POSTER ABSTRACT

Towards an Intelligent Mobile Travel Assistant

Marc Torrens and Patrick Hertzog
i:FAO's Future Lab
Scientific Park, EPFL
Lausanne, Switzerland
{torrens, hertzog}@ifao.net

Pearl Pu and Boi Faltings
Institute of Core Computing Science
Swiss Federal Institute of Technology (EPFL)
Lausanne, Switzerland
{Pearl.Pu, Boi.Faltings}@epfl.ch

ABSTRACT

Travel has many situations where context-aware computing can bring important benefits. In this paper, we describe an approach for integrating context-aware computing to a mobile travel assistant. Travel plans, generated using *reality* [2], are enriched within compact and powerful structures, called User Task Models. These structures are transferred to a mobile device enabling the support for the traveler during his trip.

1. USER TASK MODEL

When a business traveler plans a trip, he is actually trying to map goal activity requirements such as meeting schedules into a feasible plan, for example a sequence of flights and hotels for each destination. That is why we have adapted the *reality* booking engine to accept the definition of meetings or goal activities as input instead of a sequence of pairs airport-date. Once the user has selected the preferred flight combination for his trip, the user's plan is stated. In that sense, a travel plan is defined as following:

DEFINITION 1. *A* Travel Plan *is a collection of* n *goal activities* $\{G_1, \ldots, G_n\}$. *A* Goal Activity G *is mainly defined by a location* $loc(G)$, *a start date/time* $start(G)$, *an end date/time* $end(G)$, *and the activity itself* $activity(G)$ *(*e.g., *meeting, conference, seminar, etc.). The following restriction is made:* $\forall i,\ 0 \leq i < n,\ end(G_i) \leq_{prev} start(G_{i+1})$, *where* $\leq_{prev}$ *is the time precedence operator.*

*Moreover, a goal activity is decomposed in a sequence of required tasks (*e.g., *drive to airport, overnight at hotel, etc.) called means tasks. These tasks are needed to accomplish a goal activity and can be recursively decomposed in sub-tasks.*

Since a travel plan can be seen as a hierarchy of goal activities and means activities, we naturally represent it as a tree. A tree representation of a travel plan, called *travel plan tree* has the following characteristics:

- The root node contains general trip information, such as global preferences, itinerary description, and so on.
- Children of the root node represent the user's goal activities.
- Child nodes of a goal activity node represent its means activities. At the same time, these means activities can be represented recursively by another sub-plan which forms the sub-tree of the node.

Nodes at the same level are chronologically ordered by their *start* and *end* functions. The *start* of the first child of a node n cannot be anterior to the *start* of node n. Similarly, the *end* of the last child of a node n cannot be posterior to the *end* value of node n.

A travel plan tree would be enough if the traveler's plan does not change, but in practice, this is often not the case. Travel plans change because unexpected situations emerge, and context information is unpredictable. As explained below, the usage of an annotated workflow empowers significantly the support of the traveler.

DEFINITION 2. *An* Annotated Workflow *of a travel plan is the lineal execution of the different tasks following a depth-first traversal of the travel plan tree. The execution of each task is triggered by time.*

Each node of the tree includes annotations *with basically two kinds of information related to the corresponding task: a) server requests to retrieve the needed contextual information, and b) personal user preferences.*

Note that annotations in the workflow are crucial for capturing the current context of user's activities. In that sense, the server requests of a task act as sensors of the current user's state, *e.g.*, traffic situation at the highway from home to the airport. Without these annotations the only contextual information is the current user's location and current user's task which can be deduced from the travel plan and current time. On the other hand, personal preferences are used by proactive services to accurately assist the traveler according to his personal needs.

DEFINITION 3. *A* User Task Model (UTM) *is a travel plan tree with its annotated workflow. A User Task Model is* activated *when current time matches with* $start(n)$, *where* n *is the first (left bottom) leaf of the tree. On the other hand, a User Task Model is* deactivated *once current time matches with* $end(n)$, *where* n *is the last (right bottom) leaf of the tree.*

SAC'04, March 14-17, 2004, Nicosia, Cyprus

Our notion of workflow implies a lineal execution of tasks, *i.e.*, tasks are executed one by one and never in parallel. In the case we would be interested in modeling some parallelism for a traveler, *e.g.*, two tasks executed at the same time range, two different task models can be considered.

Each UTM handles the plan execution for a traveler using the system. UTMs are expressed as XML documents according to a XML Schema for better interoperability among internal or external components of the system. Moreover, in this way, UTMs can be originated and used by different software components in a convenient way.

1.1 Generation of User Task Models

The UTM is created by the *reality* application [2] once the user has finished the booking process, *i.e.*, to define his meetings and to select the products accordingly (flights, hotels, rental cars, etc.). The booking process enables building the travel plan with its tree representation, and also adding the appropriate annotations for the workflow. For example, the task `[Go from Manhattan to JFK Airport]` could be associated with a server request in order to inform the user about traffic jams in New York City. Such server request accesses web services to discover traffic information in New York City. An example of preference for that task could be that the traveler prefers going to JFK Airport by taxi rather than by public transportation.

1.2 Replanning

In practice, business travel plans often change because unexpected situations and new requirements occur. For example a meeting can last longer than initially planned, and therfore the user may want to rebook his next flight. In that sense, replanning is triggered by the traveler. The User Task Model is then used to select the task of the travel plan that needs to be replanned. Actually, the replanning must take place at the parent node of the selected task, in order to guarantee a coherent resulting model. The annotations of the affected task, especially task's preferences, are then used by the constraint-based engine (same solving engine as used by *reality* [2]) to propose a small set of possible alternative tasks. Once the user has selected his preferred alternative, the User Task Model is rebuilt consistently considering the changes.

1.3 Proactive Services

Proactive services are activated by tasks with annotations representing server requests to those services. For instance, a task `[Flight from JFK Airport to GVA Airport]` could have an annotation indicating that an automatic check-in service must be executed. Other examples include to weak-up the user automatically in order to get on time to a meeting.

When the user replans part of his UTM, the system can also alert the concerned people. For example, when the user misses a flight, an email could alert the following meeting participants about the user's delay.

1.4 Filtering Information

Workflow annotations of UTMs are also used to channel relevant information to the user. For example, one could consider annotations for filtering information related to the current user location about: political news, weather forecasts, strike announcements, traffic jams, and so on.

Figure 1: The different activities of the User Task Model are displayed chronologically from top to bottom.

2. CONCLUSION

Techniques from Artificial Intelligence have been applied in the travel planning domain [1, 2] in order to better assist travelers during the booking process. The next step is to actively support the traveler *en-route*. Current mobile devices (such as PDAs) enable advanced software development for such purpose. However, some challenges must be faced in order to truly offer a real support for the traveler. These challenges are related to context-awareness and usability issues. Our approach to overcome such difficulties is based on data structures, called *User Task Models*, providing an explicit record of user activities and intentions that serves as a basis for: a) filling in context in user-initiated actions, and b) activating relevant proactive services. Thus, User Task Models include travel plan information while providing context-awareness in a succinct way.

3. REFERENCES

[1] M. Torrens, B. Faltings, and P. Pu. Smartclients: Constraint satisfaction as a paradigm for scaleable intelligent information systems. *CONSTRAINTS: an international journal*, 7:49–69, 2002.

[2] M. Torrens, P. Hertzog, L. Samson, and B. Faltings. *reality*: a Scalable Intelligent Travel Planer, Decision Support for the Business Traveler. In *Proceedings of the Eighteenth Annual ACM Symposium on Applied Computing (SAC 2003)*, Melbourne, Florida, USA, March 2003.