Q-Aura: a Quantitative Model for Managing Mutual Awareness of Smart Social Artifacts

Seng Loke, Sea Ling, Maria Indrawan, and Eddie Leung

Abstract—**What if physical artifacts or devices can be aware of each others' physical presence and location, and interact with each other without user intervention or to enable innovative applications? We propose a model for devices to manage awareness of each other extending a spatial model of interaction previously used in virtual environments. While there has been previous work on cooperative artifacts, our model is unique in introducing a quantitative technique. Also, our model is novel in adding to proximity-based interactions among devices the concepts of: (i) *aura collision types* based on relative locations of devices, and (ii) *multiple (adjustable) levels of awareness and concealment measures* so that each device can control how much it wants to be aware of others and how much it wants to be concealed from others. Our model is general and supports awareness of devices in (sufficiently) close physical proximity and the right aura sizes. Such devices' awareness of each other facilitates or triggers interaction, and normally precedes interaction among devices (as in human communication). Our model has numerous applications, from smart soft-toy features to proximity-triggered data exchanges.**

Index Terms—**aura, social artifacts, smart devices, ubiquitous computing**

INTRODUCTION

With the emergence of devices with increasing computational and communication capabilities, there is an opportunity towards what have been called *social devices*, which might, if allowed to, exchange messages autonomously with each other, e.g. synchronizing data or cooperatively working together without users' intervention. Physical proximity might be viewed as what we call an *interaction enabler*, in that physical proximity normally facilitates mutual awareness (as typically for humans), and mutual awareness normally precedes two parties interacting. However, two physical devices, or everyday artifacts, are not usually mutually "aware" of each other even in close physical proximity. Many context-aware systems and applications have surfaced to capture location information. Using such information, location-aware systems and applications allow objects, either humans or devices, to be aware of the proximity of other objects in order to interact with each other and to provide services to each other. Physical proximity, hence, can be determined and used as a trigger or enabler for interaction (e.g., simple exchange of messages) among physical artifacts, but the extent to which proximity can be used to trigger interactions should be determined by the artifacts themselves.

An artifact or device coming close to another device might choose to make itself known to that device or it may not, i.e. there is a need to maintain autonomy and to allow each device to determine the extent to which it wants to employ this interaction enabler. This paper explores this idea of using physical proximity to trigger interactions among physical artifacts (or devices), presenting a computational model where such proximity-based interactions can be controlled by each artifact. Our model, which we call Q-Aura (short for Quantitative Device-Aura), is independent of the underlying networking technology or the underlying service discovery protocol, and we believe is a first quantitative model for prescribing levels of awareness.

The idea of interaction enablers has been successfully demonstrated in the virtual environments using the abstract concepts such as "Aura", "Focus" and "Nimbus" in the spatial model of interaction. Every device, artifact and object wishing to communicate with each other is equipped with these concepts. They provide a conceptual form of presence, awareness and concealment, respectively, to objects in the virtual environment for handling interactions. Our novel contribution in this paper is a substantial refinement of this spatial model of interaction and its adaptation to physical world environments using a quantitative measure of awareness and concealment.

Our approach is two-fold. Firstly, we define different models of aura collision. Aura is a conceptual sub-space that belongs to a device or an object to represent its presence and it is analogous to an object's ambient [2]. Intuitively, the devices can communicate with each other when their auras collide. We define several collision models, each representing how close in proximity the devices are. Only devices following the same collision model will be able to interact. After a collision is detected, the devices can use different levels of awareness (focus) and concealment (nimbus) measures so that each device can control how much it wants to be aware of others and how much it wants to be concealed from others. We envisage this approach will allow devices (and thus, users) to have control over how interactions are to be conducted.

There are numerous applications of proximity-based computing and interactions including new generations of interacting soft-toys and furniture, convenient and automated inter-device interactions in the home (e.g., living room and kitchen) and factories, and inter-device interactions in public spaces (e.g., museums, malls, etc) including interaction between PDAs and museum information services, as well as social mobile and smart mob (http://www.smartmobs.com) applications.

In the rest of this paper, we introduce the Q-Aura model of

Manuscript received August 2008. (Write the date on which you submitted your paper for review.)

Seng Loke is with the Department of Computer Science and Computer Engineering, La Trobe university, Australia (email: s.loke@latrobe.edu.au).

Sea Ling is with the Faculty of Information Technology, Monash University, Australia (email: chris.ling@infotech.monash.edu.au).

Maria Indrawan is the Faculty of Information Technology, Monash University, Australia (email: maria.indrawan@infotech.monash.edu.au).

Eddie Leung is the Faculty of Information Technology, Monash University, Australia (email: eddie.leung@infotech.monash.edu.au).

interaction and provide an overview of the concepts. We then describe our collision models depending on the relative locations of the devices. This is followed by a description of the rules of awareness and concealment, providing mutual awareness for devices. A conceptual architecture to implement the system is proposed with a description of our implemented system. We then discuss related work and conclude.

Q-AURA: A MODEL OF INTERACTION FOR DEVICES

We adapted Benford and Fahlén's spatial model of interaction [1] for the real world. The original model introduces seven key abstract concepts: Space, Objects, Medium, Aura, Focus, Nimbus and Awareness. Our model is based on these abstract concepts modified for use in the real environments, as described below:

- **Space:** The concept of *Space* is defined as an area in which it is possible to determine positions (via some form of location tracking system), expressed specifically as coordinates.
- **Objects:** *Objects* represent the items that are being tracked by the location tracking system (and therefore the people carrying them are tracked as well). These *Objects* need to have some form of processing power or are capable of transmitting information to other *Objects* that have processing power.
- **Medium:** A form of communication *Medium* is needed in the *Space* to allow interactions to take place. A *Medium* typically takes the form of a wireless communication network to enable *Objects* to interact with each other.
- **Aura:** *Aura* is a conceptual sub-space that belongs to an *Object* to represent its presence. An *Aura* can be turned on or off or adjusted like a switch. *Objects* carry their *Auras* around wherever they move and when the same type of *Aura* collides for two *Objects*, the system implementing this model is responsible for providing information to the two *Objects* about how to establish a connection with each other. *Auras* can be typed, and collisions between *Auras* of different types are disregarded and therefore interaction will not take place. Each *Object* can have any number of *Auras* and the *Auras* can be of any shape and size.
- **Focus:** The *Focus* concept is used in conjunction with *Nimbus* to represent the concept of *Awareness*. The focus of an object is, roughly speaking, how much an object wants to be aware of. An object of a larger focus is choosing to be more aware of other objects, and conversely. In real environments, the *Focus* concept can be represented as numerical values or discrete levels. An *Object* A with a higher value or level of *Focus* over the value or level of *Nimbus* of another *Object* B will be aware of *Object* B. This is only a simplified form of an implementation of *Focus*.
- **Nimbus:** The *Nimbus* concept is used in conjunction with *Focus* to obtain the concept of *Awareness*. The nimbus of an object is, roughly speaking, how much an object wants to expose itself, i.e. how much it wants others to be aware of itself. An object of a larger nimbus is choosing to be more visible to other objects, and conversely, an object of a small nimbus is choosing to be concealed from other objects. Similar to the concept of *Focus*, the *Nimbus* concept can be represented as numerical values or discrete levels. An *Object* A with a higher value or level of *Nimbus* over the value or level of *Focus* of another *Object* B, will not allow *Object* B to be aware of *Object* A.
- **Awareness:** *Awareness* (consciousness) or the absence of *Awareness* (unconsciousness) is a concept derived from the concepts of *Focus* and *Nimbus*. As *Aura* is used to determine the possibility of interaction, *Awareness* represents a secondary mechanism to determine if interactions should take place.

Figure 1 illustrates these abstract concepts when used in a real environment. It should be noted that the effective range and area where an *Aura* can be created is dependent on the location tracking system used to create the *Space* as well as the range of the wireless communication network that represents the *Medium*. Whichever the two (*Space* or *Medium*) that has the smaller range represents the effective range and area of the *Aura*.

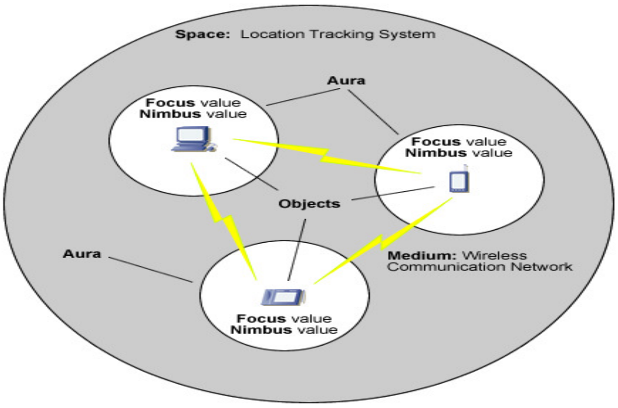

Fig. 1. Model of Interactions

A. Aura Collisions and Location Awareness

Given two objects or devices, interaction can only occur if and when their auras collide. Intuitively, they are made aware of each other's location. We envisage that aura collision can be controlled by the devices own logic (or adjusted via an interface by their users) and propose different models of collision to provide different extents of control needed by the users. As a first step for a proper collision to take place, both auras must agree on the same collision model. In cases where the collision models are different, the default action would be to disregard the collision.

In this section, we define three possible models of collisions (collision types) following the preliminary definition:

Preliminary Definition:

Given an object i, let $Aura_i$ be the set of coordinates within the Aura of i. Let Loc_i be the exact location (coordinates) of object i. Figure 2 shows the definition graphically with the assumption of a circular Aura in a two dimensional space.

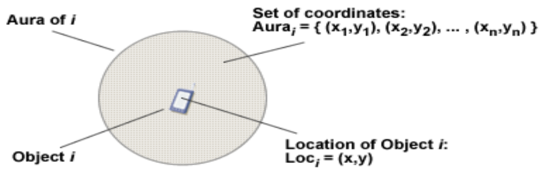

Fig. 2. Preliminary definition for the collision types.

1. Aura Boundary Collision (ABC)

This collision type considers two or more auras colliding when their boundaries intersect.

Definition 1:

For any two objects x and y,
collision is detected when $Aura_x \cap Aura_y \neq \emptyset$

An example of when collision will and will not take place for two different sized circular Auras using this collision model in a two dimensional space is shown in Figure 3.

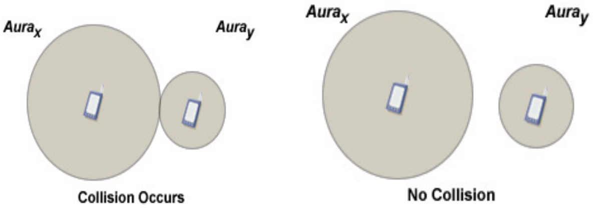

Fig. 3. Examples of collision and no collision using ABC.

ABC is useful for objects that only have controlled or fixed size auras for detecting the presence of other objects without requiring any form of interaction with the user. In some situations and especially in uncontrolled environments, an object may change the aura size to the maximum in order to obtain as many collisions as possible, but objects can still avoid collisions by decreasing their nimbus. Hence, *Focus* and *Nimbus* can be used to control the interactions, described later.

2. Object Proximity Aura Collision (OPAC) - Strict

The Object Proximity Aura Collision type is more restrictive than the Aura Boundary Collision type. OPAC requires objects to be within a certain distance before any collision can take place. There are two versions of this model, a "Strict" version and a "Relaxed" version. The "Strict" version of an Object Proximity Aura Collision is defined as:

Definition 2 (Strict OPAC):

For any two objects x and y, collision is detected
if $Aura_x \cap Aura_y \neq \emptyset$ and $Loc_x \, \square \, Aura_y$ and $Loc_y \, \square \, Aura_x$

This collision model in a two dimensional space is shown in Figure 4.

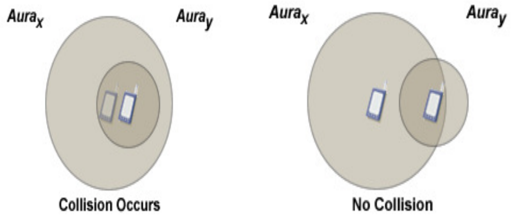

Fig. 4. Examples of collision and no collision using the OPAC - Strict.

In OPAC, to collide requires both objects to be within each other's aura. This model is therefore suitable for use in uncontrolled environments. Another added advantage this model has over the Aura Boundary Collision model is that the implementer of an application can take advantage of the knowledge that if a collision has occurred, it implicitly means that the other object is definitely within a certain specific distance, as opposed to the Aura Boundary Collision model (where additional calculation is needed). For example, if a circular Aura of an object A covers a range of five meters and a collision occurred with another object B when using OPAC, then one can assume that object B is within a distance of not more than five meters from object A. Note that the general Region Connected Calculus[1] (topological relations among regions) apply here.

3. Object Proximity Aura Collision (OPAC) - Relaxed

In the "Strict" version, OPAC requires that both object's location be within each other's aura boundaries. The "Relaxed" version on the other hand, only requires one of the two object's location to be within the other object's aura boundary.

Definition 3 (Relaxed OPAC):

For any two objects x and y, collision is detected
if $Aura_x \cap Aura_y \neq \emptyset$ and ($Loc_x \, \square \, Aura_y$ or $Loc_y \, \square \, Aura_x$).

This is illustrated in Figure 5.

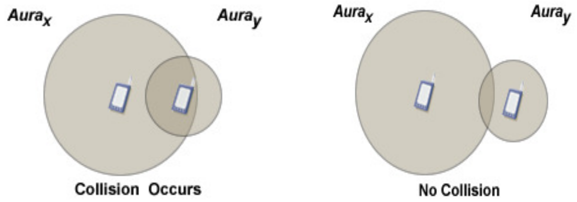

Fig. 5. Examples of collision and no collision using the OPAC - Relaxed.

The "Relaxed" version of the Object Proximity Aura Collision is similar to the Aura Boundary Collision model as there is also the possibility of increasing the size of the aura to maximize detection of other objects when using this model. An advantage of OPAC over Aura Boundary Collision is the implicit distance information collected when a collision occurs.

B. Mutual Awareness and Concealment

Having come to a consensus that a collision has occurred based on a collision type, objects are able to control the interaction by having degrees in the level of awareness between them. The level of awareness is realised by the concepts of focus and nimbus which define how one object's interaction can be redirected towards another object and how much aware it is of one object towards another [1,14]. Basically, the more an object is within your focus; the more aware you are of it and the more an object is within your nimbus, the more aware it is of you.

This means that objects can be made aware of other devices by manipulating the nimbus and focus within the shared space. Each device can control their own focus and nimbus values. By knowing the degrees of focus and nimbus between devices, awareness of devices can be determined. In previous work [11], we made the degrees of focus and nimbus more concrete by assigning distinct numerical values to the focus and the nimbus of an object. The objective is to develop a formal model for electronic devices that are not only able to communicate with each other, but also able to understand and identify its surrounding devices. Intuitively, an object A with a

---

[1] http://en.wikipedia.org/wiki/Region_Connection_Calculus

higher focus value over the nimbus value of another object B will be aware of object B. Alternatively, object A with a higher value of nimbus value over the focus value of B, will **not** allow B to be aware of A. We generalize the above to define awareness and concealment levels:

Definition 4 (Awareness and Concealment)

For every device i, let a_i be its focus (awareness) value and c_i be its nimbus (concealment) value. Given any two devices x and y, the following holds:

- $a_x \geq c_y$ if and only if x is aware of y.
- $a_x < c_y$ if and only if x is not aware y, i.e., y is concealed from x.

Previously (Ling *et. al.*, 2007), we noted that:

1. Given any three devices x, y and z such that x is aware of y and y is aware z, x is aware of z if $c_y \geq c_z$.
2. Given a set of devices $\{x^1, x^2, x^3, \ldots, x^k, \ldots, x^n\}$, for each k ($1 \leq k \leq n-1$), if $a_x^k \geq c_x^{k+1}$ and $c_x^k \geq c_x^{k+1}$, x^k is aware of x^n
3. Given a set of devices $\{x^1, x^2, x^3, \ldots, x^k, \ldots, x^n\}$, for each k ($1 \leq k \leq n-1$), if $a_x^k \geq c_x^{k+1}$ and $c_x^k \geq a_x^k$, x^k is aware of x^n.

Here, we further observe that the approach provides tremendous expressive power, in terms of possible awareness relationships possible among devices, i.e. simply by having each device (in a given collection) set their focus and nimbus values appropriately, particular awareness properties of the collection emerges. We note several possibilities below.

We first define an awareness graph as follows. An awareness graph involving a set of device D is a directed graph (D,E) with vertices D and edges E, where there is an edge from device x to device y, if and only if, x is aware of y. Each device x has a pair of values (a_x, c_x), and by setting this pair of values to appropriate values, either as determined by the *device's own preference* or *by common consensus*, appropriate collective behaviours can be enabled. For example, given a set of devices D, we have the observations:

a. All possible awareness graphs involving D can be represented if each pair of device chooses a suitable value for (a_x, c_x). Namely, for each edge coming into a device x from a device y, we set $a_x \geq c_y$ to obtain an edge from x to y, and $a_y \geq c_x$ for an edge from y to x.

b. Similarly, each set of three devices setting their awareness and concealment values will enable a transitive graph. The implication of the last two results is that as long as each device maintains the conditions in 5 and 6, the awareness relationship will be transitive, regardless of how many devices there are in the environment. Conditions 5 and 6 can be regarded as generalizations of conditions 1 and 2. Maintaining such a condition might be a "social-mile" imposed on each device in order to have a transitively aware society of devices. Such transitive awareness is an example highlighting the scalability of the approach, where as long as pairs of devices adhere to a convention in how they set awareness and concealment values, collective properties can emerge, i.e. pairwise awareness can effectively propagate for free. Other such emergent properties for the collective based on pairwise properties can be enabled in a similar way.

c. Also, if all devices use the same pair of values, and each device sets its awareness equal to concealment, it follows that we have fully-aware devices, i.e., all devices will be aware of each other.

Hence, our model allows different situation of mutual (non-)awareness to be represented. By a priori agreements as to what awareness and concealment values to use, different awareness graphs can be generated.

Similarly, cliques of devices can set their values in such a way that all devices in the same clique have transitive or total awareness, but not also between devices in different cliques.

Also, the range of awareness and concealment values can also be agreed upon a priori; the larger the range for awareness and concealment values, or if real number values are used, fine-grained control and numerous (even infinitely) levels of awareness and concealment can be supported.

An advantage of our approach lies in its simplicity, yet expressive power to generate different desired awareness graphs.

A set of devices might agree beforehand (*a priori*) about appropriate concealment and awareness values to use. Alternatively, adjustments to the level of mutual (non-)awareness can be done by adjusting the levels of awareness and concealment for each device. A device can use its own logic to determine the awareness and concealment values, and these values might be adjusted by the device itself over time. Devices dynamically self-adjusting their focus and nimbus values is important, since, in general, these values might not be preset in advance. For example, a device might vary its awareness (e.g., enlarge its focus) until it discovers particular devices, in effect increasing its "search scope" until particular devices can be detected, or a device might reduce its level of concealment gradually if not yet detected by certain devices.

C. Types of Devices: Capabilities and Resources

Different types of devices' capabilities can yield different kinds of interaction the objects support. Interaction can only happen when devices of the right capabilities have their auras "colliding".

We classify types of devices based on the *capabilities* of an object. A capability is a logical formation of a group of *resources* used to describe a specific physical capability of the object. It determines whether an interaction is possible from the physical point of view. A capability is thus formed by all resources required to support the functionalities of a specific interaction. Resources can include both software and hardware. When resources required for a capability are available, an object is considered as capable of interacting with other objects by using this capability. For instance, when an object is equipped with the resources such as screen, keyboard, appropriate networking devices and MSN messenger, the object is considered as capable of interacting with other objects with the same capability through chatting and transmitting file facilities. Although MSN messenger offers some other facilities/capabilities, only those that have the resource requirements fulfilled will be enabled. Two points should be noted:

- The set of resources that forms a capability is always unique within an object. Hence, in the process of initiating an interaction, one can determine the capability required by identifying the resources needed.

- A capability does not equal a particular functionality. Instead, it represents the functionalities that can be initiated with the support of the resources that forms the capability.

PROOF-OF-CONCEPT PROTOTYPING FOR Q-AURA

The previous section describes object/device interactions made possible by: (i) being location aware, resulting in aura collision; and (ii) being made mutually aware of each other based on the respective awareness and concealment levels. These two levels of modeling above, taken together, provide fine-grained representation of how proximity should (or should not) induce interaction. For example, the following table illustrates *six* possibilities, each depending on the relative values of awareness and concealment and the collision type.

Collision Type	**Possible relative values of awareness and concealment for Device x and Device y**
ABC	$a_x \geq c_y$ *or* $c_x > a_y$
Strict OPAC	$a_x \geq c_y$ *or* $c_x > a_y$
Relaxed	$a_x \geq c_y$ *or* $c_x > a_y$

Each possibility can be used to trigger a different behaviour. For example, consider the top-left cell,

An application might have the following rule:

```
IF Device x and Device y experience a “Strict OPAC”, AND $a_x \geq c_y$, THEN
    x should connect to y and
    x send a “hello” message to y.
```

Such *rules of engagement* can be used to fine-tune interactions. Figure 6 shows the architecture of the system. There are four categories of components, namely, end-users with their handheld or smart devices, the location tracking system, the Aura System and the client-side Aura-enabled applications on the devices.

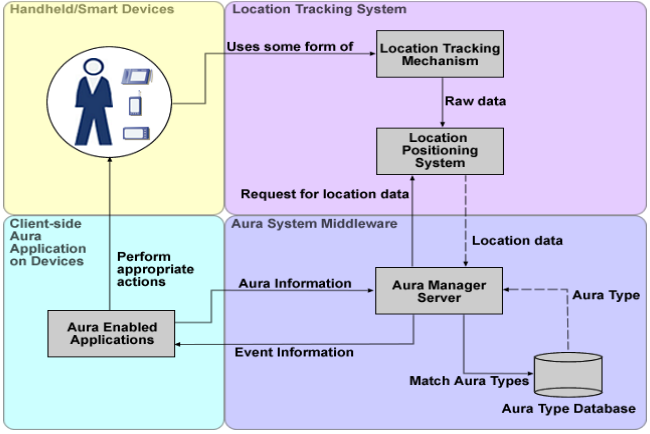

Fig. 6. Architecture of the Q-Aura prototype system.

Descriptions for each of these components are as follows:

1. **Handheld/Smart Devices:** These are the devices used by the end users and constitute the *Objects* in the proposed model. Essentially, any device such as PDAs or smart fridges that has communication and computational processing capability fall into this category.
2. **Location Tracking System:** In the conceptual model, end-user’s handheld or smart devices typically use some form of location tracking mechanism which transmits raw data to a location positioning system which then translates the raw data into proper location coordinates.
3. **Aura System Middleware:** The Aura System is a middleware, which consists of the Aura Manager Server and the Aura Type Database. The Aura Manager Server keeps track of a list of Aura-enabled applications or devices that have informed the system about their aura information. The server also periodically queries the location tracking system for location data to calculate the relative distances between devices in order to detect aura collisions. Upon detecting collision, notifications in the form of events will be sent by the server to the involved applications.
4. **Client-side Aura Application on Devices:** Client-side Aura applications are applications that sit on the end-users' devices which allow the users to customize the type of interaction and services they require with possible additional functionality such as increasing or decreasing the size of the application’s aura. Rules of engagement can be defined within such Aura applications. When the application is activated or during an update, the application will send information about the application’s aura(s) to the server. Collision notifications sent by the server are received by the application which will subsequently perform the appropriate actions depending on its implementation (i.e., the rules of engagement as pre-specified).

The Location Tracking System and the Aura Manager Server passively waits for information to be given to them before they start any form of processing. Assuming that the Location Tracking System and the Aura Manager Server have already been activated, information exchange is triggered when an Aura-enabled application is activated and sends information to the Aura Manager Server such as the device’s unique identification and the attributes of its Aura (such as size, focus and nimbus values, aura collision type to be used, which events to be notified, etc). The Aura Manager Server relays the device’s unique identification to the location tracking system in order to start collecting location information periodically which is used to detect collisions events. Upon detecting an event indicated by the application to receive, notifications will be sent to the application indicating the event type and other implementation specific information to allow the Aura-enabled application to establish its own interactions with the other applications involved in the event. This process will continue until the application notifies the Aura Manager Server that it is about to be deactivated. Figure 7 provides an illustration of the type of transmitted information within the conceptual model, describing the information that is exchanged between them. Our prototype uses Ekahau,[2] a location tracking system that uses the commonplace wireless infrastructure to track locations. The Service Oriented Architecture (SOA) in the form of web services is used to transmit information and for interoperability. Our testing site is a research area in the university’s campus. The area is covered by a private wireless network which is used for our prototype evaluation. The apparatus used in the implementation included the following:

- Ekahau, the location-tracking system creating the *Space* of our model.

---

[2] http://www.ekahau.com

- University's wireless research network as the communication *Medium* of our model.
- A Laptop running as the Aura Server, as well as a text based aura-enabled application written in Java.
- A Pocket PC running a graphical user interface version of the aura-enabled application written in .NET C#.

Our implementation provided a proof-of-concept realization of the concepts described earlier. We note that any device (even a soft-toy) embedded with networking capabilities and an embedded computer can perform the aura computations. Via aura computations, one can imagine certain soft-toys being able to be aware of and eventually interacting with other soft-toys according to the focus and nimbus settings, yielding variable interesting experiences for the toy users. Such aura settings can be factory encoded (say among soft-toys sold by the same manufacturer) and/or be adjustable by users via an interface or a switch on the device. For appliances, a mix of aura settings and device type descriptions can enable a device to only look out for certain other types of devices in close enough proximity to establish connections (e.g., a DVD player looks for a television, universal remote, and speaker system to connect to, if within close enough proximity). Note that there is little performance penalties in our model – message exchanges are low volume and simply rely on the fast wireless networking among devices (be it Wi-Fi or Bluetooth).

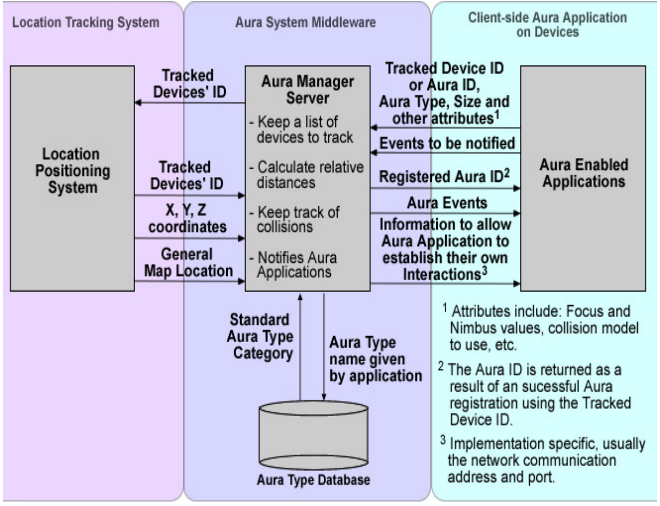

Fig. 7. Detailed information exchange diagram

RELATED WORK

Kortuem *et. al.* [10] engineered the "Relate Dongles" which are special sensor devices that can be attached to mobile computing devices via Universal Serial Bus (USB). The dongles are used to detect and measure the relative distance of other dongles using ultrasound and thus can capture detailed spatial information such as the device position and orientation as well as relationships between devices including whether they are approaching or moving away from each other. However, the system does not provide absolute location information and in terms of interaction, the most apparent disadvantage is the lack of a built-in communication medium (such as a wireless network) which is essential to provide any form of interaction between devices.

Digital Aura [18] is a thought model for spontaneous interaction between mobile devices in real environments with similar concepts of aura from. The interaction model used a single, fixed size (fixed conceptual space) aura for each object to initiate interactions upon aura collision. Digital Aura uses the proximity area (the physical range) of wireless technology such as Bluetooth, Radio Frequency Identification (RFID) or Infrared Data Association (IrDA) as the size of this conceptual aura. Therefore, the conceptual space covered by its aura is equivalent to the physical space covered by the range of the wireless technology used. Collision of the conceptual aura occurs by sensing the signal strength of other devices. Our Q-Aura model is different in that awareness and concealment are quantified and can be adjusted (that is, aura sizes are not fixed) - Digital Aura does not provide measures of awareness or fine-grained control of aura sizes as Q-Aura does.

Context-aware devices were initially researched by Schilit *et. al.* [15] whose intention is to make devices aware of other devices, the surrounding environment and to allow communication to take place. In subsequent years, several similar projects on context awareness such as MASSIVE [7], AROMA [13], CARISMA [3], EgoSpaces [9], Context Unity [9] and SOCAM [5] have evolved. Each model is targeted at different application domains such as location tracking, virtual worlds and mobility. A comparison of many of these projects with our work on mutual awareness was made previously [11].

Our work mapped the *focus* and *nimbus* concepts from the spatial model of interaction with the use of Presence which is commonly used in Instant Messaging (IM) application (e.g. Microsoft's MSN Messenger) to describe and convey the state (e.g. busy, away, etc.) of a contact. Presence takes the form of metadata which is exchanged between devices to convey different states of the device to achieve awareness and concealment of devices in real environments. This allowed devices to hide or expose themselves to other devices.

The AURA framework [16] facilitates moving computational tasks from one environment to another. The framework has been developed as part of the Project Aura at Carnegie Mellon University which, as in our work, also utilizes the notion of the personal *Aura*. Their notion of aura, however, is a proxy for the *mobile user* it represents: "when a user enters a new environment, his or her Aura marshals the appropriate resources to support the *user's task*.". The Aura also captures constraints that the *physical context* around the user imposes on tasks. The AURA framework focuses on modelling user tasks and how such tasks can be realized in different environments. Our notion of aura differs from this proxy notion – our idea of an aura is the conceptual space around a device, where awareness of other devices, or awareness by other devices, can happen.

The recent conference on the Internet of Things[3] reports on work towards RFID-tagged physical world artifacts and their detection using software infrastructures. Indeed, RFID tagging can be used to detect locations of devices to determine if two or more devices are close enough. With the employment of an accurate object positioning, focus and nimbus values can then be mapped to actual physical distances for instance. The work on social devices [17] considered wiring up artifacts with

---

[3] http://www.iot2008.org

sensors, building Internet connectivity and processing into artifacts such as “smart” potted plants, and an umbrella (which could then obtain weather information from surrounding sensors), and building Web 2.0 interfaces to objects to enable collaborative social networks of people and everyday artifacts. We consider Q-Aura enabled artifacts as social in the sense that they can be made aware of each other, but not in the sense of connectivity to the larger Internet necessarily. In contrast to our work, they do not focus on physical proximity as a trigger for applications.

Proximity and relative locations of objects have been used in [4] to determine actions. However, their work did not consider levels of awareness or collision classifications.

Companies such as Aura[4] are developing short range communication chipsets for close proximity applications - our model provides a basis for reasoning about the proximity based interactions. *More generally, our work provides a quantitative technique for proximity or spontaneous computing independently of specific networking technologies.*

In summary, we contend that a quantitative measure of focus and nimbus aids in providing fine granularity control over awareness, and more specifically, physical awareness among smart artifacts. While the notion of digital aura has been proposed in previous work, this paper makes the contribution, beyond current work to the authors' knowledge, of using quantitative measures of aura in determining which artifacts might be aware of which other artifacts.

CONCLUSION AND FUTURE WORK

We have presented a quantitative model of mutual awareness among artifacts, with numerous applications, ranging from soft-toys that interact to furniture (e.g., tables, chairs, stools, etc) that interact, as well as smart artifacts in time to come [12]. Given the right collision type and appropriate awareness and concealment values, devices are then aware of each other, afterwhich their interactions are then decided by the extent of their compatible capabilities.

Open standards for aura collision types and awareness/concealment values will be an avenue for future work, so that even devices from different manufacturers can interact (e.g., a piece of furniture can interact with the wall to adapt its colour, etc). Our approach merely involves integer comparison, which is extremely efficient. If floating values are used for awareness and concealment values, there is finer-grained modelling but devices capable of floating point computations may be required. Information exchanges involves wireless messaging but only a small number of small messages are exchanged (for each device), unless there are frequent changes to focus and nimbus values wherein more notifications to devices are required. However, further investigation is needed concerning scalability with hundreds or thousands devices.

So far, we only used location as our contextual information. If combined with other contextual information such as date and time, weather or even light, much more advanced scenarios can be produced. Instead of a number representing the size of a focus or nimbus, the size or boundary of a device's focus and nimbus can be determined by a complex set of parameters – we note, though, that using a number to represent a focus or nimbus simplifies computations and could represent a summary of a set of parameters (when a set of parameters can be used to compute the number). As future work, we will also study objects with context-dependent auras and multiple concurrent auras. Inspired by [19], another avenue of future work is to consider fuzzy aura where the boundaries are unclear, and context reasoning using a fuzzy set approach is more suitable, and to assign significance to, not just within focus or nimbus scenarios, but also outside particular focus or nimbus. Lastly, by tracking patterns in aura interactions, it may be possible to build a history of interactions among devices, facilitating automation, with prediction and anticipation of interactions (e.g., for smart homes as in [20]).

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for comments on the paper, which helped to improve the paper.

REFERENCES

[1] Benford, S. and Fahlén, L. (1993) A Spatial Model of Interaction in Large Virtual Environments. Proc. Third European Conference on Computer Supported Cooperative Work (ECSCW’93). Milan, Italy, Kluwer Academic Publishers, pp.109-124.

[2] Cardelli, L. and Gordon, A. D. Mobile Ambients (1998). Proceedings of the First international Conference on Foundations of Software Science and Computation Structure. M. Nivat, Ed. Lecture Notes In Computer Science **1378,** pp. 140–155. Springer-Verlag.

[3] Capra, L., Emmerich, W., and Mascolo, C. (2003) CARISMA: Context-Aware Reflective Middleware System for Mobile Applications. IEEE Transactions on Software Engineering **29**(10): 929-945, IEEE Computer Society Press.

[4] Strohbach, M., Gellersen, H.-W., Kortuem, G. and Kray, C. Cooperative Artefacts: Assessing Real World Situations with Embedded Technology. Proceedings of Ubicomp 2004.

[5] Gu, T., Pung, H.K. and Zhang, D.Q. (2004) Toward an OSGi-based infrastructure for context-aware applications. IEEE Pervasive Computing, 3(4), Oct-Dec 2004: 66-74.

[6] Ferscha, A., Hechinger, M., Mayrhofer, R., Rocha, MDS, Franz, M. and Oberhauser, R. (2004) Digital Aura. Pervasive Computing, Second International Conference, Pervasive 2004. Vienna.

[7] Greenhalgh, C. and Benford, S. (1995) MASSIVE: A Collaborative Virtual Environment for Teleconferencing. ACM Transactions on Computer-Human Interaction 2(3):239-261.

[8] Julien, C., Payton, J., and Roman, G.C. (2004) Reasoning About Context-Awareness in the Presence of Mobility. Electronic Notes in Theoretical Computer Science, Volume 97, 22 July 2004: 259-276.

[9] Julien, C. and Roman, G.C. (2002) Egocentric Context-Aware Programming in Ad Hoc Mobile Environments. Proc. of the 10th ACM SIGSOFT Symp. on Foundations of Software Engineering, pp. 21-30.

[10] Kortuem, G., Kray, C. and Gellersen, H. (2005) Sensing and Visualizing Spatial Relations of Mobile Devices. Proc. of the 18th annual ACM symposium on User interface software and technology, pp. 93-102.

[11] Ling, S., Loke, S.W. and Indrawan, M. (2007) Towards A Model of Interaction and Presence for Mutual Aware Devices and Everyday Artifacts. Proceedings 4th International Conference Ubiquitous Intelligence and Computing 2007 (UIC2007), Springer, pp. 84-94 .

[12] Loke, S.W. (2006) Context-Aware Artifacts: Two Development Approaches, IEEE Pervasive Computing 5(2): 48-53.

[13] Pedersen, E, R., Sokoler, T. (1997) AROMA: Abstract Representation of Presence Supporting Mutual Awareness. Conference on Human Factors in Computing Systems, Proceedings of the Special Interest Group on Computer-Human Interaction (SIGCHI) conference on Human factors in computing systems.

---

[4] http://www.auracomm.com/

[14] Rodden, T. (1996) Populating the Application: A Model of Awareness for Cooperative Applications. Proceedings 1996 ACM Conference on Computer Supported Cooperative Work (CSCW'96), pp. 87-89

[15] Schilit, B. N., Adams, N, Want, R. (1994) Context-Aware Computing Application. Proceedings of the Workshop on Mobile Computing Systems and Applications, pp. 85-90.

[16] Souza, J.P. And Garlan, D. (2002) Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments. *Software Architecture: System Design, Development, and Maintenance* (Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture) Jan Bosch, Morven Gentleman, Christine Hofmeister, Juha Kuusela (Eds), Kluwer Academic Publishers, pp. 29-43.

[17] Vazquez, J.I. and López-de-Ipiña, D. (2008) Social Devices: Autonomous Artifacts that Communicate on the Internet. Proceedings of the First International Conference on the Internet of Things (IOT 2008), Zurich, Switzerland, Lecture Notes in Computer Science 4952

[18] Ferscha, A., Hechinger, M., Mayrhofer, R., dos Santos Rocha, M., Franz, M., and Oberhauser, R. (2004) Digital Aura. In Advances in Pervasive Computing. A Collection of Contributions Presented at the 2nd International Conference on Pervasive Computing (Pervasive 2004), Austrian Computer Society (OCG), Vol. 176, pp. 405-410.

[19] Anagnostopoulos, C. and Hadjiefthymiades, S. Advanced Inference in Situation-Aware Computing. IEEE Transactions on Systems, Man and Cybernetics: Part A 39(5), 2009: 1108-1115.

[20] Rashidi, P. and Cook, D.J. Keeping the Resident in the Loop: Adapting the Smart Home to the User. IEEE Transactions on Systems, Man and Cybernetics: Part A 39(5), 2009: 949-959.