LONG PAPER

Modelling social translucency in mediated environments

**Georgios Metaxas · Panos Markopoulos ·
Emile H. L. Aarts**

Abstract This paper 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 social translucence, an essential requirement for the social embedding and acceptance of such technologies. FN-AAR abstracts away from implementation concerns, modelling the information shared and the information that is observable by actors. The model allows describing and clarifying fine nuances regarding the concept of social translucence lending clarity to earlier discussions. It is argued that building systems that support this conceptual model will allow their users to specify and configure the disclosure and display of information in terms meaningful to them rather and relevant to their concerns.

Keywords Ambient intelligence · Formal models ·
Awareness systems · Focus-nimbus · Social translucence

1 Introduction

Research in Ambient Intelligence and the related visions of Pervasive Computing and Ubiquitous Computing has, in its

G. Metaxas (✉) · P. Markopoulos · E. H. L. Aarts
Industrial Design, Eindhoven University of Technology,
Eindhoven, The Netherlands
e-mail: g.metaxas@tue.nl

P. Markopoulos
e-mail: p.markopoulos@tue.nl

E. H. L. Aarts
e-mail: emile.aarts@philips.com

majority, focused on two key elements that relate to digital systems and their integration in the environment [1]:

- *Embedding* of large numbers of networked devices into the physical environment.
- *Context-Awareness*, i.e. devices recognize you and your situational context.

Related technical advances, regarding the emphasis of related design and human–computer interaction research on the embedding of Ambient Intelligence in the social context of users, signal the need for increased efforts into the direction of adjusting system behaviour to its users, corresponding to the next two key characteristics of Ambient Intelligence [1]:

- *Personalization*, where systems can be tailored to the needs of a user.
- *Adaptivity*, where systems change in response to user activity.

Green [17] called for a 'people-driven' ambient intelligence that responds to personal tastes, habits and needs, going beyond 'mass customization' to a 'deep customization' where technology constantly evolves through interaction with a user. Such a deep customization, she argues, requires the development of open systems that will allow users themselves to define what they want and how they want it.

A key technology to fulfil Green's vision is what has been called 'end-user development' or 'end-user programming', where creative tasks traditionally handled by professional developers are handed over to non-professional programmers. The feasibility of end-user development seems to depend upon the development of domain-specific abstractions that provide appropriate mental models for users, thus bridging the abstraction gap

between a problem conception and its 'programming' solution [26].

The research reported hereby is concerned with the development of domain-specific abstractions for systems supporting awareness between individuals and groups. This class of systems, sometimes referred to as 'awareness systems' ([22], addresses typical scenarios for Ambient Intelligence that rely on context sensing to notify connected others regarding one's whereabouts, activities or availability, e.g. see D-Me scenario in [21].

In seeking to provide abstractions of awareness systems, it is necessary to consider what concerns are most relevant to their users. The model presented in this paper abstracts away from technical issues regarding context sensing [10] or the design of ambient information displays [28]. Rather, awareness systems are considered in terms of information content and the sharing of this content between connected parties. This enables the specification of concepts that are meaningful and relevant from a user perspective and focusing on the social embedding of technology. Social embedding of ambient intelligence technology means in the current context that it should be possible for individuals to act in a socially adept and intelligent manner even when part of their interactions are mediated or are affected by technology [22].

Erickson [15] points out how face to face interactions are governed by social norms, which in turn are supported by cues in the environment and cues provided by others, which allow individuals and groups to act in a socially intelligent manner. In order to transfer related social skills to technology-mediated interaction, a key requirement identified by Erickson [15] is that of *social translucence; socially translucent systems provide perceptually based social cues, which afford awareness and accountability*. In other words, not by just making information about one's actions observable, but by making this very fact observable to the persons concerned, makes both parties accountable in as far as they should apply ensuing social norms.

So far, social translucence has been discussed informally with reference to several examples relating to social interactions in the physical and the virtual worlds. This paper aims to lend some clarity to such definitions and examine this concept in more depth by modelling this property in mathematically. Following the line of reasoning by Erickson [15], it is argued that awareness systems supporting this model and related operations will allow users to directly express their needs for awareness in terms meaningful to them, which allow related social behaviours to unfold.

The remainder of the paper is structured as follows. First, awareness and awareness systems are discussed. Then, an established abstraction for awareness systems is introduced called the focus-nimbus model, a development

of which is the FN-AAR model discussed here and introduced in [24]. Then, the discussion returns to the concept of social translucence and its representation in terms of the FN-AAR model. The paper ends by discussing potential applications of the model and plans for future work.

2 Awareness and awareness systems research

In the domain of computer-supported cooperative work where awareness systems were first studied, awareness has been defined as 'an understanding of activities of others that provides a context for your own activities' [11]. In a more social context, interpersonal awareness 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.

This awareness can be supported by a broad range of systems. This includes sustained audio–video links connecting communities, often discussed as media spaces [3]. It may include buddy lists or contact lists enhanced with status information [27] and even visualizations of connected communities (also called 'social proxies') [14, 18]. In the domestic domain, the ASTRA prototype [23] studied awareness for the extended family and demonstrated that such awareness can enhance feelings of connectedness and can prompt rather than replace direct communications. The CareNet project [8] 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 [25].

Theoretical discussions motivating the design of awareness systems gravitate towards the phenomena surrounding the social aspects of using awareness. Examples include the ASTRA project [32] that examined the affective benefits and costs of using awareness systems and investigation of mobile awareness cues by Oulasvirta et al. [27], who examined how social inferences can be made through the availability of awareness information. Awareness brings about accountability, which may not always be desirable, compromising one's autonomy [7] and compromising an individual's ability to manage their own privacy borders [9, 19, 20, 29, 31] or even to achieve politeness by means of equivocation, a practice that is very common in daily face to face communication [2].

A common thread in these discussions is an essential need of individuals to manage for self-presentation and the difficulty of managing intersubjectivity in a mediated environment. Adequate control of system behaviour needs therefore to include representations of awareness-related information, but also to represent explicitly cues regarding the very presentation and sharing of this information.

3 Modelling awareness and awareness systems

The most influential mathematical conception of awareness that abstracts away from information flow or architecture issues and focuses on the communicational aspects of awareness is the *focus-nimbus* model by Benford and Fahlen [4], and Benford et al. [5, 6]. This is a spatial model of group interaction, which relies on two key abstractions for modelling levels of mutual awareness within a virtual environment.

- *Focus* represents a subspace within which a person focuses her attention. The more an object is within a person's focus, the more aware that person is of it.
- *Nimbus,* on the other hand, represents a subspace across which a person makes their activity available to others. The more an object is within a person's nimbus, the more aware it is of that person.

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.

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

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 on $j(f_{ik})$ and the nimbus of entity $j(n_{jk})$ at i.

Rodden [30] 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 with objects that are characterizing user's nimbi and foci. By estimating the awareness overlap for two users, one can evaluate the strength of awareness between two users,
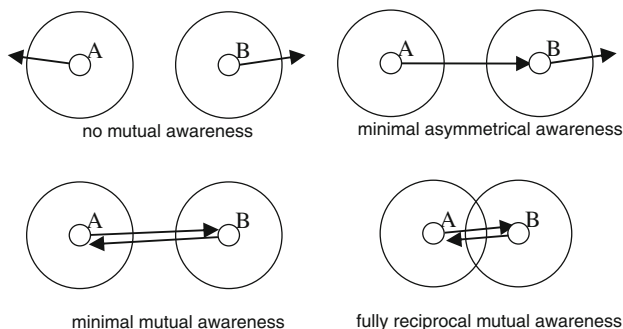

Fig. 1 Some of the discrete awareness modes (4 of 16 arrangements)

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 of empirical investigation [30]. Figure 1 shows some of the different modes of awareness between two users, when a discrete representation of awareness is considered following Rodden's focus-nimbus model.

4 Focus-nimbus-aspect-attributes-resources model (FN-AAR)

4.1 Overview

Where the original focus-nimbus model describes **how much aware** two entities are about 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 is 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 and Anna, and their young daughter Doty, use an awareness system to share with each other their 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 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 is pushing 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 corresponding entities. The family above can be thought as a community, and their house could be seen as an agent.

Aspects are any characteristics that refer to an entity's state. In our scenario 'Anna wants to be aware of John's availability'; thus, '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', 'activity', 'aspirations', or even 'focus' and 'nimbus'.

Attributes are place holders for the information exchanged between *entities.* 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 *John's availability* 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* with a way of rendering (displaying) one or more attributes about this aspect. In any situation, an entity might employ one or more resources to serve its 'interest' about certain aspects of other entities. In our example, "Anna plans to render the attributes that John provides to her about his availability by highlighting an appropriate icon on her 'aware-watch'".

Focus is also dynamic. In the example above, *Anna* assigns her watch to display *John's availability* when she presses a small button. In the proposed 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 some attributes about an aspect using a *resource*. In the above scenario, a possible observable item could be 'the highlighting of the busy icon on my aware-watch". *An observable item is the product of rendering one ore more attributes about an aspect using a specific resource.*

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'. Systemwise 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 that neither the *availability-plug-in* nor the *aware-watch* implies necessarily John's *availability*

(the plug-in may be imprecise) or that, Anna is indeed aware of it. However, one 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, *John* can choose not to let *Anna* know his *availability*, thus *John* lets *Anna* become aware of the situation by manipulating his *nimbus*.

4.2 Observable items

'John is sitting on his office reading an article. On his desk, a small lamp illuminates, indicating that Anna is currently at home.'

In the situation above, the illuminating lamp is an *observable item* that indicates to *John* whether *Anna* is at home or not. The lamp is available for observation, and it is possible (in principle) for *John* to perceive it (*John's* lamp may be switched on whether he is looking at it or not). The term *observable* does not imply a specific modality; information could be displayed in other modalities (auditory, tactile, etc.).

In any situation, there is a set of observable *items* that a given entity can observe. 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:

$$\forall i, j : Entity; \ a_{ij} : RealSituation \rightarrow \mathbb{F} \ ObservableItem;$$

Real situation is an abstraction used to encapsulate the dynamic nature of the universe to which awareness refers. The exact semantics of a_{ij} will be shaped later on based on the notions of focus and nimbus. For convenience, a_{ij}^r is used to denote $a_{ij}(r)$.

As an example of an *observable item*, a function can be considered that returns an *observable item* (light illumination):

$$lightIllumination : Lamp \times Switch \rightarrow ObservableItem;$$

It is not necessary to define *light illumination* in detail; one can imagine that different types of switches can be provided, manual or automatic, with continuous or discrete domains. As an example, *light illumination (lamp1, on)* could represent an observable item that originates from switching on *lamp1*.

In the aforementioned scenario, it can be stated that

$$a_{John, Anna}^r = \{ lightIllumination \, (lamp1, \, on) \}$$

That is, the *awareness* of *John* about *Anna* in a situation(r) is a set that includes one observable item that indicates *Anna's location* by switching lamp1 on.

4.3 Attributes, attribute providers and nimbus

The FN-AAR model sets out to address the question 'what is an entity x aware of regarding entity y'. For that, it is necessary to address the question 'what is entity y exposing to entity x', which amounts to the nimbus of this entity.

First, in any situation, an entity's state (as it is exposed to other entities) holds information about a wide range of aspects. The scheme '*attribute*' *is used* to describe a piece of information ('value') about an aspect ('aspect').

```
┌─ Attribute ──────────────────
│ aspect      : Aspect;
│ value       : Data;
│
└──────────────────────────────
```

For convenience, the idiom *(a: v)* is used to denote '$\langle aspect \rightsquigarrow a, value \rightsquigarrow v\rangle$'; i.e. the idiom *(a: v)* denotes an attribute about aspect a with value v.

Attributes were defined as place holders for information exchanged between *entities*. An entity's nimbus is populated with attribute providers; i.e. functions that given a situation return an attribute and the set of entities that this attribute is made available to. An attribute provider may return different attributes to different entities depending on the situation:

$$AttributeProvider ::= RealSituation$$
$$\rightarrow (Attribute \times \mathbb{F}\, Enitity)$$

For an instance of *attribute provider p*, p^r *is used* to denote the attribute that p returns at situation r, and $p^r.e$ to denote the set of entities that p^r is made available to.

For each entity i, it is assumed that $nimbus_i$ includes all the entity's i attribute providers:

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

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

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

One 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 ($a \in n^r_{\text{John,Anna}}$) about his 'availability'. Following the model, *John's* nimbus contains an attribute provider that depending on the situation returns the aforesaid attribute occupied by a value that corresponds to an estimation of his availability.

$$p1 : AttributeProvider; p1 \in nimbus_{John}|\forall r : RealSituation;$$
$$(p1^r.aspect = availability) \wedge (p1^r.value \in \{available, busy\})$$
$$\wedge (sw1^r.e = \{Anna\})$$

Thus, *p1* 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'.

Wrapping up John's nimbus ($nimbus_{John}$)

$nimbus_{John} = \{p1\}$

Using the definition of n_{ij}, it is verified that:

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

4.4 Resources, resource providers and focus

The previous section defined an entity's nimbus in terms of the attributes it makes available to other entities. However, the question 'What is an entity aware of regarding other entities?' is two-fold, requiring knowledge of what is available for observation to an entity, but also of, what is this entity interested in', and more particularly, how the entity can map the available attributes about another entity to observable items.

In the original focus-nimbus model, focus represents a subspace within which an entity focuses its attention; likewise, in the proposed model, it is assumed that an entity has a limited set of resources to represent the available information from other entities. The scheme *resource* is introduced below to define an aspect of interest and a function that transforms the corresponding attributes to an observable item.

```
┌─ Resource ───────────────────
│ aspect : Aspect;
│ render : 𝔽 Attribute →ObservableItem;
│
└──────────────────────────────
```

One's resources may change depending on the situation; consequently, a function-type *resource provider* is defined 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:

$$ResourceProvider ::= RealSituation$$
$$\rightarrow (Resource \times Entity)$$

For a *resource provider* instance p, we use p^r to denote the resource that p returns at the situation r, and $p^r.e$ the entity that p^r is assigned to. The focus space is populated 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 $focus_i$, f_{ij} can be defined 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:

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

Going back to the scenario we introduced earlier, *Anna's* focus on *John* can be elaborated. *Anna* can check *John's availability* by pressing a small button on her 'aware-watch'. Systemwise, *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:

$$p2 : ResourceProvider;\ p2 \in focus_{Anna}|$$
$$\forall r : RealSituation;$$
$$(buttonpressed(r) \land (p2^r.aspect = availability) \land$$
$$(\forall s : \mathbb{F}\,Attribute;\ p2^r.render(s) =$$
$$if\ (\exists p : Attribute; p \in s | p.\,aspect = availability \land p.value$$
$$= available)then$$
$$AvailableIconHighlight\ else\ BusyIconHighlight)$$
$$\land (p2^r.e = John)) \lor$$
$$(\neg buttonpressed(r) \land p2^r = \varnothing)$$

Thus $p2$ is a *resource provider* that when the button at *Anna's* aware-watch is pressed, $p2$ returns a resource, which when provided with an attribute about *availability,* it renders, 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 that when applied to a real situation r, it returns a resource that can render *John's availability.*

4.5 Focus/nimbus negotiation

Let's revisit the 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:

$$\forall i, j : Entity; a_{ij} : RealSituation \rightarrow \mathbb{F}\,ObservableItem;$$

This definition of a_{ij} is weak, since it does not specify the relation between what is available about j and how this is presented to i. This section specifies a_{ij} more strongly as a functional composition of nimbus and focus.

Figure 2 shows the attributes that an entity 'j' makes available to an entity 'i' at a situation 'r' (i.e. *a1, a2, a3*) through n_{ji}^r. The top left shows their projection (A) on the *aspect space,* i.e. the aspects they refer to. For example, the attribute a_1 contains information about the aspect Y, so its projection on the *aspect space* is Y. Also, are shown the resources that i assigns for observing j at r (i.e. *r1, r2*) through f_{ij}^r 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_{ij}^r) is the result of rendering those attributes of n_{ij}^r that project on $A \cap B$ (i.e. *a2,*and *a3*), using those corresponding resources of f_{ij}^r that project on $A \cap B$ (i.e. *r1*); therefore (see bottom of Fig. 2), a_{ij}^r includes the observable item $o1 = r2.render(\{a2,a3\})$.

This negotiation of the reciprocal foci and nimbi between two entities is generalized as follows:

$$a_{ij} ::= RealSituation;$$
$$\forall r : RealSituation;$$
$$a_{ij}(r) = \{v : ObservableItem | (\forall c : Resource; c \in f_{ij}^r \bullet$$
$$v = c.render(\{u : Attribute | (u \in n_{ji}^r)$$
$$\land (u.aspect = c.aspect)\}))\}$$

Returning to the previous example, *Anna's* observable item(s) about *John's* state is the result of rendering the value of *John's availability,* as it is provided to *John* (i.e. $p1^r$) using the resource(s) that Anna assigned for this purpose (i.e. $p2^r$).

$$\forall r : RealSituation;$$
$$a_{Anna, John}^r = \{p2^r.render(\{p1^r\})\};$$

At this point, the definitions so far can be wrapped together in a scheme that describes an awareness system using the notions introduced so far. Above and further on, the following idioms are used: $nimbus_i$ for *nimbus(i)*, $focus_i$ for *focus(i)*, n_{ij} for *n(i, j)*, f_{ij} for *f(i)*, a_{ij} for *a(i, j)*, n_{ij}^r for *n(i, j)(r)*, f_{ij}^r for *f(i, j)(r)*, a_{ij}^r for *a(i, j)(r)*.

5 Modelling social translucency in the FN-AAR model

Erickson et al. [12–14] examine the notion of social translucency and social translucent systems, i.e. *systems that 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
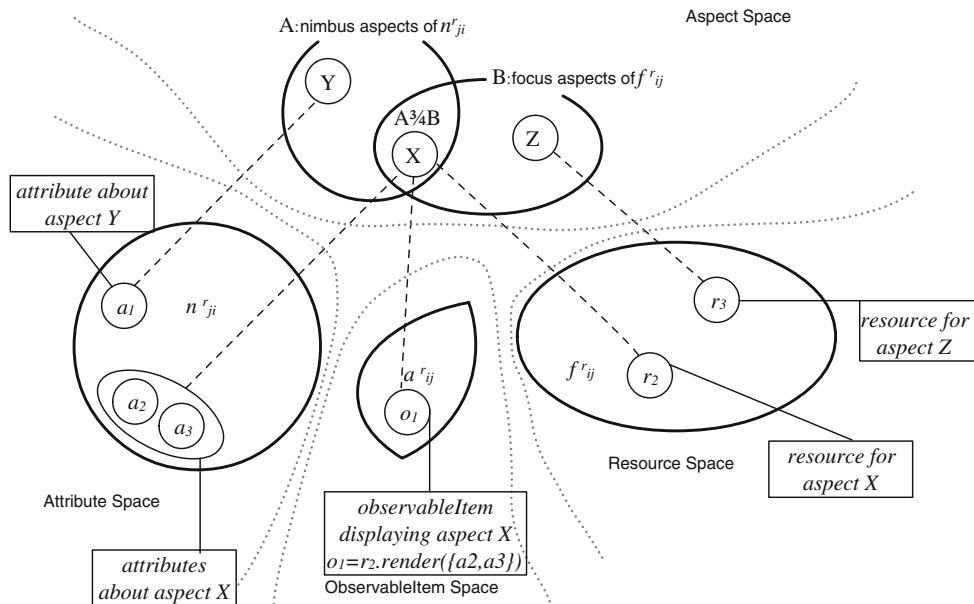
Fig. 2 Illustration of focus-nimbus negotiation between an entity i and some entity j

people's behaviour towards each other are brought to discussion.

Some of these norms can be summarized in statements like:

- Because x knows y's situation, x adjusts its behaviour accordingly.
- Because x knows that y knows x's situation, x adjusts its behaviour accordingly.
- Because x knows that y knows that x knows y's situation, x adjusts its behaviour accordingly.

To reflect on the above statements, let's 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, for example by calling him to arrange going for a walk together. Therefore, 'because **Anna knows** John's situation, she adjusts her behaviour accordingly'.

Now, the system can provide 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 nearby 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, *Anna* may know that the system provides to *John* information about 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'.

5.1 Internal translucency

The first statement, i.e. 'because x knows y's situation, x adjusts its behaviour accordingly', is already captured in the proposed 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 others' situation.

However, a non-trivial statement that is not directly addressed by the fundamental definitions of the model is the following: 'because x knows **its own** situation, x adjusts **its** behaviour accordingly'. For example, *Anna* is in her living room on a Sunday evening. If she would be aware about the bright lighting of the room that allows passers by to gaze at her, she would probably avoid a socially embarrassing situation. In situations like the above unfolding in physical, people are more or less aware of their *nimbi*, but this cannot be expected to be the case when using networked applications. Therefore, one of the properties that might apply in a mediated environment is that of 'internal translucency' or self-awareness.

Internal translucency can be summarized in the statement '*x* is aware of its nimbus'. Thus, an entity is aware of the information that it is making available to others. This statement involves both '*x* focuses on its own nimbus' and '*x* **can** be aware of its own nimbus'. The first signifies that an entity is focusing on the information that it is making available to others. The second signifies that the information about an entity that is available to others is also available to the entity itself. 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 '*x* can be aware of its own nimbus' is equivalent to the statement '*x* exposes to itself its own nimbus to *y*' or, in terms of the proposed model, that 'every attribute in *x*'s nimbus to *y* is also included in *x*'s nimbus to itself':

$\_canBeAwareOfItsNimbusTo\_ : RealSituation$
$\quad \rightarrow (Entity \leftrightarrow Entity)|$
$\forall r : RealSituation; x, y : Entity; \bullet$
$x\ canBeAwareOfItsNimbusTo\ y \Leftrightarrow$
$(x, y) \in \_canBeAwareOfItsNimbusTo\_(r) \Leftrightarrow$
$\forall u : Attribute | u \in n_{x,y}^r \bullet u \in n_{x,x}^r$

The statement '*x* focuses on its own nimbus to *y*' is equivalent to the statement 'there exists at least a resource in *x*'s self-oriented focus that renders each attribute that *x* exposes to *y*':

$\_isFocusingOnItsNimbusTo\_ : RealSituation$
$\quad \rightarrow (Entity \leftrightarrow Entity)|$
$\forall r : RealSituation; x, y : Entity; \bullet$
$x\ isFocusingOnItsNimbusTo\ y \Leftrightarrow$
$(x, y) \in \_isFocusingOnItsNimbusTo\_(r) \Leftrightarrow$
$\forall u : Attribute \left| u \in n_{x,y}^r \bullet \exists v : Resource; v \in f_{x,x}^r \right|$
$\quad v.aspect = u.aspect$

One could consider that *x* is aware of its nimbus to *y*, when both of the aforementioned statements are satisfied. However, since it cannot be assumed a priori that a focus resource presents its corresponding aspect successfully (e.g. due to poor design, an attribute is mal-presented), the relationship 'displays' can be introduced to relate an observable item to the attribute(s) it presents successfully:

$\_displays\_ : ObservableItem \times Attribute;$
$\forall o : ObservableItem;$
$a : Attribute \bullet o\ displays\ a \Leftrightarrow (o, a) \in \_displays\_$

In the trivial case, where a focus resource always presents its corresponding aspect successfully, the above relationship can be defined more strongly:

$let\ A : \mathbb{F}\ Attribute; r : Resource; r\ is\ always\ successful \Rightarrow$
$\forall a : Attribute; (a \in A) \wedge (a.aspect = r.aspect) \bullet$
$\quad (r.render(A), a) \in \_displays\_$

The use of '$\_displays\_$' can clarify the statement '*x* is aware of its own nimbus to *y*', by taking in to account whether 'the observable items that *x* can see indeed display the attributes that *x* makes available to *y*':

$\_isAwareOfItsNimbusTo\_ : RealSituation$
$\quad \rightarrow (Entity \leftrightarrow Entity)|$
$\forall r : RealSituation; x, y : Entity; \bullet$
$x\ isAwareOfItsNimbusTo\ y \Leftrightarrow$
$(x, y) \in \_isAwareOfItsNimbusTo\_(r) \Leftrightarrow$
$\forall u : Attribute \left| u \in n_{x,y}^r \times \exists o : ObservableItem; \right.$
$\quad o \in a_{x,x}^r \left| o\ displays\ u \right.$

Thus, an entity *x* is aware of its nimbus to an entity *y* when every attribute in *x*'s nimbus to *y* is also displayed to *x*, i.e. there exists an observable item (such that *x* is aware of) that displays the attribute.

The following scenario demonstrates the potential of the above formalizations in a real system:

> John and Anna, a happily married couple, use an awareness system to share with each other their daily activities. John configured the system to let Anna know his availability for a telephone communication. For that he used a simple plug-in that detects the activity at his computer which is translated (loosely) to availability. i.e. the more windows are open at John's computer the more busy he appears to be at Anna's side. 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.

In this scenario, John, by displaying on his computer his extracted availability, has engaged a strategy in which he is aware of his nimbus. Moreover, the system could also 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 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").

Therefore, both users and systems can mutually adapt to each other's behaviour to enhance the conjoint performance of the system.

5.2 External translucency

Erickson's statement 'because x knows that y knows x's situation, x adjusts its behaviour accordingly' is used here as a starting point for the concept of external translucency. This statement is broadened to 'because x knows that y knows **x's or someone else's situation**, x adjusts its behaviour accordingly'.

For example, *Anna* and *John* could use 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 whether she is focusing on *Doty;* hence, he can decide whether he also needs to check on their daughter. In other words, 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, external translucency is summarized in the statement 'I am aware of your focus'. Thus, 'x is aware of what y is focusing on x (and possibly other entities)'. This statement involves both 'x focuses on y's focus' and 'x **can** be aware of y's focus'. The first signifies that some of the focus resources of an entity are assigned to display the focus of another entity. The second signifies that the focus (i.e. the focus resources that such entity assigned to render information that other entities make available to it) is made available to those entities. Hence, an entity allows another one to observe how it is observing me (or other entities).

In more detail, the statement 'x can be aware of y's focus on x (or someone else)' is equivalent to the statement 'y exposes to x its focus on x (or someone else)', or that, there exists an attribute that indicates an entity's focus on another one included in its nimbus to such an entity (i.e. an attribute about the aspect 'y's focus on x/someone else'):

An entity y exposes $+ x$ its focus on z

_exposesTo_ItsFocusOn_ : RealSituation

$\quad \to \mathbb{F}\,(Entity \times Entity \times Entity)|$

$\forall\, r : RealSituation;\, x, y, z : Entity; \bullet$

$y\, exposesTo\, x\, ItsFocusOn\, z \Leftrightarrow$

$(x, y, z) \in$ *_exposesTo_ItsFocusOn_$(r) \Leftrightarrow$*

$\exists\, u : Attribute;\, u \in n^r_{y,x}|u.aspect = focus\, y\, on\, z \wedge u.value$

$\quad = f^r_{y,z}$

Hence, in the above definition, it is considered 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 of y on z', that has as value y's focus on z (i.e. f^r_{yz}). 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 $z;$ note that, such slight modifications could be used also as a tool for blurring the exposed focus itself.

The statement 'x focuses on y's focus' can be formalized by claiming the existence of a resource in x's focus that renders another entity's exposed attribute(s) about its own focus on the first entity (or other entities):

An entity x focuses on an enity y focus on z

_isFocusingOnTheFocusOf_On_ : RealSituation

$\quad \to \mathbb{F}\,(Entity \times Entity \times Entity)|$

$\forall\, r : RealSituation;\, x, y, z : Entity; \bullet$

$x\, isFocusingOnTheFocusOf\, y\, On\, z \Leftrightarrow$

$(x, y, z) \in$ *_isFocusingOnTheFocusOf_On_$(r) \Leftrightarrow$*

$\exists\, v : Resource;\, v \in f^r_{xy}|v.aspect = focus\, y\, on\, z$

The statement 'x is aware of y's focus on x' can be formalised similarly to the case of internal translucency:

An entity x is aware of an entitys y focus on z

_isAwareOfTheFocusOf_On_ : RealSituation

$\quad \to \mathbb{F}\,(Entity \times Entity \times Entity)|$

$\forall\, r : RealSituation;\, x, y, z : Entity; \bullet$

$x\, isAwareOfTheFocusOf\, yOn\, z \Leftrightarrow$

$(x, y, z) \in$ *_isAwareOfTheFocusOf_On_$(r) \Leftrightarrow$*

$\exists\, o : ObservableItem;$

$\quad o \in a^r_{x,y}|o\, displays \left\langle aspect \rightsquigarrow focus\, y\, on\, z,\ value \rightsquigarrow f^r_{y,z} \right\rangle$

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.

Using similar notation, a wide range of relevant statements can be formalized, such as 'x is aware of y's focus on everybody in a particular situation', or 'x can be aware of the entities y is focusing on regarding an aspect', or 'x exposes to y its focus as a whole (i.e. the set of its resource providers)', and so on.

To demonstrate the potential of implementing the above in a real system, let's build up the scenario 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 to him

his own availability takes too much space on his desktop. John asked from Anna to expose to him her focus, 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.

6 Discussion

This paper has shown how FN-AAR, a formal model of awareness systems, allows a clear definition of crucial social-related behaviours in human communication, such as deception and social translucency. The FN-AAR model abstracts away from modelling the propagation of awareness information and information flow modelling, as is the case with earlier abstractions of awareness systems, e.g. Simone and Bandini [33] and Fuchs et al. [16]. It advances the original focus-nimbus model in that it is explicit about the object of awareness, i.e. the relationship of the information that an entity can potentially provide about itself to that actually observed by another entity. This is necessary for modelling the social aspects of awareness systems, such as social translucency as shown above and deception (e.g. blurring) as shown elsewhere [24].

FN-AAR can be used both as a conceptual tool and as an analytical model that the research community can use as a foundation for building the next generation of awareness systems. Despite that a formal development process of awareness systems is not adopted by the authors, the model presented here has been implemented in an open programming environment for the implementation of awareness systems that is currently being tested in situ. The environment is built on top of an XML-based language that follows the principles of the FN-AAR model and defines web service interfaces that pursue the notions of attribute providers, resource providers, and entity-specific ontology, as instructed by the model. On the background, the system continuously invokes the entities' foci and nimbi, negotiates the intersection of their exposed and acquired attributes, invokes the corresponding 'renderers', and returns the observable items that describe the entities' reciprocal awareness, allowing at the same time end users to express social behaviours, such as those described in this paper.

The acquired experiences so far with the application of the FN-AAR model in this programming environment (to be reported elsewhere) has shown that it is powerful and flexible enough to support the implementation of a broad range of systems covering mobile domains, domestic awareness systems, context sensing, social networking applications, etc. Further, it provide the means to implement mechanisms for managing one's interactions with one's social network and the flows of information to and from others directly and in terms relevant to how users interact with each other: allowing them to lie, to ensure accountability, to negotiate symmetrical flows, etc., as predicted by the FN-AAR model.

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.

References

1. Aarts, E.: Technological visions of ambient intelligence. In: Aarts, E., Marzano, S. (eds.) The New Everyday, pp. 12–16. 010 Publishers, Rotterdam (2003)
2. Aoki, P., Woodruff, A.: Making space for stories: "ambiguity in the design of personal communication systems". In: Proceedings of CHI 2005, pp. 181–190 (2005)
3. Baecker, R.M.: Readings in Groupware and Computer Supported Cooperative Work: Software to Facilitate Human-Human Collaboration. Morgan Kaufmann Publishers, Massachusetts (1993)
4. Benford, S., Fahlen, L.E.: A spatial model of interaction in large virtual environments. In: Proceedings of ECSCW'93, Milan, pp. 109–124 (1993)
5. Benford, S., Bowers, J., Fahlen, L., Mariani, J., Rodden, T.: Supporting cooperative work in virtual environments. Comput. J. **38**(1), 653–668 (1995)
6. Benford, S., Bullock, A., Cook, N., Harvey, P., Ingram, R., Lee, O.: From rooms to cyberspace: models of interaction in large virtual computer spaces. Interact. Comput. **5**(2), 217–237 (1993)
7. Boyle, M., Greenberg, S.: The language of privacy: learning from video media space analysis and design. ACM Trans. Comput. Hum. Interact. **12**(2), 328–370 (2005)
8. Consolvo, S., Roessler, P., Shelton, B.E.: The CareNet display: lessons learned from an in-home evaluation of an ambient display. In: Proceedings of UbiComp'04, pp. 22–29 (2004)
9. Consolvo, S., Smith, I., Matthews, T., LaMarca, A., Tabert, J., Powledge, P.: Location disclosure to social relations: why, when, and what people want to share. In: Proceedings of the Conference on Human Factors and Computing Systems: CHI 2005, pp. 81–90 (2005)
10. Dey, A.K.: Understanding and using context. Pers. Ubiquit. Comput. **5**(1), 4–7 (2001)
11. Dourish, P., Belloti V.: Awareness and coordination in shared workspaces. In: Proceedings of CHI 92, pp. 117–124. ACM Press, New York (1992)
12. Erickson, T., Kellogg, W.: Social translucence: an approach to designing systems that mesh with social processes. ACM Trans. Comput. Hum. Interact. (TOCHI), **7**(1), 59–83 (2000)
13. Erickson, T., Halverson, C., Kellogg, W.A., Laff, M., Wolf, T.: Social translucence: designing social infrastructures that make collective activity visible. Commun. ACM **45**(4), 40–44 (2002)

14. Erickson, T., Smith, D.N., Kellogg, W.A., Laff, M., Richards, J.T., Bradner, E.: Socially translucent systems: social proxies, persistent conversation, and the design of "babble". In: proceedings of CHI '99, pp. 72–79. ACM, New York (1999)
15. Erickson, T.: 'Social'systems: designing digital systems that support social intelligence. AI Soc. **23**, 147–166 (2009)
16. Fuchs, L., Pankoke-Babatz, U., Prinz, W.: Supporting cooperative awareness with local event mechanisms: the GroupDesk system. In: Proceedings of ECSCW'95, pp. 247–262 (1995)
17. Green, J.: Thinking the future. In: Aarts, E., Marzano, S. (eds.) The New Everyday, pp. 22–27. 010 Publishers, Rotterdam (2003)
18. Greenberg, S., Rounding, M.: The notification collage: posting information to public and personal displays. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 514–521. CHI'01 (Seattle, Washington, United States), ACM, New York (2001)
19. Hong, J.I., Landay, J.A.: An architecture for privacy- sensitive ubiquitous computing. In: Proceedings of Mobisys'04, pp. 177–189. Boston, MA (2004)
20. Iachello, G., Smith, I., Consolvo, S., Chen, M., Abowd, G.: Developing privacy guidelines for social location disclosure applications and services. In: Proceedings of the Symposium on Usable Privacy and Security (SOUPS 2005)
21. ISTAG.: Scenarios for Ambient Intelligence in 2010. Downloaded Feb 2009 from ftp://ftp.cordis.europa.eu/pub/ist/docs/istagscenarios2010.pdf (2001)
22. Markopoulos, P., de Ruyter, B., Mackay, W. (eds.): Awareness Systems: Advances in Theory, Methodology and Design, Human-Computer Interaction Series, pp. 49–72. Springer, Berlin (2009)
23. Markopoulos, P., Romero, N., van Baren, J., IJsselsteijn, W., de Ruyter, B., Farshchian, B.: Keeping in touch with the family: home and away with the ASTRA awareness system. CHI Ext. Abstr. **2004**, 1351–1354 (2004)
24. Metaxas, G., Markopoulos, P.: 'Aware of what?' A formal model of awareness systems that extends the focus-nimbus model. In: Proceedings of Engineering Interactive Systems, LNCS 4940, pp. 429–446 (2007)
25. Metaxas, G., Metin, B., Schneider, J., Markopoulos, P., De Ruyter, B.: Daily activities diarist: supporting aging in place with semantically enriched narratives. In: Proceedings of INTERACT, LNCS, vol. 4663, pp. 390–403. Springer, Berlin (2007)
26. Nardi, B.: A Small Matter of Programming. MIT Press, Cambridge (1995)
27. Oulasvirta, A., Petit, R., Raento, M., Tiitta, S.: Interpreting and acting on mobile awareness cues. IJ Hum. Comput. Interact. **22**(1&2), 97–135(2007)
28. Pousman, Z., Stasko, J.: A taxonomy of ambient information systems: four patterns of design. In: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 67–74. AVI '06 (Venezia, Italy, May 23–26, 2006), ACM, New York
29. Price, B.A., Adam K., Nuseibeh B.: Keeping ubiquitous computing to yourself: a practical model for user control of privacy. Int. J. Hum. Comput. Stud. **63**(1–2), 228–253 (2005)
30. Rodden, T.: Populating the application: a model of awareness for cooperative applications. In: Proceedings of ACM 1996 (CSCW'96), pp. 87–96 (1996)
31. Romero, N., Markopoulos, P.: Common ground to analyse privacy negotiation in awareness systems. In: Costabile, M.F., Paternó F (Eds.) Human-Computer Interaction—INTERACT, LNCS, vol. 3585, pp. 1006–1009. Springer, Berlin (2005)
32. Romero, N., Markopoulos, P., van Baren, J, de Ruyter, B., IJsselsteijn, W., Farshchian, B.: Connecting the family with awareness systems. Pers. Ubiquit. Comput. **11**(4), 299–312 (2007)
33. Simone, C., Bandini, S.: Integrating Awareness in Cooperative Applications through the Reaction-Diffusion Metaphor. Computer Supported Cooperative Work: The Journal of Collaborative Computing, vol. 11, No. 3–4, pp. 495–530. Kluwer, Dordrecht (2002)