ENHANCED PRESENCE HANDLING

Rudi Victor[1], Andrew Rutherford[2], Reinhardt Botha[3]

[1,2,3] **Nelson Mandela Metropolitan University**
Institute for ICT Advancement
South Africa

[1]**rudivictor@gmail.com,** [2]**andrew.rutherford@nmmu.ac.za,**
[3]**reinhardt.botha@nmmu.ac.za**

ABSTRACT

The global use of the Internet and modern mobile/cellular communications networks has made ubiquitous communications possible for millions of people worldwide. However, these technologies can interrupt our daily activities through uncontrolled, unwanted disturbances. Such negative effects can be lessened by using context information and presence technology to inform others of our availability for communication. By nature context information can be sensitive and a commodity which its owner values highly. It thus becomes important to assess the impact of releasing such information on personal privacy. This can vary widely, depending on various factors. In this paper the authors review presence technology as a means to control unwanted disturbances. We consider the privacy implications and propose an enhanced presence processing model which leverages Role-Based Access Control (RBAC) principles as well as a new concept, "Availability Profiles". We present the model incrementally over three progressive and logical stages.

KEY WORDS

Presence, Presentity, Privacy, Watcher, Role-based access control

ENHANCED PRESENCE HANDLING

1 INTRODUCTION

With the evolution of networks, such as mobile/cellular communications and the Internet, the notions of ubiquitous computing and anytime-anywhere communications have become a reality for millions of people worldwide. However, the power of these networks, and related technologies, also have negative implications to consider and manage.

While people are able to work and communicate in a convenient and ubiquitous manner, the same technologies making this possible also cause uncontrolled and frequently annoying interruptions (Markus, 1994), which lead to a loss in productivity. This leads to another layer of technology being needed to manage and control these issues.

Because users would like to be available for communications, but want to minimize the disruptive effects of unwanted disturbances, information such as presence (Day, Rosenberg, & Sugano, 2000) has been used to provide clues as to the current availability of a user. However, providing such context to others can lead to potential concerns regarding privacy, which need to be considered carefully by a user.

In this paper the authors review presence technology as a means to control unwanted interruptions. We consider the privacy implications and propose a Role-Based Access Control (RBAC) (Sandhu, Coyne, Feinstein, & Youman, 1996) model to control the sharing of personal context information. A prototype implementation of the model illustrates the strengths and weaknesses of such an approach. Finally we illuminate future research needed in this area.

2 THE COMMUNICATION PARADOX

The sheer number of communication channels available today can, paradoxically, make it increasingly time consuming for human beings to establish communication amongst each other. By the same token these multiple channels of communication can increase the number of unwanted and disruptive communication attempts a person receives.

To illustrate let us view a typical communication attempt scenario from the perspective of both the caller and the callee. Alice wishes to contact Fred.

She knows Fred has an email address, Instant Messaging (IM) account, an office phone and a cellphone. Alice first sends an email message to Fred in the hopes of getting a prompt reply. Not receiving one, she proceeds to attempt to engage Fred in an IM conversation. After typing several messages with no response she calls Fred on his cellphone, which, after ringing a short period routes her to his voicemail. As a last resort she phones his office number only to hear Fred curtly informing her that he is busy and cannot talk now.

Let us now view Alice's attempts at communication from Fred's perspective. Fred is engaged in an important meeting with his company's managing director. Whilst discussing the latest sales figures his email client alerts him to a new message. He is mildly distracted for a second but continues with his discussion. Several minutes later Fred faces more distractions by the intermittent display of messages from Alice on his IM client. Choosing to ignore them, he continues with his meeting. It is at this time that Alice rings his cellphone. Fred of course had put his phone on silent thus avoiding the embarrassment of it ringing; it does however succeed in vibrating off the desk. Eventually his office line rings and he is forced to answer but abruptly informs Alice that he is busy and hangs up the phone.

From this simple example it is clear that both parties have been impacted negatively by Alice's attempts to contact Fred. She has wasted several minutes of her time in attempting to establish meaningful communication with Fred, which she is unable to do. Fred on the other hand suffered numerous unwanted interruptions which only succeeded in distracting him from a more important task.

The effects of interruption in our daily activities is a prominent research topic (O'Conaill & Frohlich, 1995; Jett & George, 2003; Rennecker & Godwin, 2005). It is clear that a need exists for a solution which controls interruptions while maintaining ubiquitous communications. Thus just switching a communications device off cannot be seen as an acceptable solution because of the loss of all communications.

The availability of user context can assist in making a more accurate communications request (Ljungstrand, 2001). This is a concept well-known from the Instant Messaging (IM) domain where presence information is available to parties. Such information can indicate basic availability as well as detailed information about the current state of a user. Research on the benefits of using such information in communications has established that it can be a useful mechanism to control unnecessary interruptions (Carroll, Neale, Isenhour, Rosson, & McCrickard, 2003; McCrickard, Catrambone, Chewar, &

Stasko, 2003; Sonnenwald, Maglaughlin, & Whitton, 2004). The architecture for the implementation of such a solution is also clear and well defined (Day et al., 2000; Rosenberg et al., 2002; Roach, 2002).

As technology improves and advanced sensors become available, the level and quality of context information that can be obtained becomes very detailed. For instance, a person's location can be determined using his cellphone or his current activity and persons with whom he is engaged can be discovered through his electronic calendar. Such detailed information can be seen as extremely sensitive and private and it becomes important to ensure the privacy of such information. Privacy encompasses the confidentiality and integrity of the information, but also controlling the access to such information by other parties. The problem of preserving the privacy of context information thus becomes an issue of access control.

Prior to presenting our enhanced model it is requisite that we should provide an overview of a current presence processing model.

3 CURRENT PRESENCE PROCESSING

The Internet Draft "A Processing Model for Presence" provides a model that describes and defines the processing operations used by presence agents in processing presence information in a SIP and SIMPLE environment (Rosenberg, 2005). The proposed model, hereafter referred to as the Rosenberg-model, is depicted by figure 1. To facilitate the discussion which follows it will be prudent to clarify two terms. A presentity is an entity that has presence information, and a watcher is an entity that is interested in the presence information of a presentity.

Watchers whose subscriptions have been accepted receive presence information notifications. To fulfill notifications the presence server must generate a presence document for each watcher. This presence document generation process, detailed by Rosenberg (2005), shall be discussed in brief.

The first step in the presence document generation process is collection. Collection involves the obtaining of all the event state information required for giving an accurate picture of a presentity's presence. This event state information can come from a variety of presence sources as illustrated by figure 1.

A composition operation now occurs which compiles all collected information to produce the raw presence document or initial view. "Raw" is appropriate since at this stage the presence document contains the full presence

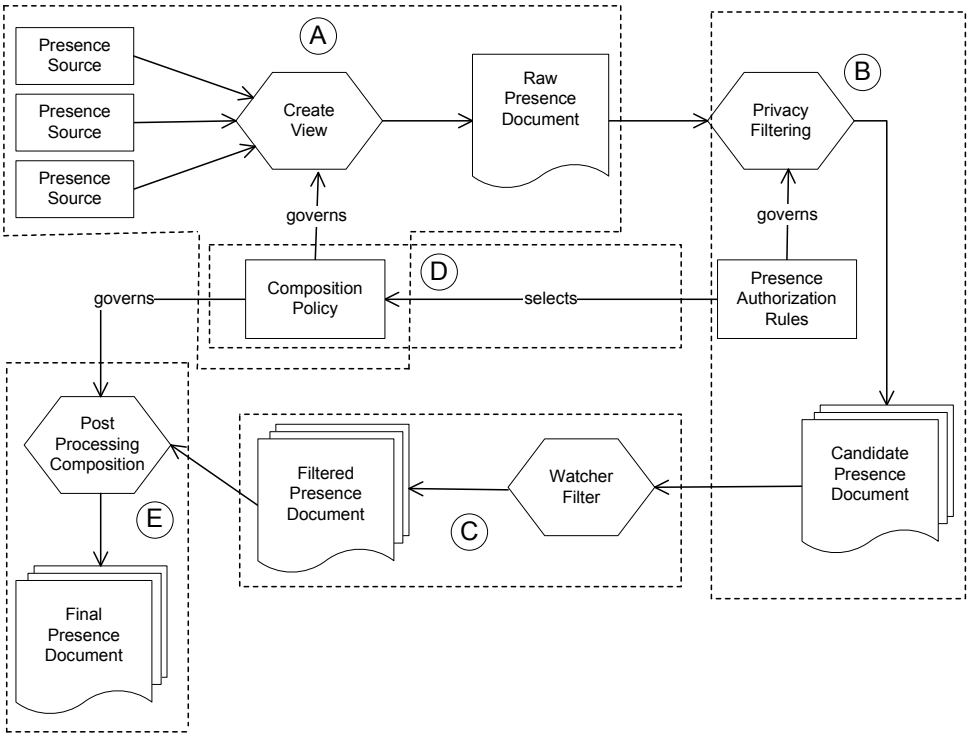

Figure 1: The Rosenberg Presence Processing Model (Rosenberg, 2005)

picture of a presentity; more information than any watcher might actually see. The composition phase makes use of several techniques to achieve its objectives. These include correlation, conflict resolution, merging and splitting. Correlation uses information in one presence document to effect information in another; conflict resolution must resolve situations where conflicting presence information is reported by multiple sources; merging involves combining different devices or services into a composite device or service; splitting is the antithesis of merging i.e. a single device or service is split into two devices or services.

Having compiled the raw presence document it is requisite to perform privacy filtering. This entails the removal of information from the raw document and thus withholding certain information about the presentity. Presence filtering is influenced by the identity of the watcher and other factors such as time of day, location and so forth. The manner in which privacy filtering is performed is determined by an authorization policy which comprises presence

authorization rules (Rosenberg, 2006).

A watcher may also control the information he receives by specifying filters along with his subscription request. At this point a presence document can be despatched to a watcher.

As a result of privacy and watcher filtering there may be information lacking to differentiate certain device and service elements from one another. In such a case further composition rules will be applied.

The presence processing model depicted by figure 1 has been partitioned into five logical groupings namely A, B, C, D and E. Grouping A represents the collection and composition phases which results in the creation of the raw presence document. A default composition policy is used to generate this raw presence document.

Grouping B represents the privacy filtering which must take place on a watcher-by-watcher basis. Presence authorization rules (Rosenberg, 2006) are applied to ensure that each watcher receives only that subset of presence information to which he is entitled i.e. the candidate presence document. It is possible at this point that the authorization policy can select a composition policy other then the default to generate the presence document sent to the watcher. This optional process is indicated by D.

Grouping C illustrates the ability of the watcher to optionally further filter the presence information he will receive, resulting in a presence document filtered according to his requirements.

Grouping E shows the optional application of further composition rules prior to the generation of the final presence document.

4 ROLE-BASED ACCESS CONTROL (RBAC)

Figure 2 shows the family of role-based access control models expounded by Sandhu et al. (1996). In order to facilitate the discussion of the application of RBAC to presence information it would be a useful exercise to map the primary RBAC terminology to those used in the presence domain.

User: In the RBAC model a user is a human being with the potential to gain access to a resource. In the presence domain **watchers** are users who wish to gain access to the presence information of a presentity.

Role: A role in RBAC typically refers to a named job junction and describes the authority and responsibility assigned to a member of the

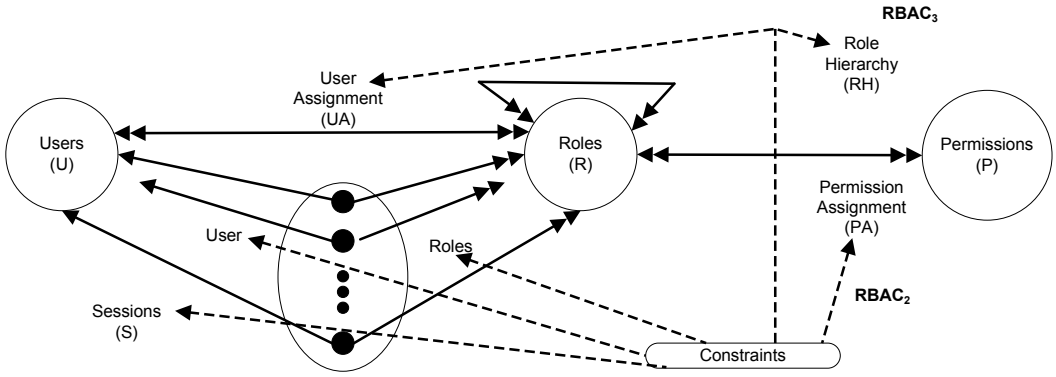

Figure 2: Role-based Access Control Models (Sandhu et al., 1996)

role. There is no presence term currently that maps to a role, but this paper suggests that the concepts of roles should be embraced in the processing and access control of presence information

Permission: A permission is said to be an authorization to a mode of access to a resource. Permissions can be equated to the components that comprise a presence authorization rule. Each rule contains conditions, actions and transformations for the purpose of controlling the presence information received by a watcher (Rosenberg, 2006). Conditions control when a particular rule is to be enforced, whilst actions outline the response a presence server should make to a watcher subscription request (Rosenberg, 2006). Transformations provide a filtering mechanism allowing presence data to be manipulated before being presented to the watcher (Rosenberg, 2006).

Session: A session provides the mapping between a user and an activated subset of the set of roles to which he is assigned. We believe that in presence processing a session will relate to the period of time during which a watcher presence document is generated. It is at this time that the subset of roles to which a watcher belongs would dictate the privacy filtering operation.

We will now proceed with the introduction of our enhanced presence processing model.

Table 1: Overview of the PH-model development

Model Progression	Administration	Processing
PH_0	watcher-based presence	presence filtering
PH_1	role-based presence	watcher-to-role mapping
PH_2	availability profiles	availability filtering

5 PROPOSED MODEL

Our enhanced presence processing model will be developed in progressive stages as shown in table 1. Each stage shall be discussed thoroughly and any shortcomings highlighted. These shortcomings will provide impetus for the next stage of our model's development. Initially, PH_0 will be constructed by inheriting the key functionality of the presence handling model of Rosenberg (2005). PH_0 will be formalized and considered as a watcher-based presence handling model.

The next step will be to modify PH_0 with the addition of role-based concepts as presented in the RBAC specification by Sandhu et al. (1996). This second step will elevate PH_0 to a role-based presence processing model (PH_1).

The final stage, PH_2, will add the concept of an *availability profile* to PH_1. The purpose of PH_2 is to improve the presentity's ability to better handle incoming messages from watchers.

5.1 PH_0: Watcher-based Presence

The presence handling model, PH_0, now proposed, is strongly based on the Rosenberg-model (Rosenberg, 2005). This model provides much more detail than is relevant for the purposes of the PH_0 model and therefore only the essential components and processes have been retained to form PH_0.

The Rosenberg-model provides for the filtering of sensitive presence information on a per-watcher basis. While not widely implemented, the Rosenberg-model is considered the status quo in a SIP/SIMPLE environment.

Figure 3 shows an architectural view of the PH_0 model, adapted from the more detailed model as represented in figure 1. Figure 3 describes the most crucial part of the Rosenberg-model, namely presence filtering on a per-watcher basis.

There are three basic concepts in the PH_0 model namely, watchers, presence, and subscriptions. These can be formalized as follows:

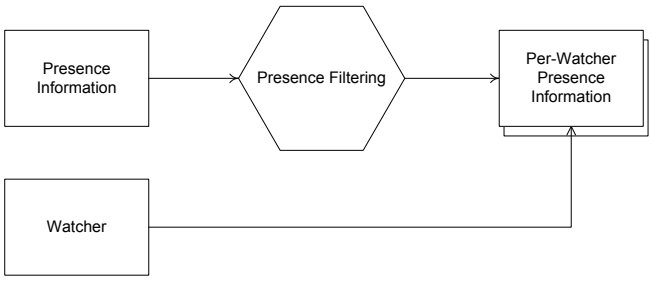

Figure 3: Architectural view of PH_0, adapted from the Rosenberg-model

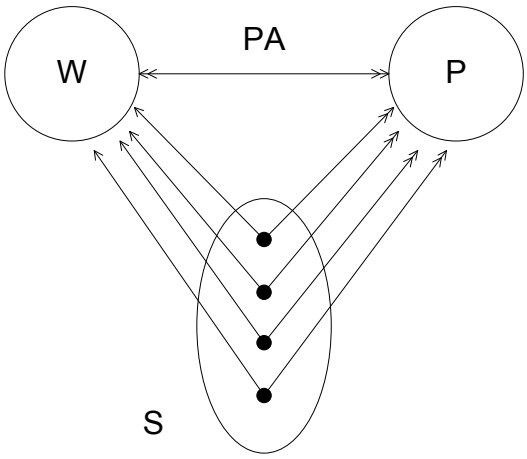

Figure 4: Graphical view of PH_0

- W, P, and S representing watchers, presence attributes, and subscriptions; and
- $PA \subseteq P \times W$ representing presence-watcher assignments.

A graphical view of PH_0 is given in figure 4 and shows that presence is directly associated with watchers in a many-to-many relationship. In figure 4 this relationship is represented by a double-headed arrow between the watchers and presence entities. In other words, a watcher can have access to various presence attributes, and every presence attribute can be provided to many watchers. The formalization of this processing can be represented as follows:

- *watcher* : $S \rightarrow W$ a function mapping each subscription s_i to a single watcher $watcher(s_i)$; and

- *presence attribute*: $S \rightarrow 2^P$ where each subscription s_i is mapped to a set of presence attributes where *presence attributes*$(s_i) \subseteq \{p \mid (p, watcher(s_i)) \in PA\}$.

A watcher obtains a subscription for a presentity's presence information after authorization by the presentity. Thus, every presence subscription is associated with only one watcher. For the duration of the subscription, the watcher is mapped to a set of presence attributes. Every presence attribute to be provided to a watcher needs to be authorized in the presence policy. Access to all other attributes is implicitly denied.

The processing of PH_0 comprises *presence information, presence filtering, per-watcher presence information, watcher filtering and watchers*. The Presence Information component maps to grouping A on the Rosenberg-model which details the composition of the current set of presence information in a presence document.

The Presence Filtering activity enforces permission-assignments and maps to grouping B on the Rosenberg-model i.e. the filtering process as being governed by presence authorization rules.

The PH_0 model addresses the one-for-all approach to handling watchers by virtue of its per-watcher filtering. However fundamental flaws still exist namely, (a) undue burden is placed on the presence server in the generation of presence documents and (b) undue burden is placed on the presentity in creating authorization rules for each watcher. This leads us to the next stage of our model's development which incorporates RBAC principles.

5.2 PH_1: Role-based Presence

In the Role-Based Access Control (RBAC) model, users are associated with roles, and roles with permissions. The mapping of RBAC concepts to those of PH_1 was shown in section 4. Therefore, the addition of role-based concepts to PH_0 can be formalized by drawing on the formalization of the RBAC model.

The most notable change in PH_1 from PH_0 is the introduction of roles. Roles are defined by Sandhu et al. (1996) as a set of permissions. Similarly, a role in PH_1 is defined as a set of presence attributes. In PH_0 such presence

attributes are assigned directly to a watcher with a subscription. In contrast, a presence attribute is assigned to a role in PH_1.

Ferraiolo and Kuhn (1992) state that the concept of roles stems from the realization that in an enterprise control is governed by an employee's role and function. Similarly a presentity behaves differently, based on the identity of the watcher. In PH_0 the control of presence information is aligned with this behavior. However, as the number of watchers increase, it is not feasible for the presentity to maintain distinct per-watcher handling. Therefore, PH_1 introduces roles to group watchers according to their "role and function" (Ferraiolo & Kuhn, 1992) with regards to the presentity. In other words, a watcher is organized according to the relationship that the presentity has with that watcher.

The architectural view of PH_1 is given in figure 5. The diagram shows that instead of a watcher directly accessing presence information, the watcher is first mapped to a role. The PH_1 concepts can be formalized as follows:

- W, P, and S are unmodified from PH_0;
- R and WA are added, representing roles and watcher-to-role assignments respectively;
- $WA \subseteq W \times R$;
- PA is modified from PH_0 to contain presence attribute-to-role mappings where $PA \subseteq P \times R$.

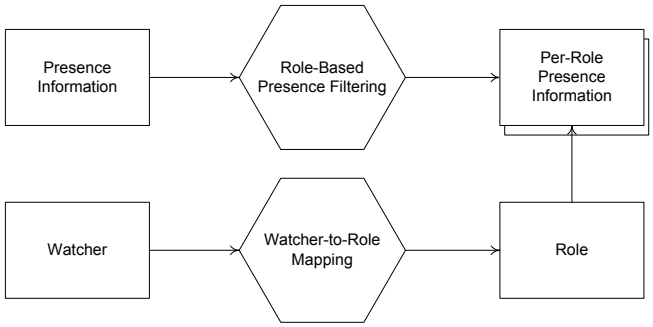

Figure 5: Architectural view of PH_1

Figure 6 graphically depicts the specifics of the PH_1 model. The figure shows that a watcher can be assigned multiple roles, and that a role may be

assigned many watchers. If a watcher activates several roles during the same subscription, scenarios are likely where multiple roles can cause conflicts in the sets of presence information approved for a watcher. The same issue exists on per-watcher filtering of presence such as in PH_0. This issue has been addressed in the IETF specification on *expressing privacy policy* (RFC 4745) but the resolution is implementation specific. Therefore, the issue need not be addressed by PH_1.

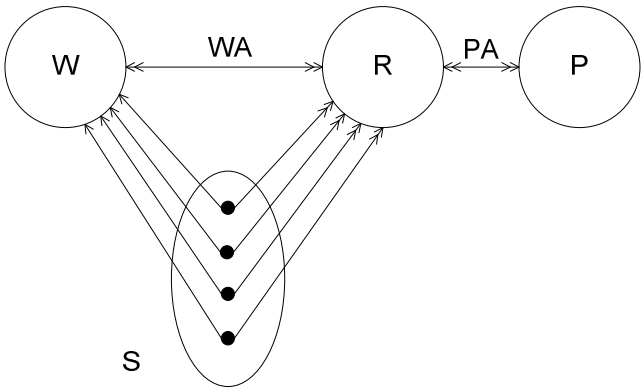

Figure 6: Graphical view of PH_1

The processing specific details of PH_1 can be formalized as follows:

- $roles : S \rightarrow 2^R$ where
- each subscription s_i is mapped to a set of roles $roles(s_i) \subseteq \{r \mid (watcher(s_i), r) \in WA\}$ and
- subscription s_i has the set of presence attributes $\cup_{r \in roles(s_i)} \{p \in (p, r) \in PA\}$.

The application of roles to PH_0 results in an improved, role-based presence handling model. It provides a presentity significant advantages in scalability and management of watchers. Furthermore, the presence server need only process presence information for the set of roles as opposed to for each watcher.

However, a presentity's presence information is in a state of flux, continuously changing and updated. In contrast, a subscription does not change as often. A distinction must be made between the presence attributes a watcher

is authorized to view within a subscription, and the actual values of the received presence attributes. The latter may change during the lifetime of a subscription. Thus while PH_1 provides role-based filtering of presence information it still does not fully take into account the needs of the presentity. Such needs are catered for by the third stage of the enhanced model.

5.3 PH_2: Availability Profiles

The low cost of sending messages via a wide range of communication channels, combined with the Internet facilitating constant connectedness, increases the chances of being disturbed, especially in the workplace.

If a presentity conveys his/her presence state as "busy", a watcher may interpret that it is not currently the best time to initiate an interaction and spare the presentity the possible interruption. The problem with providing a presence state such as "busy" to watchers, is the total reliance on the watchers' interpretation or consideration of such information. Although the presentity implies availability and willingness with a presence state, it is not enforced on the presentity's side of the interaction. The concept of *availability profiles (AP)* provides such a mechanism.

An architectural view of PH_2 can be seen in figure 7. The diagram shows that PH_1 has been extended with the addition of an availability filter connecting the watcher (W) and presentity (Pr). The input to the availability filter from the watcher-side is the presence information provided to the watcher. In particular, the presented availability information, as embedded in the presence state, is of concern. The availability filter is used to provide the presentity with an availability profile with which incoming communication from a particular watcher is handled.

An availability profile can be defined as the availability and responsiveness with which any incoming communication is handled. The purpose of an availability profile is to help the presentity handle incoming communication with the appropriate amount of responsiveness. Responsiveness in the context of our use can be viewed as *demonstrated availability*.

The PH_2 model can be defined as follows:

- PH_2 is unmodified from PH_1 except for the addition of availability profiles (AP).
- An availability profile defines how to handle an incoming watcher-message based on the availability-related presence information provided

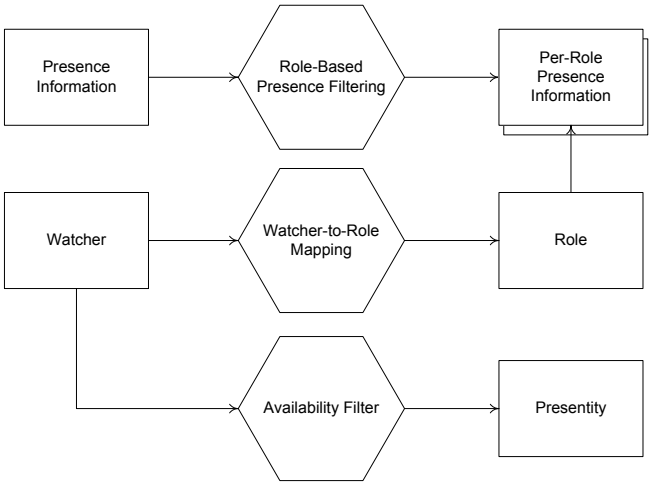

Figure 7: Architectural view of PH_2

to that watcher.

The processing of PH_2 extends PH_1 as shown in figure 8. The model does not specify how an availability profile is to be implemented. However, it can be likened to a real-life working environment where person A is working in his office. If person B enters the office unannounced, it may cause interruption even without communication. However, if person B leaves a note in front of the office, person A will be spared the interruption.

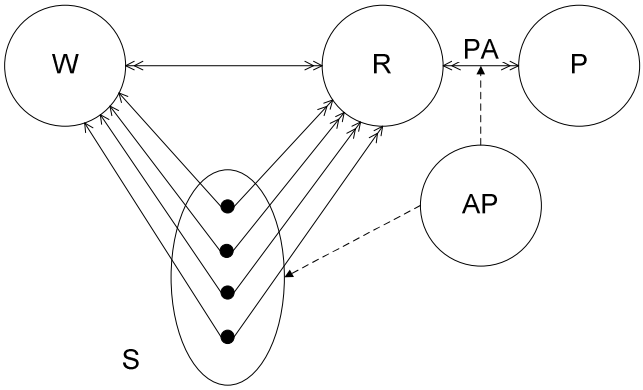

Figure 8: Graphical view of PH_2

In the example, it can be said that person A presented a presence state of "busy" to all. However, the door was not locked, leaving potential for handling emergencies as well as the potential of unwanted interruptions. An availability profile can be likened to the process of locking the door and providing the key to a specific set of people. The people that have keys, will be able to enter person A's office, implying high availability. Similarly, to the people without keys, person A can be seen as busy but can be reached by leaving a message at the door. An availability profile can thus help person A to demonstrate the appropriate amount of availability.

The availability filter uses the set of presence information as authorized by the presence-assignments of the watcher within a subscription to produce an availability profile as mentioned in the specification section. Furthermore, the availability profile may modify presence information regarding availability before it is presented to the watcher. This association between the availability profile and presence-assignments is indicated with a broken-line arrow in figure 8. Availability profiles can be likened to constraints in RBAC in that both can limit values. However, availability profiles can modify actual presence attributes and also describe watcher message handling behavior on the presentity's side, none of which can be achieved with RBAC constraints.

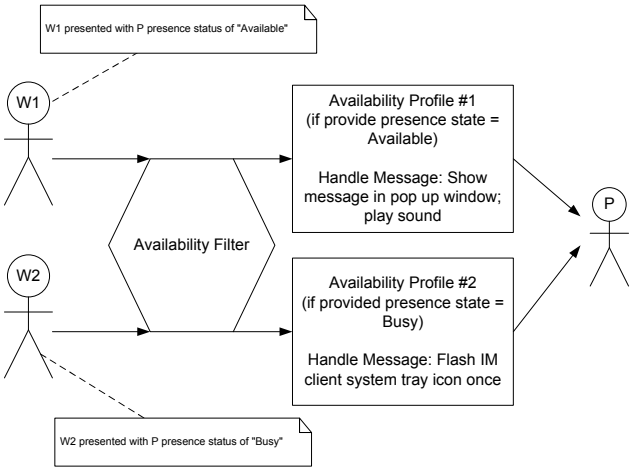

Figure 9: Example of PH_2 potential implementation

In the PH_2 model an availability profile is associated with a subscription, and with a watcher through implication. In more general terms, as long as a watcher is provided presence information within a certain combination of

authorizations, communication from the watcher will be treated in a certain way. One such implementation possibility is to associate the availability profile with the presented presence state. Furthermore, the availability profile can then use the presented presence state to handle incoming communication with a certain salience. Figure 9 illustrates such an implementation example in an IM environment. The background to figure 9 is that watcher W1, and watcher W2 are communicating with presentity P. The figure shows that different presence states have been presented to each watcher, i.e. "Available" to W1 and "Busy" to W2. The availability profiles (AP1 and AP2) handle incoming messages based on the presented presence state of P. AP1 allows a watcher's message to be displayed saliently in the form of a pop up window and a sound alert if the presented presence state is "Available" (indicating high availability). Similarly, AP2 handles incoming messages to watchers that were presented with a presence status of "Busy". Therefore messages from W2 will be handled inconspicuously by only flashing the IM client icon in the system tray once. The end result is that the presentity is helped in demonstrating the right amount of availability through the level of salience with which the message is displayed.

6 CONCLUSION

The Internet and modern mobile/cellular networks have enabled ubiquitous communications for millions of people worldwide. However, a drawback to the power of these technologies is the problem of uncontrolled and unwanted disturbances.

Inadvertent interruptions can often be attributed to a lack of knowledge regarding the current availability for communications of a person. The presence status of a person can provide valuable cues in this regard. However, such information can be highly sensitive and private, and must be distributed in a secure manner. By using an RBAC model we have shown how this can be achieved in an efficient and manageable manner. However, issues still remain. A user might have quite a number of contacts and people he/she interacts with. Automating the process of role and permission assignment is an important factor in making such a model usable and appealing. We envision the use of artificial intelligence to learn from user behaviour, and assist the user in automating and managing access control, as an important area for future work.

References

Carroll, J. M., Neale, D. C., Isenhour, P. L., Rosson, M. B., & McCrickard, D. S. (2003). Notification and awareness: synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies*, *58*(5), 605–632.

Day, M., Rosenberg, J., & Sugano, H. (2000). *RFC 2778: A Model for Presence and Instant Messaging.* Internet Engineering Task Force. (Available from: `http://www.ietf.org/rfc/rfc2778.txt`)

Ferraiolo, D. F., & Kuhn, D. R. (1992). Role Based Access Control. In *15th National Computer Security Conference* (pp. 554–563).

Jett, Q. R., & George, J. (2003). Work interrupted: A closer look at the role of interruptions in organizational life. *Academy of Management Review*, *28*(3), 494–507.

Ljungstrand, P. (2001). Context Awareness and Mobile Phones. *Personal and Ubiquitous Computing*, *5*(1), 58–61.

Markus, M. L. (1994). Finding a happy medium: explaining the negative effects of electronic communication on social life at work. *ACM Transactions on Information Systems*, *12*(2), 119–149.

McCrickard, D. S., Catrambone, R., Chewar, C. M., & Stasko, J. T. (2003). Establishing tradeoffs that leverage attention for utility: empirically evaluating information display in notification systems. *International Journal of Human-Computer Studies*, *58*(5), 547–582.

O'Conaill, B., & Frohlich, D. (1995). Timespace in the workplace: dealing with interruptions. In *Chi '95: Conference companion on human factors in computing systems* (pp. 262–263). ACM Press.

Rennecker, J., & Godwin, L. (2005). Delays and interruptions: A self-perpetuating paradox of communication technology use. *Information and Organization*, *15*(3), 247–266.

Roach, A. (2002). *RFC 3265: Session Initiation Protocol (SIP)-Specific Event Notification.* Internet Engineering Task Force. (Available from: `http://www.ietf.org/rfc/rfc3265.txt`)

Rosenberg, J. (2005). *A Processing Model for Presence* (Internet Draft). Internet Engineering Task Force.

Rosenberg, J. (2006). *A document format for expressing privacy preferences* (Internet-Draft No. Version 05). Internet Engineering Task Force.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., & Schooler, E. (2002). *RFC 3261: SIP: Session Initiation Protocol.* Internet Engineering Task Force. (Available from: `http://www.ietf.org/rfc/rfc3261.txt`)

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-Based Access Control Models. *IEEE Computer*, *29*(2), 38–47.

Sonnenwald, D. H., Maglaughlin, K. L., & Whitton, M. C. (2004). Designing to support situation awareness across distances: an example from a scientific collaboratory. *Information Processing & Management*, *40*(6), 989–1011.