Adaptive User Interfaces for Automotive Environments

Seth Rogers Claude-Nicolas Fiechter Cynthia Thompson
Daimler Chrysler Research & Technology Center
1510 Page Mill Rd
Palo Alto, CA 94304
{rogers, fiechter, thompson}@rtna.daimlerchrysler.com

Abstract

Our research group is investigating the use of adaptive user interfaces for in-car information access. These interfaces attempt to efficiently provide content the driver needs and wants, and gather feedback on these preferences through the driver's interaction with the system. In this way, the performance of the system improves as it unobtrusively builds a more accurate model of the user. The three systems presented here are the Adaptive Route Advisor for navigation, the Adaptive News Reader for news stories, and the Adaptive Place Advisor for restaurant selection. All of these systems provide useful information to a driver, and we argue they do not negatively impact safety because they are replacing other, less effective, information sources. We intend to test this hypothesis in future studies.

1 Introduction

There are few environments where the need for well-designed human-computer interfaces is more essential than the vehicle interior. Unlike desktop environments, the user can only spare short bursts of attention from his primary task, operating the vehicle in a safe manner. These bursts of attention come at unpredictable times for unpredictable durations, so it is critical that the driver receives the information he needs quickly and with minimal interaction.

Interface design for in-car systems is clearly becoming more difficult as more information sources are made available to the driver. In addition to the traditional driving controls, instrument panel, and possibly cassette radio, modern vehicles increasingly include CD players, mobile phones, and navigation computers. Before long, vehicles will arrive with wireless Internet connections, and the amount of information available to the driver will be virtually limitless.

However, the advancement of technology, particularly replacing "real" displays and controls with virtual ones on a screen, also provides an opportunity to improve the in-car interface, reducing required interactions and making important information more salient. The aim for such "intelligent" driver interfaces should be to serve the driver's information needs and, at a minimum, not increase safety risks. This paper will describe several prototypical systems whose interfaces attempt to meet these criteria. One possible argument that an interface does not decrease safety is that the interface *replaces* a pre-existing physical task, and that the distraction of operating the interface is no more distracting than the previous task. We use this argument in each of our examples.

One aspect all the following systems share is that they are capable of *adapting* to the driver. That is, the systems change the content they present based on their understanding of the particular individual they serve.[1] This lets the driver get more appropriate information quicker, because the system tailors its responses to fit the driver's needs and preferences. All of the systems encode their understanding of the driver in a *user model* [7] that represents an estimate of the characteristics of the driver relevant to the particular application. All the systems build the user model *unobtrusively*, requiring neither explicit setting of preferences or rating of the system's performance, such as [6, 8]. Instead, the systems were designed to build their user models with implicit feedback through normal interaction. This is a critical property for in-vehicle interfaces, since drivers have no attention to spare for anything more than simply interacting with an interface to find the information he needs or wants.

2 The Adaptive Route Advisor

Current systems for route advice compute solutions using a shortest-path algorithm to find the minimal-cost route from the origin to the destination. Some systems fix the cost as the estimated travel time, while others let the user choose between the shortest path, the quickest, or the "most scenic" one. In all cases, the system then describes the route to the user with little or no recourse if the driver finds the route unsatisfactory. These systems disregard the

[1] Also, future work is underway to change the *form* of presentation to best fit the individual.

fact that driving occurs in a rich environment where many factors influence the desirability of a particular route. For example, some drivers may prefer the shortest route as long as it does not have too many turns, or the fastest route as long as it does not go on the highway. The relative importance of these factors varies among individuals, and drivers may not know themselves what they value most in routes.

For this system, described more fully in [9], the user model consists of two components: general preferences and specific knowledge. General preferences are the relative importance of some global measures of a route, such as distance, estimated duration, number of left turns, and unfamiliarity.[2] Specific knowledge is particular segments on the digital map the driver has traversed, as recorded by GPS. The unfamiliarity of a route is the distance traveled along untraversed segments. The system initializes the agent with a default user model and refines this model by monitoring which route the drivers selects from a set of options.

2.1 The routing algorithm and user model

The generative component of the Adaptive Route Advisor is a routing algorithm that plans a path through a digital map from a starting point to a destination. The planner represents the digital map as a graph, where the nodes are intersections and the edges are segments of roads between intersections. Our digital maps provide four attributes for each edge: length, estimated driving time,[3] turn angle to connected edges, and road class (e.g., highway, freeway, arterial road, local road). The planner refers to these digital maps to minimize the weighted sum of fourteen route attributes using an optimized version of Dijkstra's shortest path algorithm [2].

The weight vector in the Dijkstra cost function allows attributes to have different relative importance. As the weight on an attribute increases, the cost associated with that attribute increases, making solutions with low values for that attribute more likely to be optimal. The relative importance of each attribute is exactly the preference captured in the general user model, so the user model itself serves as the weight vector.

Since it is difficult and inconvenient for a user to specify his relative preference for each attribute, our system automatically induces a driver's preferences from his route choices. We have implemented a large margin training algorithm [3] that processes a sequence of interactions with the planner and produces a weight vector that models the preferences expressed. In this way, as the driver uses the interface, it adapts itself to his preferences.

We define an interaction with the planner to be the presentation of a set of N generated routes and feedback from the user indicating which route is preferable. This is completely unobtrusive to the user, because he or she evaluates a set of routes and selects one as part of the route advice process. For training, we expand the interaction into $N-1$ pairs, representing the fact that the selected route is preferable to each of the presented alternatives. These training pairs can be used to improve the user model in a simple manner. If, out of the two routes in a training pair, the route preferred by the current user model is not the one the user selected, the adaption method increases the weights corresponding to the features in the selected route and decreases those corresponding to the features in the other route.

Once the learning algorithm finds a weight vector that best predicts preferable routes as a weighted sum of attributes, the routing algorithm uses this weight vector in its cost function. Since the routing algorithm is optimal on the cost function, the resulting route is guaranteed to have the lowest cost for that user model among all routes between the same two nodes. In other words, the routes computed are always Pareto optimal, in that there can be routes that are better along each of the dimensions (attributes) independently, but none that can be better simultaneously on all dimensions.

2.2 The interaction module

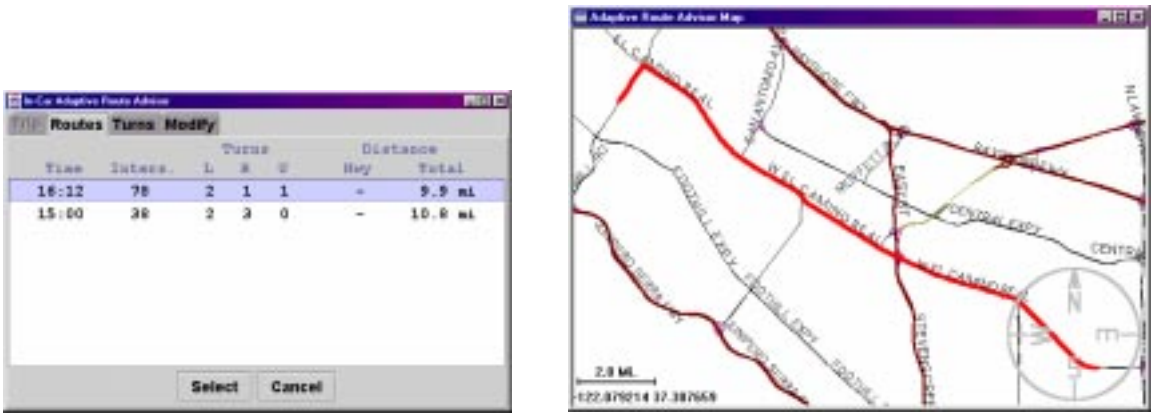
The Adaptive Route Advisor is designed for in-car use. It is a Java application that functions as a resource-light network client, suitable for mobile environments with a wireless communication infrastructure. The remote servers provide resource-intensive functions such as routing and geolocation.[4] Although the current version does not yet take advantage of information available from mobile deployment (primarily current and past locations from GPS) and the interface is not fully optimized for limited input and output resources common in vehicles, future work will more firmly embed the Adaptive Route Advisor in a mobile environment.

After requesting a route, the main interaction window appears, as displayed in Figure 1(a), providing a list of current route options and three tabs, "Routes," "Turns," and "Modify." The current routes are presented in terms of seven attributes: total time, number of intersections, number of left turns, right turns and U-turns, total distance, and distance on highways. Initially the agent presents two routes to the user. The first uses the current preference model as the weight vector for the routing cost function. The second route uses novel weights, selected from a small set of prototypical user models, in an attempt to explore

[2] We assume the driver prefers familiar segments over unknown ones, so we need to minimize unfamiliarity.

[3] Future versions will use current traffic conditions [5].

[4] Geolocation is mapping a plain English street location to its place in a digital map structure.



(a) The route selection window.

(b) The map window.

Figure 1: Initially, the user is shown two alternative routes, the best route according to the current user model being highlighted in the route selection window (a) and displayed in the map window (b).

new directions in the space of preference models.

Presenting at least two route options forces the user to make a choice and provide some feedback to the agent. The turn directions for the selected route are shown in the "Turns" tab and the map displays the selected route in a separate window, as shown in Figure 1(b). Clicking "Select" indicates that the highlighted route is satisfactory and closes the window. The route advisor assumes that the highlighted route is preferable to the alternative routes and updates the user model. Clicking "Cancel" closes the window but does not update the model.

The "Modify" tab lets the user generate a new route that is faster, shorter, has fewer turns, has fewer intersections, or has less or more highway than the selected route. The implicit assumption is that the driver is willing to accept routes that are somewhat worse on other attributes if he or she can find one that is better on the selected attribute.

We have not tested the attention requirements of our interface, but we assume that our application replaces consulting a paper map. So, we hypothesize that as long as the interface requires no more attention than unfolding a paper map, localizing on the page, and tracing a plausible route to the destination, it does not increase safety risks. In fact, navigation systems should decrease safety risks, because the system localizes and computes routes automatically. Moreover, a personalized navigation system should further reduce safety risks, because the driver is more likely to use the system efficiently and follow the suggested route.

3 The Adaptive News Reader

The Adaptive News Reader is a system that recommends and reads news stories to the driver in the car. It has been developed in collaboration between DaimlerChrysler Research and Michael Pazzani and Daniel Billsus at the University of California, Irvine [1].

The system gathers news stories from a news server on the World Wide Web and reads them to the driver via a speech synthesizer. The driver can interact with the system through a touch screen as well as through a speech interface that recognizes a number of spoken commands. Based on the driver's feedback, the system automatically adapts to the user preferences and selects news stories that the driver is likely to find interesting.

The Adaptive News Reader implements a client-server architecture similar to that of the Adaptive Route Advisor: A resource-light, in-car client interacts with the driver and communicates with a remote server through a wireless link. The remote server is responsible for gathering the news from the world wide web and rating them for the different users.

3.1 The interface module

As the system starts, a channel selection window is displayed. When the driver selects a news channel the client connects to the remote server and starts downloading stories from that channel to prepare a personalized news mix. The system uses the content of each story and a model of the driver's interests to predict whether the driver will find it interesting. The stories are then read to the driver in the order of their predicted interest. The user model is built from the driver's feedback in previous interactions with the system and is described in the next section.

Figure 2 shows the in-car display as the system reads a news story to the driver. The interface displays one sentence at a time, as it is being read, and provides buttons for

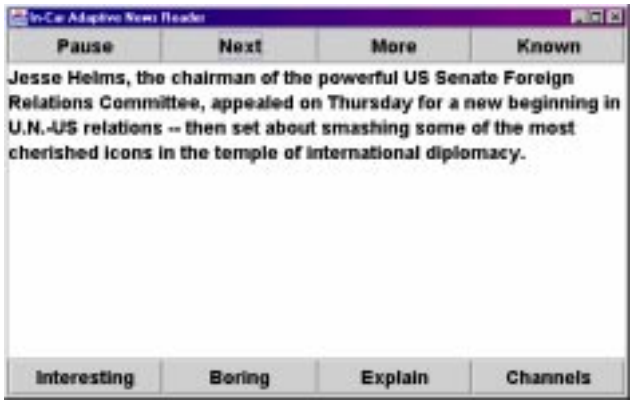


Figure 2: The news reader interface. The user can interact with the system via the touch screen and via voice commands.

the user's feedback. The driver can interrupt or skip a story at any time by pressing (or saying) "Next". He can also ask for more information about a story ("More"), in which case the system will try to find another story on the current topic and will start reading it to the driver. In addition, the driver can indicate that he already knows about a particular piece of news ("Known"), or specify directly whether he finds a story interesting or not ("Interesting" or "Boring").

3.2 The user model

The Adaptive News Reader uses these different forms of feedback to adjust its model of the driver's interests. In particular, if the driver interrupts a story early on, the system interprets that action as a negative feedback, indicating that the driver was not interested in the story. On the contrary, if the user listen to the whole story (or most of it), the system assumes that the driver found it interesting. Similarly, asking for more information about a story is also taken as a positive feedback.

The system actually maintains two complementary models of the driver's interest: a short term model and a long term model. The short term model rates a story based on its similarity with other stories that have been recently read to the user. This model lets the system capture the changing user interest about current events (e.g., a plane crash that is a major news story for a week) and is also used to avoid repeating the same information twice to the user. The long term model attempts to capture the driver's general preferences for new stories. It uses a probabilistic learning algorithm to compute a prediction for the stories that could not be rated by the short term model.

This system partially replaces another common yet dangerous driving distraction: finding interesting news radio. Rather than manually searching for in-range radio stations broadcasting personally relevant news stories, the Adaptive News Reader immediately begins reading the news it predicts the driver will find most interesting. If the story is actually uninteresting, simple spoken feedback skips to the next story and corrects the user model. As the user model becomes more accurate, less and less feedback is necessary, and the driver can concentrate more fully on driving.

4 The Adaptive Place Advisor

Finding a destination is a common task while driving. This task can be described as a process of querying a place database, with attributes describing the places and the query specifying desired values for the fields. Unfortunately, typical database interfaces are difficult to transfer directly into the vehicle. The user has to select from a list of values or even type attribute values, and the results of the query may be an unmanageably large list or too few options, so the user has to manually modify the query and send it again.

Our system, the Adaptive Place Advisor [4], demonstrates a conversational interface to a database that eliminates these shortcomings. We view the selection of destinations as an interactive process, with the advisory system proposing attributes and the human responding. The system acquires the user's needs in a conversational manner, enhances the agreed upon query with a user model, and retrieves suitable destinations from a database. The conversational nature of our system lets us create and update user models without requiring the user to provide direct feedback. The preferences of the user are instead derived from her interactions with the system.

Our prototype system aims to help drivers select a restaurant that meets their preferences, based on a database of about 2000 Bay Area restaurants. To be able to recommend a restaurant based on a conversation, the Adaptive Place Advisor engages in a dialogue with the driver. Figure 3 shows the structure of the system and the modules that enable this dialogue. The Dialogue Manager generates and recognizes utterances in the conversation. The Retrieval Engine is a case-based system that uses the query that has been generated and updated by the Dialogue Manager to retrieve restaurants from the database. The User Modeling System generates the initial (default) query from the user model and updates the user model based on the conversation history. The Speech Recognizer and the Speech Generator comprise the natural language processing part of the system. The Speech Recognizer was developed based on a commercial NLP package from Nuance, whereas the Speech Generator uses pre-recorded prompts.

4.1 The dialogue

Based on the restaurants in the database and the user model (described in the next section), the Adaptive Place Advisor

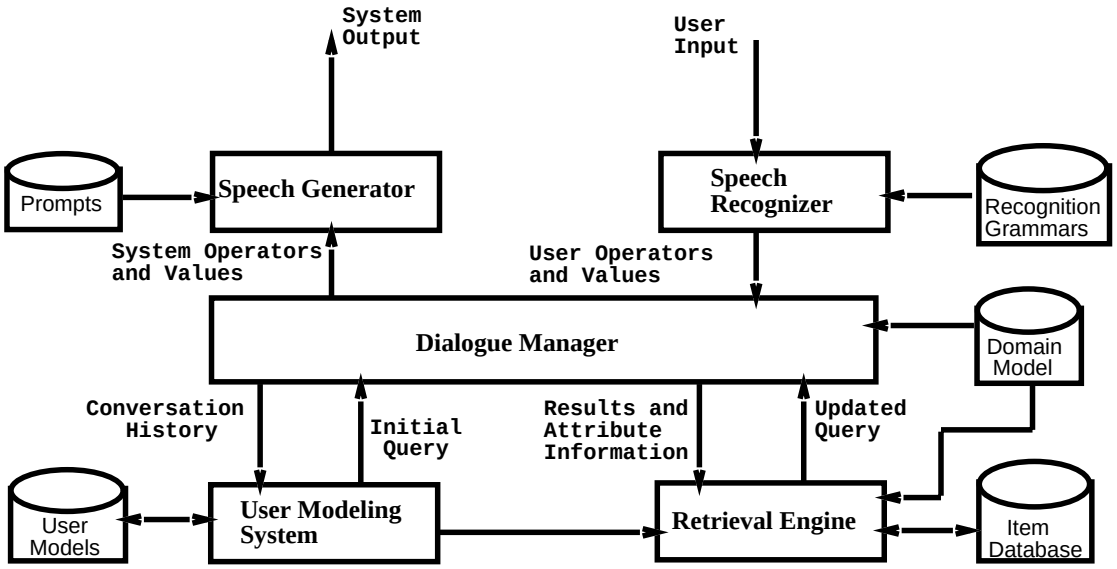


Figure 3: Overall System Architecture of the Adaptive Place Advisor.

iteratively selects a question and refines the query based on the answer. When the result set is small enough, it presents the options to the driver. The options are currently displayed on a screen, because the system has no text-to-speech generation capability. Future versions will include speech synthesis and integration with a navigation system to guide the driver to his selected restaurant.

The dialogue proceeds by generating and recognizing dialogue operators, based on the current state of the conversation. Operators available to the system include asking questions designed to constrain or relax the query, suggest possible attributes and values for an attribute, recommend an item, and clarify a user's statement. The system interprets the user's responses as providing a constraint or relaxation, rejecting a constraint or relaxation, accepting or rejecting an item, or requesting a list of possible attributes or values for an attribute. The following conversation is a sample interaction between the driver and the Adaptive Place Advisor (APA):

APA: What type of food would you like to eat today?
Driver: What types are there?
APA: Some examples are Chinese, Indian, and Mediterranean.
Driver: Oh, maybe Indian.
APA: What quality rating would you like?
Driver: I don't care, as long as it's cheap.
APA: How do you want to pay?
Driver: No, I think I'd like Chinese instead.
APA: What city do you prefer?
Driver: How about Berkeley?
[system displays a restaurant title and address]
APA: How about this restaurant?
Driver: Sure, that sounds fine.

4.2 The user model

We refer to the Adaptive Place Advisor as "adaptive" because its design includes a module that constructs user models that represent the driver's preferences from traces of conversations. However, our goal for user modeling differs from the one commonly assumed in recommendation systems, which emphasizes improving accuracy or related measures like precision and recall.

The conversation, like constraint satisfaction, will direct the system to a suitable solution. Our goal is to improve both the subjective quality of the results and the dialogue process itself. Just as interactions with a friend who knows your concerns can be more directed and produce better results than those with a stranger, dialogues with the Adaptive Place Advisor should become more efficient and effective over time.

To efficiently provide the users with the solution that matches their needs bests, it is necessary to acquire and model the preferences of the users. A user may have preferences about items in general, an attribute, a value, and combinations of certain attribute-value pairs (properties) of an item. The system learns these preferences based on the items the user selects and the direction the user steers the conversation.

Since the value preferences can be viewed as a probability distribution over the values for each attribute, the user model can be used as an initial query with default values.

In the course of the conversation, this initial query is refined and constrained with the values the user specifies for each attribute (i.e. the probability for that value for the respective attribute becomes 1). When a user finally accepts an item, the user model is updated based on the current query, also taking into account any items rejected by the user.

This system attempts to emulate a conversation with a friend. Although no natural language system is currently capable of human-level dialogue, or human-level understanding, our restaurant database is likely to be much more comprehensive, complete, and accessible than any human's memory. Because of this, conversations may be more effective and successful. In the absence of a knowledgeable passenger, the Adaptive Place Advisor replaces a paper reference like the Yellow Pages. Similar to the Adaptive Route Advisor replacing paper maps, the advantages are obvious.

5 Future Work and Conclusion

User interfaces in the car must tread the thin line between serving the user's needs and not causing unnecessary safety risks. Until and unless we attain the goal of automatic driving, safety is the ultimate responsibility of the individual driver. Since it is impossible to guarantee a "completely safe" interface, interface designers must work toward the more attainable objective of at least not worsening the safety situation, and leave the task of improving vehicle safety to others.[5]

Since it is difficult and dangerous to measure the additional safety risks of experimental interfaces in real traffic, our research group has argued that our interfaces *replace* existing tasks (with corresponding risks) with new, computer-mediated tasks, that are more knowledgeable and convenient, and therefore likely to be less risky. Moreover, we claim that *adaptive* interfaces are key elements in creating usable interfaces in the demanding vehicular environment, because they give the driver quicker access to the information he needs or wants, in a more appropriate form. These arguments notwithstanding, it is crucial to quantify safety measures as well. In our future work, we intend to install the above systems on realistic simulators and actual vehicles to test if they are no more dangerous than the tasks they replace.

As communications and computing capabilities improve, consumers will demand more information access within their vehicles. Although the standard of safety will increase with new developments in materials, sensors, and control, the introduction of richer, more distracting interfaces that do not simply replace current tasks must be undertaken with great care that safety levels do not degrade arbitrarily, perhaps even lower than current levels. It is our responsibility to provide interfaces that are as safe, useful, and convenient as possible, probably with heavy reliance on personalization. It is, and always has been, the responsibility of the driver to judge his own level of safe attention to the primary task of controlling his vehicle.

[5] Although an interface may well be part of an onboard active safety system, for example.

References

[1] Daniel Billsus and Michael J. Pazzani. A personal news agent that talks, learns, and explains. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 268–275, New York, NY, 1999. ACM Press.

[2] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, Volume 1, pages 269–271, 1959.

[3] Claude-Nicolas Fiechter and Seth Rogers. Learning subjective functions with large margins. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 287–294, Stanford, CA, 2000. Morgan Kaufmann.

[4] Mehmet H. Göker and Cynthia A. Thompson. The Adaptive Place Advisor: A conversational recommendation system. In *Proceedings of the Eighth German Workshop on Case-Based Reasoning*, pages 187–197, Lämmerbuckel, Germany, 2000. DaimlerChrysler.

[5] Peter Hadfield. Smart cars steer round traffic jams. *New Scientist*, April 26 1997.

[6] Ken Lang. NEWSWEEDER: Learning to filter news. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, Lake Tahoe, CA, 1995. Morgan Kaufmann.

[7] Pat Langley. User modeling in adaptive interfaces. In *Proceedings of the Seventh International Conference on User Modeling*, pages 357–370, Banff, Alberta, 1999. Springer.

[8] Jack Muramatsu Michael Pazzani and Daniel Billsus. Syskill & Webert: Identifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 54–61, Portland, OR, 1996. AAAI Press.

[9] Claude-Nicolas Fiechter Seth Rogers and Pat Langley. An adaptive interactive agent for route advice. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 198–205, New York, NY, 1999. ACM Press.