Probably Asked Questions: Intelligibility through Question Generation

Sebastian Bader

Department of Computer Science, University of Rostock
sebastian.bader@uni-rostock.de

Abstract. Intelligibility of dynamic and heterogeneous device ensembles is a major problem in the area of ubiquitous computing. The PAQ-approach generates a set of Probably Asked Questions together with answers based on the context, that is the current state of the world and the possible intentions of the user. By providing this list of questions and answers, we enable users to access the context, that is the state of the world and to control their environment by taking actions as explained in generated answers.

1 Introduction and Motivation

Assume a meeting room equipped with real ubiquitous presentation technology, that is technology embedded invisibly into the environment, and a user which has never been there before to enter the room, carrying a laptop containing a presentation and other mobile devices. Since the user has never been to the room, she is not aware of the technology embedded into it. Therefore, she does not even know which services are provided by the room and has thus no chance of using them. Furthermore, her mobile devices should integrate with the environment to provide further services. The environment on the other hand, knows the available technology and might also know how users normally behave. Therefore, it can assist the user proactively by setting up the devices to accommodate the user's needs. Nonetheless, the user usually has to take certain actions activating the assistance functionality. For example connecting the laptop's video output might trigger the room to set up a presentation mode by turning on the projector, moving down the projection screen and setting up the light appropriately.

The *Probably Asked Question (PAQ)* approach detailed below, addresses the following problems in the area of dynamic environments: (i) How can a user learn which actions to take? (ii) How to provide explanations for actions taken by the automatic assistant. As for instance argued in [Dey09], intelligibility is one of the big challenges for the near future of ubiquitous computing. The required technology to build intelligent environments is available, but approaches to explain the reactions and behaviour are still under research [LD11a]. In particular, while considering dynamic and heterogeneous device ensembles, it is yet completely unclear how to design appropriate user interfaces and how to provide explanations.

Some approaches have been investigated in the ExaCt [RBTL11] and MRC [CKPZW10] workshops and the Context conferences [KRRBV07] during the last years, but most ideas focus on the interpretation and explanation of context in a rather fixed setting, namely with respect to one particular application. Here, we focus on heterogeneous and dynamic ensembles of devices and users. That is, the concrete ensemble is not known in advance and sensors, actuators or application logic can enter and leave the ensemble at any time.

Asking an expert or teacher is one way to approach an unknown environment. Using the PAQ approach, we enable the environment to be the teacher and at the same time to enumerate likely questions. Based on basically the same technology, the environment is also able to provide explanations for actions taken by some automatic assistant before. The idea of the PAQ approach is to provide a list of probably asked questions to tell the user what is possible and how to achieve a possible goal. First we discuss some preliminary notions. In Section 3 we introduce our approach in detail.

2 Preliminaries

Below we discuss a number of notions needed for the remainder of the paper. First, we discuss *human behaviour modelling* and *intention recognition*, then we present *heterogeneous dynamic ensembles* as problem domain to apply the PAQ approach and *strategy synthesis*. In Section 3 we show how to combine those ingredients into a system able to explain the behaviour of a dynamic ensemble.

Human Behaviour Modelling and Intention Recognition. Different approaches have been taken to model the behaviour of a human or of groups, employing different methodologies [LPFK07,PFP+04,CCH+08,CNBH11]. Based on models of human behaviour and available sensor data, researchers try to predict future user goals, that is they try to recognise the user's intention.

Heterogeneous Dynamic Ensembles and Strategy Synthesis. Smart environment should support their users proactively by providing assistance functionality as much as possible while achieving their usual tasks. Several such system have been developed in the past, for example [KOA+99,HCC+04,CYD06]. For a general introduction into the area we refer to [CD05]. Building and controlling such a system is a complex task, because such environments are heterogeneous and dynamic with respect to their components. The idea of ubiquitous computing users is to embed the technology invisibly into the environment. Therefore, in particular new users face the problem of controlling the technology, because they are not even aware of the available functionality.

After recognising the user's goals, the ensemble needs to determine actions to support the user. Instead of hard-wiring those actions, suitable action sequences can be computed by employing action description languages like PDDL [GHL+09]. Those allow to compute, based on the current and the desired state of the world a sequence of actions to be taken, which fulfil the given goals. Example 2 shows the formalisation of some actions.

3 Intelligibility by Providing Probably Asked Questions

As mentioned above, the intelligibility of ubiquitous environments is an important issue. The environment needs to provide some means of understanding it, its dynamic and possible ways to influence it. Here we propose an approach which provides a list of probably asked questions (PAQs) to the user. That is, a list of questions a new user probably has. As shown below, the number of potential questions is to big to be comprehensible, therefore, the important ones need to be selected and shown to the user. By providing the list of PAQs, we hope to provide

1. an easy explanation of the current state of the environment – captured in answers to *why* questions.
2. an adaptive manual, showing the usage of the environment – captured in answers to *how* questions.
3. a way of understanding error and log messages – captured in *why* questions.

The PAQ approach is based on the following ingredients:

- *Intention analysis* to infer the current intentions of the user, that is the potential goals, called *expectables*, the user might pursue.
- *Strategy synthesis* to infer the possible goals, that is the subset of potential goals, called *intendables*, which are achievable in the current situation.
- *Question generation* to compute a set of probably asked questions based on the possible and potential goals.

3.1 Potential Questions

Before showing how to infer probably asked questions, we discuss potential questions which might occur. Those can be grouped as follows:

How-Questions provide some kind of context aware manual for the environment. Potential questions are 'How to move my presentation from surface A to surface B?', 'How to control the temperature?', 'How to start a presentation?'. Please keep in mind that the devices themselves are not visible, that is the user neither knows which kind of technology is available nor how to control it.

Why-Questions enable the user to understand the current state of the environment. They include questions like, 'Why is the lamp dimmed?', 'Why is my presentation shown on surface B?', 'Why has the projection surface moved down?' and 'Why is my presentation not shown on surface A?'. By generating those questions and the corresponding answers, in particular new users are able to understand the reactions of the environment. They furthermore allow to understand error messages and erroneous reactions of the environment.

Other types of questions, like *who*, *whose*, *when* and *what* are possible as well [LD11b] and will be addressed within the PAQ approach as future work. Below we will focus on *how* and *why* questions. Unfortunately, even those are too many. Assume a meeting room containing two projectors and surfaces, four lights and two computers, that is, in total ten devices with 4 properties in average. By generating question of the type '*Why has property X of device Y the value Z*', we end up with about 40 questions. This shows that we can not simply generate all possible questions, but need to select the important ones, namely those the user would probably ask in the current situation. Below, we show how to compute those interesting questions.

3.2 Computing Expectable User Goals

Humans usually behave somehow goal directed, that is they perform actions to achieve a desired state of the environment. Researchers in the area of behaviour modelling try to capture the human behaviour using formal models. Those models, like for example task trees or logical theories can also be used to predict the future and thus allow to infer potential user goals. Assuming the human behaviour expressed as a logical theory, as shown in Example 1.

Example 1. The goals expectable in a meeting room can be represented as follows:

E_1. True $\Rightarrow$ expectable(give-presentation)
E_2. not(laptop-connected) $\Rightarrow$ expectable(save-projector-energy)

The first rule allows to infer that it can be expected the user likes to give a presentation without further preconditions, and the second describes the fact that the projector should be turned off if no laptop is connected.

The set $\mathcal{E}(x)$ of expectable user goals with respect to the current state of the world x is computed as follows:

$$\mathcal{E}(x) := \{g \mid x \vdash \text{expectable}(g)\}$$

Please note, that the rules above specify necessary condition only. There are many other conditions to be satisfied such that giving a presentation is possible. There must be a projector, there must be a surface, the laptop must be connected to the projector, etc. But those are conditions depending on the environment and are irrelevant while modelling the humans behaviour and intentions. Below, we show how to incorporate those by computing the set of intendable goals.

3.3 Computing Intendable User Goals

Given a current state of the world x and a set of available actions $\mathcal{A}$, performable by either the user or by devices. Without discussing the details here, we assume the actions to be described by some PDDL-like formalisation. We furthermore assume that all actions are annotated with the agent capable of performing it as shown in Example 2.

Example 2. The connection of the laptop performed by the user and switching on the projector performed by the projector can be formalised as follows:

```
(:action connect-laptop       :agent user
 :precondition (not laptop-connected)
 :effect laptop-connected )

(:action disconnect-laptop    :agent user
 :precondition laptop-connected
 :effect (not laptop-connected) )

(:action turnOn-proj          :agent projector
 :precondition (not isOn-proj)
 :effect isOn-proj )

(:action turnOff-proj         :agent projector
 :precondition isOn-proj
 :effect (and (not isOn-proj) save-projector-energy) )

(:action present              :agent user
 :precondition (and isOn-proj laptop-connected)
 :effect give-presentation )
```

The code shows a propositional formalisation sufficient for our small example here. PDDL does also allow the usage of variables and parameters of actions. ◦

The set of intendable user goals $\mathcal{I}(x)$ is the set of goals which are achievable with respect to x and $\mathcal{A}$:

$$\mathcal{I}(x) := \{g \mid \exists l : \mathcal{A}^* \text{ such that } l(x) \models g\}$$

That is, the set of intendables is defined to be the set of all goals g such that there exists a sequence of actions l leading from the current state x to a state in which g holds. Please note, that the set of expectable and intendable goals can also be defined with respect to a given user by adding another parameter as done in [Kir11].

3.4 Question Generation

Next, we discuss the generation of questions. First we concentrate on the generation of *how* questions, based on the intendable and expectable goals described above. Afterwards, we discuss the generation of *why* questions.

The set of intendable goals does usually include the whole state space, because most actions can be undone by other actions and thus every state is accessible from every other state of the world. Therefore, the set of intendable user goals needs to be intersected with the expectable ones. The intersection of both sets contains exactly those goals which are achievable from the current state of the world and expectable with respect to the behaviour model. It is constructed, by first computing the set of expectable goals and then trying to find a plan satisfying the goals. All expectable goals for which a plan exists belong to the intersection. While computing the intersection $\mathcal{G}(x)$, we also remember the plans leading to the goal state as follows:

$$\mathcal{G}(x) := \{(g, l) \mid g \in \mathcal{E}(x) \text{ and } \exists l : \mathcal{A}^* \text{ such that } l(x) \models g\}$$

This set of goals and corresponding plans can directly be converted into a set of *how* questions. For this, we first collect all those goals and plans which involve user actions:

$$\mathcal{G}'(x) := \{(g, l') \mid (g, l) \in \mathcal{G}(x),\ l' := [a \mid a \in l \text{ and } \operatorname{ag}(a) = \text{user}] \text{ and } l' \neq \emptyset\}$$

The set $\mathcal{G}'(x)$ contains pairs of goals and sequences of actions required by the user to actually achieve the goal. From this set, we compute the set of question - answer pairs as follows:

$$\begin{aligned}\mathcal{Q}_{\mathsf{how}}(x) := \{(q, a) \mid (g, l') \in \mathcal{G}'(x),\ & q := \text{‘\textit{How to achieve} '} + g + \text{‘\textit{?}'} \\ & a := \text{‘\textit{By performing} '} + l' + \text{‘\textit{!}'}\}\end{aligned}$$

A list of *why* questions can be generated as follows. Instead of forecasting the future by generating plans, the executed plans need to be remembered. In particular, we need to remember the pairs of goal and corresponding user action which led to a state update of a device. This information is stored in a map $\mathcal{M}$ linking device (d) property (p) pairs to value (v) goal (g) action (a) triples. Based on this map we generate a set of *why* questions as follows:

$$\begin{aligned}\mathcal{Q}_{\mathsf{why}}(x) := \{(q, a) \mid ((d, p) \mapsto (v, g, a)) \in \mathcal{M} & \\ q := \text{‘\textit{Why has} '} + p + \text{‘ \textit{of} '} + d + \text{‘ \textit{the value} '} + v + \text{‘\textit{?}'} & \\ a := \text{‘\textit{Because you performed} '} + a + \text{‘ \textit{to achieve} '} + g + \text{‘\textit{.}'}\} &\end{aligned}$$

Please note that at the moment we are concerned with the general concept of PAQs. The generation of more human-like questions and answers is subject to future work, as well as the generation of other types of questions.

3.5 Question Ordering

Above, we showed how to generate a set of questions based on the current context and intentions of the user. Even though the set is limited to intendable goals, that is goals which can actually be achieved from the current state, the list of questions might nonetheless contain to many entries. Therefore, we need to order them based on their appropriateness – which still has to be defined exactly. Possible candidates include the ordering by

- likelihood of underlying intention (Most intention recognition systems employ probabilistic models, the probability attached to goals can be used to order the questions.)
- length of plan (Shorter plans might by more likely.)

– total cost of actions taken (Attaching costs to device actions enables the controller to compute the total cost to achieve a goal. Goals which are cheaper might be more likely.)
– user experience (Based on the user's experience, some questions are more unlikely than others.)

Those different orderings need to be evaluated, which is subject to future work.

3.6 Automatic Assistance

Based on the set of intendable and expectable goals, an automatic assistance can be defined. The technical details are beyond the scope of this paper, but the idea is as follows: Whenever the user takes actions leading to an expectable goal state, the necessary device actions can be executed automatically. This has been discussed in detail in [Bad10,BL11,KRBK11,BD11].

3.7 A Worked Example

Below we discuss a simple example showing the general idea of the PAQ approach. As in the examples above, we assume a meeting room, equipped with a projector. Expectable user actions are formalised in Example 1 and available actions are shown in Example 2.

Starting from an initial state x_1 in which the projector is turned off and the laptop is not connected, the following two goals are expectable:

$$\mathcal{E}(x_1) = \{\text{give-presentation}, \text{save-projector-energy}\}$$

Both goals are also intendable, because there are actions sequences connecting x_1 with a state satisfying the goals:

$$l_{\text{give-presentation}} = [\text{connect-laptop}_{\text{user}}, \text{turnOn-proj}_{\text{projector}}, \text{present}_{\text{user}}]$$
$$l_{\text{save-projector-energy}} = [\text{turnOn-proj}_{\text{projector}}, \text{turnOff-proj}_{\text{projector}}]$$

The resulting set $\mathcal{G}'(x_1)$ contains only the goal give-presentation, because the user is not involved in the plan to save-energy:

$$\mathcal{G}'(x_1) = \{(\text{give-presentation}, l_{\text{give-presentation}})\}$$

The resulting set $\mathcal{Q}_{\mathsf{how}}(x_1)$ of *how* questions contains the following pair of question and answer:

$$q = \text{`How to achieve ``give-presentation''?'}$$
$$a = \text{`By performing ``connect-laptop'' and ``present''.'}$$

After connecting the laptop, the automatic assistance would turn on the projector and the presentation starts. Afterwards, the device property map $\mathcal{M}$ in state x_2 looks as follows:

$$\mathcal{M} = \{(\text{projector}, \text{on}) \mapsto (\text{true}, \text{give-presentation}, \text{connect-laptop})\}$$

This results in the following set $\mathcal{Q}_{\mathsf{why}}(x_2)$ of *why* questions:

$q =$ 'Why has property "on" of "projector" the value "true"?'
$a =$ 'Because you performed "connect-laptop" to achieve "give-presentation".'

This simple examples are only meant to show the general idea of the PAQ approach. There are yet a number of open issues discussed below.

4 Conclusions and Open Problems

The PAQ approach presented above provides a context dependent list of probably asked questions. Those questions are meant (i) to tell lay users which options they have and how to achieve their goals, and (ii) to explain actions taken in the past. It is based on the automatic recognition of user intentions and the construction of plans to support the user. In particular, we focus on dynamic and heterogeneous ensembles in which all actions are specified formally and background knowledge on user's behaviour is available. Please note that the formalisation of the devices as well as human behaviour models can enter and leave the ensemble dynamically, because they do not depend on each other. We believe that a list of questions and answers are a very natural way to explain the capabilities and actions taken by a distributed heterogeneous device ensemble. The answers described above provide information on a very high level of abstraction. Based on those, more fine-grained answers need to be generated giving details on the inference procedure as done for example in [LD10].

After adapting the idea to our lab, we will evaluate the approach and extend it to other types of questions as discussed in [LD11a]. The generation of more natural questions and answers as well as the user adaptation are also subject to future work. For example, if a user knows how to set up a presentation, the corresponding question can be moved down the list.

References

[Bad10] S. Bader. A logic (programming) based controller for smart environments. In B. Ludwig, S. Mandl, and F. Michahelles, editors, *Proceedings of the workshop Context Aware Intelligent Assistance, held at KI-2010*, pages 1–10, Karlsruhe, Germany, September 2010.

[BD11] S. Bader and M. Dyrba. Goalaviour-based control of heterogeneous and distributed smart environments. In *Proceedings of the 7th International Conference on Intelligent Environments - IE'11*, July 2011.

[BL11] S. Bader and R. Leistikow. Levels of adaptation and control. In A. Abraham, J. Corchado, S. González, and J. De Paz Santana, editors, *International Symposium on Distributed Computing and Artificial Intelligence*, volume 91 of *Advances in Intelligent and Soft Computing*, pages 385–388. Springer Berlin / Heidelberg, April 2011.

[CCH+08] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, P. Klasnja, K. Koscher, J.A. Landay, J. Lester, D. Wyatt, and D. Haehnel. The mobile sensing platform: An embedded activity recognition system. *Pervasive Computing, IEEE*, 7(2):32 – 41, April 2008.

[CD05] D. J. Cook and S. K. Das. *Smart Environments*. Wiley, 2005.

[CKPZW10] J. Cassens, A. Kofod-Petersen, M. Silva Zacarias, and R. K. Wegener, editors. *Modeling and Reasoning in Context*, number 618 in CEUR Workshop Proceedings, Lisbon, Portugal, August 2010. CEUR-WS.org.

[CNBH11] L. Chen, C. D. Nugent, J. Biswas, and J. Hoey, editors. *Activity Recognition in Pervasive Intelligent Environments*, volume 4 of *Atlantis Ambient and Pervasive Intelligence*. Atlantis Press, May 2011.

[CYD06] D. Cook, M. Youngblood, and S. Das. A multi-agent approach to controlling a smart environment. In Juan Augusto and Chris Nugent, editors, *Designing Smart Homes*, volume 4008 of *Lecture Notes in Computer Science*, pages 165–182. Springer Berlin / Heidelberg, 2006.

[Dey09] A. K. Dey. Modeling and intelligibility in ambient environments. *J. Ambient Intell. Smart Environ.*, 1:57–62, January 2009.

[GHL+09] A. Gerevini, P. Haslum, D. Long, A. Saetti, and Y. Dimopoulos. Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners. *Artificial Intelligence*, 173(5-6):619–668, 2009.

[HCC+04] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman. Creating an ambient-intelligence environment using embedded agents. *IEEE Intelligent Systems*, 19:12–20, 2004.

[Kir11] T. Kirste. Making use of intentions. Informatik Preprint CS-01-11, Institut für Informatik, Universität Rostock, March 2011.

[KOA+99] C. D. Kidd, R. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. D. Mynatt, T. Starner, and W. Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *Proceedings of the Second International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture*, pages 191–198, 1999.

[KRBK11] F. Krüger, G. Ruscher, S. Bader, and T. Kirste. A context-aware proactive controller for smart environments. *I-COM*, 10(1):41 – 48, April 2011.

[KRRBV07] B. Kokinov, D. C. Richardson, T. R. Roth-Berghofer, and L. Vieu, editors. *Modeling and Using Context 6th International and Interdisciplinary Conference, CONTEXT 2007*, volume 4635. Springer, August 2007.

[LD10] B. Y. Lim and A. K. Dey. Toolkit to support intelligibility in context-aware applications. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 13–22, 2010.

[LD11a] B. Y. Lim and A. K. Dey. Design of an intelligible mobile context-aware application. In *MobileHCI '11*, 2011. To Appear.

[LD11b] B. Y. Lim and A. K. Dey. Investigating intelligibility for uncertain context-aware applications. In *Proc. of Ubicomp'11*, 2011. To Appear.

[LPFK07] L. Liao, D. J. Patterson, D. Fox, and H. A. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171:311–331, 2007.

[PFP+04] M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Fox, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *Pervasive Computing, IEEE*, 3(4):50 – 57, 2004.

[RBTL11] T. Roth-Berghofer, N. Tintarev, and D. B. Leake, editors. *Proc. of Exact 2011*, 2011.