

# Learning User Preferences in Ubiquitous Systems: A User Study and a Reinforcement Learning Approach

Sofia Zaidenberg, Patrick Reignier, Nadine Mandran

► **To cite this version:**

Sofia Zaidenberg, Patrick Reignier, Nadine Mandran. Learning User Preferences in Ubiquitous Systems: A User Study and a Reinforcement Learning Approach. Harris Papadopoulos and Andreas S. Andreou and Max Bramer. Artificial Intelligence Applications and Innovations, Oct 2010, Larnaca, Cyprus. Springer, 339, pp.336-343, 2010, IFIP Advances in Information and Communication Technology. <10.1007/978-3-642-16239-8\_44>. <hal-00788028>

**HAL Id: hal-00788028**

**<https://hal.inria.fr/hal-00788028>**

Submitted on 13 Feb 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Learning User Preferences in Ubiquitous Systems: a User Study and a Reinforcement Learning Approach

Sofia Zaidenberg<sup>1</sup>, Patrick Reignier<sup>1</sup>, and Nadine Mandran<sup>2</sup>

<sup>1</sup> INRIA,

Sofia.Zaidenberg@inrialpes.fr,

WWW home page:

<http://www-prima.imag.fr/prima/people/zaidenberg/index.php>

<sup>2</sup> LIG

**Abstract.** Our study concerns a virtual assistant, proposing services to the user based on its current perceived activity and situation (ambient intelligence). Instead of asking the user to define his preferences, we acquire them automatically using a reinforcement learning approach. Experiments showed that our system succeeded the learning of user preferences. In order to validate the relevance and usability of such a system, we have first conducted a user study. 26 non-expert subjects were interviewed using a model of the final system. This paper presents the methodology of applying reinforcement learning to a real-world problem with experimental results and the conclusions of the user study.

**Keywords:** Ambient Intelligence, Context-Aware Computing, Personal Assistant, Reinforcement Learning, User Study.

## 1 Introduction

This work fits into the frame of ambient intelligence (AmI) where a ubiquitous system tries to adapt its behavior to the perceived context. Through the assistant, the system aims at helping the user to perform his everyday tasks and at reducing his cognitive load. The assistant could make sure for instance that its user doesn't miss any important meetings by forwarding him reminders from his calendar. If the reminder pops up while the user is away from his computer, the assistant can find him in the building and find a means of informing him of the upcoming event. Defining preferences for such a system is a fastidious task for the user. It would be difficult for him to describe all the situations of interest and to assign an action to them. The solution proposed by [1] is to learn these user preferences.

We study the automatic acquisition of user preferences in ubiquitous environments by applying known machine learning techniques to AmI. In this paper, we will first introduce a qualitative user study conducted on a model of the system we want to build. We will then present our learning agent approach that takes into account the constraints revealed by the user study.

## 2 State of the Art

Numerous researchers have worked on *context-aware computing*. For instance, one of the first generic framework for building context aware applications is the *Context Toolkit* [2]. This toolkit is based on *context widgets* and a distributed infrastructure that hosts the widgets. As an example of context-aware application let us cite the *context broker* [3]. The broker maintains a context model of the part of the environment that it is responsible for and shares this information with other agents or services while respecting several confidentiality levels. The context broker is supplied with information from different sources (sensors, other agents, devices, etc.) and merges this information into a consistent model.

This infrastructure enables applications like “EasyMeeting” [4], where the ubiquitous system helps the user Alice to conduct her seminar by automatically projecting her slides in the meeting room. In this work, the user is offered context-aware, ubiquitous services, but these services are hard-wired and not personalized. A similar example is the *PersonisAD* framework [5]. This distributed framework also maintains a hierarchical context model using reasoning on context proofs provided by applications. Components of the system can be notified of context changes in order to adapt themselves to the new context, but user preferences are defined manually.

Some papers try to inject user models into context-aware systems in order to deal with user preferences. For instance, [6] shows how the two closely related concepts of user modelling and context-awareness could be combined. [7] build user profiles of net surfers capturing their preferences and interests for page suggestions. These profiles are learned using classification techniques.

Our work is closely related to the field of *intelligent inhabited environments* which are living spaces equipped with embedded intelligent systems taking care of the users needs. As stated in [8], adaptation to the user is essential in AmI. [9] use fuzzy logic to learn rules representing the user’s behavior with the aim of automating his usual actions. But this system does not offer new services to the user. In addition, this system learns rules by observing the user, and not by being trained by the user.

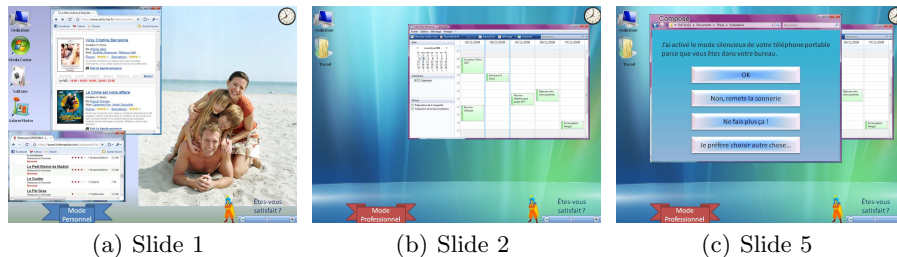
## 3 User Study

The main direction at the beginning of AmI was to build seamless, invisible applications that *weave themselves into the fabric of everyday life until they are indistinguishable from it* [10]. But around 2005, researchers started noticing that *calm computing* is maybe not what people really want [11–13]. *UbiComp technologies are designed not to do things for people but to engage them more actively in what they currently do* [12]. To build smart ubicomp applications, [13] suggested that the best approach is to study user habits and their ways of living and create technologies supporting that and not the opposite, as it was done at the time [14]. This approach was, for instance, adopted by [15, 16].

For this reason, we believe that conducting user studies to validate our research in AmI is fundamental.

The goal of this user study was to measure the expectations and needs of users with regard to an ambient personal assistant. Subjects were 26 active persons, 12 women and 14 men, distributed in age categories as follows: 9 subjects between 18 and 25, 7 between 26 and 40, 7 between 40 and 60, and 3 over 60. None of the subjects had advanced knowledge in computer science.

The study was based on  $\sim 1$  hour interviews with every subject. The interviewer followed a predefined script. The script started with a few open questions about information and communication technologies to evaluate the subject's general knowledge, but also his perception and his uses of such technologies. Then, the interviewer presented our ubiquitous system using a model (an interactive power point presentation: some of the slides are shown figure 1). This interacting powerpoint was exposing a simple scenario about the user's laptop. The scenario starts in the morning and the user is at home, browsing for movies and restaurants (figure 1(a)). When he arrives at work, the laptop automatically switches to the user's "work" setting (figure 1(b)). Then, the assistant turns the user's cellphone to vibrate and displays a message about this action. The user can ask for an explanation about this action and choose to undo it or select another action for this situation (figure 1(c)). At the end of the day, the system switched back to the "home" setting. The interviewer explained also orally other examples of services that could be offered by the assistant.



**Fig. 1.** A few slides from the model used to present our system to the subjects.

After the presentation, the subject was asked for his opinion about such a system. He could freely express the advantages and drawbacks of what he saw and the situations in which he thought the assistant was particularly useful or interesting. This gave him the opportunity to talk about ubiquitous assistants in general and about what their usage implies for his everyday life. Another goal of the conversation was to determine the acceptability of the system. The interviewer asked the following questions:

- “If the assistant learns badly, if it offers you wrong services at wrong times, what would be your reaction?”
- “If the assistant makes mistakes, but you know that he is learning to adapt to your behavior, would you give him a chance?”

- “Would you accept to spend some time to answer questions to make the assistant learn more quickly?”
- “What would you gain from getting an explanation about the assistant’s decisions?”

We were also interested in finding out if the subjects would feel observed and fear that their privacy was in jeopardy. If they would not bring the subject up themselves, we would ask questions about this.

### 3.1 Results

After analyzing all the interviews, it appeared that 44% of subjects were interested in our assistant, and 13% were conquered. Interested persons share the same profile: they are very active, very busy in their professional as well as personal lives, they suffer from cognitive overload and would appreciate some help to organize their schedule. Other noticeable observations standing out from the interviews are the following:

- Having a *learning* assistant is considered as a plus by users. In fact, subjects felt a learning system would be more reliable since it would respond to their own training.
- Users prefer a gradual training versus a heavy configuration at the beginning.
- This training must indeed be simple and pleasant (“one click”).
- The initial learning phase must be short (one to three weeks).
- It is absolutely necessary for the assistant to be able to explain its decisions. This aspect was particularly discussed by [17].
- The amount of interactions wanted between the user and the assistant varies from one subject to another. Some accept only to give one-click rewards while others would be happy to give more inputs to the system. This confirms that people are interested in *engaging* systems, as stated by [12]. For those users, we could add an optional debriefing phase where the assistant goes through the learned behavior and the user corrects or approves it.
- Mistakes made by the system are accepted to some extent as long as the user knows that the system is learning and as the system is useful enough to the user. But errors must not have critical consequences. Users always want to remain in control, to have the last word over the system and even have a “red button” to stop the whole system at any time.
- Some subjects pointed out that the assistant could even reveal to them their own automatic and subconscious customs.
- A recurrent worry expressed by interviewees is the fear of becoming dependant of a system that cares for them and becoming unable of living without it (what if the system is broken-down?).

This user study justifies our ubiquitous assistant since a sizeable part of interviewed subjects were prone to using it. Points listed above give us constraints to respect in our system. They will be listed in the next section that will present our approach.

## 4 The Ubiquitous Assistant

### 4.1 Constraints

We want to build a personal assistant whose behavior is learned from user's inputs. We have to respect several constraints:

- (a) The system must not be a black box. As detailed in [17], a context-aware system can not pretend to understand all of the user's context, thus it must be responsible about its limitations. It must be able to explain to the user what it knows, how it knows it, and what it is doing about it. The user will trust the assistant (even if it fails) if he can understand its internal functioning.
- (b) The training is going to be performed by the user thus it must be simple, non intrusive and it must not put a burden on the user.
- (c) The training period must be short, unless the user changes preferences.
- (d) The system must have an initial behavior that is not incoherent.

Several learning techniques have been considered to solve this problem. For instance, [18] uses supervised learning where the user provides offline feedback on the sequence of events that happened during the day, providing for every sequence the correct action. We prefer to gather user feedback online (in context), collecting only positive or negative feedback to simplify the interaction. Reinforcement learning (see [19] for instance) is a possible solution, but it must be adapted to our particular constraints.

### 4.2 Reinforcement Learning of User Preferences

Our goal is to learn the behavior of a ubiquitous assistant based on the satisfaction or disapproval of the user toward the assistant's actions. Reinforcement learning (RL) approaches are based on state space exploration. It needs many experiences (and many of them will seem inappropriate because of exploration) to converge to the correct solution. As the user is directly implied in those experiences, he may quickly reject the system. To minimize user interaction in the learning process, we have used indirect reinforcement learning.

Indirect RL was proposed by Sutton [20] and consists in learning a model of the environment, a *world model*, in the guise of the transition function ( $\mathcal{P}$ ) and reward function ( $\mathcal{R}$ ). A part of the trials is then done in the world model instead of the real world (this is the DYNA-Q algorithm [20]).

The world model is learned using supervised learning on a history of real interactions between the user, the assistant and the environment. This history contains tree kinds of triplets:  $(s, a, s')$ ,  $(s, e, s')$  and  $(s, a, r)$ , where  $s$  is the previous state of the environment,  $a$  is an action taken by the assistant,  $e$  is an event beyond the assistant (for instance the user coming inside the office or receiving an email). Both  $a$  and  $e$  cause a change in the environment's state, the next state being  $s'$ . Finally,  $r$  is the reward given by the user when the assistant took action  $a$  in state  $s$ . The user can give an explicit reward using a

slider on a graphical interface. He does not have to worry about the numerical value of  $r$ , but only to its relative position between a minimum and a maximum. Implicit reward could also be gathered from clues or from recognizing the user’s emotional state after an action was taken. For example, if the assistant opens the email client and the user closes it straight away, the implicit reward is negative. Constraint (b) defined section 4 is thus respected.

Our state is modeled by a set of first order predicates. Each predicate transcribes an observable part of the environment, (the output of one of our sensors). For instance, we have the predicates `entrance(isAlone, friendlyName, btAddress)`<sup>3</sup> and `hasUnreadMail(from, to, subject, body)`. These predicates are readable by humans: the users can understand the internal state of the assistant at all times. This assures the respect of constraint (a) defined section 4. Our states space is huge. For instance, the states “`entrance(friendlyName = Bob, ...)`” and “`entrance(friendlyName = Alice, ...)`” are two different states that must be explored both to learn a behavior. To speed up the learning process, we must generalize the observed examples. We generalize states by replacing values with wildcards: “`<+>`” means any value but “`<null>`” and “`<*>`” means any value. Initially, we merge all these similar states and we deal with “super-states” such as “someone entered the office”. Eventually, we will need to split some of these super-states and adapt different behaviors for example when the boss sends an email and when a newsletter sends an email. This way, we make best use of every example. Indirect RL and states generalization accelerates the learning phase, respecting constraint (c). More details about this approach are given in [21].

## 5 Experimental Results

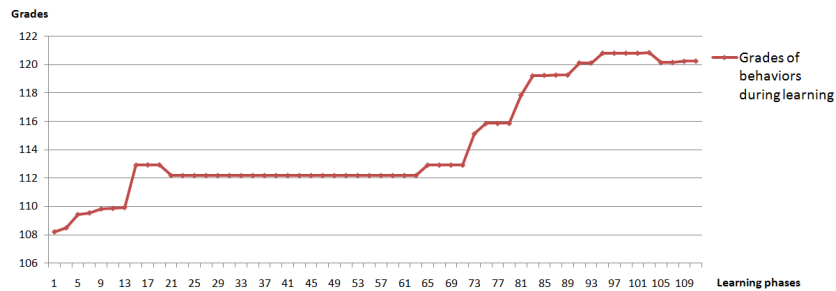
We created a prototype of an AmI system and the assistant. An experimenter is put in the position of the user and creates interactions with the environment using a graphical interface for convenience. Based on sensor events sent by the user through the interface, the assistant learns a behavior and puts it into practice. The experimenter, who gives rewards, grades each behavior by indicating if he agrees with every action that has been associated with a situation by the RL algorithm. Those grades are presented figure 2. This curve shows that when the assistant has the opportunity to observe new situations, it quickly learns what to do (phases 1 to 20 and 65 to 110). When nothing new happens, the behavior stays stable (phases 20 to 65).

## 6 Conclusion

The aim of this research is to investigate AmI systems and their acceptability by users. We exploit a ubiquitous system to provide personalized, context-aware services to users. The personalization of the system is achieved by learning user

---

<sup>3</sup> The arguments correspond to attributes of a bluetooth device, which is how we detect the presence of users for now.



**Fig. 2.** Grades of behaviors produced by the RL algorithm.

preferences during interactions. In order to validate the relevance of such an application, we conducted a user study. This study approved our work and revealed constraints for the acceptability of such a system. We developed a method and a prototype respecting those requirements and showed the correct functioning of our ubiquitous assistant.

## References

1. Maes, P.: Agents that reduce work and information overload. *Commun. ACM* **37**(7) (July 1994) 30–40
2. Dey, A.K., Abowd, G.D.: The context toolkit: Aiding the development of context-aware applications. In: *The Workshop on Software Engineering for Wearable and Pervasive Computing*, Limerick, Ireland (June 2000)
3. Chen, H., Finin, T., Joshi, A.: A context broker for building smart meeting rooms. In: *Proceedings of AAAI Spring Symposium*, Stanford, California, AAAI Press, Menlo Park, CA (2004) 53–60
4. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review* **18**(3) (2004) 197–207
5. Assad, M., Carmichael, D., Kay, J., Kummerfeld, B.: PersonisAD: Distributed, active, scrutable model framework for context-aware services. In LaMarca, A., Langheinrich, M., Truong, K.N., eds.: *Proceedings of PERVASIVE 2007*. Volume 4480 of *Lecture Notes in Computer Science.*, Springer (2007) 55 – 72
6. Byun, H.E., Cheverst, K.: Exploiting user models and context-awareness to support personal daily activities. In: *Workshop in UM2001 on User Modeling for Context-Aware Applications*. (2001)
7. Godoy, D., Amandi, A.: User profiling for web page filtering. *Internet Computing, IEEE* **9**(4) (2005) 56–64
8. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.: Scenarios for ambient intelligence in 2010. Technical report, ISTAG (2001)
9. Doctor, F., Hagaras, H., Callaghan, V.: An intelligent fuzzy agent approach for realising ambient intelligence in intelligent inhabited environments. In: *IEEE Tr on Systems, Man and Cybernetics, Part A*. Volume 35. (2005) 55–65
10. Weiser, M.: The computer for the 21st century. *Scientific American* **265**(3) (1991) 66–75



11. Dourish, P.: What we talk about when we talk about context. *Personal Ubiquitous Comput.* **8**(1) (2004) 19–30
12. Rogers, Y.: Moving on from weiser’s vision of calm computing: Engaging ubicomp experiences. In: *UbiComp 2006*, Springer (2006) 404–421
13. José, R.: *UbiComp 2.0: From envisioning a future to being part of a new reality.* Plenary Talk at the UCAmI’08 conference (2008)
14. Barton, J., Pierce, J.: Quantifying magic in ubicomp systems scenarios. In: Position Paper for *UbiSys 2006*, Orange County, California (2006)
15. Pascoe, J., Thomson, K., Rodrigues, H.: Context-Awareness in the Wild: An Investigation into the Existing Uses of Context in Everyday Life. *Lecture Notes in Computer Science*. In: *OTM 2007 Workshops*. (2007) 193–202
16. Taylor, A.S., Harper, R., Swan, L., Izadi, S., Sellen, A., Perry, M.: Homes that make us smart. *Personal and Ubiquitous Computing* **11**(5) (2007) 383–393
17. Bellotti, V., Edwards, K.: Intelligibility and accountability: human considerations in context-aware systems. *HCI* **16**(2) (2001) 193–212
18. Brdiczka, O., Crowley, J.L., Reignier, P.: Learning situation models for providing context-aware services. In: *Proceedings of HCI International*. Volume 4555 of LNCS., Springer (2007) 23–32
19. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press (1998)
20. Sutton, R.S.: Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In: *ICML1990*. (1990) 216–224
21. Zaidenberg, S., Reignier, P., Crowley, J.L. In: *Reinforcement Learning of Context Models for a Ubiquitous Personal Assistant*. Volume Volume 51/2009 of *Advances in Soft Computing*. Springer Berlin / Heidelberg (september 2008) 254–264