A Case-Based Personal Travel Assistant for Elaborating User Requirements and Assessing Offers[1]

Lorcan Coyle, Pádraig Cunningham, Conor Hayes

Department of Computer Science
Trinity College Dublin
{Lorcan.Coyle, Padraig.Cunningham, Conor.Hayes}@cs.tcd.ie

Abstract. This paper describes a case-based approach to user profiling in a Personal Travel assistant (based on the 1998 FIPA Travel Scenario). The approach is novel in that the user profile is made up of a *set* of cases capturing previous interactions rather than as a single composite case. This has the advantage that the profile is always up-to-date and also allows for the borrowing of cases from similar users when coverage is poor. Profile data is retrieved from a database in an XML format and loaded into a case-retrieval net in memory. This case-retrieval net is then used to support the two key tasks of requirements elaboration and ranking offers.

1 Introduction

Case-Based Reasoning (CBR) belongs to the lazy school of Machine Learning and thus has the defining characteristic of deferring processing to run-time. Whatever the run-time penalties associated with this, it does ensure that the data used for inference is as up-to-date as possible [1]. This is of particular benefit in situations where data is scarce and where data is being updated continuously. In this paper we describe a user profiling problem that has these characteristics and describe a case-based solution to this user-profiling problem. This solution contrasts with the normal use of CBR in user profiling where each user profile is represented by a single case (e.g. the PTV scenario presented by Smyth and Cotter [9]). By contrast, in the solution described here, the user profile is made up of several cases, each capturing a previous interaction episode with the system [11]. This approach is also used by Bradley et al. [3] in CASPER, a case-based assistant operating in on-line recruitment.

The scenario we refer to was introduced in 1998 by FIPA (Foundation for Intelligent Physical Agents) as one of a set of benchmark scenarios to drive research on intelligent agents. In this scenario a Personal Travel Assistant (PTA) would help a user in organizing travel. The PTA could operate on a central server and users would interact with this through a mini-PTA (mPTA) that might operate on a WAP-enabled

[1] The Support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged

mobile phone. Personalization is to be used to overcome the inherent limitations of using these devices (i.e. poor text-inputting facilities, high network latency).

The main role of the PTA is to negotiate with a Broker Agent to book flights on behalf of the user. We have identified two tasks within this negotiation that require some 'intelligence' from the PTA; these are:

- Elaboration of the user's requirements based on the user profile.
- Evaluation and ranking of offers from the Broker Agent based on the user profile.

In this paper we present an integrated case-based solution to these two tasks. This approach uses a history of past interactions to generate a case-based user profile to complete the task at hand. The FIPA Travel Scenario is described in the next section and then the case-based solution is described in section 3. Then the effectiveness of the solution is described with the aid of two examples in section 4.

2 The FIPA Travel Scenario

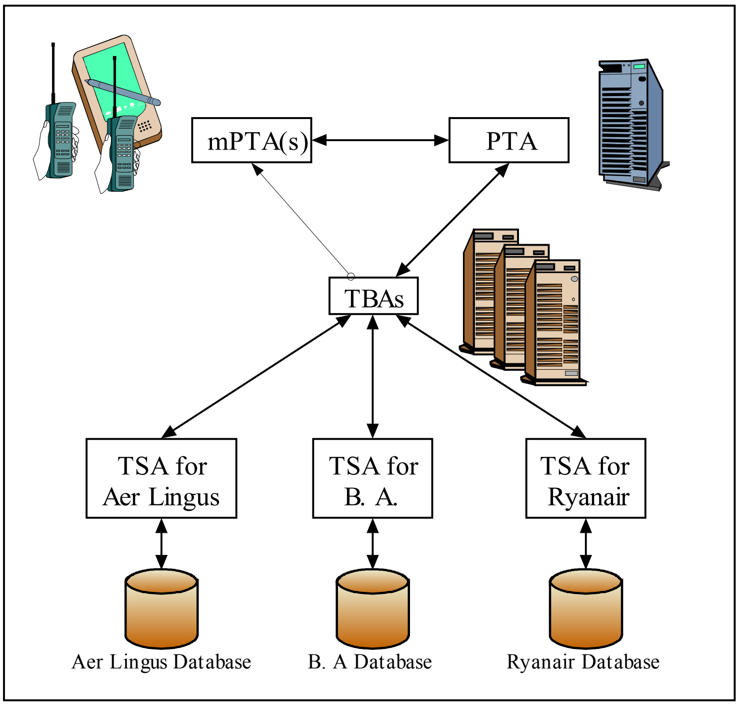


Fig. 1. The Personal Travel Market

FIPA's PTA specification [5] describes an open marketplace – the Personal Travel Market (PTM) – where software agents broker deals between travel vendors and customers. Figure 1 shows an architecture for a Personal Travel Market, showing the different agents and the flows of communication between them.

The PTM consists of the following agents:

- The PTA is the user's representative in this market; it is responsible for carrying out the user's instructions and learning the user's preferences. It receives travel requests from the user (through the mPTA) and forwards these to the TBAs in the system. On return of travel offers from the TBAs it sends the most suitable of these (based on user preferences) to the mPTA to present to the user. If any of these offers are accepted it has the authority to book them with the TBA. Users cannot access the PTA except through an mPTA interface.
- mPTA: This provides a user-interface to the PTA and is concerned mainly with presentation of requests and offers. It communicates mainly with the PTA; its only communication with the other agents would be simple requests for information, e.g. allowing the user check on the status of a flight by making a request directly to a TBA.
- TBA(s) (Travel Broker Agent): These are the middlemen of the PTM. They take travel requests from the PTAs and negotiate travel solutions on their behalf with the TSAs. They do this by decomposing travel requests into segments and negotiate with suitable TSAs for these segments. They then compose these into travel offers and return them to the PTA.
- TSAs (Travel Service Agents): These provide a wrapper around legacy travel databases and systems. They are responsible for maintaining the data access, interpretation and delivery to the TBA(s).

Our goal is to implement the PTA and mPTA. An interface to a TBA, with some simple code underneath to simulate the remainder of the PTM will also be developed. FIPA have already created their own agent communication language [4], and use this to define the transactions within a PTM system. This simplifies our goal of creating a fully compliant PTA-mPTA system, capable of interfacing with any FIPA compliant PTM. There are a number of publicly available implementations of agent platforms that conform to the FIPA Specifications. The LEAP (Lightweight & Efficient Application Protocols) project [2] appears to be the most promising of these; it allows deployment of intelligent agent systems on computers, PDAs and mobile phones.

The two main interactions between a user and the PTA are:

- Making a travel request on behalf of a user (Figure 2)
- Recommending suitable travel offers to the user (Figure 3).

These diagrams show timelines for each component in the system and outline what transactions occur between them. The bulk of the personalization computation occurs in the boxed areas along the PTA timeline.

Request Elaboration:

According to FIPA's PTA ontology, for a PTA to make a travel request to a TBA it must send it a `trip-summary` object, filling all mandatory parameters, some of which are shown below in Table 1. These parameters may be absolute, e.g. `'Origin = London'`, or constrained, e.g. `'Budget < €800'`. The goal of the request-elaboration task is that the user should not need to manually fill all of these parameters, that the PTA should be able to determine user preferences from previous interactions. Ideally the PTA will send the mPTA a simple form with minimum entry fields to present to the user, which the user will complete and return. This information makes up a *skeletal request.* The PTA should be able to use similar requests from that user's history to fill in the remaining parameters. This elaborated request represents a final travel request and is forwarded to the TBAs. Alternatively, the PTA may return it to the user for confirmation that it represents his/her intentions. The PTA can learn more about the user's preferences from this confirmation/rejection.

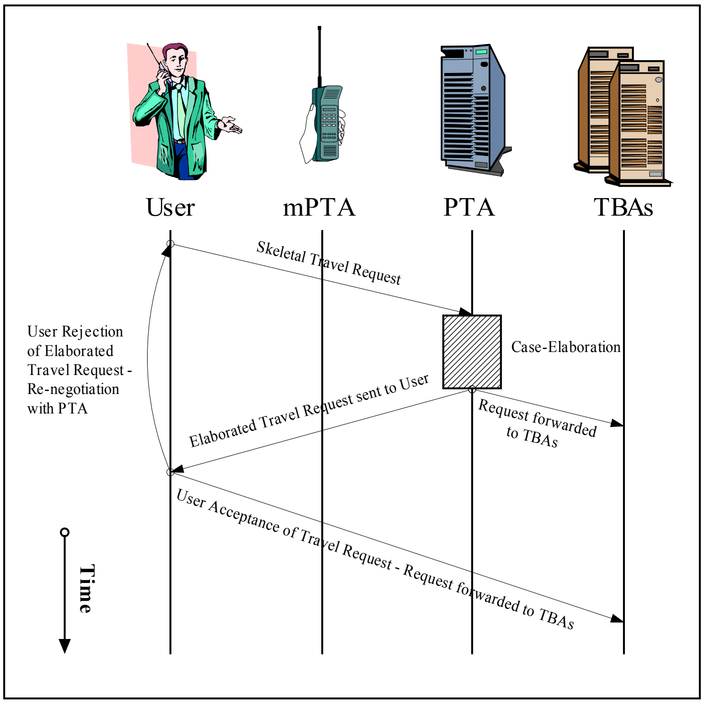

User
mPTA
PTA
TBAs
Skeletal Travel Request
Case-Elaboration
User Rejection of Elaborated Travel Request - Re-negotiation with PTA
Elaborated Travel Request sent to User
Request forwarded to TBAs
User Acceptance of Travel Request - Request forwarded to TBAs
Time

Fig. 2. Request-Elaboration

Table 1 A selection of the `trip-summary` parameters mentioned in this paper. The PTA specification covers other optional parameters, some of which are outside the flights domain.

Parameter	**Description**	**Presence**
Origin	The origin of the trip	Mandatory
Via	A list of via locations of the trip	Mandatory
Destination	The destination of the trip	Mandatory
Departure Time	A list of start dates and times for the trip	Optional
Return Time	A list of return dates and times for the trip	Optional
Budget	The currency and preferred price range of the trip	Optional
Selection pref.	The selection preference, e.g. by cost or comfort	Optional
Class	The class of ticket, e.g. Business, Economy	Optional

Depending on the capabilities of the mobile device, extra contextual information could be sent to the PTA on logon, e.g. if the device has a GPS module, the user's geographical location. The PTA could use this to determine the nearest airport or rail station to the user.

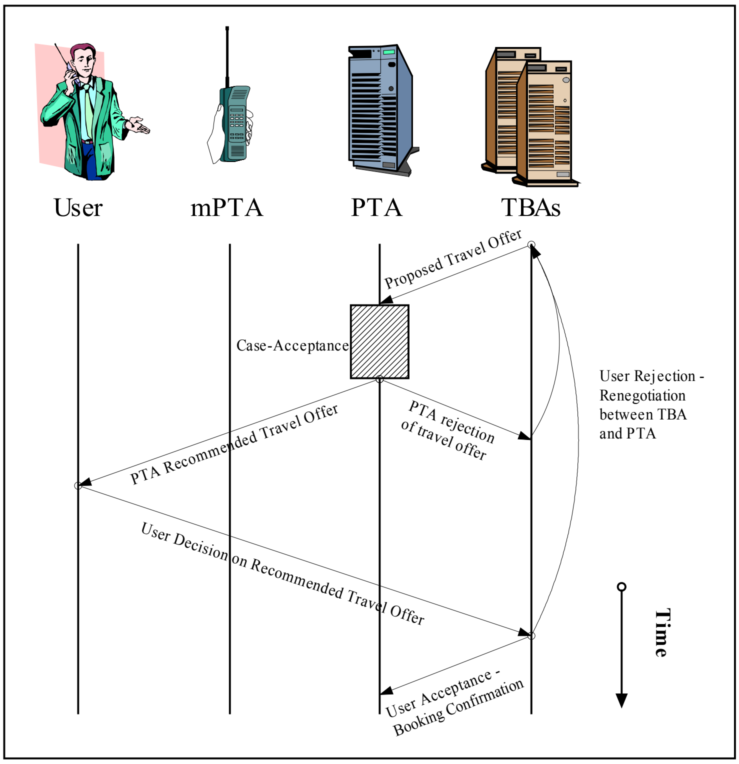

User
mPTA
PTA
TBAs
Proposed Travel Offer
Case-Acceptance
User Rejection - Renegotiation between TBA and PTA
PTA Recommended Travel Offer
PTA rejection of travel offer
User Decision on Recommended Travel Offer
User Acceptance - Booking Confirmation
Time

Fig. 3. Offer-Recommendation

Offer Recommendation:
TBAs reply to a `trip-summary` object with a number of `trip-details` objects. These include the original `trip-summary` object, replacing any constraints passed by the PTA with specific values. They also contain information relating to the trip, most importantly, the specific details of the trip (i.e. all plane hops or train stops) as a sequence of travel segments. Since these offers could be quite numerous, the PTA should rate them, and reject unsuitable ones. It does this by examining user's previous interest in similar offers in the past, whether similar offers were ignored, examined or reserved. It may reject segments (and entire offers) that it believes will be rejected by the user and attempt to negotiate full travel-solutions with the TBAs from acceptable segments. It then forwards acceptable offers to the mPTA to present to the user. If the user rejects these, it renegotiates with the TBA for a better offer by amending the original travel request. There is also an upper limit to the number of offers that will be presented to the user. A diversity function like those used by [10] may be used to limit the number of offers presented without sacrificing similarity.

These personalization tasks require the PTA to have a level of autonomy. This autonomy should be backed up by transparency. To ensure user confidence in the system, the PTA should be capable of explaining the motives behind its actions to the user.

3 The Case-Based Solution

In order to offer personalized service to every user, the PTA stores a profile for each user of the system. This profile comprises the following types of information:

- Personal details that each user fills in at registration time, e.g. name, address, preferred language, credit-card details, etc.
- Transaction History. This includes all the previous user transactions with the system, and their context, i.e. it links transactions that occurred within a single interaction.

The PTA extracts useful information from the transaction history and generates a case-base specific to the task at hand by looking at the problem (target) case. Within the request elaboration task the target-case is the skeletal request, and within the offer-recommendation task it is the travel offer. It would be convenient to develop a single retrieval process that would operate along the same lines for both tasks. This is done by first extracting the useful information from the transaction history and generating a case-base from this. This case-base is expressed in XML format. This case-base generation process is similar to defining a CBR-view on the data [6]. The target case and this case-base are then passed into a similarity retrieval module.

There have been systems developed with distributed CBR in mind [7], [12]. This may become more important in the context of this system as the abilities of the target mobile devices increase. It may be appropriate for the retrieval module to be implemented on the client side. The PTA could then be responsible for data maintenance and the mPTA for carrying out the personalization tasks. It may eventually be possible for the mPTA to become a PTA and deal directly with the TBAs thus cutting out the middleman.

The retrieval module for both of these tasks is a CRN. These are described in more detail in the following section. It should be noted that, that under certain circumstances the CRN solution may be excessive as a retrieval mechanism for the request-elaboration task for situations where a users preferences in travel requests are very consistent. In such circumstances a more straightforward process may be appropriate.

3.1 Case-Retrieval Nets

CRNs are used to retrieve similar cases to a presented problem-case. They borrow ideas from neural networks and associative memory models. A CRN is a structured net constructed from the case-base [8]. CRNs are used because they are easy to extend, are flexible and content-sensitive. They are made up of the following components:

- Case-nodes – These represent stored cases.

- Information Entity Nodes (IEs) – These represent feature-value pairs within cases
- Relevance-Arcs – These link Case Nodes with the IEs that represent them. They have weights relating to the level of importance between the connected IE and the case.
- Similarity-Arcs – These connect IEs that refer to the same features, i.e. all IEs referring to the `budget` feature are interconnected via similarity arcs. These have weights relating to the similarity between the values of connected IEs.

Case-nodes and IEs can be *activated*, which means given a score relating to their usefulness. The arcs in the system are used to spread activation across the net. Any score passed across an arc is attenuated by the weight of the arc. The idea behind the CRN architecture is that if a target case is connected to the net via a set of relevance arcs, and activated, this activation will spread across the net. Eventually each of the other case-nodes will accumulate a score. In this way, the case-node with the highest activation represents the most similar or relevant case to the test case.

The main difficulty in implementing and maintaining CRNs is in assigning weights to the arcs in the net. The similarity arc weights must be determined before cases are presented to the CRN. The onus is on the CRN developer to initialize and maintain these weights. Features have two types of values, numeric and symbolic. These values are constrained in advance, numeric features reside within a range and symbolic values are elements of a predefined set. Functions can be defined that will calculate the similarity between numeric IEs. This allows us to assign similarity weights to new IEs added to the net without difficulty. We don't have the same problem with symbolic values because of the fixed set of possible values; we just need to assign similarity weights to every possible IE. However, determining this similarity in the first place is difficult because the similarity between groups of symbolic features can be quite subjective. Some features are inherently problematic by their very complexity, e.g. the travel-time features (`DepartureTime` and `ReturnTime`). This can be overcome by breaking the time down into a number of micro-features, e.g. time of the day, day of the week and day of the year. We can generalize on each of these features in ways we can't with their super-feature.

Relevance arcs weights are assigned in a number of ways. The first few times the PTA serves a user, there are minimal records on users' previous travel requests and evaluations on travel offers. Therefore, the PTA does not have sufficient confidence to guess the importance of features in a travel request or travel offer. In this situation, the PTA can assume an even distribution of relevance weights. On the other hand, if a user indicates a feature in the travel request to be important, a larger relevance weight can be put on this feature while others share an even relevance distribution. Over time, as the PTA gets to know the user's preferences, the relevance arcs are weighed accordingly. In the request elaboration task, the relevance arcs for elaborated features are copied from those of the retrieved-case.

The CBR system automatically learns the relevance arc weights during the renegotiation stage. If the PTA's decision is rejected, the differences between target and solution cases must have been more important than their relevance arcs weights suggested, so these weights must be increased, and vice versa. This will lead to greater accuracy over time.

Because the PTA is acting without supervision, it must act with confidence. This confidence comes from looking at the activation scores of retrieved cases. The PTA will not use a case unless it scores above a predefined threshold. These thresholds are useless if the CRN only returns poorly scoring case solutions. The PTA must then decide whether to use these or to widen the search by relaxing constraints on the target case. Clearly this risks offering a poorer alternative.

3.2 Collaborative CBR

Every user has their own personal case-base based on their previous interactions with the system. CBR is only useful if the solution for a problem can be found within this case-base. This poses a problem when dealing with first-time users, who begin using the system with empty personal case-bases. If we note that many PTA users will have similar travel preferences, we can share cases between users. If it were possible to determine these groupings, the PTA could direct its collaborative CBR search within a small group of similar users – the user's neighbours. If a problem lies outside the experiences of a user (as all interactions will, for first-time users) we can consult the case-bases of his/her neighbours for a solution.

The most difficult part of implementing collaboration is finding the neighbourhood groupings. Ideally users are assigned to groupings on the basis of the similarity between their cases. With this solution, new users still present a problem, but this could be solved if users' registration details in some way related to their similarity. By including a number of profiling questions in the registration form, we could assign new users to preexisting groups of neighbours.

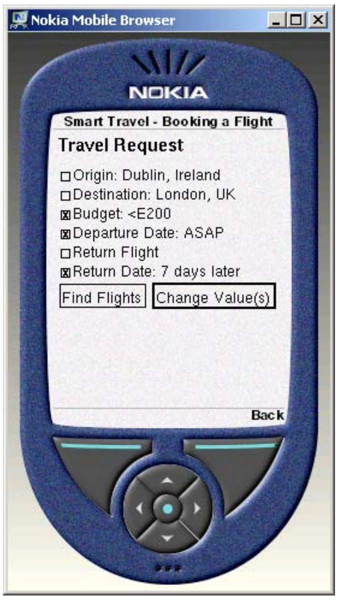
Collaborative CBR is used reluctantly; we attempt to solve a problem using only the user's own case-base, if this cannot find a solution or only poorly scoring ones, we repeat the retrieval-process, but extend the retrieval to include cases from the user's neighbourhood group. This increases the capacity of the CRN and should thus increase the probability of finding a good solution.

4 Evaluation

To illustrate the operation of the PTA-mPTA system developed here, we include two sample interactions. These interactions show the usefulness of using personalization in the context of this project. All screenshots were generated from Nokia's Mobile Internet Toolkit™. The device shown is a simulation of an XHTML browser. The next generation of Nokia WAP browsers will follow on this model.

Example 1- John

John is a first time user of the system. He makes a request for a flight to London for a weekend trip. He has filled in the registration details.

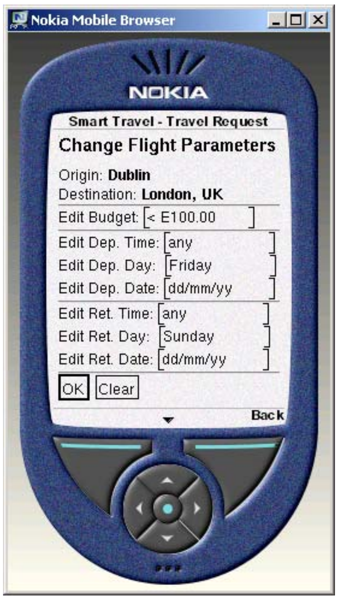


Smart Travel - Booking a Flight

Travel Request

□ Origin: Dublin, Ireland
□ Destination: London, UK
☒ Budget: <E200
☒ Departure Date: ASAP
□ Return Flight
☒ Return Date: 7 days later

Fig. 4. Initial Travel Request. The trip parameters are shown with default values. John either selects the parameters he wants to change or accepts the defaults and continues to the next screen.



Smart Travel - Travel Request

Change Flight Parameters

Origin: **Dublin**
Destination: **London, UK**
Edit Budget: [< E100.00]
Edit Dep. Time: [any]
Edit Dep. Day: [Friday]
Edit Dep. Date: [dd/mm/yy]
Edit Ret. Time: [any]
Edit Ret. Day: [Sunday]
Edit Ret. Date: [dd/mm/yy]

Fig. 5. Changing Initial Parameters. The parameters John wants to change are shown in more detail. He changes the `Budget, DepartureTime` and `ReturnTime parameters.`

John connects with the PTA, which brings up a page giving him an initial travel request screen as shown in Figure 4. This contains a form with the mandatory `trip-summary` parameters; ordinarily these parameters are given default values based on his previous travel requests, however, since he is a first-time user he has none. He has filled in his registration form fully and it fits him into a category of users. This group comprises people who prefer to fly with budget airlines. Let's assume that he fits accurately into this group (since the assigning of users to groups is outside the scope of this paper). He is offered the default choice of a return flight from Dublin (which is his home city) to London (the most popular destination within his grouping) leaving as soon as possible with a budget of under €200. John is happy with this travel request, except he would like to reduce his budget to less than €100 and would like to fly out on a Friday and return on a Sunday. He does this by selecting the parameters he wishes to change and editing them in another screen (Figure 5).

The skeletal travel request, as sent to the PTA is now:

`Origin` =	Dublin, Ireland
`Destination` =	London, UK
`Budget` =	< E100
`DepartureTime` =	Friday (`DayofWeek`)
`ReturnTime` =	Sunday (`DayofWeek`)

This request is not sufficient for the PTA to complete a trip-summary object so it initializes the personalization process with the above parameters as a target case. The PTA composes cases from John's neighbours' transactions (since John has no previous interactions). It uses the origin and destination as constraints on the raw data, only composing cases with the values Dublin-London (or London-Dublin) to get the offer in context; it would not be appropriate to use requests for transatlantic flights to solve this problem (although it may be appropriate to use European flights, this will become clearer with user feedback). This returns a case-base from the most relevant transactions in the system. Cases based on initial travel requests that yielded good travel offers that interested the user are given increased importance.

The CRN retrieval lists the final activation of all the cases in its case-base. The PTA uses the cases with the highest activation to decide on a final travel request. The travel-time parameters - `DepartureTime` and `ReturnTime` are composed from the micro-features `DayofWeek` and `TimeofDay`. John specified `DayofWeek` himself and `TimeofDay` is elaborated from the solution cases. All that remains is the actual dates used. Because John left the date field blank in his skeletal travel request the PTA assumes that he is looking for a suitable offer in the near future. The PTA will therefore be comfortable making a list of suitable travel times that fit those criteria to include in the trip-summary object.

The solution cases will either be quite similar, in which case we can easily compose a single final travel request from them; or they will be different, in which case we can make several travel requests to cover all possibilities. John's elaborated travel request looks like this:

`Origin` =	Dublin, Ireland
`Destination` =	London, UK
`Budget` =	< E100
	(1) After 20/9/02 14:00, before 20/9/02 23:00
`DepartureTime` =	(2) After 27/9/02 14:00, before 27/9/02 23:00
	(3) After 4/10/02 14:00, before 4/10/02 23:00
	(1) After 22/9/02 18:00, before 22/9/02 00:00
`ReturnTime` =	(2) After 29/9/02 18:00, before 29/9/02 00:00
	(3) After 06/10/02 18:00, before 06/10/02 00:00

Example 2 – Margaret

A corporate user, Margaret, has arranged to attend a conference in her company's headquarters in San Francisco. She is based in London and uses the PTA to arrange the travel arrangements to fit in with the times of the conference. She is a regular user of the PTA service. She has filled in an initial travel request, which was elaborated to this final travel request:

Origin =	Any Airport, London (LON), UK
Destination =	San Francisco Intl. Airport (SFO), San Francisco, Ca. USA
DepartureTime =	After 29/09/02 18:00, before 30/09/02 16:00
ReturnTime =	After 05/10/02 10:00, before 07/10/02 00:00
Selection =	By Comfort
FareType =	First Class

The PTA sends this single request to several TBAs, who return a number of travel offers. The PTA filters these by the final travel request itself (it's possible that a TBA would make an offer that does not satisfy the original request). It then begins the personalization process to filter the offers further.

Retrieval begins before the offers are received. It uses the final travel request to assemble a case-base where the cases contain previous offers that fit the travel request and the level of user-interest they drew when they were received, i.e. whether they were offered to the user, examined by the user or booked by the user. This case-base is stored in a CRN and as travel offers arrive, they are presented to it. The following travel offer was presented to the CRN:

Origin =	Gatwick Airport (LGW), London, UK
Destination =	San Francisco Intl. Airport (SFO), San Francisco, Ca. USA
Carrier =	British Airways
DepartureTime =	29/09/02 11:15 (Sunday)
ReturnTime =	06/10/02 16:40 (Sunday)
Price =	£6531.90Stg
FareType =	First Class

The target case is created from this offer; it is made up of the offer less the constrained parameters (since all offers will meet these).

Case n4 accumulated the highest activation. The offer associated with case n4 had been presented to Margaret before but she wasn't interested in it; she didn't request further information on it or book it. Because Margaret ignored n4's offer the PTA will give a relatively low recommendation score to the current offer. All offers received from the TBAs are given a recommendation score in the same manner. When all the travel offers have been received (either by all the TBAs responding or timing-out) the highest scoring travel offers are presented to Margaret as recommended offers. There is a scoring threshold above which an offer will be recommended. This figure may change depending on how discriminating Margaret is over time.

The possibility of several recommended offers combined with the lack of screen space means that offers should be condensed for presentation. Only the most important features offers are displayed to allow easy comparison between offers. These features are chosen based on user preferences in much the same way relevance arc weights are chosen in CRN maintenance. The presentation of Margaret's offers is shown in Figure 7. These offers are added to Margaret's personal profile with the interest she showed towards each offer. Offers that were rejected by the PTA are not stored. Figure 8 shows the screen presented to Margaret when she requests further information on the offer mentioned previously.

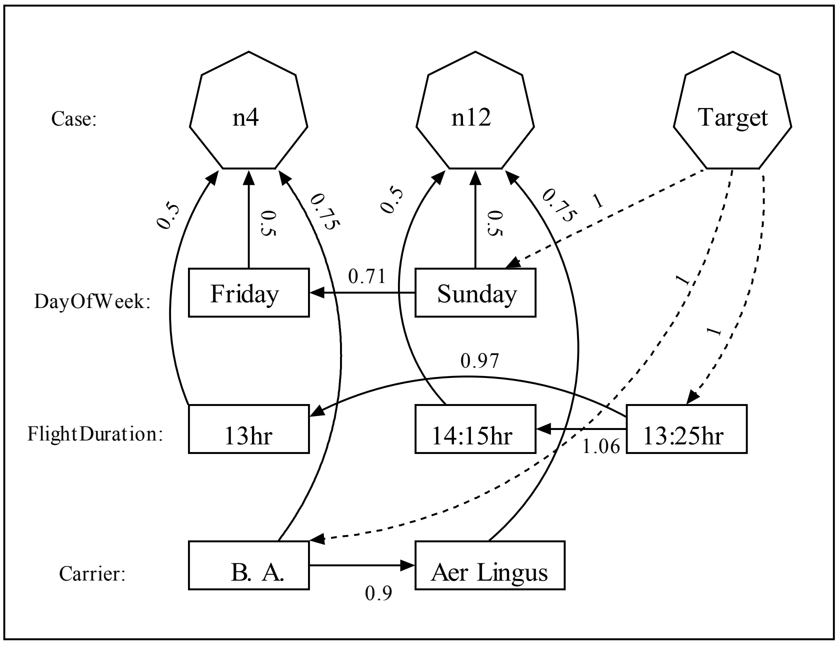


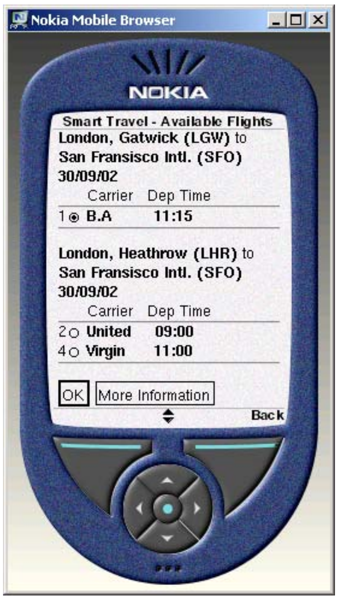
Fig. 6. A section of the CRN. For the purposes of clarity, this diagram only shows two cases and three features: `DepartureTime.DayOfWeek`, `FlightDuration` and `Carrier`. The target case is connected to the net as shown. It is activated and the scores at cases n4 and n12 are accumulated. The arc weights are shown but not the similarity functions. The final activation scores are: n4 = 2.205, n12 = 1.945

5. Summary and Conclusions

In this paper we have described the FIPA PTA scenario. This involves two personalization tasks: request elaboration and offer-recommendation, which the PTA could perform to ease user interaction with the overall travel system. We have outlined how a case-based user profile can support this personalization process and how CBR tools could be used to perform these tasks. This case-based approach has the advantage that the profile is always up to date and has the potential to borrow cases from similar users when coverage is poor. Further, the strengths of the CRN architecture in case completion are particularly useful in the request elaboration task.

In operation, the system has stored a personal profile for each user. In order to make a travel request, the user makes a skeletal travel request outlining the most important trip parameters. The PTA elaborates this offer using the user's travel preferences into a full travel request and sends it to the PTM on the user's behalf. The

PTM returns a number of travel offers that satisfy this request. The PTA filter these using the user's preferences and send the most suitable to the mPTA to presenst to the user.



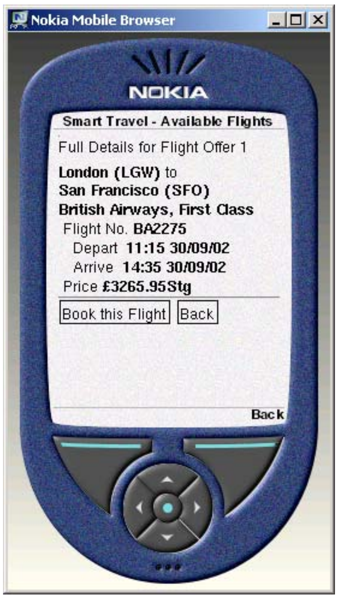
Smart Travel - Available Flights

London, Gatwick (LGW) to
San Fransisco Intl. (SFO)
30/09/02

	Carrier	Dep Time
1 ◉	B.A	11:15

London, Heathrow (LHR) to
San Fransisco Intl. (SFO)
30/09/02

	Carrier	Dep Time
2 ○	United	09:00
4 ○	Virgin	11:00



Smart Travel - Available Flights

Full Details for Flight Offer 1

London (LGW) to
San Francisco (SFO)
British Airways, First Class
Flight No. BA2275
Depart 11:15 30/09/02
Arrive 14:35 30/09/02
Price £3265.95Stg

Fig. 7. Presented Travel Offers. This screen shows travel offers with Margaret's most important features. She can proceed to request more information on an offer and book it.

Fig. 8. Examined Offers. Margaret selected this offer for further information. This screen shows the full details for this travel offer and allows her to instruct the PTA to book it.

Two example interactions with the system are described to further demonstrate the personalization tasks and their implementation. The first interaction demonstrated request-elaboration and how collaborative CBR could be used to allow first-time users of the system to benefit from the experiences of their peers. The second interaction demonstrated offer-recommendation and how travel offers are rated against previous cases that were offered to a user based on that user's preferences towards similar offers in the past.

The scenarios described here show that the CBR solution can perform request elaboration and offer recommendation. The next step in this research is to evaluate the utility of this approach. Since the success criterion is user satisfaction, the standard Machine Learning-type evaluation is not appropriate. This evaluation will need to be done as user trials on a real system.

References

1. Aha, D. W. (1997). Editorial on Lazy Learning. Artificial Intelligence Review, 11, 7-10.

2. Bergenti, F. & Poggi, A.. LEAP: A FIPA Platform for Handheld and Mobile Devices. In *Proc. Eighth International Workshop on Agent Theories, Architectures, and Languages (ATAL-2001)*, Seatle, CA, 2001.
3. Bradley, K., Rafter, R. & Smyth B. (2000) Case-Based User Profiling for Content Personalisation. In Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, (AH2000), Trento, Italy.
4. Foundation for Intelligent Physical Agents – ACL Message Structure Specification (Document No. XC00061E).
5. Foundation for Intelligent Physical Agents - Personal Travel Assistance Specification (Document No. XC00080). Carried forward from FIPA 1997 Specification 4 V1.0.
6. Hayes, C. & Cunningham, P. (1999) Shaping a CBR View with XML. *Third International Conference on Case-Based Reasoning*. ICCBR-99 Seeon Monastery, Germany, pp.468-481
7. Hayes, C., Cunningham, P., & Doyle, M.. Distributed CBR using XML. In Proceedings of the KI-98 Workshop on Intelligent Systems and Electronic Commerce, number LSA-98-03E. University of Kaiserslauten Computer Science Department, 1998.
8. Lenz, M., Auriol, E. & Manago, M. (1998) Diagnosis and Decision Support. In M. Bartsch-Sporl, H. D. B., and Wess, S., eds., Case-Based Reasoning Technology: From Foundations to Applications, volume 1400 of Lecture Notes in Computer Science. Springer-Verlag.
9. Smyth, B. & Cotter, P., (1999) Surfing the Digital Wave: Generating Personalised TV Listings using Collaborative, Case-Based Recommendation, in Proceedings of *3rd International Conference on Case-Based Reasoning* eds K-D. Althoff, R. Bergmann, L. K. Branting, Lecture Notes in Artificial Intelligence 1650, V pp561-571, Springer Verlag.
10. Smyth, B. & McClave, P. (2001) “Similarity vs. Diversity”, Proceedings of the 4th International Conference on Case-Based Reasoning. Vancouver, Canada.
11. Waszkiewicz, P., Cunningham, P. & Byrne, C., (1999) Case-based User Profiling in a Personal Travel Assistant, *User Modeling: Proceedings of the 7th International Conference, UM99,* Judy Kay, (ed).pp. 323-325, Springer-Wien-New York.
12. Watson, I. & Gardingen, D. (1999). *A Distributed Case-Based Reasoning Application for Engineering Sales Support*. In, Proc. 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99), Vol. 1: pp. 600-605. Morgan Kaufmann Publishers Inc. ISBN 1-55860-613-0