A Cognitive Architecture for Personal Networks

Yunfei Wu and Ignas Niemegeers

WMC group, Department of Electrical Engineering,
Delft University of Technology,
Mekelweg 4, 2628 CD Delft, the Netherlands
{y.wu, I.G.M.M.Niemegeers}@ewi.tudelft.nl

Abstract. This paper proposes a three-layer cognitive architecture for pervasive and intelligent computing of personal networks. The key element of the proposed architecture is the cognitive layer, which consists of five components, namely the context cognition, the personalization cognition, the resource cognition, the network cognition, and the cognition management. In order to demonstrate the purpose of each one of the cognition component, we present a motivation example on session mobility and show that the proposed architecture enables proactive configuration and thus hides the latency of configuration. Finally, this paper identifies the research issues that need to be addressed in order to implement the cognitive architecture for personal networks.

1 Introduction

A Personal Network (PN) [1] is a distributed personal environment where people interact with various companion, embedded, or invisible computers not only in their close vicinity but potentially anywhere. By creating a personal distributed environment, a PN extends and complements the concept of ubiquitous computing with a user-centered view. Figure 1 illustrates the concept.

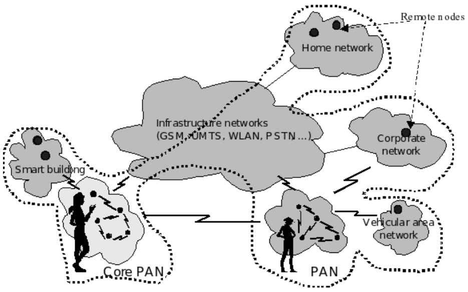

Fig. 1. An example of a personal network.

Mark Weiser, the originator of the concept of ubiquitous computing, has envisioned that computing technologies would disappear into the background and weave themselves into the fabric of everyday life [2]. In the case of a distributed personal environment, such disappearance requires context awareness and the ability of self-organization. Personal environments need to be smart to acquire and apply knowledge about a person and his surroundings in order to learn his preferences, to be proactive under uncertainty, and to be secure from malicious attacks. This requires some intelligent behavior in the personal environment.

As a consequence, a new research topic, the cognitive networking, is emerging, exploiting machine learning in networking. An example is the knowledge plane [3], a concept aiming to utilize cognitive techniques (e.g., representation, learning, and reasoning) to make the network aware of what is going on in the network and respond. A PN, however, not only needs to be smart and intelligent in managing itself, but also in providing personal services. Therefore, it must be capable of learning the preferences of its owner, reasoning about what he intends to do, and acting proactively. The goal of this paper is to explore how cognitive technology could be applied in a PN for providing personal services and managing itself. To accomplish this, we propose a high level abstraction of the three-layer cognitive architecture, namely *the device layer, the cognitive layer, and the service layer*. The key new element of the proposed architecture is the cognitive layer. It consists of five components: *context cognition, personalization cognition, network cognition, resource cognition, and cognition management*. The proposed cognitive architecture provides the functionality to manage intelligent components, enable communication between them, and facilitate the interfaces between a person and his environment.

This paper is organized as follows. Section 2 presents some related work to this paper. Section 3 describes the main characteristics of PNs. Section 4 presents the main capabilities the cognitive architecture should provide in a PN environment. Section 5 proposes a cognitive architecture, stressing the functionalities to be provided and the interfaces among the different components. Section 6 illustrates the role of the architectural components. Section 7 discusses open questions regarding the implementation of the proposed cognitive architecture. In Section 8, we draw some conclusions.

2 Related Work

There are a number of research being worked on cognitive networking. IBM [4] has proposed autonomic computing systems: they manage themselves and free system administrators from many of today's routine management tasks. Self-management in their view includes four aspects: self-configuration, self-healing, self-optimization, and self-protection. Dietterich et al [5] examine various aspects of machine learning that are useful for a cognitive approach to networking and they review three capabilities in cognitive networking. These are anomaly detection and fault diagnosis, responding to intruders and worms, and rapid configuration of networks. Works have been going in various projects to design pervasive computing systems and applications that adapt themselves to their users, e.g., Project Aura at Carnegie Mellon University [6], Easyliving project at Microsoft [7], and MIT Project Oxygen [8]. However, our work aims at designing an intelligent environment specifically for PNs with the knowledge of the characteristics of PNs.

3 The Main Characteristics of Personal Networks

Before we describe the cognitive architecture design, let us present those characteristics of a PN that have a strong influence on its design.

- **Heterogeneity:** Heterogeneity exists in a PN in a number of ways, such as the variety of devices, the geographical distribution of its components, the diversity of wireless and wired interfaces, and the different resource constraints. A PN consists of heterogeneous devices including diverse sensors and actuators, digital cameras, mobiles, PDAs, computers, home appliances, etc. These devices are distributed over different locations, at home, in the office, or in the car. Some devices may have one or more wired or wireless interfaces. The PN should support ubiquitous connectivity via infrastructure-based or ad hoc networks and integrate the heterogeneous devices in a seamless way. Moreover, the devices may have different resource constraints regarding energy supply, memory, and processing capability.
- **Personalization:** PNs are user-centric. For example, many people would like their favorite program to show up immediately when they turn on the TV. When the phone is ringing, they would like the volume of the TV to be automatically turned down. When they roam from one room to another, their favorite program should follow them on the displays that are present in different rooms. Already now many devices can be programmed to suit personal preferences. We expect that future personal devices will be able to understand a person's needs, be aware of what he wants to do and try to accommodate the person's likes, habits, and situations. Therefore, a PN should offer personalization by understanding a person's needs; e.g., it should ensure that a person receives the right information at the right time in the right way.
- **Autonomy:** Autonomy has two aspects. The first one is that the devices constituting a PN, should be able to organize themselves with minimal human intervention. A PN consists of multiple clusters [9] which are geographically distributed, at home or wherever the person happens to be. The cluster topology is dynamic in the sense that nodes can join or leave a cluster, clusters can merge, or a cluster can be split up. These dynamics requires that the PN can discover these changes, self-organize components of clusters, and maintain the services or devices automatically. The second aspect of autonomy is that PNs should make their own decisions with minimal intervention by the user.
- **Security:** PNs are vulnerable to attacks by malicious intruders. The concept of a PN will only inspire trust and be accepted by its users when a sufficient level of security is guaranteed. Since PNs are centered on the needs of an

individual, measures will have to be taken to protect the privacy of the owner. It requires that a PN is able to detect and react to attacks in a timely fashion. When an attack has happened, the PN should find suitable software to directly respond to this attack, update security software automatically, and filter out all packets coming from the origin address of this attack.

The characteristics outlined above require some kind of intelligent behavior in a PN. Under intelligence, in this context, we understand the capability to gather information about the personal environment, to learn from the information, from the prior knowledge of the environment, and from personal preferences, to adapt to personal preference automatically, to reason under uncertainty, and to react in advance.

4 The Capabilities of Cognitive Architecture for Personal Networks

In order to provide intelligent behavior in a PN, we need to design a cognitive architecture, a structure that forms the framework for the immediate processes of cognitive performance and learning [10]. The cognitive architecture should describe the detailed mechanisms on how cognition functions are implemented and how actions are performed. Therefore, the cognitive architecture of a PN concerns 1) the ability to access as many sources of knowledge as possible, 2) the ability to select proper AI techniques for the obtained knowledge, 3) the ability to make right decisions to actions, and 4) the ability to achieve its goals. Figure 2 illustrates the relationship between the cognitive architecture, knowledge, AI techniques, actions, and goals. We will discuss each of these in details as follows.

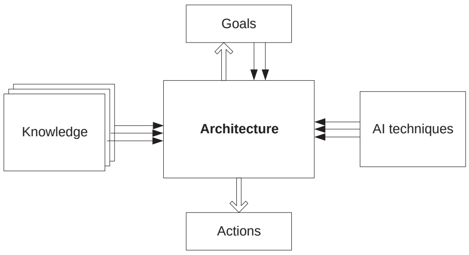

Fig. 2. The relationship between architecture, knowledge, AI techniques, goals and actions.

- **The capability to manage knowledge:** Learning is a process of knowledge transfer. Therefore, the cognitive architecture should be able to manage all sources of knowledge of PNs as efficient as possible. Generally, There are three different sources of knowledge applicable to PNs. First, knowledge about environments of PNs can be obtained directly from local personal devices, e.g., from sensors on a person's movement and on temperature etc. Second, some knowledge may not be obtained directly from external sources, but from internal sources generated by problem solving, such as a person's destination predicted from the person's movement. Third, some knowledge may be obtained from communication of distributed personal devices over a PN due to its distributed environment as mentioned before. Moreover, some knowledge may provide the whole information of a personal environment, some may carry inaccurate or partial information of the personal environment, and some knowledge may change over time under a dynamical environment. These characteristics put forward a challenge on the design of a cognitive architecture for PNs. Therefore, the cognitive architecture should be able to not only efficiently access and organize the knowledge, but also deal with uncertainty, such as noisy data and dynamical environments. The architecture should be able to facilitate component interaction in order to exchange data or access each other's services.

- **The capability to select and use proper AI techniques:** There are many existing AI techniques and algorithms developed for learning and reasoning. Generally, the task of the cognitive architecture is to perform a particular task using a proper AI technique with the knowledge obtained. Therefore, instead of having one-for-all solution, the architecture should be able to choose among the AI techniques that are appropriate for various types of learning tasks and different knowledge obtained. For example, the architecture selects a learning algorithm to learn knowledge about the past (like a user's routine) or a reasoning algorithm to infer the future situation based on the current knowledge. The architecture should be able to decide which algorithm is good for the limited/enough knowledge obtained and evaluate how the performance varies with the amount of knowledge.
- **The capability to make decisions:** The architecture is not only able to perceive its environment through sensors, but also able to act upon the environment through actuators. This is exactly the same procedure as human beings. When he sees something to touch, he needs to stretch out his hands to do so. The decision made by the cognitive architecture should be rational, which includes three aspects. First, it is able to do the right thing, for example, sending message, playing music, or displaying something on the screen. Second, it is able to trigger the right personal device. Third, if the personal device is remote, the architecture is able to detect the availability of resources for communications with the remote device. Given the knowledge obtained and the proper AI algorithms used, the architecture first infers what actions can be made, then calculates the utility to be expected on taking each alternative action according to the probabilities of the user's expectation and devices available to the user, and finally selects the alternative with the highest expected utility. The right decision should be made at the right time. In these cases, the architecture should be able to optimize when to make the decision considering resource costs, communication delay, etc.

- **The capability to achieve its goals:** Learning arises from goal-oriented activities. The architecture should have the ability to bring to bear all the knowledge that it has in the service of its goal. We envision that the goals of the cognitive architecture for PNs are to provide personal services and network management. Personal services means not only to specify the preferences of users, but also to infer similar services a user may have encountered or from uncertainty. Network management services enable personal networks to manage themselves. Network management includes fault management, performance management, configuration management, and security management. In order for the architecture to maximize the performance, the architecture might need some goal information to acquire and revise the knowledge through learning or to combine the knowledge obtained to make a decision. Imagine configuring the network at the destination when a user is moving. The goal is to make sure the network is ready when the user reaches the destination. The architecture needs to acquire knowledge on where the user is, what the destination can be, what the network he needs to use, and how long it would take to configure it. Taking these factors into account, the architecture can make the decision.

In order to achieve these capabilities, two major challenges arise in designing the architecture. One is the interaction issue, which includes two aspects. The interaction between the architecture and the environment should be as flexible as possible in order to acquire as many sources of knowledge as possible and manage personal devices as efficiently as possible. The interaction between the architecture and the user should be as easy and convenient as possible for the user. The second issue is that the architecture should be open to be extendable and easily modified with newly emerging functional entities.

5 The Cognitive Architecture of PNs

As illustrated in Fig. 3, the proposed cognitive architecture consists of three layers: the device layer, the cognitive layer, and the service layer. The *device layer* provides context information to the cognitive layer from heterogenous devices (such as sensors, mobile, cameras, and computers etc.). It also takes actions invoked by the cognitive layer. The *cognitive layer* is responsible for obtaining context data from the device layer and to deal with uncertainties to better serve the current situation of the owner of a PN. It consists of five components, namely *context cognition, personalization cognition, resource cognition, network cognition, and cognition management.* The *service layer* uses information (e.g., reasoning and inferring results) from the cognitive layer to adapt PNs for providing two kinds of services: personal services and network management. Let us discuss each of these components.

5.1 The Context Cognition

Context has been defined in many publications (e.g., in [11] [12]). A primary functionality of the existing context awareness system is to sense both the state

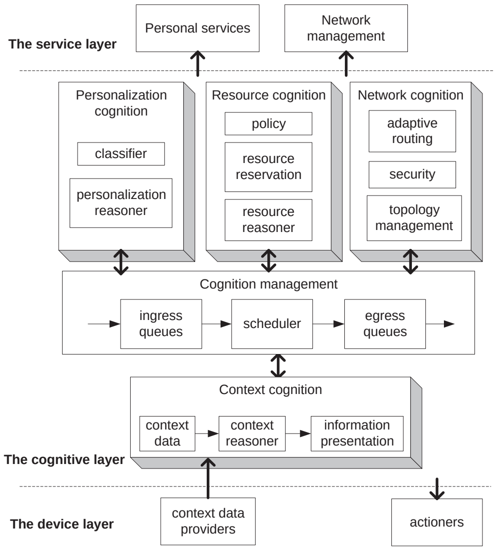

Fig. 3. The cognitive architecture overview.

of the resources internal to the devices (e.g., the amount of memory, energy level of batteries) and external resources (e.g., location, velocity, available services in the proximity of the user, and user activity). We extend this functionality with the capability of learning by generating rules from the context history, reasoning under uncertainty, and providing context information to other components. For example, suppose that, most of time when one comes to the office, one turns on the light, the heating and computer and then reads. However, this is not always the case. Sometimes, one reads emails with a cup of coffee. Sometimes, one comes to the office just to pick up something one forgot. This presence of uncertainty radically changes the way the context cognition makes decisions. It needs to infer from these uncertainties, make a rational choice and act under uncertainty. As illustrated in Fig. 3, the work flow of the context cognition consists of three steps. First context data is collected from the context data provider in the device layer (e.g., from a sensor network). Then the collected data are reasoned about to deduct context information, such as the user's activity based on his movement. Finally, context information is generated and presented to actors in the device layer or other components of the cognitive layer.

5.2 The Personalization Cognition

The personalization cognition's task is to build a generic model by understanding the needs of each individual. Personalization cognition is responsible for providing two functionalities: classification and personalization reasoning. The *classifier* examines incoming data from the context cognition and classifies the data into a set of personal profiles. This can be based on a system of rules. We provide a method to generate a user profile and classify the user profile data for personalization. The general structure of the method for classification is in the following:

DEFINE PROFILE_NAME AS
((attribute [operator] value)
OR (attribute [operator] value)
AND (attribute [operator] value)
...)

The attributes are typically location, activity, person, and time. The operators are typically =, !=, >, < >=, and <=. A number of (attribute [operator] value) sets is connected by OR or AND operation. Let us give examples of such a set of rules:

- *(Location = office) AND (Activity = come in) AND (Time >= 9:00AM) -> office manager.*
- *(Person ! = hosts) AND (Location = in front of the door) OR (Activity = knocking at the door) -> Guest assistant.*
- *(Sender = Maria) OR (date received <= today) AND (subject = appointment) -> Personal scheduler.*

The *personalization reasoner* mainly deals with learning personal preferences, reasoning about personal preferences to build personal profiles, and invoking actors in the device layer. Figure 4 illustrates an example of using a Bayesian network [13] to infer personal preferences for a home assistant PN application. In this model, we describe six types of services: data services, telephone services, audio services, video services, diet-related services, and monitoring and controlling services. Each service may trigger different devices involved. If some services can be spontaneously used together, there is a directed link between them. Suppose that while watching TV, one likes to enjoy a cup of tea or juice, so, in this model there is a directed link from TV to either the tea or juice maker. The set of directed links or arrows specify the conditional dependent relationships. They are independent if there is no directed link between nodes. For example, we represent that *time* is independent of *location* by the absence of a directed link. *Location* has a directed link to *phone services* or *data services* and some other nodes. We call *location* the parent of these nodes. When one uses a phone service, one may choose a mobile phone, or a cordless telephone, or a PC to communicate. They are conditionally independent given the use of phone services. Using a Bayesian network, we can calculate the full joint probability distribution of each entry from the conditional probability distribution of each node given its parent.

5.3 The Resource Cognition

Resource cognition has three functionalities: *policy generation, resource reservation and resource reasoning.* The *resource reasoner* keeps track of each device

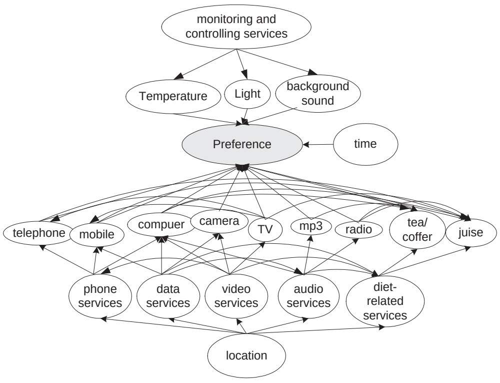

Fig. 4. An example of a Bayesian network for inferring personal preferences.

by monitoring its resource constraints and presenting a reasonable estimate of its status. As mentioned before, a PN consists of a variety of devices with different resource constraints: energy rich or energy poor, nomadic or fixed, etc. Knowing these constraints is very important in selecting a node as an intermediary in adaptive routing or as a platform to run a particular application. For example, to provide high quality media streaming for entertainment, sensors cannot be selected as intermediaries because of their energy poverty and low processing capability. If a node is frequently moving from one place to another, it is not a good candidate for an intermediary either. Figure 5a depicts that the resource reasoner gains knowledge of each device based on its five resource constraints (energy level, memory, processing capability, bandwidth, and mobility) and produces a reasonable estimate of its possible status. The *resource reservation* reserves the devices based on the estimates given by the resource reasoner and releases the devices afterwards. We define three types of device states: idle, reserved, and busy. Figure 5b depicts a state diagram of resource reservation. The *policy generation* defines and generates a set of rules for resource reservation. For example, the small-world policy where devices have only very few relations to a large number of other devices, but have a lot of contacts with a limited number of devices. This is similar to one's address book, where one has a lot of names in the list. Most of the contacts are rarely sought, but with some contact sought very often. The small-world policy can provide priority information of each device for resource reservation and enable resource cognition to easily reserve the resources of a proper device.

We see that resource cognition works as a *a knowledge gainer* of a network and as *a controller* to network cognition.

5.4 The Network Cognition

The primary goal of *network cognition* is to enable the network to organize itself, detect problems, repair or recover itself, and optimize itself. Network cognition

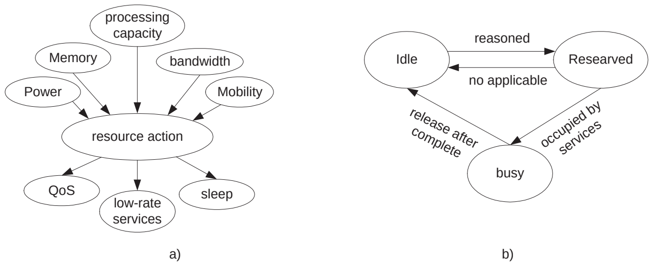

Fig. 5. a) Reasoning from resource constrains for resource reservation. b) State diagram of resource reservation.

mainly consists of three functionalities: *adaptive routing, security management and topology management.* Adaptive routing uses the feedback information from resource cognition, plans a reasonable route and optimizes the route for performance or reliability. Figure 6 illustrates an example of a cumulative learning process for adaptive routing based on the feedback from resource cognition. Consider the following example. Suppose the case of a node A sending packets to a node B via a node C. On knowing that node A needs to send to node C first, it concludes that all packets destined to node B need to be sent to node C. Yet when it knows that node C has been moved out of its (radio) range, which is a feedback from resource cognition, it does not consider node C any longer as an intermediary to node B. The feedback from resource cognition allows node A a faster (e.g., real-time) adaptation, thus preventing unnecessary traffic. Security management should detect possible attacks in real-time, perform vulnerability assessments, and forecast threatening tendencies. Compared to traditional security management which reacts to an attack and eliminates the consequences, this security management enhances this strategy with forecasting and prevention. In case the attack is not prevented, the second react strategy is used to find suitable software to update the system and the system should learn from this experience. Topology management manages the dynamics of a PN, e.g., nodes leaving or joining a cluster. Topology management uses cognitive monitoring to collect data on the current situation of the network and to predict the future situation of the network. It speeds up the adaptation of the network's behavior to accommodate changes and achieves a more efficient use of resources in controlling the potentially large dynamics of the PN.

5.5 Communications: the Cognition Management

The four above mentioned cognition components can work alone for an application. But most often, they need to work together. Cognition management is responsible for the coordination and the interaction of the four cognition components. It consists of an ingress queue, a scheduler, and an egress queue. The scheduler schedules the requests or responses to the cognition components, such as to context cognition, or to resource cognition. In order to guarantee real-time

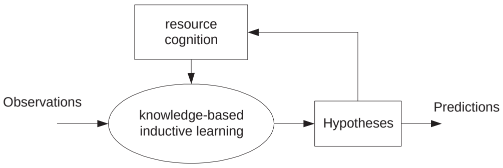

Fig. 6. A cumulative learning process for adaptive routing.

requirements, the scheduler is based on a priority queue and priority threads. The cognition management can also be used by the cognitive layer to discover its peers, to negotiate and to co-operate with them since they are distributed over a PN. For example, a context cognition entity in one place can send its context information to a resource cognition entity in another place where the node is moving to and to reserve resources there. The advantage of using cognition management is that it enables the whole system to be modular and extendible in the sense that the cognition components can be replaced or updated independently without impacting the other components. Further, cognition components can be added to this layer without impacting the integrity of the whole architecture.

6 Services Scenarios

In order to demonstrate the intended functionality of the proposed framework, in this section, we illustrate some service scenarios based on this architecture.

Figure 7 demonstrates how the cognitive architecture is used to manage session mobility (sessions moving between personal devices) in PNs. The user is using his computer and an attached microphone and speaker to have a video conference (which includes a video and audio session, and might contain a data session for text chat.) with his team workers in his office. Around 5:00PM, his personalization cognition reminds him that he needs to pick up his child from school. The context cognition in the office discovers that his mobile with an embedded camera can be available for the continuity of the video conference sessions. The cognition management sends a handoff query to the mobile. The resource cognition of the mobile reasons about the possibility of available resources for the handoff and reserves the resources. The network cognition of the mobile binds the sessions with their network interfaces. Finally, the mobile sends a handoff notification to the computer with the IP address and port numbers of the network interfaces. Before he leaves his office, the sessions have been merged (from three devices to one device) into his mobile assuring continuity. When he is on the move, the context cognition in his mobile predicts that the user is going to his car. The cognition management of the mobile sends again a handoff query to the cognition management of the car. The context cognition of the car discovers that either his mobile or the GPS display with a Bluetooth camera and headset are available for the continuity of the sessions. The resource cognition reasons that the mobile might run out of the battery power and concludes that the GPS display with the camera and the headset is better for providing the service. So the resources of the GPS, the camera and the headset are reserved and the network cognition of the car cluster binds each session with its IP address and port number of the device. Thus the sessions have been split across these three devices and the network cognition manages the communications of these devices. When he is driving home, the personalization cognition of the home cluster reasons that the user would like to use his TV for the video service and his stereo system for the voice service. In the same way, the handoff query is negotiated and the relevant resources at home are reserved. In addition, the personalization cognition at home infers the user might need a cup of coffee. So the coffee machine is ready. Meanwhile the heating is turned to his favourite temperature. When he reaches home, the light is on and the sessions have been migrated to the TV, the webcam and the stereo system.

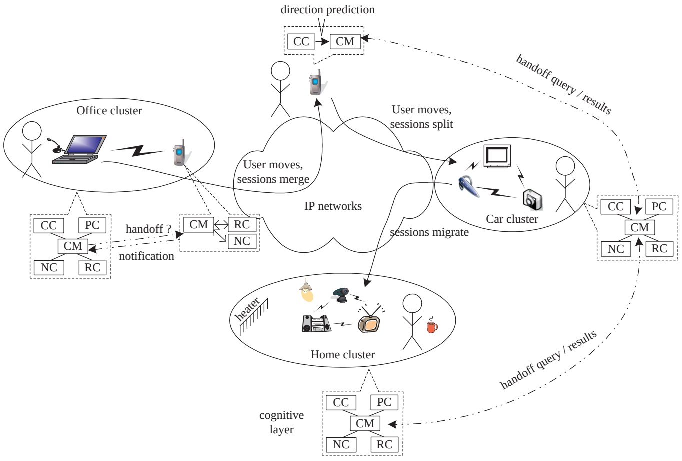

Fig. 7. A scenario of session migration in personal networks.

From the above scenario, we can see that by using cognitive techniques, the proposed architecture enables proactive configuration and adaptation before session migration, merge, and split across devices. Many existing systems suffer handoff delays due to too slow configuration speed (e.g., device start-up and high setup latency). This is exacerbated when stringent security requirements and QoS come into play. In particular for personal applications the maximum allowed configuration time, e.g., to get a device that is turned on to become part of the PN is determined by the patience of the user (often of the order of seconds). The cognitive approach hides the latency by predicting when a configuration will be needed and setting up available resources in advance without the user or applications being aware of this.

7 Research Issues on the Cognitive Architecture Implementation

When implementing the cognitive architecture for PNs, the following issues need to be addressed.

- How are functional entities of the cognitive layer mapped on personal devices of a particular domain? From a device point of view, due to the heterogeneity of personal devices, some devices may be capable of supporting full functional entities, some may not. From the functionality point of view, it may not be necessary to implement all entities in each device. Therefore, whether it is highly centralized or fully distributed in one domain is one of the considerations.
- If it is fully distributed in one domain, what can the minimal set of functionalities be installed on a device? How are the devices capable of providing adaptation in terms of modification of existing features or adding new ones?
- whether the implementation of the cognitive layer is hidden from the service layer or not is a concern. Hiding all the details of the implementation will lead to a very large software code, which is often considered difficult to construct and maintain. Transparency of some functionalities to the service layer will lead to a lightweight implementation in terms of computational complexity and amount of software code. However, it will raise the question on how the service layer can monitor and adapt the behaviors of the cognitive layer according to its needs.
- How smart should the cognitive architecture be? On the one hand, a proactive system can enhance productivity, safety, awareness, and efficiency [14]. On the other hand, a proactive system can also annoy a user and thus defeat the goal.
- How fast are learning algorithms? In order to learn from a person's preferences and to make decisions, the person's behavior needs to be observed and a hypothesis representation needs to be generated from the data. How much observed data is sufficient? How accurate is the hypothesis representation and how much time is required?
- How efficient can the architecture be in managing distributed information over PNs? Since cognition components in the cognitive layer are distributed within a PN environment, data collection and dissemination is a challenge. Can the interaction of the cognition components be tuned to achieve optimal performance? Does the cognition management work efficiently among its peers?
- What the cost of implementing this new functionality on these devices is also a major concern for consumers.

Solutions to these issues should be researched and developed in the future. But at first, we will map proposed reasoning and learning functionalities on personal devices and investigate their efficiency and performance. Meanwhile, we will explore more possible cognitive techniques for personal networks.

8 Conclusions

In this paper, we have explained and given a motivation of why cognitive networking is needed in personal networks. We proposed a cognitive architecture which explores how cognitive technology could be applied into PNs for both personal services and network management. We illustrated some examples which can benefit from using the cognitive techniques and the proposed architecture. This architecture develops a framework for pervasive computing applications and enables distributed intelligent processes to manage personal networks. Finally, we outlined some research issues on implementation of the architecture.

Acknowledgment

The work presented here was funded by the Dutch Ministry of Economical Affairs under the Freeband PNP2008 project. This work expresses the view of the authors and not necessarily the general view of PNP2008.

References

1. Niemegeers, I., de Groot, S.H.: Research issues in ad-hoc distributed personal networking. Special Issue of the journal Wireless and Personal Communication **26**(2-3) (2003) 149–167
2. Weiser, M.: The computer for the 21st century. Scientific American **265**(3) (1991) 94–104
3. Clark, D.D., Partridge, C., Ramming, J.C., Wroclawski, J.T.: A knowledge plane for the internet. In: SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. (2003) 3–10
4. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. IEEE Computer Society **36**(1) (2003) 41–50
5. Dietterich, T., Langley, P.: Machine learning for cognitive networks: Technology assessment and research challenges. Technical report, Oregon State University (2003)
6. : (Carnegie mellon university project aura) http://www.cs.cmu.edu/ aura/.
7. : (Microsoft research project easyliving) http://research.microsoft.com/easyliving/.
8. : (Mit project oxygen) http://oxygen.lcs.mit.edu/Overview.html.
9. Jacobsson, M., Niemegeers, I.: Privacy and anonymity in personal networks. In: Proceedings of the 2nd International Workshop on Pervasive Computing and Communication Security (PerSec'05). (2005)
10. Newell, A.: Unified Theories of Cognition. Harvard College (1990)
11. Schilit, B., Hilbert, D., Trevor, J.: Context-aware communication. IEEE Wireless Communications **9** (2002) 46–54
12. Abowd, G.D., Anind K. Dey, e.a.: Towards a better understanding of context and context-awareness. In: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing. (1999) 304 – 307
13. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall (2003)
14. Kahn, K.C., Culler, D.E.: Ad hoc sensor networks: Proactive computing is coming. Technical report, Intel Corporation (2002)