A Smart Shopping Assistant utilising Adaptive Plan Recognition

Michael Schneider
University of Saarland
D-66123, Saarbruecken, Germany
mschneid@cs.uni-sb.de

Abstract

In this paper we describe an adaptive shopping assistant system utilising plan recognition. Radio Frequency Identification (RFID) sensory is used to observe a shopper's actions, from which the plan recogniser tries to infer the goals of the user. Using this information, an automated assistant provides help tailored to the shopper's concrete needs. We discuss why it is crucial to make the plan recognition process itself user adaptive and present ideas how to realise this through modification of existing plan recognition approaches.

1 Introduction

While support systems for online shoppers like review databases, comparison shopping, or "Customers who bought this also bought that" hints are standard today, less effort has been made in the past to support customers during their offline shopping tour. A well known attempt to transfer some of the advantages of online shopping help systems into real world stores was the installation of information kiosks, presenting applications like a product catalogue or a product search. But as it can be observed in most shops, these systems are rarely used. Reasons for this may be, that using these kiosks is unintuitive or (at least seems to be) to difficult, or because standing static in front of some display inhibits the fun of strolling around in a shop.

Having this background in mind, the motivation behind our smart shopping assistant is to provide a support tool that is mobile and user adaptive, delivering support when and where the user needs it. To make this support as transparent and unobtrusive to the user as possible, we observe the user's actions and employ plan recognition techniques to infer the user's goals and thus his needs. Utilising this information, support is provided on a display mounted to the user's shopping basket.

That our work on the shopping assistant is relevant and may have practical impact in the near future was demonstrated impressively a few weeks ago: During our implementation of the smart shopping assistant, Metro Group announced the opening of their *future store* in Rheinberg, Germany. This store puts RFID technology on the job in a large scale. Read more about this interesting project in section 2.

In the following we first discuss some relevant properties of the shopping domain, before we take a closer look at the problem of plan recognition.

1.1 Shopping Scenario

The shopping domain has some characteristics that make it especially well suited for the application of plan recognition driven adaptive support systems.

Most user actions relevant for identifying a user's goal include physical interaction with products available in the store. These actions like taking a product out of some shelf, putting it back or in the shopping basket can easily be observed by RFID sensory. Other actions like moving around or staying in front of some display may also be observed by according sensory. For a detailed description of the sensory used in our shopping assistant implementation the reader is referred to section 3.2.

Besides the availability of significant sensor data, another advantage is the common understanding of possible user goals and their limited number of realisations. Because a traditional plan recognition system has to enumerate all possible ways to achieve a goal that should be recognised, plan recognition can become quite difficult or even intractable if there exists no common scheme to reach a certain goal (in which case it is intractable even for humans). In the shopping domain, most goals fall into one of two main categories (at least for honest shoppers): Either the act of buying or of gathering information. Characteristic behaviour for both goals can be observed in any store every day. Nevertheless, plan recognition is not trivial given the limited observations delivered by the RFID sensory in the shopping domain, and the interpretation of competing plan hypotheses depends heavily on the individual shopper's preferences and background. A deeper look on the used plan library is taken in section 3.3.

1.2 Plan Recognition

Plan recognition is the process of inferring a user's plans and goals by observing his actions. Most plan recognition systems reason on the basis of a *hierarchical plan library*, linking subactions to abstract or compound higher order actions. Plans are represented by top-level actions and user goals are ascribed to these plans.

The generic plan recognition process works as follows: When a user action is observed, the plan library is searched for higher order actions explaining this observation. If there exists more than one explanation for a series of observations, the plan recognition process should deliver some hints how to interpret these results. A common way to get some rating of competing explanations is to apply a probabilistic model. Such a model assigns an a-priori probability of being executed to each top-level action and to certain decompositions of higher order actions into subactions being chosen by the user. Given such a model the likelihood of discovered explanations can be computed.

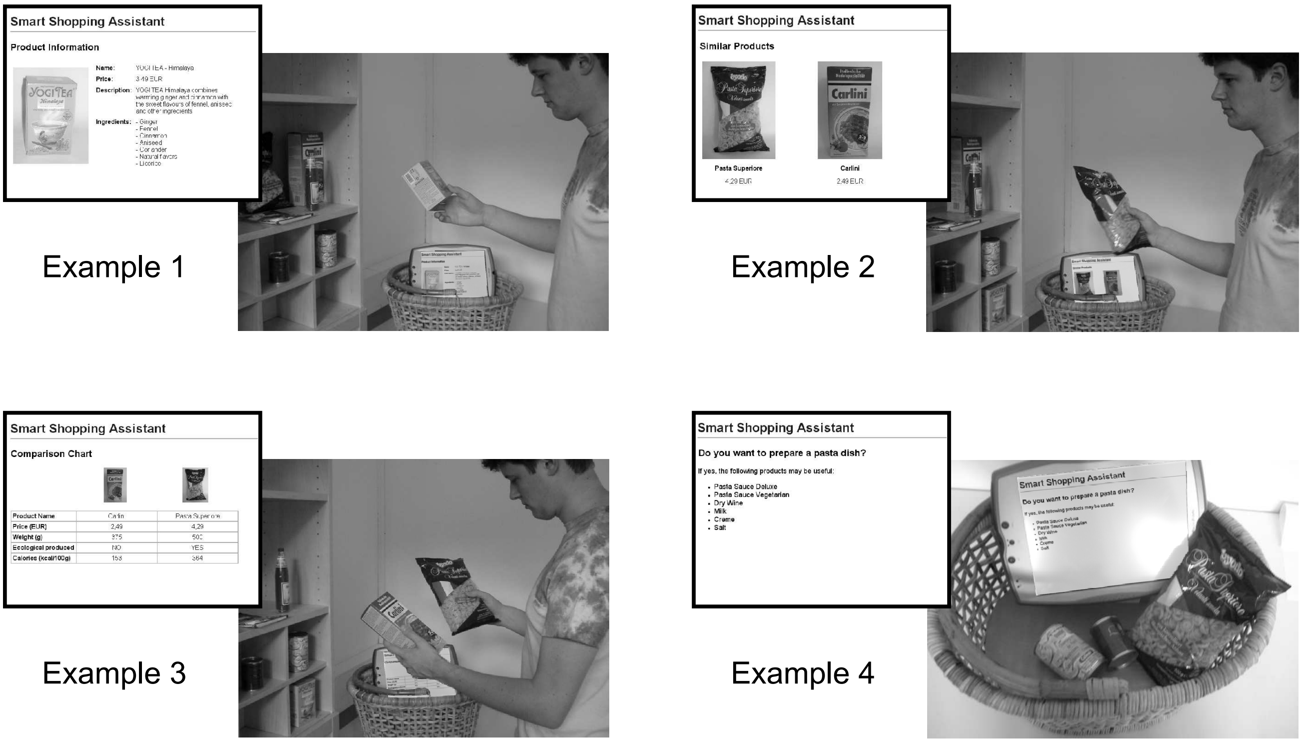

Figure 1: The Smart Shopping Assistant in Use

2 Related Work

The work closest related to our's is the work of the *Future Store Initiative* [FutureStore, 2003]. In this project major players from the IT and consumer goods industry have united to develop new ways of supply chain management and customer support. A few weeks ago the first future store was opened for private customers. Like in our implementation, the future store employs RFID technology to tag single products, and a *personal shopping assistant* (PSA) is attached to each shopping cart in order to assist the customer. As yet, the PSA fulfils only very simple tasks like keeping the shopping list or delivering on demand product information. As far as we know the PSA is not user adaptive besides loading the personal shopping list and keeping the last items bought.

Another project that uses RFID technology and some very simple kind of "plan library" is *RFID Chef* [Langheinrich *et al.*, 2000]. In this system products are tagged with RFID transponders to be recognised by an instrumented kitchen table. On the basis of the products present on the table, the RFID Chef searches a recipe database and presents a list of meals to the user preparable with the given products. By clicking on one of the proposed meals the user can retrieve a detailed recipe from the database. The sorting of presented meals depends on the amount of products actually needed out of the given to prepare a certain meal.

3 Smart Shopping Assistant

In this section we describe our smart shopping assistant implementation. At first we will give an overview of the shopping assistant from the user's perspective, demonstrating the support provided in a test scenario. After this we will have a closer look at the technical aspects, before we will explain the computational insides of our system together with the integration of the plan recognition component.

3.1 The Shopping Assistant in Use

At the entrance of the shop the user picks up one of the smart shopping baskets. He registers himself with the basket by holding his customer card near the basket display for a second. After the shopping assistant has welcome him the user starts shopping.

In our first example the user takes a package of tea out of the shelf, holding it in his hand for a while. Lets assume the user's experience with this kind of products is very sparse, the shopping assistant infers that the user may have a need for in-depth product information and probably is reading the wrapping of the product right now. Therefore the shopping assistant decides to display appropriate product information on the basket display.

In our second example the user again takes a product, but now we assume that he is an expert with this kind of product. Anyhow he has not yet decided to buy this product[1], so the shopping assistant infers that the user may be searching for a similar product that better suits his needs. The assistant decides to display a list of similar products available at the store.

Next the user takes out two products simultaneously, holding them in his hands for a while. The shopping assistant again infers that product information may be needed, but now regarding the two products in comparison. He displays a comparison chart to the user merging product information of both products.

Our last example deals with the situation, where the user has already chosen some products and has put them in his shopping basket. Utilising a recipe book the shopping assistant infers that the user may want to prepare some kind of pasta dish. The assistant decides to display a list of products that may also be useful to cook a pasta meal. Because

[1]Because we have not yet seen him putting it in his shopping basket

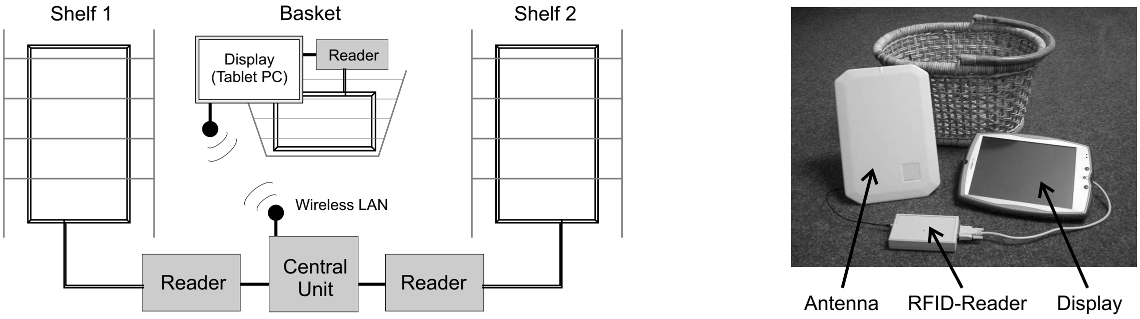

Figure 2: Setup of test environment (left) and components of the smart shopping basket (right)

the user is vegetarian, the assistant thereby omits dishes containing meat or fish.

3.2 Technical Plattform

In our implementation of the Smart Shopping Assistant we employ passive Radio Frequency Identification (RFID) sensory to observe the shoppers behaviour. This sensory works as follows: A reader sends commands over a reader antenna to a passive RFID tag. The tag contains a smaller antenna and a microchip with onboard memory. Possible commands are reading from and writing to this memory. The tag uses the energy induced by the reader antenna to process the request and send the response back to the reader. Every tag has a unique ID that is included in it's response. The maximum communication range depends on the size of the reader and tag antenna and on the transmitter power of the reader. A typical range is up to 1 meter for a credit card sized tag and a reader antenna of 80x60 centimetre. The exact range can be fine tuned by adjusting the output power of the reader.

For our test setup we attached RFID tags to all products in our shop. Additionally, each customer card is marked by a tag. Readers and antennas were mounted to our shelves and the shopping basket. We mounted the antennas and adjusted the transmitting power in a way that make the readers almost only scan the area inside our shelves respectively inside the basket. The display and reader of the shopping basket were connected to the rest of the infrastructure via Wireless LAN. Support information is generated as dynamic HTML pages and pushed to a web browser running at the basket display. Figure 2 shows the overall setup used by our implementation (left side) and the technical components built into the shopping basket (right side).

As described in section 3.1, the shoppers "login" takes place by bringing his customer card near the basket display, thus making the embedded RFID tag recognisable by the reader attached to the basket. The ID stored on the tag is read and sent to the central unit, which loads the user model from an external service like U2M [Heckmann, 2003].

At the beginning of the user's shopping tour, all products of interest are still in their shelves and therefore are visible to the RFID readers attached to these shelves. If the user now takes a product out of some shelf, this is registered by the according RFID reader and signalled to the central unit. Thereby, two information items are delivered: The product which was taken is identified by the unique ID of the attached RFID tag, and the location of this interaction is given by the location of the object the according antenna is mounted to. Analogous we observe if the user puts a product back into a shelf or puts a product into or gets a product from the basket.

To summarise, we use RFID technology as an unobtrusive, simple and reliable way to check the physical presence of well defined objects within a certain area. By the use of more than one RFID reader we track the "spatial flow" of objects between multiple locations. We interpret this flow as manifestation of a user's *get-product* and *put-product* actions in a shopping scenario.

3.3 Inside the Shopping Assistant

To decide what kind of support to offer the user, the shopping assistant tries to figure out the intentions behind the user's actions. The assistant does so by utilising a probabilistic plan recognition system, which is fed with the observed user actions. Figure 3 shows the internal structure of the central shopping assistant unit.

A *Sensor Data Interpreter* collects and analyses the data delivered by the RFID sensory. It translates this data into symbolic ground level user actions which are fed into the plan recognition engine. In the shopping domain these are *get-product* and *put-product* actions.

The *Plan Recognition Engine* has to infer potential user plans explaining the observed actions. Beside the actions observed other input parameters to the plan recogniser are the library of possible user plans, a probabilistic model describing the probability distribution over all explanations, and the user model containing individual user characteristics and preferences.

While the first three are typical parameters for generic probabilistic plan recognition systems as described in section 1.2, the user model is typically not explicitly included in the reasoning process. It is undoubtful, that the likelihood of certain plans being executed depends on the user's knowledge and experiences, his personal preferences and even his current context. All probabilistic models have to (and mostly do) account for these factors when specifying a-priori probabilities for certain plans. Unfortunately, in general these factors are hidden within the general probabilistic model, given somehow implicit by the specified probabilities for top-level plans. While this is applicable to systems used by only one user, this is not acceptable for systems required to adapt to many individual users, as our shopping assistant does. Therefore, it is absolutely necessary to explicitly represent user specific dependencies in the probabilistic model and link them to a concrete user model provided at runtime.

This requirement can be met by all probabilistic plan recognition approaches by making slight enhancements to the probabilistic model. The key idea is to leave user specific parts of the probabilistic model unspecified, then defining some kind of probabilistic interface which describes the requirements a probabilistic user model have to

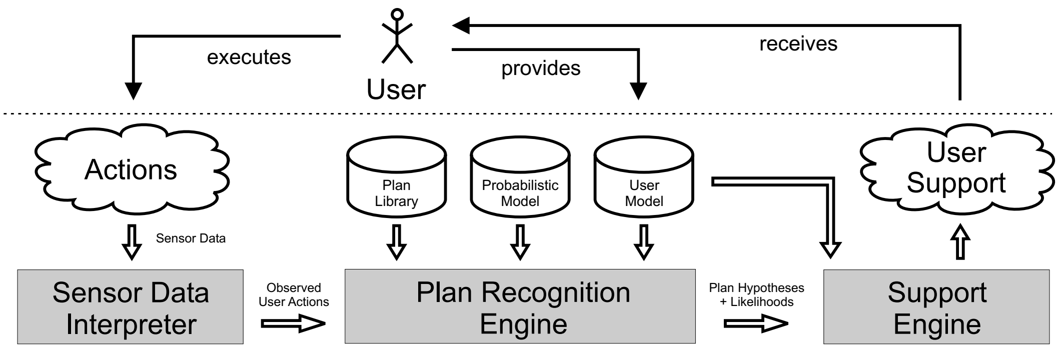

Figure 3: Structure of the Central Shopping Assistant Unit

meet in order to be linkable to the basic domain model. When the user logs in, his personal user model is melted with the given domain model and afterwards can be processed by the plan recogniser in the common way. Unfortunately, because every plan recognition system may use its own representation of the probabilistic model, we can give no general formalism how to realise this interface.

Although to our knowledge every probabilistic plan recognition system can be modified as described above, we are currently developing a plan recognition system called *OPRES* that supports distributed domain models more naturally. *OPRES* is designed with a focus on applications in dynamic and open domains and uses object-oriented probabilistic networks for knowledge representation and probabilistic reasoning. First results will be published shortly, so long the interested reader may find a rough overview in [Schneider, 2003]. As plan recognition engine in the implementation of our smart shopping assistant we used an early version of our OPRES system.

The *plan library* used in our implementation is build upon the basic user actions of getting and putting a product from/to some place. From these actions we infer the higher order actions of buying a product (taking it out of a shelf and putting it in the basket) and gathering information (taking it out of a shelf and putting it back after a while). From seeing multiple inform or buy actions in a row we infer top-level plans like gathering in-depth product information, searching similar products, comparing products, or buying a certain bunch of products (for example in order to prepare a pasta dish). Each action can have conditions assigned limiting their applicability, e.g. restricting *compare* actions to products of the same kind.

The *probabilistic model* used in our implementation defines the likelihood of top-level actions being executed to depend on the user's experiences and preferences. The probability that a "gather product information" plan is active for a certain product for instance is higher if the experiences with this kind of product are low and vice versa. On the other side, the plan to look for alternative products becomes more likely if the experiences regarding the product handled are high. As another example, the probabilities of plans to prepare a certain meal depend on the eating preferences, like being a vegetarian, being on diet or having allergies. The state of subactions generally depends deterministic on the state of their parent actions; they are executed if and only if at least one of their parent actions is executed.

At the end of our processing chain, the *Support Engine* delivers support adaptive to the inferred goal with the highest likelihood of being pursued by the user as delivered by the Plan Recognition Engine. For our shopping assistant, we already discussed the different forms of support provided by the system in chapter 3.1.

4 Conclusion and Outlook

In this paper we presented a smart shopping assistant implementation utilising Plan Recognition. We showed how to use RFID sensory to track the movement of real world objects and how to explain them by means of user actions. We proposed a generic way to separate probabilistic user knowledge from probabilistic domain knowledge, enabling a plan recognition system to adapt its recognition process to the current user's preferences.

While *get-product* and *put-product* actions observed by our implementation so far are important to reason about user plans in a shopping scenario, they are still limited to deliver evidence only for plans including physical interaction of the user with some objects (here products). Other sensory capturing non interactive behaviour would be useful to discover plans without such interactions. In the next version of our shopping assistant we plan to integrate spatial sensor information, as delivered by camera sensors, active RFID tags[2] or WLAN triangulation. Non interactive plans recognisable with this additional sensory for instance are product search (aimlessly moving around in the store) or gathering product information by reading a poster (resting in front of some advertising display). Further sensory possible to apply includes biometric sensors for measuring increases or decreases of the user's heartbeat frequency or other stress indicators.

References

[FutureStore, 2003] FutureStore. Future store initiative, June 2003. Official website: http://www.future-store.org.

[Heckmann, 2003] Dominik Heckmann. Ubiquitous user modeling for situated interaction, August 2003. Website and online demo: http://www.u2m.org.

[Langheinrich *et al.*, 2000] Marc Langheinrich, Friedemann Mattern, Kay Römer, and Harald Vogt. First steps towards an event-based infrastructure for smart things, 2000. Ubiquitous Computing Workshop at PACT.

[Schneider, 2003] Michael Schneider. Opres: An object-oriented plan recognition system, June 2003. Short project description on website: http://w5.cs.uni-sb.de/~mschneid/opres/.

[2] Active RFID tags have their own power supply and therefore a higher reading range