The Architecture of Future Automotive Applications based on Web Technologies

Luka Bradesko[1], Marko Grobelnik[1], Andreas Harth[2],
Achim Rettinger[2], and Sebastian Speiser[3]

[1]Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana, Slovenia
Email: firstname.lastname@ijs.si
[2]Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany
Email: firstname.lastname@kit.edu
[3]Karlsruhe Service Research Institute (KSRI), KIT, Germany
Email: firstname.lastname@kit.edu

October 29, 2012

Abstract

In this paper we propose the building blocks of a driver assistant system. The system includes i) a conversational interface for communication between user and system ii) a virtual knowledge base which consists of a stable core of world knowledge (Cyc) and a large pool of instance data (Linked Data from the web) and iii) an interoperation layer which interconnects the knowledge base and individual dynamic data sources (from vehicle sensors to social network streams) to support the driver in tasks such as finding and reserving parking or other points of interest, and staying in touch with other drivers via social networks.

1 Introduction

Driving vehicles is rarely done for its own sake. Rather drivers want to achieve specific targets that are often more complex than coming from A to B, e.g., car drivers who want to find the cheapest petrol station on their way to work; truck drivers who want to dynamically adjust their delivery route according to traffic; camper van drivers who want to optimise their time of being autonomous. In order to achieve their goals, the drivers have to interact a lot with the vehicle, e.g., handling the radio, following the navigation system, answering phone calls, or adjusting and monitoring vehicle settings.

Because the most important thing while driving and interacting with the vehicle is safety, it is very important that the interaction activities do not interfere with the driver's focus on the road. The natural way to achieve non-interference is to move as much interaction as possible to the speech and natural language domain. Thus, it is possible to introduce new and more complex ways to interact with the vehicle without affecting safety. Speech interfaces can be even more beneficial in bigger vehicles such as trucks or camper vans which usually travel longer distances and drivers practically live in them.

Our plan is to develop a system that will serve as a driving assistant. The assistant will be aware of the context of the user, including the current location, the status of the vehicle, the purpose of the travel, etc. The assistant will be able to connect to the car's sensors, external sources of data and other users as well, to provide an additional and a social way of getting new information and interacting with the vehicle and other drivers. We are using web technologies to create a system, which is open to new information sources, and supports the scalable integration of the heterogeneous data from these sources.

Figure 1 shows the building blocks of the system. In the following sections, we discuss the main building blocks in more detail.

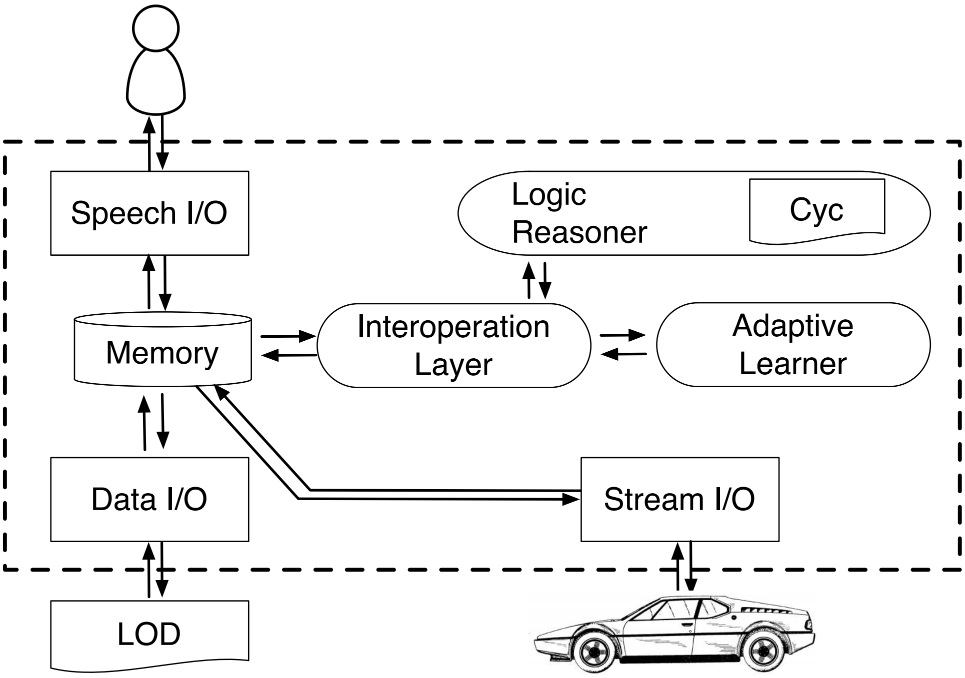

Figure 1: High-level architecture.

2 Speech I/O: Conversational Interface

For the conversational part of the system, we will start with a novel approach which will not be focused around patterns, as it is common in current state of the art conversational clients including AIML [11] and ChatScript engines [12]. We plan to use patterns only to help us converting text to logic [6]. Once the user's questions or statements will be encoded in logic, we can use a reasoning engine to control the conversation and construct the replies. For that we will use Cyc's reasoner and its logic-to-language translation capabilities [3]. When moving chat engine focus from patterns to logical reasoning, we get deeper reasoning capabilities immediately and we get rid of the need to emulate reasoning through lots of prepared and canned patterns. We will first focus on English language, but the system is set up for possible extensions with other languages as well. Thus, we will be able to some extent automatically cover multiple languages with only small amount of patterns.

3 Adaptive Interaction Component

Existing systems typically react in a static predefined manner to user input. We propose a learning system for symbolic interaction of the driver assistant with the human users. A learning system addresses several problems: out of the many potential actions of the driver assistant the system learns to execute the one that satisfies the individual user needs best. Instead of simple one-step action-reaction patterns the system learns sequences of actions leading to optimal states. The component not only learns when and how to react, the system also can learn the meaning of concepts that are not yet grounded in the knowledge base.

To achieve a flexible learning system, the Adaptive Interaction Component on the one hand relies on the rich background knowledge provided by the knowledge base and the deductive reasoning component. This is crucial to identify the set of consistent and safety conform actions for the assistant at all times. On the other hand it analyzes the constant input stream from the speech component and car sensors representing user behavior. By correlating the reactions of the human with the actions executed by the driver assistant, using an approach based on ideas from [7].

4 Memory: Virtual Knowledge Base

We build the memory of our system as a virtual knowledge base by using the REST [5, 9] and Linked Data [1] principles. By building on the principles underlying the web architecture, we benefit from established and proven technologies that a large community of users and experts is using to build and operate large distributed systems. The basic abstraction in our virtual knowledge base is the resource. Different representation of the same resource (e.g., an image can be returned in PNG or GIF format for the image itself, or XML or RDF for the image metadata) using content negotiation. The different representations are useful for example when considering a point of interest (POI): the multimedia display can show a picture of the POI to the driver using a PNG representations, whereas the navigation system uses a geo metadata representation for computing the optimal way to reach the POI. Operations on resources are create/read/update/delete. Resources return references (links) to other resources which are related (such as leading to the next step in a series of operations).

In combination with the resource abstraction we plan to employ RDF as a knowledge representation format, which enables domain modelling using a graph-structured format.

5 Interoperation Layer

The interoperation layer provides mechanisms to integrate information from different data sources and services. Besides the data sources on the web, we will integrate measurements from vehicle sensors as well as input from the driver given in response to questions asked by the system. Input from the sensors and users can provide information missing in the knowledge base that is useful either for enriching the knowledge base for later use, or for completing the requested task at hand. Creating workflows to achieve user goals requires both querying information as well as reacting by triggering actions. Query operations are based on a series of read operations on the relevant resources. We support deductive reasoning in the interoperation layer to integrate information from different sources and discover situations in which the execution of actions is necessary. We trigger actions by manipulating the state representations of the corresponding active resources.

6 Related Work

There has been work to use web technologies and Linked Data in cyber-physical systems, i.e., systems that bridge the virtual and the real world. Wagner at el. describe the challenges in data management and integration in Smart Grid systems [10].

The automotive industry has investigated the use of semantic technologies in several scenarios. Chevalier and Servant of Renault describe a system that uses Linked Data to encode product information about cars [2]. Parundekar and Oguchi of Toyota describe a system that supports drivers with finding points of interest [8]. The system learns driver preferences which are used in the recommendation and navigation process to personalise results. Our proposed system goes beyond the existing approaches by integrating live stream data from the car sensors and leveraging web data in a speech interface.

There were approaches in making conversational clients using Cyc before [4], but the approach was still pattern focused and it used Cyc just as a tool to get the answers on pre-defined questions which were still encoded in AIML patterns.

7 Conclusion

Vast amounts of data and knowledge that are relevant to driver assistant system are available to be accessed in the car, including web data and sensor measurements. In order to make some use of the valuable knowledge in principle accessible to the driver, there is a clear need for reducing its complexity and decide on its usefulness, before we can present it to the user and thus overcome the information bottleneck. We propose a system that is able to understand and process the available data and give users the possibility for interaction via a speech interface, which both brings benefits for usability and safety of driving.

Acknowledgments

This work was partially funded by the iZEUS project which is financially supported by the German Federal Ministry of Economics and Technology.

References

[1] T. Berners-Lee. Linked Data – Design Issues, 27th July 2006. Available at `http://www.w3.org/DesignIssues/LinkedData`, accessed July 22nd 2012.

[2] E. Chevalier and F.-P. Servant. Product Customization as Linked Data. In *Proceedings of the 9th Extended Semantic Web Conference (ESWC'12)*, pages 603–617. Springer-Verlag, 2012.

[3] E. Coppock and D. Baxter. A Translation from Logic to English with Dynamic Semantics. In K. Nakakoji, Y. Murakami, and E. McCready, editors, *New Frontiers in Artificial Intelligence*, volume 6284 of *Lecture Notes in Computer Science*, pages 197–216. Springer-Verlag, 2010.

[4] K. Coursey. Living in CyN: Mating AIML and Cyc together with Program N. In *Daxtron Laboratories, Inc.*, 2004.

[5] R. Fielding. *Architectural Styles and the Design of Network-based Software Architectures.* PhD thesis, University of California, Irvine, 2000.

[6] D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.* Addison-Wesley, Reading, Massachusetts, 1990.

[7] M. Nickles and A. Rettinger. Towards interactive relational reinforcement learning of concepts. In *NIPS 2011 Workshop; Learning Semantics.* NIPS Foundation; Inc., Neural Information Processing Systems, Dezember 2011.

[8] R. Parundekar and K. Oguchi. Learning Driver Preferences of POIs Using a Semantic Web Knowledge System. In *Proceedings of the 9th Extended Semantic Web Conference (ESWC'12)*, pages 703–717. Springer-Verlag, 2012.

[9] L. Richardson and S. Ruby. *RESTful Web Services.* O'Reilly Media, May 2007.

[10] A. Wagner, S. Speiser, O. Raabe, and A. Harth. Linked data for a privacy-aware smart grid. In *GI Jahrestagung*, pages 449–454, 2010.

[11] R. Wallace. The elements of aiml style. In *ALICE A.I. Foundation*, 2003.

[12] B. Wilcox. Beyond Façade: Pattern Matching for Natural Language Applications. 2011.