him to check his e-mail. It is John who initiates the automatic discovery of the internet connection (by turning on his notebook), just like eventually it is John that will initiate the interaction with the *HotelClub* community at a moment appropriate for him. The same implicit interaction that in the first scenario connects John to the Internet and is experienced as beneficial regarding John's goal to check his e-mail proves catastrophic (presumably) in the second scenario.

As A. Schmidt (2000) puts it -'*Implicit human computer interaction is an action, performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input*'-, implicit interaction is an apparent pattern that appears throughout the domain of information technology. The above examples indicate that such patterns have radically different impact on the user experience. Interaction design is critically concerned with anticipating and managing such implications. We argue that context-aware toolkits and application frameworks should support developers, designers, and end-users in the same direction.

In the last scenario, able to foresee the implications of a visitor's implicit interaction, the designer decides to opt for explicit subscription to the application rather than auto discovery. The end-user on the other hand is supported in developing a socially intelligent profile. The system not only detects the existence of a symmetry constraint(or even proposes it), that protects him from exposure to fraud, but can even detect whether at any time the *HotelClub* is not fair to him (e.g., a dishonest community could hide from him that there are people expressing to the *HotelClub* that they are feeling lonely he is not informed about it, and actually use his disclosed information for some other purposes).

The above remarks and similar observations (e.g., Kindberg & Fox, 2002) indicate that context-aware applications are dynamic evolutions rather than static configurations of services and functions, and that their dynamics should be seen as the results of both implicit and explicit interaction with the user. Consequently, supportive frameworks for the development of context-aware applications should allow the design of the interactive mechanisms through which end-users can control, direct, and advance the lifecycle of such systems, rather than channel the interaction design through their architectural constraints.

In the tour-guide application and in ambient awareness applications (e.g., Metaxas et al., 2007) the role of the end-user is typically foreseen as a passive one. Users' implicit interaction with their environment directly affects their experience. However, their control over the consequences of such implicit interactions is typically limited to the level of information acquisition and visualization. Belloti and Edwards (2001) discussed the interaction design challenges context aware systems present. They argued that such systems should not simply act on behalf of users but, rather, should involve users in action outcomes, allowing them to understand, explore, and even define the underlying mechanism governing the behaviour of such systems. According to Belloti and Edwards, two key features for such systems are *intelligibility* and *accountability*.

Whereas intelligibility and accountability may not be crucial for the case of a mobile tour guide, in the more social context of the hotel community they are crucial. It is only recently that the focus of the research community has shifted in this direction, see (Dey & Newberger, 2009), yet existing infrastructures are not able so far to support efficiently these requirements. State of the art approaches manage to address intelligibility and accountability at the level of designers and software developers rather than at the level of actual end-users.

While in the previous chapter, we presented tools and mechanisms that support these requirements at the level of the user interface, the Amelie framework discussed below aims at addressing these requirements at an architectural level.

An influential concept for the development of the Amelie framework is the *recombinant computing* approach. Edwards et al. (2001) pointed out that ubiquitous computing research has considered enabling technologies in isolation, e.g., location sensing, multi-device user interfaces, ad hoc network protocols, and so on, overlooking fundamental software architectural issues relating to their composition. They outline an approach that they call *"recombinant computing"* which allows the dynamic extension of computational entities through the use of mobile code. As they point out existing component frameworks (e.g., JavaBeans, DCOM, UPnP) are insufficient to enable arbitrary software interconnection because the users of such frameworks (i.e., application developers) are required not only to have knowledge of the interconnected components' interfaces but also of their extensive semantics.

The recombinant computing approach proposes a limited set of *recombinant interfaces* that provide the foundation for allowing components to interact with one another dynamically. For that to succeed it is the user who provides the semantic understanding of what components actually do. The initial prototypical architectural framework supporting recombinant computing, Speakeasy (Edwards et al., 2002), outlays three general functional requirements namely connection, context and control: connection refers to the mechanisms that allow components to exchange information with each other, context refers to how components reveal information about themselves, and control refers to how components allow users and other components to effect changes.

4.3 Amelie as recombinant framework

Amelie uses a single interface with a few methods to support notions such as feature-extraction, automatic-discovery, information dissemination, or aggregation. More specifically, Amelie services are implemented as SOAP (*Simple Object Access Protocol*) services over HTTP. SOAP is a widely adapted protocol specification that is supported by all modern programming languages and application development kits. Moreover its close binding to XML makes it inherently suitable for supporting a generic and unrestricted representation of context[12]. Amelie services support connection, context and control by implementing four methods, namely *'execute'*, *'getrange'*, *'describe'*, and *'register'*, each presented in turn below.

4.3.1 Amelie service methods

For any Amelie service, the *'execute'* method encapsulates its functionality, regardless of whether this service extracts, disseminates, or consumes information. *'Execute'* can be invoked with any parameters and may return any content that the service desires. The interpretation of the invocation parameters is specific to each service, and a single service's output may be deterministic if a service always returns the same result any time it is called with specific parameters or, otherwise, nondeterministic if a service may return different results each time it is called with specific parameters.

Each Amelie service implements the *'getrange'* method to allow others to examine what its outcome could be and whether its outcome can be used by another service. For that, *'getrange'* can be invoked with any parameters and should return the range of effective output that an invocation of execute with the same parameters would have. The obvious choice to use standard web notations such as XSD to describe the effective outcome range of a service was abandoned in favour of a simpler notation that can readily be translated to natural language and allows the evolution of efficient end-user tools that support intelligibility, accountability and control.

[12] The choice for SOAP was to a big extend a pragmatic decision emerging from the iterative design process of Amelie; however, as Amelie turns out to employ fundamentally a single interface one could certainly opt for a much more lightweight communication protocol.

Our proposed notation, along with its impact on the end-user side was described in detail in the previous chapter (chapter 3).

Despite that there is no way to assess a priori whether the result of a service's '*execute*' method falls within the range described by '*getrange*', it is quite straightforward to falsify such a mal-behaving service at runtime. Amelie as a recombinant approach inherently discourages services to use over-generalization and requires programmers to specify the effective outcome of each service as precisely as possible since otherwise it becomes practically impossible both for end-users and other services to make use of it.

While the *'getrange'* method is sufficient for others to query a service's possible outcome, it is insufficient for explaining how and why this outcome is produced. Given the aforementioned requirements of intelligibility and accountability the latter is crucial for users to understand the impact of interactions with their environment. Thus, the *'describe'* method can be invoked with any parameters expected to return any content that the service considers appropriate to describe its behaviour given the specified parameters, or a content that both describes and provides interactive mechanisms for modifying the service's behaviour. Using the HTTP-header information the service may adjust its content to the client that triggers the method. If, for example, the client accepts only plain text, the service may return a pure text content describing its behaviour, while if the client is a web-browser and accepts HTML, the service could return an HTML formatted description of its behaviour, which provides a link to modify it, or even provide inline tools that allow its modification.

The parameters passed to the above methods can be seen as instances of Amelie services that may reside in any context, such as a service configuration, an application, or any client that invokes an Amelie service. The *'register'* method is the notification mechanism that allows services to negotiate such instantiations. A client can invoke the *'register'* method of a target Amelie service notifying it that an instance of it is being registered in some context for later use. The service can respond whether it accepts or declines the operation, or even whether it has an alternative proposal of its own regarding its parameters

Let us consider a trivial service example, named *'random-echo'*. As the name suggests, when the *'execute'* method of this service is invoked with some parameters, the service returns randomly one of the invocation parameters. When the *'getrange'* method is invoked, the service is supposed to return its range of output given that *'execute'* would be invoked with the same parameters; apparently, in this case the '*execute'* method is deterministic when either a single parameter or no parameters at all are passed to it, whereas it is nondeterministic when at least two parameters are passed to it; hence *'getrange(∅)'* yields *'∅'* , *'getrange(X)'* yields *'X'* while *'getrange(X,Y)'* yields *'one(X,Y)'*. The *'describe'* method could return a string depending on its parameters such as *'this service does nothing'*, *'this service echoes X'*, or *'this service returns randomly either X or Y'* respectively for the parameters *'∅'*, *'X'*, and *'X, Y'*. Finally the *'register'* method could simply return *'accept'* denoting that it accepts being registered in some context regardless of its parameters.

4.3.2 Amelie profile-processor service

Be it an actor, an agent, a community, or an application, Amelie allows a set of services to interoperate and contribute in a common resultant component configuration forming an *Amelie profile*. The structure and semantics of an Amelie profile allows other services to interconnect and cross-reference each other in order to compile some behaviour. The Amelie framework provides the necessary component infrastructure, which we term *Profile-Processor,* itself an Amelie service, which handles the semantics of any profile.

Let us walk through the features of an Amelie profile, using the following simple profile that uses only instances of the trivial service *'random-echo'*. The profile portrayed in the XML snippet below declares three such service instances, each of which defines its persistent parameters that will be passed to the specified service URL when any of the Amelie service methods is invoked.

```
<aml:profile xmlns:aml="…" xmlns:inv="…">
   <inv:service id="e1" url="http://random.echo.service">
      <foo/>
   </inv:service>
   <inv:service id="e2" url="http://random.echo.service">
      <inv:service ref="e1"/>
      <bar/>
   </inv:service>
   <inv:service id="e3" url="http://random.echo.service">
      <baz><inv:service ref="*"/></baz>
   </inv:service>
</aml:profile>
```

The first *'inv: service'* element declares *'e1'*, an instance of the *random-echo* service that should be invoked with the XML element *foo* as parameter. The second element declares *'e2'*, yet another instance of the *random-echo* service that should however be invoked with parameters the XML node-set that consists of the output of *'e1'* and the element *'bar'*. The last element declares *'e3'* also as an instance of the *random-echo* service; this element however denotes that its parameter should be the element *'baz'* containing as children the outputs of all the other services in the profile.

One possible result of invoking the *execute* method of the profile-processor with the above profile as input could be the node-set *{foo, bar, baz {foo}}*. On the other hand invoking the *getrange* method would return a range corresponding to the possible outputs of execute (i.e. *{foo, one (foo, bar), baz {foo, one (foo, bar)}})*.

In more generic terms, when the *execute* method of the Profile-Processor is invoked with a profile passed as a parameter to it, the Profile-Processor needs to iteratively replace each of the service instances with the result of their invocation after resolving their parameters. The latter, in turn, may also be references to or instances of other services. In this process, service instances that depend on their own output (i.e. that contain circular references or contain other instances with circular references) are replaced by their last known (cached) output[13] when they appear as parameters. At the end of this process, and having all service instances invoked, the cache is updated using their new output.

In the following figure(figure 1), for example, service instances *'A'*, *'B'*, and *'C'* are engaged in a circular reference loop because the first refers to the second which refers to the third, which in turn refers back to the first. All these service instances are replaced by their last known output when they appear as parameters of other instances. Consequently, *'A'* is invoked using the cached output of *'B'*, *'B'* using the cached output of *'C'*, which in turn is invoked using the cached output of *'A'*. For the same reason, the service-instance *'D'* is invoked using as parameters on one hand the cached output of *'A'* and on the other the result of the invocation of service instance *'E'*, since the first (*'A'*) is involved in a circular reference loop while the latter (*'E'*) is not.

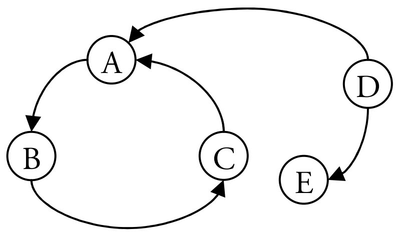


Figure 1. An example service-instance configuration that contains a circular reference loop.

When the '*getrange*' method of the *Profile-Processor* is invoked, the same procedure as above is followed, but '*getrange'* is invoked instead of '*execute'* while processing the *'inv:service'* elements. Hence the result of '*getrange'* returns the effective output range of the Profile-Processor for a specific profile.

When the *'describe'* method of the *Profile-Processor* service is invoked the service generates a graphical interface, following the same iterative procedure. Depending on each of the services, the graphical interface allows the user to understand, explore, and modify the profile. In figure 2 a snapshot from the description of the author's profile is displayed; the rendition is the result of invoking the Profile-Processor's *'describe'* method passing to it the author's profile as input, which iteratively generates the corresponding descriptive HTML content. On the left pane, the profile-processor generates an interface that allows end-users to view and modify their profile, while on the right pane each of the profile's registered services generates its description based on its specific parameters.

Overall, the *Profile-Processor* is an Amelie service that allows, through its semantic simplicity, the composition of context-aware applications as a set of services forming component architecture diagrams. Importantly for ambient intelligence applications, this enables the evolution of the composite service to match developments of its component services, using dynamic composition and loose binding.

---

[13] This pragmatic choice, avoids infinite recursions, while allowing service instances to contain cross references within a profile and evaluate their output in a declarative fashion.

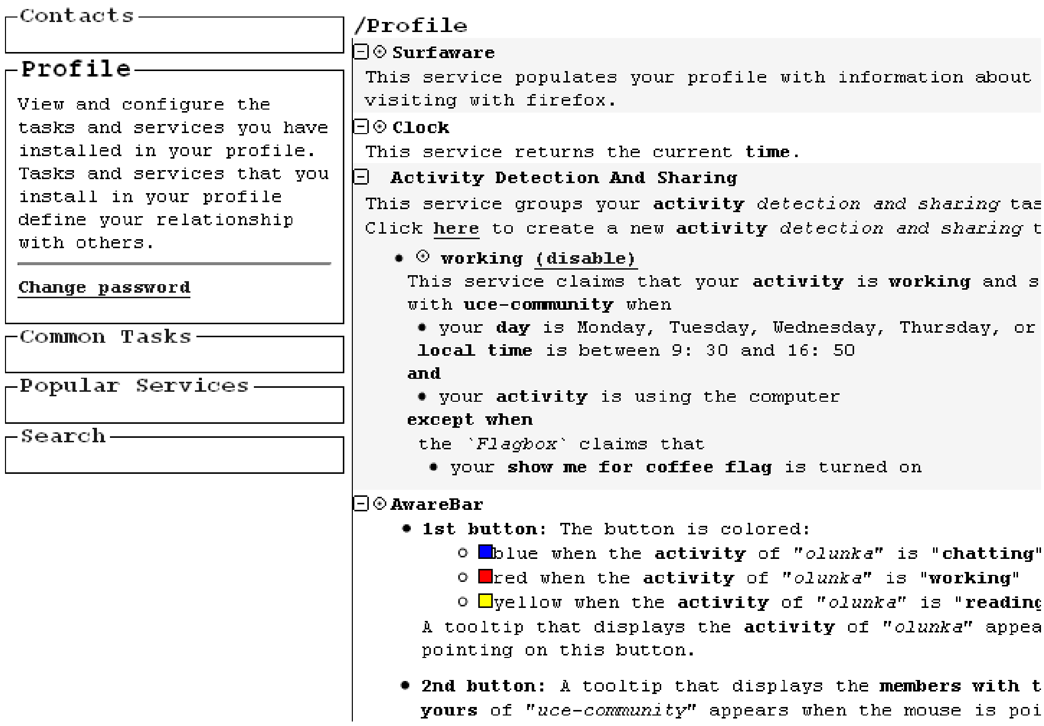


Figure 2. A snapshot of the author's Amelie profile rendition by the *Profile-Processor*'s *'describe'* method.

4.3.3 Registration, mobility and application sharing

In the previous section we have shown how Amelie services provide the necessary set of methods that, on one hand, allow the formation of higher levels of abstraction to evolve within a compositional approach, and, on the other hand, to expose their reasoning and internal functionality to end-users. Here we investigate the mechanisms that enable designers and end-users of context-aware applications to separate implicit and explicit interaction in application deployment, addressing the problems we discussed in the introduction of this chapter. Moreover, we examine how the same mechanism is eventually sufficient to support mobility of code and application sharing.

A designer may deploy her context-aware application through a client (be it a web, pervasive, mobile, or desktop application) that guides the registration of a service-bundle that accounts for her context-aware application. In the example scenario that was used in the introduction, such a client would have been a web application that allows users to register within the hotel club. This web application would guide the registration of a set of services in the user's profile, which carry the task of keeping the user informed about who else of the hotel visitors is in the mood to have a drink at the hotel lobby and the task of exposing whether the user herself is in such a mood. As far as the designer is concerned, her context-aware application is a set of interconnected services that is deployed to end-users through a client of her choice (a web application in this example). For a client to register a bundle of services in a profile, it needs to prepare the XML code describing the instances of the services as they will appear in the target profile. Then it has to post this XML data to a registration service that takes care of the rest of the procedure.

Our prototypical registration service analyzes the service-bundle data by querying the bundle's effective outcome, its relationship with other services, and its required credentials, in order to inform the user about the consequences of her actions. The user then has the option to either decline or grant access to the service bundle. For example, figure 3 below shows a screenshot from the registration service notifying the user that such a service bundle is requesting registration in a user‘s profile.

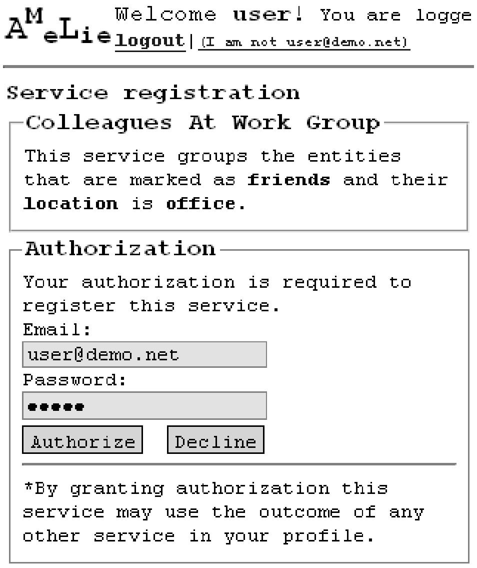


Figure 3. The user is notified that a service bundle is requesting registration in her profile

In order to grant access to the service-bundle registration, authentication is needed from the user, yielding the user's target profile and allowing the registration service to register the service-bundle in the resolved profile. Eventually, if this user grants access, all services of the bundle are notified about the outcome of the operation (by invoking their 'register' method). Finally, regardless of whether the user granted or declined the service registration, the client who initiated the interaction is also notified about the result through a client specified callback-URL.

During registration a service-bundle may include a list of credentials that a service requires in order to function properly. For example a service that generates a textual description of a user's profile could require read-only access to the profile, while a service such as an alternative graphical user interface or an automatic-discovery service that needs to be able to freely modify the user's profile would require to have full access on it. Consider for example the following service bundle that is posted by a client to the registration-service, initiating the registration procedure:

```
<package callback=”any.client.defined.url”>
   <inv:service id=”foo” url=”…”>
      <inv:credentials>
         <read-profile/>
         <write-profile/>
      </inv:credentials>
   </inv:service>
   <inv:service id=”bar” url=”…”>
      <inv:service ref=”mailbox”/>
   </inv:service>
</package>
```

Once the client application requests to register the above service bundle and before proceeding, the registrar service invokes the register methods of *foo* and *bar* to establish whether they accept such an instantiation. Then it processes the bundle and besides querying *foo* and *bar* for their effective outcome, it warns the user that by granting authorization to the client, the first service (*foo*) will gain full access to her profile, while the second (*bar*) will be able to receive as input during invocation the output from the *mailbox* service (if any) of the target profile. The user has a clear view of the consequences of registering the service bundle and can decide whether she will grant or decline access to it.

If the user declines access, the registrar service invokes the client-specified callback-URL, passing to it the result of the operation. For example, in the above situation the registrar service could notify the client application about the rejection using the following snippet, which denotes that the registration was declined because the user declined write access to her profile, while she accepted the second service.

```
<declined>
   <package callback=”any.client.defined.url”>
      <inv:service id=”foo” url=”…”>
         <inv:credentials>
            <read-profile/>
            <declined><write-profile/></declined>
         </inv:credentials>
      </inv:service>
      <accepted>
         <inv:service id=”bar” url=”…”>
            <inv:service ref=”mailbox”/>
         </inv:service>
      </accepted>
   </package>
</declined>
```

Additionally, the registrar may propose to the client an alternative package that could be the result of the interactive-user's manipulation of the client's original service-bundle:

```
<proposed>
   <package callback=”a url of the client app decision”>
      <inv:service id=”foo” url=”…”>
         <inv:credentials>
            <read-profile/>
            <write-instance id=”foo”/>
         </inv:credentials>
      </inv:service>
      <inv:service id=”bar” url=”…”>
         <inv:service ref=”mailbox”/>
      </inv:service>
   </package>
</proposed>
```

In the above pseudo-XML snippet the registrar counter-proposes to the client an alternative bundle hopping that the interactive user would accept registering in her profile. In the proposed alternative the instance of the first service has write-access only to the *'foo'* service instance itself and not to the whole profile. In this iterative process, the client now can choose whether the end-user's modifications are a sufficient alternative or it can propose for registration a new service bundle.

Eventually let's imagine that this negotiation is fruitful. The registrar service creates the URL monikers that resolve the requested profile access credentials (if any), registers the bundle in the user's profile and after notifying *foo* and *bar* about the registration result, it notifies the client-provided callback-URL about the outcome:

```
<accepted>
   <package callback="any.client.defined.url">
      <inv:service id="foo" url="…">
         <inv:credentials>
            <read-profile>
             http://url.moniker.to.get/the/profile
            </read-profile>
            <write-instance id="foo">
             http://url.moniker.to.set/foo/paramss
            </write-instance id="foo">
         </inv:credentials>
      </inv:service>
      …
   </package>
</accepted>
```

As one can notice from the above XML snippets, an application in terms of Amelie is literally a service bundle that is no different than any Amelie profile. This inherent feature of the Amelie framework allows, apart from programmers and designers, end-users to create and share applications using the same tools and interaction mechanisms that allow them to create any Amelie profile. Interestingly, the registration procedure of a service-bundle and its negotiation mechanism allows potentially end-users not only to accept or decline an application as it is proposed, but furthermore, to tailor it to specific needs even prior to installation, exactly because the XML semantics used to describe a service-bundle are identical with those used for defining an Amelie profile. On the other hand, application deployment can be initiated or even completed automatically, by an already installed service (e.g., an autodiscovery agent) with sufficient credentials to modify the target profile, rendering a priori dedicated architectural mechanisms for automatic discovery and registration a redundant requirement in terms of Amelie.

Reflecting on Speakeasy (Edwards et al., 2002) the original prototypical recombinant computing framework, Amelie follows quite a different approach as to what essentially is context-aware application code, and how can such code become "mobile". The design of Amelie realizes that it is not realistic to promote mobility of code, when requiring services and applications to be written in a common general programming language such as Java. Instead, within Amelie, services need to be reflected only at the semantic layer that refers to their dependence and interconnection to other services. In this respect, the behaviour of a context-aware application is guided by this exact semantic-level and can be sufficiently described using XML in terms of Amelie-profiles and service-bundles. Whether a client initiating service registration, or services themselves require additional components and how such components should be implemented is irrelevant and transparent to Amelie. The framework provides the interaction points (during service registration, or even within the *'describe'* method) where a service could initiate the deployment of its specific components and allows service designers to use any development toolkit for their implementations.

In this section Amelie has been sketched out as a generic framework for supporting recombinant computing and its prototypical implementation has been presented as a supportive infrastructure. This is sufficient and not constraining for implementing any context aware system, while it overcomes some of the limitations of earlier models, such as excessive number and complexity of software interfaces, domain specificity, and architectural assumptions that constrain end-user control and propagate inflexibility. In the following section Amelie is positioned in the domain of ambient awareness systems by integrating the notions of the FN-AAR model (discussed in chapter 2). We will then continue to illustrate how relevant social patterns of communication can be addressed at an architectural level using several examples and case studies.

4.4 Recombinant implementation of the FN-AAR

The Amelie framework as a recombinant computing approach, embraces the domain of ubiquitous computing and context-aware systems. Services that extract, aggregate, and visualize contextual information, can be implemented on Amelie, having the ability to interoperate and to be combined in compositional structures, thus allowing aggregations created to be treated in an identical manner as constituents of higher level composition structures.

What, however, sets an awareness system apart from other context-aware or ubiquitous computing applications, is neither its sensing, nor its dissemination and information visualization capabilities. Rather, it is embedding of the system in social interactions between individuals and groups. The technical infrastructure challenge is assessed by the extent to which the system does not hinder or even supports social norms, socially intelligent behaviours, and gives salience and control over the social aspects of system behaviour. Following the discussions of chapter 2, these could pertain to the system accounting for and supporting socially salient implications of information disclosure among individuals, communities, and artificial agents. In terms of Amelie, it is the ability and requirement to express and manifest socially intelligent behaviour that differentiates entities from other services, rather than an a priori evolution of component framework architecture that makes explicit such a classification.

In chapter 2 we have introduced FN-AAR, a model of awareness serving both as a conceptual tool to support the coherent expression of socially salient behaviours pertaining to mediated communication and as an analytical tool to help reason about the objective of awareness.

The FN-AAR model is populated by the notions of *entities*, *aspects, attributes, resources and observable items.* Entities are representations of actors, communities, and agents. Aspects are any characteristics that refer to an entity's state. Attributes are place-holders for the information exchanged between *Entities* by binding *aspects* with values, allowing an entity to expose its state to other entities using attributes.

Resources are bindings of *aspects* with ways of rendering (displaying) one or more relevant attributes. In any situation an entity might employ one or more resources to express its focus on certain aspects of other entities. Finally, observable items are the result of displaying some attributes about an aspect using a *resource.* Roughly speaking, an *observable item* contains the answer to the question *"How are these attributes displayed to you?".*

To reflect the dynamic nature of awareness, the FN-AAR model populates one's *nimbus* with *attribute-providers*; i.e. functions that return those attributes that one makes available to other entities in a specific situation. In a similar fashion, *focus* is populated with *resource-providers*; i.e. functions that return one's *resources* that display information about other entities in a specific situation. The negotiation of the reciprocal *foci* and *nimbi* of two entities in a given situation (i.e. the corresponding 'produced' *attributes* and *resources*) is a function which returns the *observable-items* that are displayed to the two entities about each other's states, effectively characterizing their respective awareness of each other.

Mapping the aforementioned notions to executable semantics following and benefiting from the recombinant computing approach discussed above, we aim to let designers and end-users support quite nuanced social aspects of mediated interaction in terms of simple Amelie services.

Attribute and Resource providers (i.e. the functions that return one's focus and nimbus in a situation) are abstracted with the same single recombinant interface and implemented as standard Amelie services. An entity's profile comprises a set of service instances that interact with one another, effectively characterizing an entity's focus-on and nimbus-to other entities. Rendering functions are also abstracted as Amelie services that by implementing the same simple interface can consume and visualize acquired information. One may conclude whether a service behaves as an attribute provider, a resource provider, a renderer, or any of the above by queering it for its effective outcome using the service method '*getrange'*.

Most importantly the focus/nimbus negotiation function, that defines the reciprocal awareness among entities, is also abstracted as an Amelie service, allowing entities to communicate and interoperate by exchanging attributes or resources which are the actual carriers of information within an awareness system. In contrast to and apart from the fundamental Profile-Processor service which allows services within a profile to freely form component compositions in arbitrary configurations, the focus/nimbus composition provides the means for expressing and implementing social norms and allows entities to manifest socially intelligent behaviours.

4.4.1 Mapping of the FN-AAR elements on Amelie

The FN-AAR model introduces the notion of attributes, resources, and observable items; the first are the elements of information exchanged between entities hence defining one's nimbus in some situation, resources are elements of information comprising one's focus on other entities, and observable items are elements describing how the information is rendered on some medium. Below we will describe how these notions are mapped onto Amelie.

Mapping attributes

In terms of the FN-AAR model, an attribute is a binding of an aspect to a value. Amelie represents the model's notion of attributes through a simple XML schema which we call Awareness Mark-up Language (AML). Below we can see an example of a simple attribute denoting that someone's activity is walking for the last hour:

```
<aml:attribute>
   <aml:aspect>activity</aml:aspect>
   <aml:value>walking for 1 hour and 25 minutes</aml:value>
</aml:attribute>
```

By default, the value of an attribute is considered as simple text. This allows heterogeneous services to present attributes in a human-readable way. Attribute values may also be defined as structured types. Below we see the same attribute value as above extended with richer semantics of a custom type.

```
<aml:value type="aml-state-duration">
   <state duration="1h25m" state="walking">
    walking for 1 hour and 25 minutes
   </state>
</aml:value>
```

The above declaration is richer semantically for services that are able to handle the "*aml-state-duration*" type. E.g., a graphical rendering or even some ambient information display can interpret the detailed semantics to present the duration. On the other hand, a service that does not support the introduced value-type can still use the text part of the value only; this eliminates type-errors in the information propagation for either services and addresses to an extent, alongside with the ability of services to describe their range of effective outcome, *"the tyranny of types"*, one of the problems that recombinant computing is aimed at tackling (Edwards et al., 2001).

For an entity to express its nimbus (i.e. the attributes that it exposes to others), attributes are adorned with a list of entities that they are exposed to. For example the attribute declaration below instructs the system to expose to *"John"* and *"Anna"* that this entity's location is downtown[14].

```
<aml:attribute>
   <aml:aspect>location</aml:aspect>
   <aml:value>downtown</aml:value>
   <aml:access>
      <aml:entity>John</aml:entity>
      <aml:entity>Anna</aml:entity>
   </aml:access>
</aml:attribute>
```

[14] We explain how access to an attribute is enforced in a later section (4.4.5), where we present the implementation of focus-nimbus composition on Amelie.

Mapping resources

Resources as described in the FN-AAR model are bindings of aspects to ways of rendering information. Similarly to attributes, resources are also described as AML elements; below a simple resource is declaring one's focus on *John's* activity.

```
<aml:resource>
   <aml:entity>John</aml:entity>
   <aml:aspect>activity</aml:aspect>
   <aml:renderer target="http://home-server/picture-frame"/>
</aml:resource>
```

The above declaration instructs the system that we are acquiring John's activity, and given that he is exposing to us any information (i.e. attributes) concerning this aspect, we would like to display it using the specified renderer (in this case a "picture-frame"). In order to allow richer interaction with the rendering services the *aml:renderer* tag allows also the declaration of render-specific parameters. In the example below the same resource is also populated with renderer specific parameters that instruct the picture-frame to colour code different activities.

```
<aml:resource>
   <aml:entity>John</aml:entity>
   <aml:aspect>activity</aml:aspect>
   <aml:renderer target="http://home-server/picture-frame">
     <color for="walking">blue</color>
     <color for="driving">green</color>
   </aml:renderer>
</aml:resource>
```

Mapping observable items

Observable items as described in the FN-AAR model are the effect of displaying some attributes related to an entity about an aspect using a resource. Taking this in consideration, an observable item in terms of Amelie is a binding of any content to the attribute(s) it visualizes. Below we see such a possible rendering:

```
<aml:observable-item content-type="image/jpeg"
   href="url.pointing.to/the_actual_redition.jpg">
   <aml:original entity="John">
      <aml:attribute>
         <aml:aspect>activity</aml:aspect>
         <aml:value type="aml-state-duration">
            <state duration="1h25m" state="walking">
            walking for 1 hour and 25 minutes
            </state>
         </aml:value>
      </aml:attribute>
   </aml:original>
   <aml:effective entity="John">
      <aml:attribute>
         <aml:aspect>activity</aml:aspect>
         <aml:value type="aml-state-duration">
            <state state="walking"/>walking</state>
         </aml:value>
      </aml:attribute>
   </aml: effective>
</aml:observable-item>
```

This last AML snippet declares that the original attribute about John's activity was rendered as a JPEG image; the effective result, however, indicates that the image rendition displays only the state of John's activity (i.e. that John is walking) and not its duration (i.e. that John is walking for the last hour and 25 minutes).

4.4.2 Enabling seamful design

A requirement, pointed out in previous chapters, for ambient awareness systems pertains to what has been termed seamful design (Chalmers & Galani, 2004). In most cases the 'seams' by which technological components are aggregated do not interest users and, therefore, should be transparent; however, the complex nature of Ambient Intelligence environments and the uncertainty that is bound to be associated with sensing and networking infrastructures, mean that it is often important to allow users to inspect and understand the nature of these seams and even to exploit them in the design of such systems. Metaxas et al. (2007) argued how letting users inspect the basis of the inferences made from sensed data can let them better deal with erroneous information and can be essential for their use; in their study users where positive towards the systems ability to expand its logic beyond conclusions such as *'your father had a somewhat calm sleep'* with statements such as *'2 short interruptions where detected by the system'*.

The *'describe'* method of any Amelie service addresses exactly this purpose on a generic scope, by revealing their overall internal behaviour in relation to their instantiation. However as services within an entity's profile expose their seams to end-users, the same may be desirable when referring to extracted attributes, or when entities are involved in the focus/nimbus negotiation procedure. To do so, AML attributes contain an optional confidence index and a section that contains information about the attribute's seams, both of which are populated by the underlying services that generate the attributes. In the example below an attribute exposed to *Anna* is adorned with seams that carry out the specific reasoning of the service that generated the attribute in natural language:

```
<aml:attribute confidence="0.9">
   <aml:aspect>activity</aml:aspect>
      <aml:value>walking</aml:value>
      <aml:seams content-type="text/plain">
         the activity was detected as walking because the
         accelerometer is detecting that there is frequent
         change of the accelaration vector.
      </aml:seams>
</aml:attribute>
```

Given that Anna is acquiring the above attribute, a renderer in her profile could not only illustrate that the activity of her contact is walking, but even further could also detail the seams of the system behind this information.

4.4.3 Supporting symmetrical constraints

The aforementioned semantics of attributes and resources are sufficient to define one's focus and nimbus at some point in time with regard to others. At first sight, the services that generate such information offer adequate protection for one's privacy by choosing when, what, and to whom information should be exposed-to and when, what, and how information should be acquired-from other entities. Although this approach has been traditionally followed by relevant application-frameworks (e.g., Hong & Landay, 2004), it is not sufficient (as we discussed in chapter 2) to address privacy control when this is considered at a social level rather than as an issue of security or access control.

Consider for example the following scenario:

"*John, an office worker, seeks some distraction and a break from work. He would like his colleague Anna to know it, in case she wishes to join him. He hesitates however, as he does not want to be perceived as lazy. He would like to know Anna's mood, but would prefer his interest not to be known or to be public, so that he will not be perceived as prying.*"

This scenario describes guardedness to disclose what characterizes many social interactions (e.g., dating, confiding) and even business transactions. Reciprocity is paramount and there is high social and emotional cost at revealing intentions to others, especially in cases where these are unmatched. The Amelie framework supports the application of constraints that are applied prior to the exchange of information among entities, and within the boundaries of the focus-nimbus composition, in order to support this kind of requirement regarding disclosure. In the scenario above a constraint could require the system to check whether Anna and John are sharing their feelings, before actually exposing any related information to each other.

To apply such equity constraints, attributes and resources may be adorned with relevant semantics. For example, the attribute declaration below, instructs the system to expose John's mood to Anna given certain symmetry constraints regarding Anna's nimbus to John are met.

```
<aml:attribute>
   <aml:aspect>mood</aml:aspect>
   <aml:value>bored</aml:value>
   <aml:access><aml:entity>Anna</aml:entity></aml:access>
   <aml:contstraints>
      <ctx:symmetry xmlns:ctx="aml:focus-symmetry">
         <aml:attribute>
            <aml:aspect>mood</aml:aspect>
            <range:any>
               <aml:value>bored</aml:value>
               <aml:value>tired</aml:value>
            </range:any>
         </aml:attribute>
      </ctx:symmetry>
   </aml:contstraints>
</aml:attribute>
```

The above declaration uses a contextual symmetry constraint that defines an affirmative symmetrical constraint regarding sharing John's mood with Anna. The *'aml:contstraints'* tag provides the necessary executable semantics for the interpretation of the constraints, instructing the system to enforce that the depicted attribute should not be exposed to Anna unless she is also exposing to John that she feels bored or tired. The constraints are described using the same notation as when describing the possible outcome of a service to define the context that needs to be matched. Similarly, we could define a negative symmetrical constraint regarding the exposure of John's desire for a break to Anna, in order to reassure that John's desire for a break will only be exposed to Anna if she is not busy.

Like attributes, resources may be adorned with symmetrical constraints. We could also, for example, adorn John's resources focusing on Anna, to inquire about her mood, only if she is focusing on his location.

Such level of symmetrical constraints both on the side of the observed and on the side of the observer can support the definition of nuanced participation structures addressing privacy requirements of individuals and groups, rather than just different levels of access control.

4.4.4 A real world example profile

The limited set of functional requirements along with the choice to follow well established internet standards and protocols make the development and integration of new services within Amelie a relatively straightforward and simple process. By allowing developers and designers of such services to use any toolkit, programming language, or medium that they feel more comfortable with, Amelie provides enough power and flexibility to focus on the design concepts, and the involved social implications.

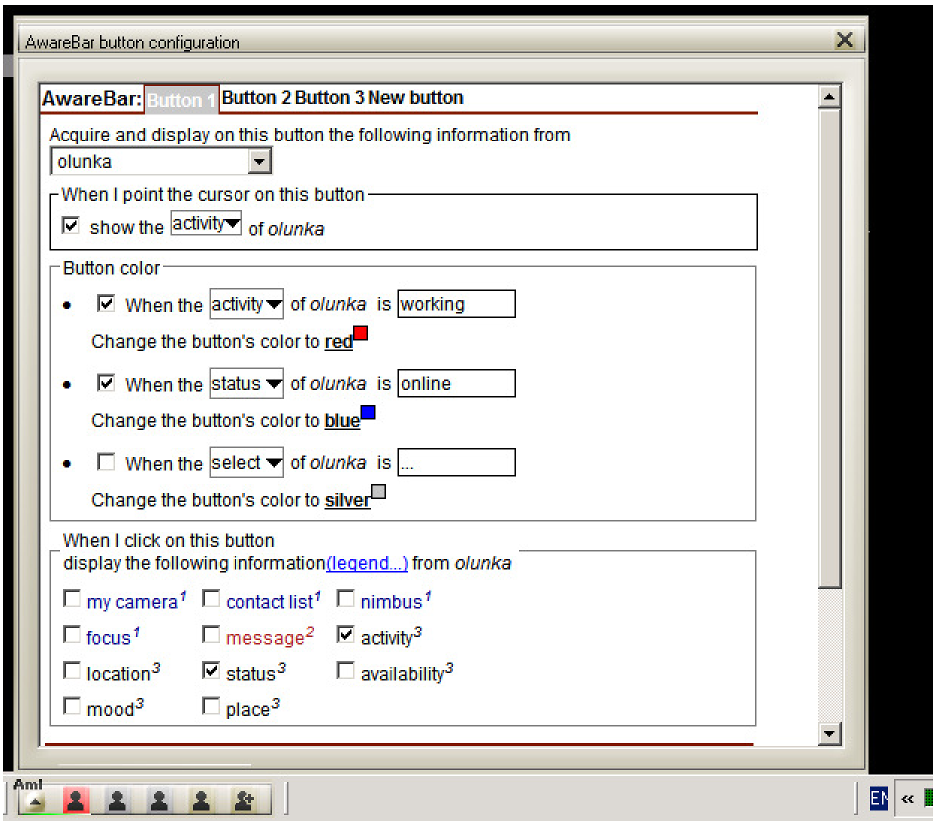


Figure 4. AwareBar, is implemented as a component of the Windows taskbar

Services that extract contextual information through wireless -sensor networks, that define and manage one's social network, that extract activities and status from one's desktop, that capture images and videos from one's context, that extract location using existing Wi-Fi infrastructure, that allow information decoration through artefacts, that allow end-users to define rules and constraints for generating meta-information, sharing and acquiring information from others, are only a small fragment of services that have been implemented on the Amelie framework addressing various end-user requirements for building up awareness systems in the past 3 years. These services have been used extensively in relevant studies, demonstrations (e.g., Khan et al., 2008; Khan et al., 2010) and several student projects.

For example, consider the service instance AwareBar that can be configured using the desktop client depicted in figure 4. AwareBar is implemented as an integrated component of the Windows desktop, occupying a band on the Windows task bar, and is populated with several push buttons. Each button can be assigned individually to a contact of the user, and programmed to change color depending on the acquired information. For example a user can state that the first button should be assigned to her father, and turn to blue when he is online. Additionally each button can be configured to display a call-out window about specific aspects from the focused contact when the user's mouse is hovered over it, and the information to be displayed in a pop-up window when she clicks the button.

Figure 5. Configuring the Today-plugin for Smartphones to acquire and display several aspects from one of the user's contacts (left); here the user is opting to inquire several aspects relevant to a specific contact. The acquired information is displayed on the wall-paper of the Smartphone (right).

In a quite diverse pervasive awareness application (Khan et al., 2010) a service had to be developed to display information on smart-phone devices as an integrated part of the smart-phone's operating system. For that a "Today plugin" service was developed, able to render any acquired awareness information as a background 'wallpaper' of smart-phone devices, while at the same time allowing end-users to configure it as if it is an integrated part of the Windows Mobile platform(figure 5).

In general, and following the concept of the Profile-Processor presented earlier, the behaviour of an entity within an awareness system can be defined by interconnecting services such as those outlined above, composing an entity's profile. Consider for example the following simple profile that represents the focus and nimbus of some entity:

```
<aml:profile xmlns:aml=”…” xmlns:inv=”…”>
   <inv:service id=”gps-location” uri=”http://device.to.gps“>
     <deviceid>01-234-56-789</deviceid>
   </inv:service>
   <inv:service id=”myweather” uri=”http://weather.forecast”>
      <location>
         <inv:service ref=”*”
            select=”aml:attribute[aml:aspect=’location’]”/>
      </location>
   </inv:service>
   <inv:service id=”forjohn” uri=”http://expose.some.info”>
      <attributes>
         <inv:service ref=”myweather” select=”aml:attribute”/>
      </attributes>
      <contacts>John</contacts>
   </inv:service>
   …
</aml:profile>
```

The profile above declares three service instances, each of which defines its persistent information and its relevant service URL. The first declaration is a service instance *"gps-location"*, that instructs the profile processor that the service should be instantiated for some device with an identifier *"01-234-56-789"*. The *<deviceid>* is specific to the *gps-location* service, and its semantic value is transparent to the profile processor: the profile serves as a store for the service specific parameters. The *deviceid* tag will be used internally by the service residing at the URL "*http://device.to.gps"* to retrieve the GPS coordinates from the specified device. Using the service declaration the profile processor prepares the connection with the service and invokes it with the provided parameters; consequently the GPS-location service returns the coordinates of the device "01-234-56-789":

```
<aml:attribute>
   <aml:aspect>location</aml:aspect>
   <aml:value type=”latitude-longtitude”>
      <lat value=”51.4366“>51 degrees,43.66 minutes North</lat>
      <long value=”5.4780“>5 degrees,47.80 minutes East</long>
   </aml:value>
</aml:attribute>
```

In a similar manner the profile processor is instructed to instantiate the second service (*myweather*) with its specific parameters. We can imagine that this service implements a weather service that returns an attribute describing the weather conditions and forecast at some geo-location. Yet, notice the contents of the tag '*<location>':*

```
<inv:service ref="*"
   select="aml:attribute[aml:aspect='location']"/>
```

This declaration instructs the profile processor to instantiate "*myweather*" passing to it the result of all services *(<inv:service ref="*"…)*, that return attributes regarding *"location" (aml:attribute[aml:aspect='location'])*. Notice that the Amelie *Profile-Processor*, supporting XPath expressions, allows services to interoperate beyond the typical boundaries of component frameworks. Services form dynamic compositions and are not required to refer explicitly to each other; this allows in the abovementioned case the weather service to form an implicit connection with the GPS-location service, by declaring that during its execution it wants as input any location-related attribute. Given that the service 'gps-location' returns a matching attribute, the weather service could return an attribute such as:

```
<aml:attribute>
   <aml:aspect>weather</aml:aspect>
   <aml:value>partly cloudy, 26° C</aml:value>
</aml:attribute>
```

The last declaration in the profile instructs the processor to instantiate the service '*forjohn*' passing to it the attributes that the service '*myweather*' returned previously, and an entity identifier '*John*'. Once invoked, the service at *"http://expose.some.info"* could then adorn the above attributes using the specified contact-list to define that the weather information should be exposed to John.

4.4.5 Implementing focus-nimbus composition on Amelie

The carrier of the communication objective, both in the original focus-nimbus model, and in the FN-AAR model is the focus-nimbus composition function; i.e. the function that negotiates the foci and nimbi of entities and defines the communicational outcome among them. The actual implementation of this function lies in the core of the Amelie framework, the *Awareness-Manager*. The Awareness Manager polls periodically the foci and nimbi of its registered entities pushing appropriate attributes to the identified renderers. As one can guess by now, the Awareness Manager is itself an Amelie service whose '*execute*' method implements the focus/nimbus negotiation among its registered entities.

The profile storage of an entity and its underlying mechanisms of expressing the entity's focus and nimbus are not relevant to the awareness manager. Of importance are only the information each entity exposes to others (nimbus) and the information each entity inquires from other (focus) entities in a given situation. Each entity registers in the awareness manager, a service URL that identifies the entity's current focus and nimbus (typically its profile processor). The awareness manager, invokes periodically the registered entities' profile-processor service URLs, and combines their instances in order to identify and invoke the appropriate renderers according to the FN-AAR model.

Moreover, entities registered within the *Awareness-Manager* can optionally provide a URL that points to a web service which identifies the entity's ontological associations regarding the space of awareness information. This design choice not only allows an entity to observe others in its own view, but also allows the awareness manager to protect one's privacy efficiently. E.g., it could be that Anna exposes to John that she is in the kitchen; at the same time it could be that John is focusing on Anna's activities. The ontology of John could facilitate the inference that any person who is in a kitchen is probably cooking. This way John would become aware that Anna's activity is cooking (leaving aside for the moment whether this inference would be sound or not). By allowing each entity to register its own relevant ontology, Amelie provides flexibility with regards to using simple or more complex ontologies that relate available information to information that is needed or can be inferred. With regards to implementation, third party services for defining and managing ontologies can be integrated transparently into Amelie.

The following XML fragment defines a set of entities that can be passed to the Awareness Manager execute method, triggering the focus/nimbus negotiation procedure:

```
<entity id="entity1">
   <profile url="profile-processor.svc">…</profile>
   <ontology url="entity.1.inference.engine.svc">…</ontology>
</entity>
…
<entity id="entityN">
   <profile url="profile-processor.svc">…</profile>
   <ontology url="entity.N.inference.engine.svc">…</ontology>
</entity>
```

Below we summarize how the *Awareness-Manager* invokes the appropriate renderers while applying the focus of an entity *x*, on an entity *y*.

- Initially, f_{xy}, the last pulled focus of the entity *x* on the entity *y* is recovered (i.e. the resources that *x* occupies for observing *y*)
- Then, n_{yx}, the nimbus of *y* to *x* is recovered (i.e. the attributes that *y* is exposing to *x*), and the ontology of *x* is applied on it, transforming these attributes using the entity's *x* point of view.
- In case there are constraints of symmetry, the same procedure is followed for f_{yx} and n_{xy}.
- Having the reciprocal foci and nimbi of *x* and *y*, symmetry constraints are applied on the attributes that entity *y* exposes to entity *x*, and the resources that entity *x* uses to inquire information from *y*.
- The rendering service of each one of these resources is invoked passing to it as parameters the set of corresponding attributes that were obtained after applying the symmetrical constraints, and the resulting observable item(s) is stored.
- Depending on its configuration the awareness manager exposes to the involved actors the observable items that define their reciprocal awareness.

Consider for example that the last known instance of John's focus at some time inquires about Anna's activity, while at the same time Anna's nimbus corresponds to an attribute that exposes to John that her location is in the kitchen. Given that John would have registered some ontology corresponding to an inference engine like the one described earlier, the *Awareness-Manager* would populate Anna's nimbus with the attribute that exposes to John that her activity is cooking.

Consequently, based on the fact that the inferred attribute is exposed to John and its aspect(i.e. "activity") matches John's resource that focuses on Anna, and since there are no constraints that the manager needs to validate, the manager would invoke the service pointed by john's resource (e.g., "*http://some.actuator.at.home*") passing to it the abovementioned attribute. The resulting observable item from the rendition of the original attribute about Anna's location in the kitchen is reflected as if her activity were cooking.

4.5 Examples and case studies

In the remaining sections we will discuss briefly some examples and demonstration applications that have been developed on the Amelie platform, which illustrate its use and capabilities.

4.5.1 Enabling social translucency

Erickson et al. (2002) examine the notion of social translucency and socially translucent systems; e.g. systems which provide perceptually based social cues which afford awareness and accountability. They state the need to make socially salient information visible in applications mediating social interactions, enabling the application of social norms that influence people's social behaviour. In chapter 2 we have been discussing extensively about these concepts which can be summarized in statements such as *"because I know your situation, I adjust my behaviour accordingly"*, *"because I know that you know my situation, I adjust my behaviour accordingly"*, or even *"because I know that you know that I know your situation, I adjust my behaviour accordingly"*.

In this section we will examine how the Amelie framework allows seamless integration of social translucency in awareness applications. Consider for example the following situation: Anna exposes to John her current activity. However for John to apply some resources for rendering Anna's activity (i.e. for John to focus on her activity), he primarily needs to be aware of the fact that Anna is exposing to him her activity. The typical solution would be to announce the kinds of information exposed with a dedicated API (information and application specific); however within Amelie this is done through a simple service that exposes to John Anna's exposed aspects. For that, Anna's profile has to instantiate the following AML attribute through a service:

```
<aml:attribute>
   <aml:aspect>nimbus to john</aml:aspect>
      <aml:value type="aml-aspect-list">
         <aml:aspect>activity</aml:aspect>
         <aml:aspect>location</aml:aspect>
         …
      </aml:value>
      <aml:access>
         <aml:entity>John</aml:entity>
      </aml:access>
</aml:attribute>
```

The service instance should be able to locate all the attributes that Anna is exposing to John and therefore construct an attribute that contains the list of the involved aspects (such as the above attribute). This task can be carried out in quite a straightforward way thanks to a simple declaration in Anna's profile neither longer nor more complex than the following:

```
<inv:service id="trivial-translucency" uri="http://...">
   <exposed-aspects>
      <inv:service ref="*" select="aml:attribute[
           aml:access/aml:entity='John']/aml:aspect"/>
   </exposed-aspects>
</inv:service>
```

The above declaration is sufficient to query Anna's profile for all the attributes that she is exposing to John, extract their involved aspects and pass this information to the service itself. The only thing that the '*trivial-translucency*' service has to do is to format its parameters (i.e. the extracted aspect-list) into an AML attribute such as the aforementioned.

We can imagine that Anna, apart from exposing information to John, also acquires some information (e.g. mood) from him. Typical social translucency behaviour would not only apply a resource that focuses on John's mood but also exposing to him Anna's focus (i.e. the fact that Anna is acquiring his mood). Similarly to the previous example and in an equally concise way a translucency service that exposes to John Anna's focus on him can be implemented. Such a service could only require as input the list of aspects that one's resources are acquiring from some other entity, and in the above situation would be declared in Anna's profile with the following statement:

```
<inv:service id="focus-translucency" uri="http://...">
   <acquired-aspects>
      <inv:service ref="*"
         select="aml:resource[aml:entity='John']/aml:aspect"/>
   </acquired-aspects>
</inv:service>
```

The above declaration in Anna's profile, instructs her profile processor to extract all of her resources that focus on John, and pass to the service the list of their aspects, hence allowing the service to generate an attribute that exposes to John the acquired information.

As we mentioned above, the above system's behaviour could be implemented as a dedicated mechanism. But then, to understand the value of Amelie, apart from its brevity and clarity, consider the following situations that involve similar social translucent patterns.

1. *"Anna wants to let John know when she is focusing on their daughter's activity"*
2. *"Anna wants to let John know what she is exposing to her Boss"*

To incorporate the above situations and enable the required system behaviour in Amelie, nothing more complicated than the previous examples should be declared in her profile. In fact a simple declaration that instructs Anna's profile to pass as input to the translucency service her focus on *Daughty* would be enough for the first problem. A declaration that instructs Anna's profile to extract Anna's exposed attributes to the Boss would be enough for the second. The seamless integration of AML with XPath makes the solution to the above problems almost trivial as one can notice by the following XML snippet which could serve to instantiate a service implementing such complex behaviours.

```
<inv:service id="…" uri="http://...">
   <expose-to>
      <aml:entity>John</aml:entity>
   </expose-to>
   <as-aspect>nimbus to the boss</as-aspect>
   <attributes>
     <inv:service ref="*"
        select="aml:attribute[aml:access/aml:entity='boss']"/>
   </attributes>
</inv:service>
```

In the following figure (figure 6), the interactive user configures our prototypical translucency service to expose to the user's colleagues at work and busy friends what she is exposing to her boss[15].

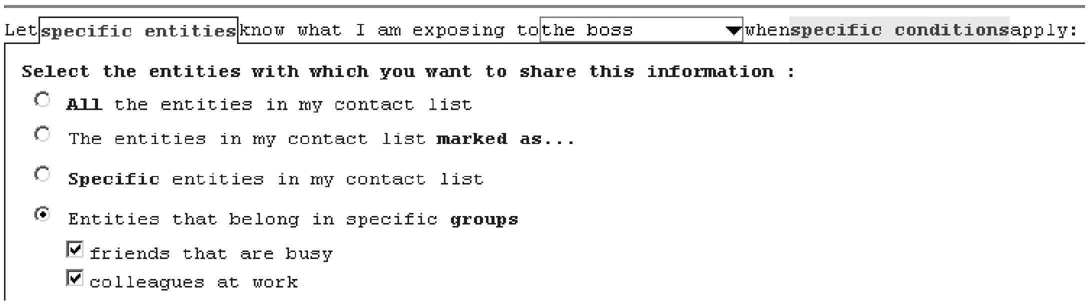


Figure 6. The user configures an instance of the translucency service.

4.5.2 Enabling deception

In earlier work (Metaxas & Markopoulos, 2007) and in chapter 2 we have shown how we can reason about deceptive patterns, such as lying, blurring, and hiding in terms of the FN-AAR model. Deception is clearly different than privacy regulation in terms of selecting who has access to what information; apart from such explicit mechanisms of control, there is a need to take into account the implications that rise from ontological associations that an entity may employ while forming its model of the situation of others.

In a simplified view one could register a service that enfolds and filters the output from one or more services or even a profile. Such a service could for example

[15] Notice that both the terms 'colleagues at work' and 'busy friends' are referring to other Amelie services in the user's profile, while the conditions under which the service should act are employing the instantiation of a service that is allowing the generation of logical expressions relevant to the user's state.

implement or embed a user-specific ontology, allowing the service to apply deceptive patterns based on the inferences that the user herself would have made.

For example, in the snippet below, a deceptive filter service enfolds *Anna's* profile within the *Awareness Manager* in order to apply deceptive filters on the entity's nimbus. In terms of the *Awareness Manager,* the user's focus and nimbus originates from another Amelie service that replaces the original profile of Anna; this service however is the deceptive filter that now invokes the original profile-processor for the entity's profile, allowing it to apply a set of deceptive rules and filters, taking into account the original focus and nimbus of the user and the implications arising from the associations defined in the ontology associated with her profile.

```
<entity id="anna">
   <profile url="deceptive-filter.svc">
      <enfold>
         <actual-profile url="profile-processor.svc">
            …
         </actual-profile>
      </enfold>
      <ontology url="anna.inference.engine.svc">…</ontology>
      <rules>…deception rules…</rules>
   </profile>
   <ontology url="anna.inference.engine.svc">…</ontology>
</entity>
```

One could comment, that such a deceptive-filter service should also use the ontologies of the entities focusing on the user (Anna in this case), hence the deceptive filters should be applied on a same level and within the same mechanism similar to the one used to impose symmetrical constraints. However, this implies that every entity would be able to use the inferences that the rest could make about it without their consent, where in fact Amelie allows entities to permit or decline sharing even this information through the focus-nimbus composition.

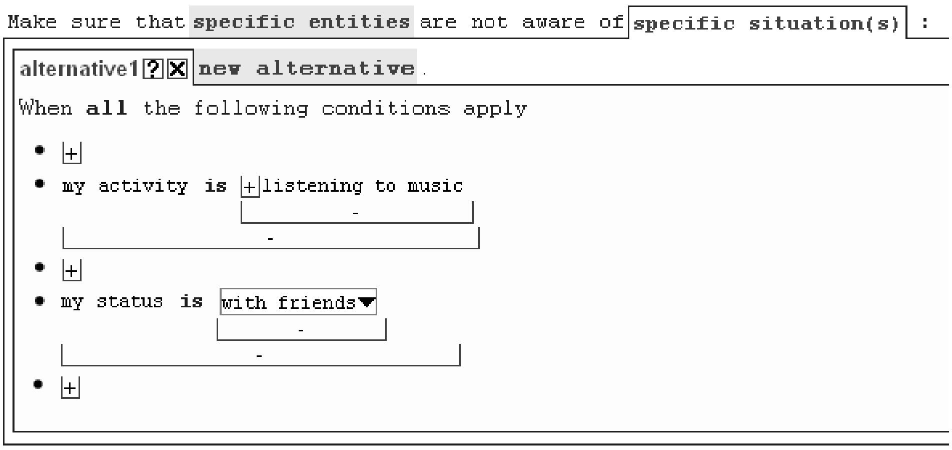


Figure 7. Configuring an instance of a deceptive filter service employed to hide a specific situation

In practice, our prototypical deceptive-filter service allows entities to choose any context-range (using the same semantics as those that services use to describe their effective output) that should be hidden from others. Moreover this service inquires (through the entity's focus) the ontologies that others are using; this allows the service to apply its deceptive filters using both the entity's own ontology and the counterpart's ontologies (if available). In the figure above (figure 7) the user employs the service to ensure that her family is not aware that she is partying with her friends. The service will either act as a cloaking filter (see chapter 2, section 2.6.3) when she is exposing (implicitly or explicitly, depending on the acquired ontologies) information about her activity that is inferring the specified situation, or as a blurring filter otherwise (see chapter 2, section 2.6.1).

Although the aforementioned facility supports deception and privacy considerations to a big extent, it is not yet enough to protect users from certain disclosure hazards. Consider the following application:

"The cupboards at Anna's kitchen contain sensors that detect whether the cupboards are open or closed. Based on this information, the system determines kitchen activity, and Anna is eventually exposing to John her activity in the kitchen."

Several privacy-related scenarios may be derived from the above application:

- *One evening, she opens the cupboard with the candies for the fifth time in the last hour. When she closes the cupboard, it is already too late to hide from John her greediness. Anna blames herself instead of the system that disregards people who forget.*
- *One evening, just when Anna is ready to open the cupboard with the candies, for the fifth time in the last hour, she thinks that John would worry being aware of her greediness. She realizes her activity should be hidden from John before opening the cupboard, and restored just after she is done. Anna does not open the cupboard to avoid the embarrassment, while at the same time she is thinking: 'Awareness systems hate what real people love: bad habits'.*
- *One evening, she opens the cupboard and picks the fifth candy in the last hour. When she closes the cupboard, she thinks that it is not too late to hide from John her greediness. She recalls that she has installed a hide-and-delay service that delays the propagation of the cupboard's activity within the system for two minutes. She activates the hide-functionality of it, yet she does not realize that John who also has access to the cupboard, is now aware that the cupboard is open, and even worse can anticipate her attempt to lie.*
- *One evening, she opens the cupboard and picks the fifth candy in the last hour. When she closes the cupboard, it is not too late to hide from John her greediness. The hide button on the cupboard's handle has a small light which indicates that the cupboards state is not yet propagated through the system. Anna presses the 'hide button', and the candy is melting in Anna's mouth while she recalls with relish: 'Awareness systems remember that people forget'.*

In the first scenario, Anna is completely exposed by the system because she is not aware early enough of the implications of her interaction with the otherwise 'smart' environment. In the second scenario, it is apparent that she has to divert her activities one way or another. In the third scenario, a generic solution such as a 'delay and hide' filter-service can enfold any number of services within an entity's profile. However, as the scenario demonstrates, such a solution could only apply on services that don't share their output among different entities or otherwise could lead to socially inconvenient situations. In the last scenario, in contrast, the service that causes the sequence of events provides a sufficient and socially appropriate mechanism that allows Anna to avoid the consequences of her implicit interaction with the system.

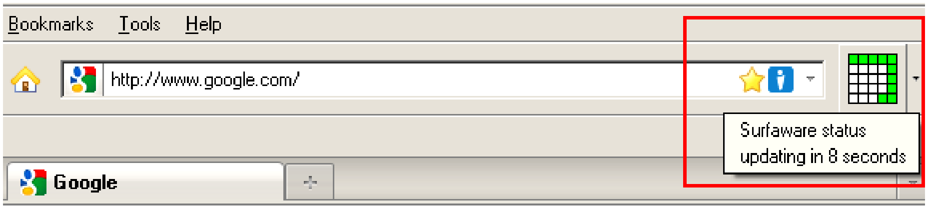


Figure 8. Surfaware is an Amelie service that provides a hide-and-delay functionality to protect the user from potentially embarrassing situations

It becomes apparent that to address the social implications of implicit interaction, services that extract information directly from the environment should themselves provide sufficient mechanisms (such as the *'hide-and-delay'*- button attached on the cupboard handle in the last scenario). For example, Surfaware is an Amelie service implemented as a Firefox plug-in that extracts the active website that the user is visiting with her browser. When the user switches among the browser's tabs or visits a website, the service populates the user's context with an attribute corresponding to the last visited URL. The service however, uses a delayed update feature (figure 8) and peripheral notification before updating the extracted URL. This provides sufficient time to end-users to roll-back from accidentally privacy-threatening situations.

In any case, services that by design may potentially propagate information to more than one user could reduce the privacy related risks by exposing indirectly their extracted information through an agent which takes into account the implications of sharing such information to multiple entities and regulates the information flow. The discomfort that may result in the third example, due to the fact that the cupboard sensor can be accessed both by Anna and by John, can be avoided provided that the designers of such services are aware of the implications.

For example, in our prototypical implementation of a wireless sensor network, a variety of wireless sensors has been developed, each extracting different features (accelerometers, light, weight, sound, and electrical-current sensors can be placed in pockets, cupboards, closets, under chairs and sofas, on tables and desks, or attached to any electrical appliance returning their rudimental states, and the list of other sensors within proximity). Each sensor is implemented as an Amelie service; however, these services are not directly accessible by actors. Instead, a network of such sensors is implemented as an agent-entity which serves as a mediator between its sensors and its registered end-users. The agent employs auto-discovery mechanisms to register the available sensor within its profile, while at the same time it can be configured, just like any other Amelie entity, to expose the state of sensors to its users in a social intelligent and privacy respecting way. For example, end-users can program it to expose the state of those sensors that are within proximity of a user's token, or to deny access to sensors that belong in the private space of some other entity. To the end-user this is a transparent mechanism, since the profiles of end-users that register within the agent are populated by service counterparts of the actual sensor services.

4.5.3 Enabling communities and artificial agents

As the FN-AAR model predicts, there is no need for special treatment of communities within an awareness system. A community in terms of the FN-AAR model is just another entity that is related with Actors in a similar fashion. This approach is followed closely by Amelie, and supported by several services that enable entities to act as communities. To illustrate how this is performed through the Amelie framework let us consider a demonstration application based on the scenario below.

"Eric, Nora and Killa all stay at the same hotel. The hotel provides its guests with an online community "HotelClub" that allows them to come easier in contact. The community lets its members to know how many of them are at various places of the hotel, but to preserve their privacy it doesn't disclose to each other their identities. On the other hand, it lets guests know who else is coming from the same country to make them feel at comfort."

HotelClub in the above scenario is implemented as an entity. What makes the *HotelClub* a community and differentiates it from the involved actors, is only its profile configuration and the services it involves. We describe briefly the key elements of such an implementation: The *HotelClub* acquires from its registered members (i.e. the hotel guests, Eric, Nora and Killa) their country of origin and their current location. For that, a simple service should be inserted in its profile that focuses on the desired aspects ("*origin*", and "*location*") of *Eric, Nora* and *Killa*. Besides, an instance of another service aggregates the acquired locations in order to generate the summary of people that are at all the distinct locations (for example it could be that one person is at the restaurant and two at the hotel's swimming pool). The service is instructed to expose the generated attributes to all involved entities. A second instance of the same service, with different parameters, extracts the country of origins as exposed by the involved entities and exposes back to each of them, people of the same origin (for example Nora, and Eric are both from the Netherlands). In terms of interaction, the crucial part in the above application is the registration of the involved actors in the community, and the mechanisms through which the actors expose and acquire information from the *HotelClub*. The community designer may offer through a web client a service bundle to the actors using the mechanisms described earlier in section 4.3 of this chapter; when the bundle is registered in their respective profiles, it ensures that the actors have a default way to both expose and acquire information from the community. The actors then are free to modify their participation level, as their profile and its services provide the interactive tools necessary to control the system.

Let us now consider another demonstration example from an earlier case study (Khan et al., 2008) that involves a photo sharing community which enables automatic capturing and distribution of information.

"Alf and Didix are registered in a camera sharing community that has various cameras in the city. The community allows them to take automatic snapshots of themselves as they walk through the city, but at the same time it allows them to define privacy contexts for the various camera locations to protect their privacy."

One can guess by now that what brings the above scenario to life is the configuration of the AML profiles of the various involved entities. The actors involved (i.e. Alf and Didix) expose their location to the *autophoto* community along their privacy considerations for their locations[16]. On the other hand the involved agents (the various cameras at different locations) expose to the *autophoto* community their location and their current snapshots in various levels of detail. The services that the actors use to detect their location and expose it to the community as well as the services that the agents use to take snapshots and detect their location are encapsulated in their respective AML profiles and are concealed from all the rest. The community is only interested in the information exposed to it by Alf, Didix, and the camera agents; hence its profile focuses on the location and location-privacy-settings of its members (Alf, and Didix) on one hand, and on the snapshots and location of its registered camera agents on the other. The acquired information (if any) is processed by a service that matches the contexts (in this case location) of the actors with the agents, and taking into account their respective privacy settings exposes to each actor the proper snapshots in the accounted level of detail (effectively reducing the accuracy or precision of the information presented to account for privacy). It is up to the actors then again to focus on the snapshots that the *autophoto* community is exposing back to them and further decide how the acquired pictures will be used.

4.6 Conclusions

We have presented Amelie, a framework for developing Awareness Systems based on the FN-AAR model (Metaxas & Markopoulos 2006) and the notion of recombinant computing (Edwards et al., 2001). The Amelie framework provides the necessary semantics to directly implement systems that ensure socially salient properties for awareness systems, namely symmetry, divergent ontologies, and social translucency.

Typically, component frameworks, e.g. 'Sens-ation'(Gross et al., 2006), in the domain of context-awareness and ubiquitous computing rely on classifying and differentiating among numerous component types which in itself induces complexity for the application developer and makes system structure quite rigid, hindering the extension of functionality.

Amelie overcomes the inherent problems of these approaches using a single interface type. Whether it is an actor, an agent, a community, a service that

[16] It could be for example that Alf exposes to the *autophoto* community that his location is in the central square, and that he considers as a private space his office and his home

disseminates, aggregates, or consumes information, it is always an Amelie service with exactly the same functional requirements. Services can communicate and identify each other's behaviour regardless of their specific implementation.

At the same time, a substantial part of the research community's focus falls on the device/service discovery problem. Frameworks such as the Context Toolkit (Dey et al., 2001) and E-Gadgets (Kameas et al., 2003) describe mechanisms which components employ to discover other components, to subscribe in each others functionality, mechanisms to notify each other for changes in their state and so on.

Contrary to these solutions, Amelie supports an implicit mechanism that allows automatic discovery and registration at various levels. Communities, agents, and actors may behave as intermediates for the process of automatic discovery, while trusted services can have permission to edit one's profile. However, this level of system automation is not achieved at the expense of privacy and social intelligence (as was the case in the examples discussed at the beginning of this chapter) exactly because Amelie is designed to support social behaviours and patterns pertaining to human communication.

Amelie embraces desktop, pervasive, and ubiquitous services both for context-sensing and for information-decoration; the recombinant design of Amelie allows seamless and rapid integration of third-party services: they only need to support a very small set of functional requirements to complement and benefit from the framework.

Amelie respects and promotes the designers' and end-users' role in the development of applications, allowing the first to explore the design space more easily, and the latter to control their participation and project their view on the system's intelligence.

Amelie's strength lies in its theoretical foundation that has been discussed in this and in previous chapters, and in its language and platform independent implementation that allows the composition of heterogeneous services. Its focus upon modelling awareness systems provides a foundation for enabling end-users to programme awareness systems, addressing several of the challenges of awareness systems, and context-aware systems in general.

Chapter 5

Conclusions

5.1 Summary

In the first chapter of this thesis three challenging aspects for advancing the field of awareness systems were identified: 1) how we can model the socially salient patterns pertaining to human communication 2) what are the proper tools and mechanisms in order to enable sustainable control and intelligibility in the domain 3) how the infrastructure itself can support the development of awareness systems not only by designers but by end-users themselves.

We argued that a key challenge was endowing social intelligence and enabling social skills and norms to be exercised in this domain. For this we should be able to interpret and predict the success or failure of such systems in relation to their communicational objectives and their social implications. We therefore developed a formal model that, by capturing the general characteristics of the awareness systems domain, allows reasoning regarding the social interaction patterns related to using this medium.

We recognized that harnessing the benefits of context awareness can be problematic for end-users and other affected individuals, who may not always be able to anticipate, understand, or appreciate system function and may feel their own sense of autonomy and privacy threatened. Hence, we pointed out the importance of a set of tools and mechanisms that can uniformly address end-user control, system intelligibility and accountability by minimizing the cognitive effort to handle the increased complexity of such systems, and by increasing the ability of people to configure and maintain their intelligent environments.

We indicated that context-aware applications overall can not be examined as static configurations of services and functions, and that they should be seen as the results of both implicit and explicit interaction with the user. Consequently, supportive frameworks for the development of context-aware applications should encourage the design of the interactive mechanisms through which end-users can control, direct and advance the lifecycle of such systems, rather than channel the interaction design through their architectural constraints.

In this concluding chapter, these challenges are reviewed in terms of the research conducted for this thesis, the contribution and limitations of this research are summarized, and a reflection on the most important results is presented.

5.2 Contribution

In chapter 2, FN-AAR, a formal model of awareness systems was introduced as a research tool that allows predictions about the social implications of awareness applications. Based on Benford's focus-nimbus model of awareness and Rodden's generalization of it, and inspired by works such as those of K. Schmidt (2002) and Boyle and Greenberg (2005) on awareness, the FN-AAR model acknowledges the need to capture the communicational objectives of awareness systems. Where the original focus-nimbus model answers the question *'how aware are two entities of each other in a particular medium'* the FN-AAR model answers the question *'what are two entities aware of, regarding each order, in a particular situation'*. This shift enables the unambiguous and comprehensive discussion of socially salient patterns pertaining to mediated communication and allows coherent predictions of their implications. As a formal representation, it brings clarity in the discussion of awareness systems, allowing clear relations to be drawn in a way previously not possible among theoretical concepts that are overlapping and related. The FN-AAR model is unique in its analytical capacities as it has been shown able of covering concepts such as social translucency, deception, and symmetry in terms of the information exchanged and the ontological associations relevant to the involved entities. The model intentionally masks away the underlying mathematical and information propagation concepts in order to leave enough space both for analysing existing applications, and for later developments and implementations.

In chapter 3, we suggested a set of tools and mechanisms that can uniformly support intelligibility, accountability, and control of context-aware systems by minimizing the cognitive effort required for handling the inherently high complexity of such systems, and by increasing the ability of people to configure and maintain their intelligent environments. For that, we introduced a context-range notation that allows on one hand services to describe the range of their outcome, and on the other hand facilitates compositionality and allows services to describe the premises that manage their behaviour. The notation can be presented to users both as natural language and by means of a structured editor that extends the paradigm of 'what you see is what you mean'. By adopting human cognitive skills and limitations, and applying a set of heuristics, it was pointed out how such semantic information –otherwise concealed from end users- can be reflected on users' mental state and effectively support intelligibility, accountability and control.

An experimental evaluation of this editor showed that non programmers could understand and formulate logical expressions of context of realistic complexity and are empowered to answer questions such as “how does the system behave”, “why is something happening”, “how would the system behave in response to a change in context”, and “how can the system’s behaviour be altered”. In the second section of this chapter, the assumptions regarding cognition that underlie the heuristics, which we used for the presentation of context-ranges and their manipulation, were further validated in a laboratory experiment. This experiment investigated the role of term-affinity in the spontaneous usage and comprehension of disjunctive and conjunctive normal forms, extending relevant findings of the Mental Models theory.

Amelie, a framework for the implementation of awareness systems was proposed, in chapter 4. By adopting a recombinant computing approach towards the implementation of awareness applications, it was argued that Amelie forms a solid foundation so that a set of recombinant services can be employed for and by end-users in the development of awareness systems. Within Amelie, a single recombinant interface was argued to be a sufficient abstraction yet not an overgeneralization of awareness systems’ services, which can engage in compositional structures forming awareness and context-aware applications. By following the tenets of the FN-AAR model the Amelie framework provides the necessary semantics to directly implement systems that ensure socially salient properties, such as symmetry, deception, and social translucency. The minimalistic functional requirements of Amelie services allow rapid integration of third-party services that extract, disseminate, and consume information, while at the same time enable the composition of services that implement higher order logic operations. Furthermore, Amelie addresses at an architectural level the requirements of intelligibility, accountability, and control and allows the dynamic composition and maintenance of applications both through implicit and explicit interaction mechanisms. Amelie realizes and uplifts the designers’ and, more importantly, the end-users’ roles in the development of applications. It allows the first to exploit the design space without considering architectural limitations, and the latter to control and project their view on the system’s intelligence while effectively participating in the system’s design.

5.3 Limitations & future work

5.3.1 Inherent (physical) awareness

The FN-AAR model homes in on the question *“what are the entities aware of regarding each-other in a particular situation”* in terms of the information changed among entities, leaving aside the question of whether real world entities (i.e. actors) actually do perceive the ‘observable items’ and therefore are physically (inherently) aware of them. This limitation causes an uncertainty while exploring or defining the social implications (e.g. in the case of social translucency) that awareness systems impose on the end-users themselves.

In chapter 2 we discussed how an appropriate extension of the model could be done by incorporating associations of observable-items with their inherent nimbus, associations of entities with their inherent focus, and a function that quantifies the question *"how aware (physically) is a physical entity (e.g. an actor) of an observable-item?"*.

With the qualification 'inherent' we wish to draw a distinction between what is made visible to the user through some resource and what this person actually perceives and becomes cognizant of. This link to user perception and cognition was not elaborated further; it was suggested, however, that the approximation for an entity's inherent focus can be done using its nimbus, while the association an observable-item with its inherent nimbus could be an intrinsic property. For example, if from the nimbus of some person one can infer that this person is very likely to be in front of a screen, we can (with some probability of error) assume that this person is also having the screen in their focus. In our later development of the Amelie framework, however, and in our case studies we didn't further explore mechanisms that could generalize such quantifiers.

On the other hand, one may notice that entities, within Amelie, are eligible to define the dynamics of their applied focus on others using tools such as the context-range-editor presented in chapter 4. For example a user could express her focus on someone else with a statement *'acquire and display a snapshot of my father on the TV when my location is in the living room'*. Such statements, although implicitly approximate the physical awareness of an entity (the above statement implies that the user will be physically aware of her father's picture given that she is in the living room and that her father is exposing to her his snapshot), are far from allowing the system to assess the entities' physical awareness of the observable-items in their environments and leave an open space for exploration and further research.

Romero and Markopoulos (2005; 2009) point out the need for lightweight interactive mechanisms by which individuals can collaboratively 'ground' needs for interaction, i.e., establish through communication acts that they are shared. Such interaction mechanisms serve to overcome the inevitable limitations of pattern recognition technology and to preserve the autonomy of both individuals. Inline with their findings, perhaps the way forward passes through the realization that such capabilities will always undergo the user's contribution; hence, instead of looking into mechanisms and tools that automatically approximate the users' inherent awareness of their acquired information we should be looking at explicit mechanisms that support users in expressing that they are physically aware of observable-items and their depicted attributes.

5.3.2 Ontological associations

An essential prerequisite of FN-AAR for capturing effectively social salient patterns of communication is the presence of a function that summarizes the ontological-associations of entities regarding the information space.

In terms of Amelie, an entity's ontology is yet another service that can be applied to any context, such as an entity's focus and nimbus instance, to yield the inferences that can be deducted. In a simplified view, a prototypical service of this latter kind was implemented, to iteratively validate the context against a set of inference-rules, until no more deductions can be made.

As a development of this inference-engine one could propose to employ the context-range editor (chapter 4) which was developed to support relevant end-user programming tasks. The context-range editor, however, is not a sufficient tool by itself-alone to allow a generic description of ontological associations for several reasons. On one hand the tools presented in chapter 4, apply on dynamic yet known contextual-ranges to allow users to define logical premises through subtractive-operations and direct-manipulation. Hence, to apply these tools for the purpose of defining ontological associations, a priori abstractions of the context space are needed. Such abstractions may be the outcome of user studies and further research to elicit domain or even population specific ontologies. On the other hand ontological associations may involve complex generic mappings and transformations that are out of the end-user programming scope. Associations of this complexity could cover for example inferences such as *"when an entity's 'history of any aspect X' is known then her current X is also known"*. Amelie provides a placeholder and opens up the space for realizing a wide range of different approaches that can be followed to address the aforementioned issues, including higher order logic deductions, such as temporal, modal, fuzzy, and deontic logic.

5.3.3 User Acceptance Studies

An ample part of this thesis has focused on developing tools and mechanisms that support the development and maintenance of awareness applications by end-users, following requirements that have been identified by earlier influential studies. Laboratory experiments were conducted to validate the effectiveness and efficiency at a cognitive level of the end-user-programming tools that we present in chapter 3.

Further studies may contribute to our findings and extend the community's understanding of the actual involvement of end-users in the development of awareness applications aiming at higher environmental validity. Among others, we have identified three interesting directions for future research. Firstly, a field study in a small established user community could help identify the prominent high level programming tasks that would naturally emerge through users' interactions. Secondly, based on existing literature scenarios, a walk-through study method could reveal to which extent the users are able to develop or maintain awareness applications. Finally, a field study in a broader social environment (e.g. by adopting Amelie to existing social-networking tools) could expand our understanding of the impact of end-user-programming on the emergence of social patterns.

5.4 The future-proof future

The domain of awareness systems -as a class of computer mediated communication systems that help individuals or groups build and maintain a peripheral awareness of each other- is typically considered as a field overlapping with relevant contemporary domains such as context-awareness, ubiquitous computing, pervasive computing, and ambient intelligence, exercising their technological tenets.

All these relevant fields, however, face scepticism regarding their cognitive and social related implications. This scepticism becomes even more intense when 'smart' technology is transferred on awareness systems, because the potential threats involve human relationships themselves. This scepticism which is often expressed informally through reflect-reactions of both experts and non experts regarding the above technological visions, surrounds notions of privacy, control, the nature of the social relationships arising; see for example Wright (2005).

The FN-AAR model foresees that the social patterns pertaining to human-human communication may apply also to artificial agents and communities. This initial observation is not simply transferred in the Amelie framework but more importantly becomes the distinctive factor separating 'services' from 'entities'. In fact the same rules and interactive mechanisms that allow an actor to build up her awareness of others can be applied also while designing artificial agents, or communities.

What transforms a component of an awareness system from a socially agnostic service to a socially-intelligent entity is its ability to comply with social behaviours, its proficiency to converse its internal principles, its flexibility to accept changes, and its capability to contribute in complicated social structures. This social intelligence pertains to the long standing debate of what artificial intelligence might stand for.

Consider *'42'* which is the answer to '*the Ultimate Question of Life, the Universe, and Everything from the supercomputer, Deep Thought, specially built for this purpose*' which is rendered in the classic novel *'The Hitchhiker's Guide to the Galaxy'* by Douglas Adams (1979). Serving science fiction comedy, Adams elegantly and deliberately presents *'Deep Thought'* as a socially incompetent computer that is not able to give any reasoning behind the answer.

Interestingly, it is Adams himself who at a latter interview of 1993 brings accountability to 'Deep Thought' by giving a simple explanation to why the number *'42'* is the 'Answer':

"The answer to this is very simple. It was a joke. It had to be a number, an ordinary, smallish number, and I chose that one. Binary representations, base thirteen, Tibetan monks are all complete nonsense. I sat at my desk, stared into the garden and thought '42 will do.' I typed it out. End of story." (Google groups, 2010)

Exercising socially-intelligent skills, two acquaintances of Adams give different, probably deceptive versions regarding the same question. Stephen Fly, claimed that

Adams told him *"exactly why 42"*, and that the reason is *"fascinating, extraordinary and, when you think hard about it, completely obvious"* while he has vowed not to disclose the secret (BBC News, 2008). On the other hand, John Lloyd in a different occasion, claimed that Adams has simply called 42 *"the funniest of the two-digit numbers"* (Lloyd, 2008).

Transferring simple behaviours, such as Adams himself and his acquaintances express above, to *'Deep Thought'* and to other fictional artefacts of Adams's book, would be sufficient to transform them to social-intelligent entities.

In this respect this thesis proposes a new viewpoint for the domain of awareness systems and the relevant domains. Awareness systems should be viewed as enfolding rather than overlapping with the adopted technological domains such as ambient intelligence and ubiquitous computing. The consequences of not taking this approach are already evident both in the external and evermore in the internal criticism that the research community has to confront, and the immense efforts to bypass it. The truth is that for every newly appearing disappearing-device, for every evolving intelligent-service, for every invading pervasive-system, a new layer that allows practicing social skills and that supports intelligibility, accountability, and control is needed before end-users can accept it and adopt it.

Over the history of computing, simple protocols and 'recombinant' architectures have survived their more complex analogues. In the 20th century the Public Switched Telephone Network (PSTN) served with its architectural simplicity a vast number of higher level applications to evolve (e.g. peer to peer voice calls, conference calls, faxing, data connections etc.). In contrast, its visionary-designed descendant, the Integrated Services Digital Network (ISDN), a set of communications standards for simultaneous digital transmission of voice, video, data, and other network services over the traditional circuits of the PSTN, is pushed to extinction. This is often attributed to the revolution that the Internet brought to communications, and even more to the appearance of Digital Subscriber Line (DSL). Quite interestingly, however, DSL was initially a part of ISDN, yet eventually PSTN adopted DSL as well and prevailed over ISDN. It was the complexity and rigidity of ISDN and its inherent architectural constraints that doomed it eventually to oblivion both in relation to the traditional telephony network and against the Internet Protocol (IP).

In the 21st century the Hypertext Transfer Protocol dominates the internet, and is expanding over all kinds of application layer protocols. Its architectural simplicity allowed developers to adopt it to such extent that nowadays it embraces, among others, applications such as e-mailing, video streaming, messaging, and file management.

If we are to keep on our vision towards a next generation of intelligent computing, a new layer, that allows end-users to take in their own hands the design and expression of systems' intelligence, has to be adopted by the research community.

The research effort presented in this thesis paves the way towards a future-proof future, where services can be employed in unforeseen compositional structures, devices can be controlled and assigned new meanings by users themselves, artefacts can adopt to a dynamically changing future environment, while engaging in social behaviours and acting -as well as being perceived as acting- intelligently.

References

Aarts, E. (2003). Technological Visions of Ambient Intelligence. In Aarts, E. and Marzano, S. (Eds.), *The New Everyday* (pp. 12-16), Rotterdam: 010 Publishers.

Aarts, E., De Ruyter, B. (2009). New research perspectives on Ambient Intelligence. *Journal of Ambient Intelligence and Smart Environments*, IOS Press, 1(1), 5–14.

Abowd G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R. & Pinkerton, M. (1997). Cyberguide: A mobile context-aware tour guide. *ACM Wireless Networks*, 5(3), 421-433.

Abowd, G. D. and Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(1), 29-58.

Ackerman, M., Darrell, T., and Weitzner, D.J. (2001). Privacy in context. *Special Issue on Context-Aware Computing, Human-Computer Interaction*, 16 (2-4), 167-176.

Adams, D. (1979). *The Hitchhiker's Guide to the Galaxy*. London: Pan Books.

Aoki, P. and Woodruff, A. (2005). Making Space for Stories: "Ambiguity in the Design of Personal Communication Systems". In *Proceedings of the Conference on Human Factors and Computing Systems, CHI '05*, (pp. 181-190), New York: ACM Press.

Baecker, R.M. (1993). *Readings in Groupware and Computer Supported Cooperative Work: Software to Facilitate Human-Human Collaboration*. Morgan Kaufmann Publishers.

Barkhuus, L. and Dey, A.K. (2003). Is context-aware computing taking control away from the user? Three levels of interactivity examined. In *Proceedings of International Conference on Ubiquitous Computing, UbiComp 2003*, (pp. 149–156), New York: ACM Press.

BBC News - Magazine - What on earth is 42? (2008). 2008-03-07. http://news.bbc.co.uk/ 1/hi/magazine/7283155.stm. Retrieved 2010-05-05.

Bederson B. B. (1995). Audio augmented reality: A prototype automated tour guide. In *Proceedings of the Conference on Human Factors and Computing Systems, CHI '95*, (pp. 210-211), New York: ACM Press.

Bellotti F., Riccardo B., de Gloria A., Margarone M. (2002). User Testing a Hypermedia Tour Guide. *IEEE Pervasive Computing*, 1(2), 33-41.

Bellotti, V, Sellen, A. (1993). Design for privacy in ubiquitous computing environments. In *Proceedings of The Third European Conference on Computer Supported Cooperative Work, ECSCW'93*, (pp. 77-92), Milan: Kluwer Academic Publishers.

Bellotti, V. and Edwards, K. (2001). Intelligibility and accountability: Human considerations in context-aware systems. *Human-Computer Interaction*, 16 (2–4), 193–212.

Benford, S., Fahlen, L. (1993). Spatial Model of Interaction in Large Virtual Environments. In *Proceedings of The Third European Conference on Computer Supported Cooperative Work, ECSCW'93*, (pp. 109-124), Milan: Kluwer Academic Publishers.

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

Benford, S., Bowers, J., Fahlén, L. E., and Greenhalgh, C. (1994). Managing mutual awareness in collaborative virtual environments. In *Proceedings of the Conference on Virtual Reality Software and Technology*. (pp. 223-236).River Edge, NJ: World Scientific Publishing Co.

Benford, S., Bowers, J., Fahlen, L., Mariani, J., and Rodden. T. (1995). Supporting Cooperative Work in Virtual Environments. *The Computer Journal*, 38 (1), 653-668.

Blackwell, A.F. and Green, T.R.G. (1999). Investment of Attention as an Analytic Approach to Cognitive Dimensions. In T. Green, R. Abdullah & P. Brna (Eds.) *Collected Papers of the 11th Annual Workshop of the Psychology of Programming Interest Group (PPIG-11),* (pp. 24-35)

Blackwell, A. F. and Hague, R. 2001. AutoHAN: An Architecture for Programming the Home. In *Proceedings of the IEEE 2001 Symposia on Human Centric Computing Languages and Environments (Hcc'01)* (pp. 150-157), Washington, DC: IEEE Computer Society.

Blackwell, A.F. (2002). Applying attention investment to end-user programming. In *Proceedings of the IEEE 2002 Symposia on Human Centric Computing Languages and Environments (HCC'02),* (pp. 28-36), Washington, DC: IEEE Computer Society Press.

Bly, S., Harrison, S., Irwin, S. (1993). Media Spaces: Bringing people together in a video, audio, and computing environment. *Communications of the Association of Computing Machinery*, 36 (1), 29-47.

Boyle, M. and Greenberg, S. (2005). The Language of Privacy: Learning from Video Media Space Analysis and Design. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12 (2), 328-370.

Brave, S. and Dahley, A. (1997). inTouch: a medium for haptic interpersonal communication. In *Extended Abstracts of the Conference on Human Factors and Computing Systems, CHI '97*, (pp. 363-364), New York: ACM Press.

Byrne, R.M.J. and Johnson-Laird, P. N. (1992). The Spontaneous Use of Propositional Connectives. *The Quarterly Journal of Experimental Psychology Section A: Human Experimental Psychology*, 1464-0740, 45 (1), 89 – 110.

Chalmers, M. and Galani, A. (2004). Seamful Interweaving: Heterogeneity in the Theory and Design of Interactive Systems. In *Proceedings of Designing Interactive Systems, DIS 2004,* (pp. 243-252), New York: ACM Press.

Chalmers, M., Maccoll, I. and Bell., M. (2003). Seamful design: Showing the seams in wearable computing. In *Proceedings of IEEE Eurowearable*, (pp. 11–17), Washington, DC: IEEE Computer Society Press.

Cheverst, K., Byun, H.E., Fitton, D., Sas, C., Kray, C., Villar, N. (2005). Exploring Issues of User Model Transparency and Proactive Behaviour in an Office Environment Control System. *User Modelling and User-Adapted Interaction*, 15 (3-4), 235-273.

Consolvo, S., Roessler, P., Shelton, B. E. (2004). The CareNet Display: Lessons learned from an in-home evaluation of an ambient display. In *Proceedings of International Conference on Ubiquitous Computing, UbiComp 2004*, (pp. 22-29), New York: ACM Press.

Consolvo, S., Smith, I., Matthews, T., LaMarca, A., Tabert, J., Powledge, P. (2005). Location Disclosure to Social Relations: Why, When, & What People Want to Share. In *Proceedings of the Conference on Human Factors and Computing Systems, CHI '05*, (pp. 81–90), New York: ACM Press.

Davidoff, S., Lee, M.K., Yiu, Zimmerman, C. J. and Dey, A. K. (2006). Principles of smart home control. In *Proceedings of International Conference on Ubiquitous Computing, UbiComp 2006*, (pp. 19–34), New York: ACM Press.

Davies N., Mitchell, K., Cheverst, K. & Blair, G. (1998). Developing a context-sensitive tour guide. In *Proceedings of the 1st Workshop on Human Computer Interaction for Mobile Devices*, (pp. 64-68), Glasgow, Scotland.

Dey A., Abowd G. D., and Salber D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of contextaware applications. *Human-Computer Interaction,* 16(2–4), 97–166.

Dey A. K. (1998). Context-aware computing: The CyberDesk project. In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, (pp. 51-54), Menlo Park, CA: AAAIPress.

Dey A. K., Salber, D., Abowd, G. D. (1999). A context-based infrastructure for smart environments. In *Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99)*, (pp. 114-128), Heidelberg, Germany: Springer Verlag.

Dey, A. K. and Newberger, A. (2009). Support for context-aware intelligibility and control. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '09*, (pp. 859-868), New York: ACM Press.

Dey, A. K., Hamid, R., Beckmann, C., Li, I., and Hsu, D. (2004). a CAPpella: programming by demonstration of context-aware applications. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '04,* (pp. 33-40), New York: ACM Press.

Dey, A.K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing*, 5 (1), 4-7.

Dey, A.K., Sohn, T., Streng, S., and Kodama, J. (2006). iCAP: Interactive Prototyping of Context-Aware Applications. In *Proceedings of Pervasive 2006*, (pp. 254-271), Heidelberg, Germany: Springer Verlag.

Dix, A. J. (1990). Information processing, context and privacy. In D. Diaper, D. J. Gilmore, G. Cockton, and B. Shackel, (Eds.), *Proceedings of the IFIP Tc13 Third international Conference on Human-Computer interaction*, (pp. 15-20), Amsterdam: North-Holland Publishing Co.

Dix, A., Finlay, J., Abowd, G., Beale, R. (1998). *Human Computer Interaction*, (pp. 128-129), London: Prentice Hall.

Dix, A., Sheridan, J., Reeves, S., Benford, S., O'Malley, C. (2005). Formalising Performative Interaction. In *Proceedings of Designing interactive systems: processes, DSV-IS 2005, LNCS 3941,* (pp. 15-25), Heidelberg, Germany: Springer Verlag.

Dourish, P. and Bellotti, V. (1992). Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work, CSCW '92*, (pp. 107-114), New York: ACM Press.

Dourish P., Bly, S. (1992). Portholes: supporting awareness in a distributed work group. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '92*, (pp. 541-547), New York: ACM Press.

Edwards, W. K., Newman, M. W., and Poole, E. S. (2010). The infrastructure problem in HCI. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '10,* (pp. 423-432), New York: ACM Press.

Edwards, W.K, Newman, M.W., Sedivy, J.Z. (2001). The Case for Recombinant Computing. *Technical report CSL-01-1*, Palo Alto Research Center.

Edwards, W.K., Grinter, R.E. (2001). At Home with Ubiquitous Computing: Seven Challenges. *Ubicomp 2001: Ubiquitous Computing Lecture Notes in Computer Science*, Volume 2201/2001, (pp. 256-272), Heidelberg, Germany: Springer Verlag.

Edwards, W.K., Newman, M.W., Sedivy, J., Smith, T., Izadi, S. (2002). Challenge: Recombinant Computing and the Speakeasy Approach. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom 2002)*, (pp. 279-286), New York: ACM Press.

England, P., Allen, R., Underwood, R. (1996). RAVE: Real-time services for the Web, Computer Networks and ISDN Systems. In *Proceedings of the Fifth International World Wide Web Conference*, 28 (7-11), 1547-1558.

Erickson, T., Smith, D. N., Kellogg, W. A., Laff, M., Richards, J. T., Bradner, E. (1999). Socially translucent systems: social proxies, persistent conversation, and the design of "babble". In *Proceedings of the Human Factors in Computing Systems Conference, CHI '99*, (pp. 72-79), New York: ACM Press.

Erickson, T. and Kellogg, W. (2000). Social Translucence: An Approach to Designing Systems that Mesh with Social Processes. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7 (1), 59-83.

Erickson, T. (2009). 'Social' systems: designing digital systems that support social intelligence. *AI & Society,* 23 (2), 147-166.

Erickson, T., Halverson, C., Kellogg, W. A., Laff, M., and Wolf, T. (2002). Social translucence: designing social infrastructures that make collective activity visible. *Communications of the ACM*, 45 (4), 40-44.

Feiner S., MacIntyre, B., Hollerer, T. and Webster, T. (1997). A Touring Machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. *Personal Technologies*, 1(4), 208-217.

Fels S., Sumi, Y., Etani, T., Simonet, N., Kobayshi, K. and Mase, K. (1998). Progress of C-MAP: A context-aware mobile assistant. In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, (pp. 60-67), Menlo Park, CA: AAAI Press.

Fernado, O., Adach, K., Cohen, M. (2004). Phantom Sources for Separation of Listening and Viewing Positions of Multipresent Avatars in Narrowcasting Collaborative Virtual Environments. In *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04),* (pp. 170-175), Washington, DC: IEEE Computer Society Press.

Fields, B., Merriam, N., Dearden, A. (1997). DMVIS: Design, Modelling and Validation of Interactive Systems. In *Proceedings of Design, Specification, Verification of Interactive, Springer Computer Science*, (pp. 29-44), Heidelberg, Germany: Springer Verlag.

Fish, R.S., Kraut, R.E., Root, R.W. et al. (1992). Evaluating video as a technology for informal communication. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '92)*, (pp. 37–48), New York: ACM Press.

Fitzpatrick, G., Mansfield, T., Kaplan, S., Arnold, D., Phelps, T., and Segall, B. (1999). Augmenting the workaday world with Elvin. In *Proceedings of the Sixth Conference on European Conference on Computer Supported Cooperative Work* (ECSCW), (pp. 431-450), Norwell, MA: Kluwer Academic Publishers.

Friedman, B., Kahn, P.H. Jr, Hagman, J., Severson, R.L., Gill, B. (2006). The Watcher and the Watched: Social Judgments About Privacy in a Public Place. *Human-Computer Interaction*, 21 (2), 235-272.

Fuchs, L. 1999. AREA: a cross-application notification service for groupware. In *Proceedings of the Sixth Conference on European Conference on Computer Supported Cooperative Work* (ECSCW),(pp. 61-80), Norwell, MA: Kluwer Academic Publishers.

Fuchs, L., Pankoke-Babatz, U., Prinz, W. (1995). Supporting cooperative awareness with local event mechanisms: The GroupDesk system. In *Proceedings of the 1995 Fourth European Conference on Computer Supported Cooperative Work , ECSCW'95*, (pp. 247-262), Milan: Kluwer Academic Publishers.

Gaver, G.W., Beaver, J., Benford, S. (2003). Ambiguity as a resource for design. In *Proceedings of the Conference on Human Factors in Computing Systems,CHI '03*, (pp. 233-240), New York: ACM Press.

Gaver, W. (1992). The affordances of media spaces for collaboration. In *Proceedings of the Conference on Computer-Supported Cooperative Work, CSCW '92*, (pp. 17–24), New York: ACM Press.

Gaver, W., Moran T., MacLean, A., et al. (1992). Realizing a video environment: EuroPARC_s RAVE system. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '92,* (pp. 27–35), New York: ACM Press.

Gaver, W., Sellen, A., Heath, C., and Luff, P. (1993). One is not enough: multiple views in a media space. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93),* (pp. 335-341), New York: ACM Press.

Google groups, (2010). Why 42? (1993). http://groups.google.com/group/alt.fan.douglas-adams/msg/d1064f7b27808692?dmode=source. Retrieved 2010-05-05.

Green, J. (2003). Thinking the Future. In Aarts, E., and Marzano, S., (Eds.) *The New Everyday*, (pp. 22-27), Rotterdam: 010 Publishers.

Green, T.R.G. and Petre, M. (1992). When visual programs are harder to read than textual programs. In *Proceedings of the Sixth European Conference on Cognitive Ergonomics (ECCE-6),* (pp. 167-180), Budapest, Hungary.

Green, T.R.G., Petre, M., and Bellamy, R.K.E. (1991). Comprehensibility of visual and textual programs: A test of superlativism against the "match-mismatch" conjecture. In J. Koenemann Belliveau, T. G. Moher and S. P. Robertson (Eds.), *Empirical Studies of Programmers: Fourth Workshop*, (pp. 121-141), Ablex.

Green, T.R.G. and Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' approach. *Journal of Visual Languages and Computing.* 7 (2), 131-174.

Greenberg, S., Rounding, M. (2001). The notification collage: posting information to public and personal displays. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '01,* (pp. 514-521), New York: ACM Press.

Gross, T., Egla, T., Marquardt, N. (2006). Sens-ation: A Service-Oriented Platform for Developing Sensor-Based Infrastructures. *International Journal of Internet Protocol Technology,* 1 (3), 159-167.

Grudin, J. (1994). Groupware and social dynamics: eight challenges for developers. *Communications of ACM,* 37 (1), 92-105.

Gutwin, C., Roseman, M., Greenberg, S. (1996). A usability study of awareness widgets in a shared workspace groupware system. In M. S. Ackerman (Ed.). *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work CSCW '96*, (pp. 258-267), New York: ACM Press.

Gutwin, C. and Greenberg, S. (1996). Workspace awareness for groupware. In *Conference Companion on Human Factors in Computing Systems: Common Ground* (pp. 208-209) New York: ACM Press.

Heath, C. and Luff, P. (1991). Disembodied conduct: Communication through video in a multimedia office environment. In *Proceedings of Human Factors in Computing Systems, CHI '91*, (pp. 99–103), New York: ACM Press.

Hindus, D., Mainwaring, S.D., Leduc, N., Hagström, A.E., Bayley, O. (2001). Casablanca: Designing social communication devices for the home. In *Proceedings of Human Factors in Computing Systems, CHI '01*, (pp. 325-332), New York: ACM Press.

Hong, J.I., Landay, J.A. (2004). An Architecture for Privacy- Sensitive Ubiquitous Computing. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services Mobisys '04*, (pp. 177-189), New York: ACM Press.

Hudson, S.,E., Smith, I. (1996). Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems. In *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work, CSCW'96*, (pp. 248-257), New York: ACM Press.

Humble, J., Crabtree, A., Hemmings, T., Åkesson, K., Koleva, B., Rodden, T., and Hansson, P. (2003). "Playing with the bits": User-configuration of ubiquitous domestic environments. In *Proceedings of International Conference on Ubiquitous Computing, UbiComp 2003*, (pp. 256–263), New York: ACM Press.

Iachello, G., Smith, I., Consolvo, S., Abowd, G.D., Hughes, J., Howard, J., Potter, F., Scott, J., Sohn, T., Hightower, J., and Lamarca, A. (2005). Control, Deception, and Communication: Evaluating the Deployment of a Location-Enhanced Messaging Service. In *Proceedings of International Conference on Ubiquitous Computing, UbiComp '05*, (pp. 213-231), New York: ACM Press.

Iachello, G., Smith, I., Consolvo, S., Chen, M., Abowd, G. (2005). Developing Privacy Guidelines for Social Location Disclosure Applications and Services. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS 2005),* (pp. 65-76), New York: ACM Press.

Ji Gao-feng, Tang Yong, Jiang Yun-cheng (2006). A Service-Oriented Group Awareness Model and Its Implementation. In *Proceedings of the Knowledge Science, Engineering and Management, Second International Conference, KSEM 2006,* (pp. 139-150), Heidelberg, Germany: Springer Verlag.

Jiang, X., Hong, J.L., Landay, J.A. (2002). Approximate Information Flows: Socially-based Modelling of Privacy in Ubiquitous Computing. In *Proceedings of International Conference on Ubiquitous Computing, UbiComp 2002*, (pp. 176-193), New York: ACM Press.

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Harvard University Press.

Johnson-Laird, P. N. (1994). Mental Models, Deductive Reasoning, and the Brain. In Gazzaniga, M. S. (Eds.), *The Cognitive Neurosciences,* (pp. 999–1008), Cambridge, MA: MIT Press.

Johnson-Laird, P. N., and Byrne, R. M. J. (1991). *Deduction*. Hillsdale, NJ: Lawrence Erlbaum.

Jones, S., Mcinnes, S., and Staveley, M. (1999). A graphical User Interface for Boolean Query Specification. *International Journal on Digital Libraries*, 2 (2-3), 207-223.

Kameas, A., Bellis, S., Mavrommati, I., Delanay, K., Colley, M., Pounds Cornish, A. (2003). An architecture that treats everyday objects as communicating tangible components. In *Proceedings of the 1st IEEE international conference on pervasive computing and communications (PerCom 2003),* (pp. 23–26), Washington, DC: IEEE Computer Society Press.

Khan, V., Markopoulos, P., Eggen, B., IJsselsteijn, W., de Ruyter, B. (2009). Reconexp: a way to reduce the data loss of the experiencing sampling method. In *Proceedings of MobileHCI '09*, (pp. 471-476), New York: ACM Press.

Khan, V., Markopoulos, P., Eggen, B., and Metaxas, G. (2010). Evaluation of a pervasive awareness system designed for busy parents. *Pervasive Mobile Computing,* 6 (5), 537-558.

Khan, V., Metaxas, G., Markopoulos, P. (2008). Pervasive awareness. In *Proceedings of the 10th international Conference on Human Computer interaction with Mobile Devices and Services, MobileHCI '08,* (pp. 519-521), New York: ACM Press.

Kindberg, T., Fox, A. (2002). System Software for Ubiquitous Computing. *IEEE Pervasive Computing,* 1(1), 70–81.

Klein, J. (1956). *The Study of Groups*. London: Routledge and Kegan Paul.

Kruchten, P. (1995). The 4+1 View Model of Architecture. *IEEE Software,* 12 (6), 42-50.

Ko, A., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrence, J., Lieberman, H., Myers, B., Rosson, M.B., Rothermel, G., Shaw, M., and Wiedenbeck, S. (2010). The State of the Art in End-User Software Engineering. *ACM Computing Surveys* , to appear.

Lederer, S., Mankoff, J., Dey, A.K. (2003). Who Wants to Know What When? Privacy Preference Determinants in Ubiquitous Computing. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems,* (pp. 724-725), New York: ACM Press.

Leichti, O., Ichikawa, T. (2000). A digital photography framework enabling affective awareness in home communication. *Personal Technologies*, 4 (1), 6-24.

Lim, B. Y., Dey, A. K., Avrahami, D. (2009). Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '09,* (pp. 2119-2128), New York: ACM Press.

Lloyd, J. (2008). Speaking at the 30th Anniversary Hitchhiker's recording at Douglas Adams Memorial Lecture on Wednesday 12th March 2008 at The Royal Geographical Society.

Markopoulos, P. (2005). Designing ubiquitous computer human interaction: the case of the connected family. In Isomaki, H., Pirhonen, A., Roast, C., Saariluoma, P. (Eds.), *Future Interaction Design*. (pp. 125-150), Heidelberg, Germany: Springer Verlag.

Markopoulos, P. (2009). Awareness systems and the role of social intelligence. *AI & SOCIETY*, 24, (1), 115-122.

Markopoulos, P., Romero, N., van Baren, J., IJsselsteijn W., de Ruyter, B, Farshchian, B. (2004). Keeping in touch with the family: home and away with the ASTRA awareness system. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems,* (pp. 1351-1354), New York: ACM Press.

Markopoulos, P., Rowson, J., Johnson, P. (1997). Composition and Synthesis with a Formal Interactor Model. *Interacting with Computers*, 9 (2), 197-223.

Markopoulos, P., Bongers, B., Van Alphen, E., Dekker, J., Van Dijk, W., Messemaker, S., Van Poppel, J., Van der Vlist, B., Volman, D. and Van Wanrooij, G. (2006). The PhotoMirror appliance: affective awareness in the hallway. *Personal Ubiquitous Computing,* 10 (2), 128-135.

Mavrommati, I., Kameas, A., Markopoulos, P. (2004).An editing tool that manages device associations. *Personal and Ubiquitous Computing,* 8 (3-4), 255-263.

Mehandjiev, N., Sutcliffe, A.G. and Lee, D. (2004). Organisational views of end user development. In Lieberman, H., Paternò, F., and Wulf, V. (Eds.*),* *End User Development: Empowering People to Flexibly Employ Advanced Information and Communication Technology*, (pp. 371-399), Dordrecht: Kluwer Academic Publishers.

Metaxas, G., and Markopoulos, P. (2007). 'Aware of What?' A Formal Model of Awareness Systems That Extends the Focus-Nimbus Model. In *Proceedings of Engineering Interactive Systems 2007, LNCS 4940*, (pp. 429-446), Heidelberg, Germany: Springer Verlag.

Metaxas, G., Markopoulos, P. (2009). Abstractions of Awareness. In Markopoulos, P., de Ruyter, B., Mackay, W. (Eds.*), Awareness Systems: Advances in Theory, Methodology and Design, Human-Computer Interaction Series*, (pp. 149-172), Heidelberg, Germany: Springer Verlag.

Metaxas, G., Markopoulos, P., Aarts, E. (2009). Amelie: a recombinant computing framework for ambient awareness. In *Proceedings of AMI'09, 3rd European Conference on Ambient Intelligence*, (pp. 88-100), Heidelberg, Germany: Springer Verlag.

Metaxas, G., Metin, B., Schneider, J., Markopoulos, P., De Ruyter, B. (2007). Daily Activities Diarist: Supporting Aging in Place with Semantically Enriched Narratives. In *Proceedings of INTERACT 2007*, (pp. 390-403), Heidelberg, Germany: Springer Verlag.

Miller, T., Stasko, J. (2002). Artistically Conveying Information with the InfoCanvas. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI2002),* (pp. 43-50), New York: ACM Press.

Modica, S. and Rustichini, A. (1994). Awareness and Partitional Information Structures. *Theory and Decision*, 37 (1), 107-124.

Mynatt, E. D., Rowan, J., Craighill, S., and Jacobs, A. (2001). Digital family portraits: supporting peace of mind for extended family members. In *Proceedings of the Conference on Human Factors in Computing Systems*, *CHI '01*, (pp. 333-340), New York: ACM Press.

Nardi, B. (1995). *A Small Matter of Programming*. MIT Press.

Neustaedter, C. and Greenberg, S. (2003). The design of a context-aware home media space for balancing privacy and awareness. *Report 2003-722-25*, Department of Computer Science¸ University of Calgary.

Ormerod, T.C. (2000). Mechanisms and strategies for rephrasing. In Schaeken, W. et al. (Eds.), *Deductive Reasoning and Strategies*, 131–151.

Oulasvirta, A., Petit, R., Raento, M., Tiitta, S. (2007). Interpreting and acting on mobile awareness cues. *International Journal of Human Computer Interaction,* Taylor & Francis, 22 (1&2), 97-135.

Oulasvirta, A. (2004). Finding meaningful uses for context-aware technologies: the humanistic research strategy. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '04,* (pp. 247-254), New York: ACM Press.

Palen, L. and Dourish, P. (2003). Unpacking "privacy" for a networked world. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '03,* (pp. 129-136), New York: ACM Press.

Pane, J.F., Myers, B.A. (2000). Tabular and textual methods for selecting objects from a group. In *Proceedings of VL 2000: IEEE International Symposium on Visual Languages,* (pp. 157–164), Washington, DC: IEEE Computer Society Press.

Paul, P., Evans, R., Cahill, L., and Tipper, N. (2000). Natural language generation in the MILE system. In *Proceedings of the IMPACTS in NLG Workshop,* (pp. 33-42), Schloss Dagstuhl, Germany.

Pederson E. R. and Sokoler, T. (1997). AROMA: Abstract representation of presence supporting mutual awareness. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '97,* (pp. 51-58), New York: ACM Press.

Petre, M. (1995). Why looking isn't always seeing: readership skills and graphical programming. *Communications of ACM,* 38 (6), 33-44.

Evans, R., P. Piwek, L. Cahill and N. Tipper (2008). Natural Language Processing in CLIME, a Multilingual Legal Advisory System. Journal of Natural Language Engineering, 14(1):101-132.

Pousman, Z., Stasko, J. (2006). A taxonomy of ambient information systems: four patterns of design. In *Proceedings of the Working Conference on Advanced Visual interfaces (AVI '06),* (pp. 67-74), New York: ACM Press.

Power, R., Scott, D., and Evans, R. (1998). What You See Is What You Meant: direct knowledge editing with natural language feedback. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98),* (pp. 180–197), Wiley, Chichester/London/New York.

Price, B.A., Adam, K., Nuseibeh, B., (2005). Keeping ubiquitous computing to yourself: A practical model for user control of privacy. *International Journal of Human-Computer Studies*, 63 (1-2), 228-253.

Prinz, W. (1999). NESSIE: An Awareness Environment for Cooperative Settings. In *Proceedings of the 1999 Sixth European Conference on Computer Supported Cooperative Work, ECSCW'99*, (pp. 391-410), Milan: Kluwer Academic Publishers.

Repenning, A. (1993). *Agentsheets: A Tool for Building Domain-Oriented Dynamic, Visual Environments*. Doctoral dissertation, University of Colorado, Boulder.

Rodden, T. (1996). Populating the Application: A Model of Awareness for Cooperative Applications. In *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work, CSCW '96*, (pp. 87-96), New York: ACM Press.

Romero, N., Markopoulos, P. (2005). Common Ground to Analyse Privacy Negotiation in Awareness Systems. In Costabile, M.F., Paternó, F. (Eds.). *Human-Computer Interaction – INTERACT 2005*, *LNCS 3585*, (pp. 1006-1009), Heidelberg, Germany: Springer Verlag.

Romero, N. and Markopoulos, P. (2009). Grounding interpersonal privacy in mediated settings. In *Proceedings of the ACM 2009 international Conference on Supporting Group Work (GROUP '09)*, (pp. 263-272), New York: ACM Press.

Romero, N., Markopoulos, P., van Baren, J., de Ruyter, B., IJsselsteijn, W., Farshchian, B. (2007). Connecting the Family with Awareness Systems. *Personal and Ubiquitous Computing,* 11 (4), 299-312.

Root, R.W. (1988) Design of a multi-media vehicle for social browsing. In *Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work, CSCW '98*, (pp. 25-38), New York: ACM Press.

Salber, D., Dey, A. K., and Abowd, G. D. (1999). The context toolkit: aiding the development of context-enabled applications. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI '99,* (pp. 434-441), New York: ACM Press.

Schmidt, A. (2000). Implicit Human Computer Interaction Through Context. *Personal Technologies*, 4(2-3), 191-199.

Schmidt, K. (2002). The Problem with 'Awareness' Introductory Remarks on 'Awareness in CSCW'. *Computer Supported Collaborative Work*, 11(34), 285-298.

Simone, C., Bandini, S. (2002). Integrating Awareness in Cooperative Applications through the Reaction-Diffusion Metaphor. *Computer Supported Cooperative Work: The Journal of Collaborative Computing,* 11(3–4), 495–530.

Sloutsky, V.M. and Goldvarg, Y. (1999). Effects of externalization on representation of indeterminate problems. In Hahn, M. and Stones, S. (Eds.). *Proceedings of 21st Annual Conference of Cognitive Science Society*, (pp. 695–700), Hillsdale: Erlbaum.

Sloutsky, V.M. and Johnson-Laird, P.N. (1999). Problem representations and illusions in reasoning. In Hahn, M. and Stones, S. (Eds.). *Proceedings of 21st Annual Conference of Cognitive Science Society*, (pp. 701–705), Hillsdale: Erlbaum.

Sohlenkamp, M., Fuchs, L., Genau, A. (1997). Awareness and Cooperative Work: "The POLITeam Approach". In *Proceedings of 30th Annual Hawaii International Conference on System Sciences, HICSS'30,* (pp. 549-558), Washington, DC: IEEE Computer Society Press.

Spivey, M. (1992). *The Z Notation, 2nd ed*. Prentice Hall International: Hemel Hempstead, England.

Stefik, M. J., Bobrow, D. G., Foster, G., Lanning, S., Tatar, D. (1987). WYSIWIS revised: early experiences with multi-user interfaces. *ACM Transactions on Office Information Systems,* 5 (2), 147-167.

Strong, R., Gaver, B. (1996). Feather, Scent and Shaker: Supporting Simple Intimacy. In *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work, CSCW '96*, (pp. 29-30), New York: ACM Press.

Tollmar, K., Sandor, O., and Schömer, A. (1996). Supporting social awareness @ work design and experience. In *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work, CSCW '96*, (pp. 298-307), New York: ACM Press.

Truong, K., Huang, E., Abowd, G. (2004). CAMP: a magnetic poetry interface for end-user programming of capture applications for the home. In *Proceedings of international conference on ubiquitous computing, UbiComp 2004*, (pp. 143–160), New York: ACM Press.

Voida, A., Voida, S., Greenberg, S., and He, H. A. (2008). Asymmetry in media spaces. In *Proceedings of the ACM 2008 Conference on Computer Supported Cooperative Work, CSCW '08,* (pp. 313-322), New York: ACM Press.

Weiser, M., Brown, J.S. (1996). The coming age of calm technology. http://www.ubiq.com/weiser/calmtech/calmtech.htm. Retrieved 15/06/2008.

Wright, D. (2005). The dark side of ambient intelligence. *The Journal of Policy, Regulation and Strategy for Telecommunications*, 7(6), 33-51.

Publications

Journals

Metaxas, G., Markopoulos, P., Aarts, E.H.L (to appear). Modelling social translucency in mediated environments. *Festschrift für Norbert Streitz*. In Press.

Khan, V., Markopoulos, P., Eggen, B., and Metaxas, G. (2010). Evaluation of a pervasive awareness system designed for busy parents. *Pervasive Mobile Computing,* 6 (5), 537-558.

Proceedings

Metaxas, G., Markopoulos, P., Aarts, E. (2009). Amelie: a recombinant computing framework for ambient awareness. In *Proceedings of AMI'09, 3rd European Conference on Ambient Intelligence*, (pp. 88-100), Heidelberg, Germany: Springer Verlag.

Metaxas, G., and Markopoulos, P. (2007). 'Aware of What?' A Formal Model of Awareness Systems That Extends the Focus-Nimbus Model. In *Proceedings of Engineering Interactive Systems 2007, LNCS 4940*, (pp. 429-446), Heidelberg, Germany: Springer Verlag.

Metaxas, G., Metin, B., Schneider, J., Markopoulos, P., De Ruyter, B. (2007). Daily Activities Diarist: Supporting Aging in Place with Semantically Enriched Narratives. In *Proceedings of INTERACT 2007,* (pp. 390-403), Heidelberg, Germany: Springer Verlag.

Khan, V., Metaxas, G., Markopoulos, P. (2008). Pervasive awareness. In *Proceedings of the 10th international Conference on Human Computer interaction with Mobile Devices and Services, MobileHCI '08*, (pp. 519-521), New York: ACM Press.

Book chapters

Metaxas, G., Markopoulos, P. (2009). Abstractions of Awareness. In Markopoulos, P., de Ruyter, B., Mackay, W. (Eds*), Awareness Systems: Advances in Theory, Methodology and Design, Human-Computer Interaction Series*, (pp. 149-172), Heidelberg, Germany: Springer Verlag.

Metaxas G., Markopoulos P. (2009). Awareness of Daily Life Activities. In Markopoulos, P., de Ruyter, B., Mackay, W. (Eds*), Awareness Systems: Advances in Theory, Methodology and Design, Human-Computer Interaction Series*, (pp. 351-365), Heidelberg: Springer Verlag.

Summary

End User Programming of Awareness Systems
Addressing cognitive and social challenges for interaction with aware environments

The thesis is put forward that social intelligence in awareness systems emerges from end-users themselves through the mechanisms that support them in the development and maintenance of such systems. For this intelligence to emerge three challenges have to be addressed, namely the challenge of appropriate awareness abstractions, the challenge of supportive interactive tools, and the challenge of infrastructure.

The thesis argues that in order to advance towards social intelligent awareness systems, we should be able to interpret and predict the success or failure of such systems in relationship to their communicational objectives and their implications for the social interactions they support. The FN-AAR (Focus-Nimbus Aspects Attributes Resources) model is introduced as a formal model which by capturing the general characteristics of the awareness-systems domain allows predictions about socially salient patterns pertaining to human communication and brings clarity to the discussion around relevant concepts such as social translucency, symmetry, and deception.

The thesis recognizes that harnessing the benefits of context awareness can be problematic for end-users and other affected individuals, who may not always be able to anticipate, understand or appreciate system function, and who may so feel their own sense of autonomy and privacy threatened. It introduces a set of tools and mechanisms that support end-user control, system intelligibility and accountability. This is achieved by minimizing the cognitive effort needed to handle the increased complexity of such systems and by enhancing the ability of people to configure and maintain intelligent environments. We show how these

tools and mechanisms empower end-users to answer questions such as “how does the system behave”, “why is something happening”, “how would the system behave in response to a change in context”, and “how can the system’s behaviour be altered” to achieve intelligibility, accountability, and end-user control.

Finally, the thesis argues that awareness applications overall can not be examined as static configurations of services and functions, and that they should be seen as the results of both implicit and explicit interaction with the user. Amelie is introduced as a supportive framework for the development of context-aware applications that encourages the design of the interactive mechanisms through which end-users can control, direct and advance such systems dynamically throughout their deployment. Following the recombinant computing approach, Amelie addresses the implications of infrastructure design decisions on user experience, while by adopting the premises of the FN-AAR model Amelie supports the direct implementation of systems that allow end-users to meet social needs and to practice extant social skills.

Biography

Georgios Metaxas was born in Kozani, Greece on March 30, 1977. After graduating from the department of Computer Science of the University of Ioannina in Greece he worked for several years as a software engineer at Runet Software, a company that develops structural design software for civil engineers and architects. His work there stimulated his interest in the field of Human Computer Interaction, which he pursued further in the Netherlands, where he graduated from the post Masters program of User System Interaction (USI) in the Eindhoven University of Technology. As part of his final project at the USI he had the opportunity to work in Philips research in the field of awareness systems, focusing on supporting elderly living alone to communicate with their social intimates. His further involvement in a doctoral dissertation in the field of awareness systems was a natural consequence.

Acknowledgements

I owe my deepest gratitude to prof. Dr. Emile Aarts and prof. Dr. Panos Markopoulos for their encouragement, guidance and support throughout this project and for their supervision that made it possible to understand and develop the subject.

I would like to thank the reading committee of the thesis for their time and effort in providing constructive comments, and express my appreciation especially to prof. Dr. Alan Dix for the fruitful and inspiring conversations that helped shape the outcome of this project.

Acknowledgement is also due to the Astra consortium, funded by the European Community under the "Information Society Technologies" programme, for the academic and financial support without which this thesis would not have been possible.

I wish to express my gratitude to my colleagues in the Department of Industrial Design at Eindhoven University of Technology, who provided a dynamic research environment, rewarding philosophic discussions, and courageous company. I should however acknowledge especially the help of Nikos Batalas, who carried out the unbearable task of proof reading this thesis, and of Dr. Vasilios Javed Khan, who provided precious feedback while this work was under development.

I would like to thank my parents Charalampos and Margarita and my brother Christos for supporting me morally and materially throughout my studies, and even more I would like to thank my true motivating power and most influencing instructor and friend Dr. Constantinos Georgiadis.

Finally, special credits go to Dr. Olha Bondarenko for her strong moral support and academic assistance throughout this project. To my soul mate and companion, Olha, who tolerated my eccentricity and my peculiarities up to the submission of this thesis, my gratitude can not be described in words.