Decision-Theoretic Planning of Navigation Instructions:
Theoretical and Practical Aspects

Thorsten Bohnenberger, Andreas Butz
Department of Computer Science, University of Saarland
P.O. Box 15 11 50, D-66041 Saarbrücken, Germany
{bohnenberger | butz}@cs.uni-sb.de

Abstract

We describe a decision-theoretic approach to providing user oriented and resource-adapted navigation instructions within buildings. Although information is broadcast by infrared senders in a unidirectional manner, we found it possible to treat the problem of assigning each sender both a user-oriented and resource-adapted broadcast program as a fully observable Markov decision problem.

1 Introduction

Navigation within large public buildings like airports or exhibition centers can be facilitated by navigation systems making use of mobile computing devices, so called hand-held computers or personal digital assistants (PDAs). The quality of the instructions given by these devices increases, if they consider the context they are given in, i.e. the current situation of the user. The instructions become even more valuable, if possible future contexts are taken into account.

The projects READY[1] and REAL[2] both focus their work on resource-adaptive cognitive processes. A shared scenario of the projects is a user-oriented airport information system, which considers aspects of resource-adaptation for the guidance of the user through an airport building. The navigation information for passengers and visitors is broadcast by infrared senders and can be received by hand-held computing devices, such as the 3Com $PalmPilot^{©}$ or the IBM $WorkPad^{©}$.

In section 2, we describe how decision-theoretic planning can be exploited to take into consideration current and future contexts, thus a user being instructed in a resource-adapted manner. The chapter includes results achieved by checking the model for the implementation of a simple example. As the more practical aspect of this article, section 3 presents how the necessary communication between the central system and the mobile devices can be realized.

2 Resource-adaptation

To achieve a resource-adapted navigation information broadcast, we need to compute adequate "broadcast programs" for the infrared senders. Let us assume a sender topology as simple as described in figure 1. It contains 6 senders in strategically important locations of a simplified airport building. Each sender will

[1] http://w5.cs.uni-sb.de/∼ready/
[2] http://w5.cs.uni-sb.de/real/

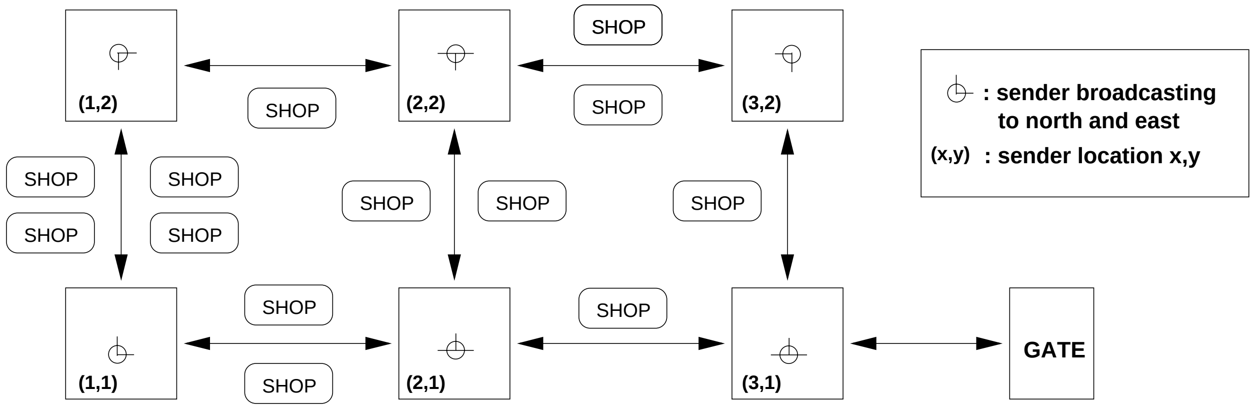

Figure 1: Infrared senders in strategically important locations of a simplified airport building

broadcast different information into several directions, but we are assuming that the presentation mode (i.e. with or without maps or texts) the sender chooses is the same for all directions in any given program.

Let us further assume (for simplicity) that there are exactly two presentation modes: one we want to refer to as "quick mode", which makes a sender transmit only arrows pointing into the direction a user is supposed to take, and the other we want to refer to as "info mode", which gives the user a small and simple map of the area he is currently located in, including an indication where to go next, but also additional information such as the location of nearby shops.

We assume that information given in quick mode (possibly supplemented by the calling to hurry up) in fact makes a user go faster, while the info mode map enriched by additional information rather invites the user to rambling and watching shop windows.

Let us now just pick one passenger whose goal it is to reach a certain gate for boarding. To make the problem more interesting, we assume the system knows that this user wants to complete a certain task, say to buy a gift, before going to the gate. How can we determine his navigation "program" provided the user's preferences concerning an early arrival at the gate and the completion of his task before the departure?

2.1 Model

Obviously there is a trade-off between getting to the gate very quickly, possibly not getting to buy the gift, and quite certainly getting to buy the gift, but reaching the gate barely in time. If the task to be completed is not very important to the user or the time pressure is very high, the guidance to the gate should be rather direct. If the task is very important or the user has got plenty of time left, the guidance can include indirections, making the user passing more shops (if the task is to buy a gift), indicating the shops by making use of the info mode, thus increasing the probability of the user to get the intended task completed.

To solve this problem, we modelled the domain by means of Markov decision processes (MDPs)[3]. MDPs resemble finite state automata with the state transitions (which represent actions) annotated with probabilities, as MDPs are used to model stochastic dynamical systems. What makes the transition models "Markov" is that they meet the Markov property, which claims that the transition probabilities from any given state depend only on the state and not on previous history. In our model, each state in the transition model contains as one feature the location of a sender or the current position of the user respectively (just as you want to view it). Another feature represents the information whether the user has completed the

[3]Our approach was inspired by a simple example given in section 'Making Complex Decisions' of [3] (pp. 498 – 507).

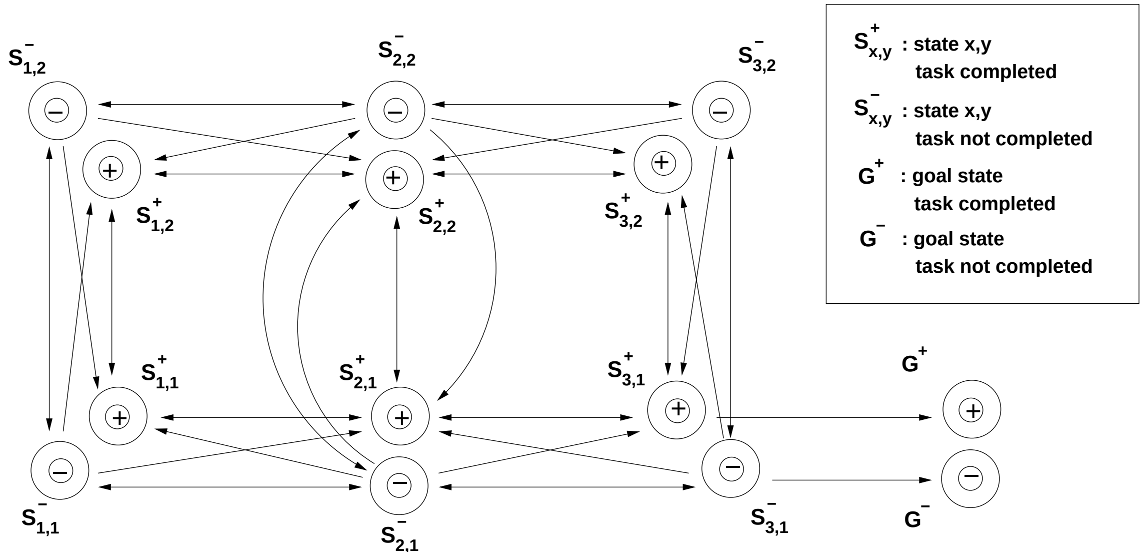

Figure 2: Sketch of the transition model

intended task (+) or not (-). Thus in our simple example, there are two nodes for each location (figure 2). The actions we introduce integrate the information about the direction, the user is to be guided to and the presentation mode, in which the information is given, i.e. an example of an action would be 'go left, quick mode' or 'go ahead, info mode'. Note that we later refer to the directions as 'north', 'east', 'south' and 'west', which have to be mapped to 'left', 'right', 'ahead' and 'back' relative to the direction in which the information is broadcast. The costs of actions can be interpreted as the time that we expect a user to need to get to an adjacent sender. The costs of instructions given in different modes differ indeed, since we can expect a user invited to hurry to reach the adjacent sender quicker than one who was provided with a map including additional information and not inviting to hurry. On the other hand using the info mode increases the probability for the user to get his task completed by indicating adequate shops on the map (figure 3). How successful a certain mode for the completion of a task really is (depending on the shop density of the according location) could be determined by empirical studies, e.g. the evaluation of log-files of prototypical users. The results could be incorporated by methods from the area of machine learning. This is beyond the scope of this article.

After we have modelled our domain as described above, we can use any of the standard algorithms, e.g. *value iteration* or *policy iteration* [3, 2], to determine the resource-adapted "program" (*broadcast policy*) for a user in his current situation. The broadcast policies have to be recomputed as the resource situation changes over time.

The basic idea of the *value iteration* algorithm is to calculate a utility (or value) for each state; these state utilities can then be straightforwardly used to select the optimal action in each state. The utility of a goal state is simply the reward associated with it. The utility of a nongoal state is defined as the sum of (a) the cost (a negative quantity) of the optimal action in that state and (b) the expected utility of that action—which is in turn determined by the utilities of the states reachable through that action, weighted by the probabilities that they will be attained. This definition of a state utility is summarized in equation1:

$$V^*(s) = R(s) + \max_{a \in A}\{C(a) + \sum_{s' \in S} Pr(s'|a,s)V^*(s')\}, \quad (1)$$

where $V^*(s)$ is the utility for state s, $R(s)$ is the reward for s, a is a possible action with costs $C(a)$ and $Pr(s'|a,s)$ the probability that the user gets to state s' if he does action a in state s.

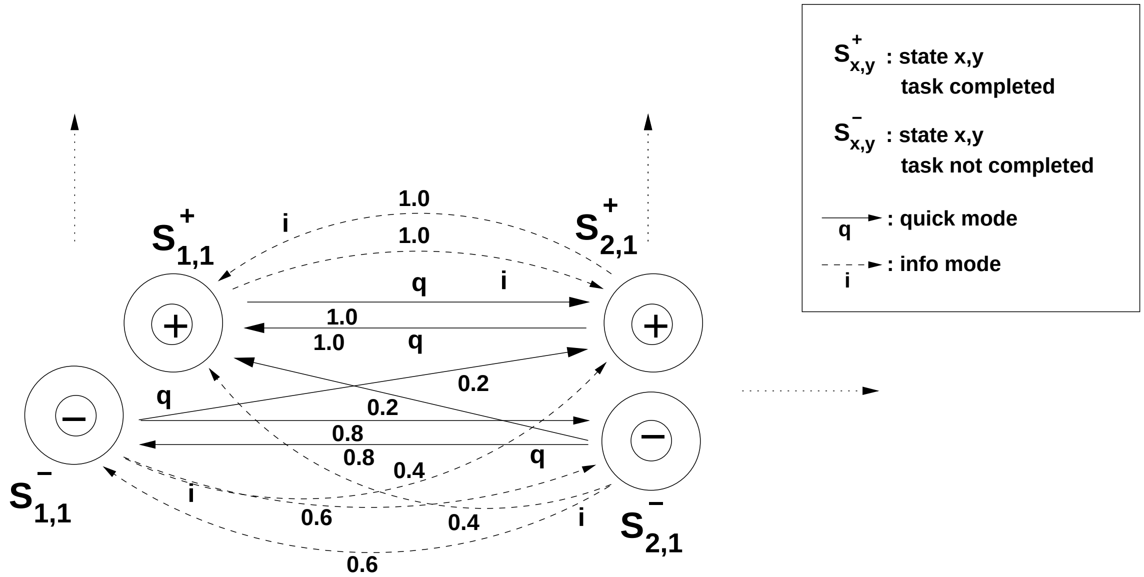

Figure 3: Transition probabilities for selected transitions

Since utility terms appear on both sides of the equation, we have to use an interative procedure to determine them, starting with initial values of 0: In each iteration, the right-hand side of equation1 is used to compute a new utility for each state.

In our application, without loss of generality we assume that the reward for reaching the gate without a present is 0 and the reward for reaching it with a present is some positive number that can be set by the user. The cost of an action is assumed to be the time that the action requires. Thus the desirability of reaching the gate earlier rather than later is taken into account by the fact that quicker routes have lower costs.[4]

2.2 Results

We implemented *value iteration* for the simple scenario described in figure 1. We modelled the 12 non-terminal states and the 2 goal states of figure 2 and linked them with edges for the appropriate directions and presentation modes as indicated. The edges are annotated in a way that they indicate higher probability for getting the task of buying a gift done via paths along many shops. I.e. it would be more likely for a user to get a gift bought using the path between $(1, 1)$ and $(1, 2)$ than using the path between $(2, 1)$ and $(2, 2)$. In our tests, we assumed a cost of 1.8 time units (minutes) for an instruction given in info mode and 1.0 time unit for an instruction given in quick mode.

Figure 4 describes the resulting policies for different rewards, which were assigned to G^+. It can be interpreted in a temporal manner: If the difference between the rewards for G^+ and G^- is high (6.0 in the example), then the policy suggests info mode for 4 out of 6 senders in case the task is not completed yet (first line in the boxes for the states $S_{x,y}{}^-$) and at the same time leads the user to an area with high probability of getting the task completed (between $(1, 1)$ and $(1, 2)$). If the user completed the task (states $S_{x,y}{}^+$), then he is led to the gate in a direct way and in quick mode. If the importance of getting the task

[4]If we want to take into account, e.g., the possibility that the user will miss his plane if he arrives too late, we will need a more complex modeling of time. The current modeling simply assumes that the user should arrive ”the sooner the better”, e.g., in order to get a good seat assignment. The relative importance of this time factor is determined implicitly by the utility assigned to the goal of arriving with the gift.

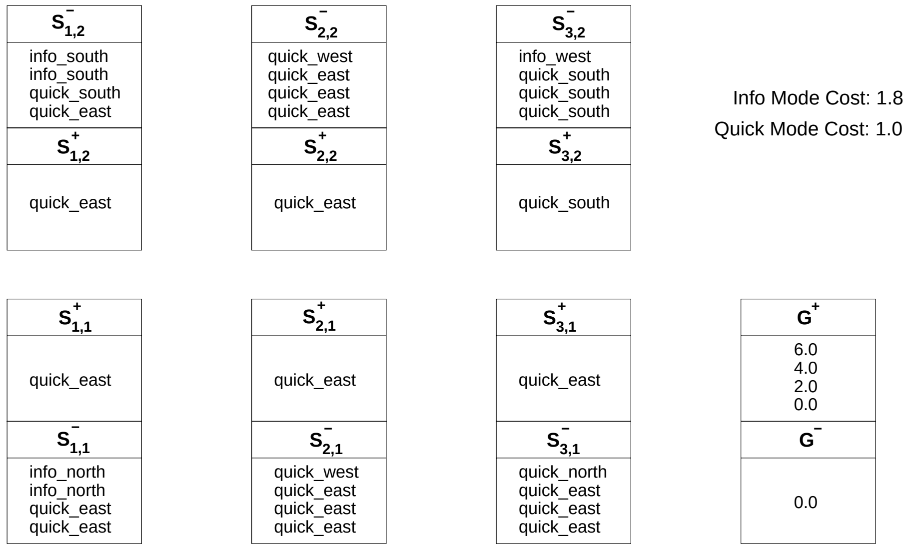

Figure 4: Policies for four different situations

completed is decreased (consider the difference between G^+ and G^- to be 4.0, 2.0 and finally 0.0; in other words: it becomes more important to get to the gate quickly), the policies more and more tend to lead the user to the gate directly and in quick mode, even if the task is not completed (rows 2, 3 and 4 in the boxes for the states $S_{x,y}{}^-$ of figure 4).

3 User-orientation

The navigation infrastructure in the airport scenario consists of a number of strong infrared transmitters mounted to the ceiling in strategically important locations of the building, such as crossections or landmarks. A sample installation has been done in our lab's offices and hallways, where visitors can choose from a list of employees and then are guided to the respective office. Since the mobile devices had to remain unmodified and their builtin IrDA infrared ports can only transmit over distances of $1-2m$, the only technical solution was to raise light intensity on the sender side and use the mobile devices in a passive way, i.e. just collecting and filtering data from the infrared streams.

For one user it is quite obvious how one can always send the right presentation (consisting of arrows, maps, or more generally vector graphics and text) at any given time and location and have it displayed on the device's screen. For up to ten or twenty users, presentations could be marked by user IDs and share the available infrared bandwidth in a time multiplexing scheme. In an airport scenario, however, we have to consider tens of thousands of users passing the building at the same time. Since users are not being tracked, we cannot just broadcast their respective presentations only in places where they just happen to be at a certain time, but instead have to send data for all users in all locations all the time.

This at first sight seems to render the scenario absurd, but by structuring the information space differently, it can be overcome very easily. If we think of all users with the same navigation goal (i.e. the same flight to reach) as being in one user group, we can already reduce the number of different presentations to broadcast by roughly two to three orders of magnitude, but still there might be hundreds or thousands of flights per day. The trick here is to look at what the single senders will end up broadcasting. In the case of a rectangular crossection, for example, there are basically just four useful choices: You either have to turn left, right, walk on, or turn around because you might have gone wrong before. Every user group will end up being sent one of those four possible types of information, so if we just broadcast four different arrows, each annotated with a list of some thousands of user groups, the necessary bandwidth will only grow very slowly with the number of users. Even broadcasting two or three stylistic variations of each arrow (for example with or without a schematic floor plan or a supporting text) will easily fit into this scheme.

Since the mobile devices filter out only the fitting packets from the data stream, users have the impression to receive a presentation taylored to them personally. In fact the thousands of “broadcast programs” share the same building blocks, just composed in different ways. By using these techniques, we can effectively broadcast adapted navigation instructions to thousands of users in the scenario described above. More details on the system and the underlying broadcast protocol can be found in [1].

4 Conclusion and Future Work

In section 2 we showed how to use decision-theoretic planning to deal with certain aspects of resource-adaptation for a navigation system which uses unidirectional infrared broadcast of information to mobile computing devices within buildings. The example we chose can occur in different variants in other scenarios and indicates the practical relevance of the explored issue. Section 3 linked the results to a concrete system based on self-made infrared senders and 3COM $PalmPilots^{©}$ and the IBM $WorkPads^{©}$.

As indicated in section 2, we could not exploit empirical data for the adjustment of the probabilities in our example. Realistic probabilities would be a necessary requirement for the practical use of the approach.

Acknowledgements

The research described in this paper was funded by ’Deutsche Forschungsgemeinschaft’ DFG within their collaborative research program 378 on “resource-adaptive cognitive processes”. Palm Pilots for the test installation were kindly provided by Deckarm & Co., St. Ingbert, Germany. We thank Anthony Jameson for helpful suggestions.

References

[1] Jörg Baus, Andreas Butz, and Antonio Krüger. One way interaction: interactivity over unidirectional links. In *Proceedings of I3 workshop on Adaptive Design of Interactive Multimedia Presentations for Mobile Users, Sunday, March 7, 1999*. www.i3net.org, 1999.

[2] C. Boutilier, T. Dean, and S. Hanks. Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. *Journal of Artificial Intelligence Research*, 11, 1999.

[3] S. Russel and P. Norvig. *Artificial Intelligence – A Modern Approach*. Prentice Hall International, Inc., 1995.