Profiling and Service Delivery in Internet-enabled Cars

M. Cilia and A. P. Buchmann
Databases and Distributed Systems Group, Department of Computer Science
Darmstadt University of Technology - Darmstadt, Germany
<lastname>@informatik.tu-darmstadt.de

Abstract

This paper presents our experience applying web services and active database technology in an automotive scenario. Particularly, we concentrate on developing an infrastructure for customizing the user's experience in a vehicle. The motivation of this work is to allow the users to carry with them a personalized environment and apply their preferences not only to instrument adjustments and to maintenance and diagnostic information handling services but also to location-based services, infotainment and a briefing service that links to the user's office environment. We describe the infrastructure, the realization of the internal as well as the application services and discuss work in progress related to privacy and billing issues.

1 Introduction

Similar to other pervasive computing environments, cars will see a convergence of Internet, multimedia, wireless connectivity, consumer devices, and automotive electronics [5]. Assuming this context, wireless links between the car systems and the outside world open up a wide range of telematics applications. Automotive systems are no longer limited to information located on-board, but can benefit from a remote network and service infrastructure. Consider an automotive scenario, where vehicles, persons and devices have a web presence. Within this scenario new possibilities emerge, for example preferences can be stored and maintained in portals and then used for customization purposes in a uniform way. This avoids the known problem of defining many times your preferences in different systems. For illustration purposes just consider the adjustment of instruments in a vehicle. This can be done according to the personal settings stored in your portal. This approach not only provides the possibility of adjusting instruments of one vehicle, but the possibility of applying these settings (or at least part of them) to different cars. A frequent traveler can then use any rented car and it automatically adjusts its instruments (display of units of measurement, radio stations, car internal temperature, seat settings, etc.) according to the driver's preferences. But not only instruments can be adjusted, services can be personalized too. Services such as, "read my e-mail on the way to the office", "download/play my favorite music", "find and set the route to the next gas station", or "book an appointment to change oil" can take into account driver, car and company preferences.

This paper presents a case study of using an active functionality service for customizing user experience and invoking Web services. The active functionality, implemented as internal services, such as event detection

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

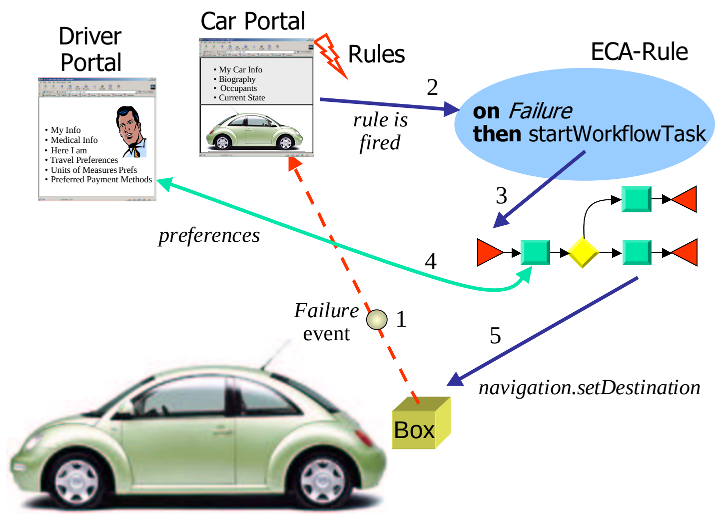

Figure 1: Car scenario

service, notification service and action invocation service, is used to coordinate and invoke external, application-specific Web-services. Information integration is accomplished through the use of ontologies. This paper is organized as follows: In Section 2 the automotive scenario and a set of related services are described. In Section 3 the requirements imposed by this scenario are presented and the infrastructure we have developed is described. Section 4 presents the ongoing and future research.

2 Scenario

This section describes a set of services that can be invoked as reaction to predefined situations according to the user's preferences.

Maintenance Services:

Part failure: The car detects a problem or reports a warning of a likely failure. A workflow is started to make the proper appointment (and depending on the severity of the problem) to find and set the route to the closest repair shop. This workflow takes into account the car's current geographical position, driver's (or car's) preferences. The workflow may also order a courtesy car or a taxi. Figure 1 illustrates this case.

Low fuel: When the car is running out of fuel a sensor signals this happening. As a reaction, a location service is invoked to find the next gas station considering current geographical position, destination and driver's preferences, e.g. the preferred vendor.

Adjustment of instruments: Once the driver gets into the car all her preferences (like, the units used for temperature, distance and velocity, the format of date/time, radio stations or music preferences, etc.) are automatically loaded. This is achieved because the driver uses her identification (e.g. badge, smartkey, smartcard) when getting into the car and as a reaction, driver's instrument preferences are read and the car's instruments are adjusted accordingly.

Briefing services:

Driving to work: Consider the case of a commuter and assume the following situation where the driver gets into the car. It is a workday and the current time is between 8:00am and 9:00am. As a reaction, the driver is requested to confirm the detected situation and the following set of actions can be performed: the best route to work is computed (avoiding traffic jams) and is passed on to the navigation service; today's scheduled meetings are checked; company news and other personalized news are obtained; and e-mails are read. Because drivers should concentrate on driving, all this information can be read out by using a text-to-speech service.

Location-dependent services:

Bilingual dictionary: When the car crosses the border of a country and taking into account the driver's profile (in particular spoken languages) a bilingual dictionary (and/or a currency convertor) is loaded in the driver's PDA and/or car's computer.

Restaurant locator: This service combines personal preferences, similar to well-known profiling services provided by portals like Yahoo!, but adjusted as a function of location. The internal ECA service can direct the choice to the on-board navigation service.

3 Infrastructure

The telematic scenario presented in the previous section illustrates many of the issues that should be considered when designing and building a customization infrastructure:

- Data heterogeneity: It is well-known that data from different sources can only be properly interpreted when sufficient context information about its intended meaning is known. For this reason, to exchange and process data from independent sources in a semantically meaningful way explicit information about its semantics in the form of additional metadata is required. [Preferences are applied to diverse vehicles.]
- Active mechanism: In many applications related to context-awareness the "automatic" detection and response to context changes is desired. To accomplish this, active database technology can be applied in order to avoid hard-wired code scattered in various applications and sensors. The required functionality is made explicit in form of rules that are under the responsibility of a rule engine. [getting into the car .. the adjustment of instruments and services occur automatically; low fuel .. find a gas station]
- Event handling: Events identify happenings of interest (e.g. part failure or driver gets into the car). These events may be consumed from a set of applications subscribes to them. For that purpose a notification service with publish/subscribe capabilities and asynchronous communication support is required.
- Service invocation: The interaction with third-party services is mandatory in scenarios where it is intended to interact with different services the user is habituated to use.
- Profile management: The user profile must be managed in order to satisfy two needs: semantic data representation and privacy aspects. The first issue is also related with the first bullet where relevant customization data must be interpreted by diverse systems. The second relevant issue relates basically to access control. Because the profile contains sensible information the access to this data must be strictly controlled.

With these requirements in mind, an ontology-based infrastructure was proposed and developed. On its foundation it relies on a self-describing data model, MIX [1] which uses shared concepts (ontologies) expressed through common vocabularies as a basis for interpretation of data and metadata. This provides the possibility to support the explicit definition of context information which describes the semantics of exchanged data in the form of additional metadata. Additionally, cleanly integrated conversion functions can be used for converting data coming from diverse systems and services.

Ontologies in the infrastructure are organized in three levels: a) the basic level, where elementary ontology functionality and physical representation is defined; b) the infrastructure level, where basically concepts of the active functionality domain are specified; and c) the domain-specific level, where concepts of the subject domain (e.g. telematics) are defined. An *ontology service* is used to store, manage and provide a common access point to the concepts of the underlying ontologies.

A *notification service* based on a publish/subscribe paradigm is responsible for delivering events to interested consumers. This service provides asynchronous communications, it naturally decouples producers and consumers, it makes them anonymous to each other, it allows a dynamic number of publishers and subscribers, and it provides location transparency without requiring a name service. The notification service uses *concept-based addressing* in order to provide a higher and common level of abstraction to describe the interests of publishers and subscribers. This addressing approach is based on the subject-based addressing principles where the subject name space is hierarchically organized. Here concepts are mapped to the subject name space where the concept name and some of its attributes form part of the subject.

Events produced by different applications are integrated by *event adapters* that convert source-specific events into semantic events (represented by ontology-based concepts enriched with semantic contexts). Events are disseminated to interested consumers by using the notification service.

On this foundation an *active functionality service* was developed. But instead of applying traditional Event-Condition-Action (ECA) engines [6], the ECA-rule processing mechanism was decomposed into its elementary and autonomous services. These services are responsible for complex event detection, condition evaluation and action execution. The rule processing is then realized as a composition of these elementary services according to the rule definition. Interactions among elementary services involved in the processing of a rule are based on the notification service. Elementary services that interact with external systems or services use plug-ins for this purpose. Besides that, plug-ins know how to communicate with their respective services. They are also responsible for maintaining the target context of the system they interact with making possible the correct exchange of data. The active functionality service exposes a web service interface in order to provide to external clients the possibility to define, activate, deactivate, and remove ECA-rules. Figure 2 shows an abstract overview of the overall active functionality service. Details about this approach can be found in [2].

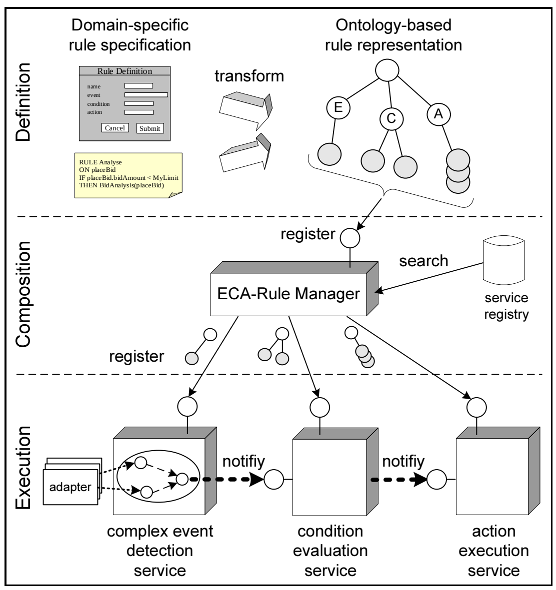

Figure 2: Overview of the active functionality service

3.1 Scenario-related Technology

Under the CoolTown model [4], people, places and things have a “web presence”, that extends the “home page” concept to include all physical entities and to include automatic system-supported correlation of the home page or *point of web presence* with the physical entity. This web presence (or portal) provides current information and services relevant to its representative. This model supports nomadic users, based on the convergence of web technology, wireless networks and portable devices.

The profile manager functionality is incorporated to the portal manager. They can run at any server in the Internet or simply at a small device like a PDA or the car box.

It is assumed that vehicles are equipped with a GPS receiver and a box. This box plays the role of a mediator between the vehicle itself and the external world. It can access a vehicle’s electronic and diagnostic interfaces, and it is responsible for announcing status changes (events) to its portal, keeping it (always) up to date. To take advantage of these changes/happenings the portal manager is extended with the ability to react to them. This is realized by integrating the active functionality service mentioned before. In this way, portals can provide the possibility to define ECA-rules that react according to happenings of interest and user preferences.

A prototype implementation of this telematic scenario was built on top of the proposed infrastructure [3].

4 Outlook

The active functionality service we developed allows the seamless integration of previously isolated automotive applications with Web services. It consumes the events generated by the car's internal sensors without the risk of interfering with them. It relays them to the manufacturer for maintenance advice or to the on-board rule mechanism for user guidance. These services can be realized immediately. The customization of the dashboard or creature comfort settings must be done strictly through the car manufacturers, since no manufacturer will risk an external system that is not under its control to modify vital components of the car. We are interacting on these issues with car manufacturers.

While car manufacturers are beginning to offer limited services, such as navigation support in response to traffic, these are closed systems. The infrastructure we developed complements the existing on-board systems with the possibility to add and combine new services, particularly the integration of the driver's or passenger's home or office environment and a wide infotainment offering.

At present one of the main issues is the source of payment of these services. Different payment models are emerging that range from one time payments included in the car's price to monthly subscriptions or pay-per-use. We are expanding the infrastructure to handle pay per use for Web services. Another issue we are currently addressing is privacy and security of the profile information. Since trust and human perception that sensitive data is under user control are key issues, we are experimenting with different forms of distribution of the information across external servers, the car's on-board box and the user's PDA. Finally, we are moving from a proprietary to an open platform based on J2EE and OSGi for the internal service architecture. The external services will continue to be accessed as Web services through the standard interfaces.

Acknowledgements

The authors would like to thank Peer Hasselmeyer, Rama Penumarthi and Christof Bornhövd for their collaboration with this project. This work has been partially supported by Hewlett-Packard Laboratories and Hewlett-Packard German Innovation Center.

References

[1] C. Bornhövd and A. Buchmann. A Prototype for Metadata-Based Integration of Internet Sources. In *Proceedings Intl. Conference on Advanced Information Systems Engineering (CAiSE)*, volume 1626 of *LNCS*, pages 439–445, Heidelberg, Germany, June 1999.

[2] M. Cilia, C. Bornhövd, and A. Buchmann. Moving Active Functionality from Centralized to Open Distributed Heterogeneous Environments. In *Proc. of CoopIS'01*, *LNCS* 2172, pages 195–210, Trento, Italy, September 2001. Springer.

[3] M. Cilia, P. Hasselmeyer, and A. Buchmann. Profiling and Internet Connectivity in Automotive Environments. In *Proceedings of the International Conference on Very Large Data Bases (VLDB'02)*, pages 1071–1074, Hong-Kong, China, August 2002.

[4] T. Kindberg et al. People, Places, Things: Web Presence for the Real World. In *Proceedings of the Intl. Workshop on Mobile Computing Systems and Application (WMCSA 2000)*, Monterey, CA, 2000.

[5] U. Hansmann, L. Merk, M. Nicklous, and T. Stober. *Pervasive Computing Handbook*. Springer, 2001.

[6] N. Paton, editor. *Active Rules in Database Systems*. Springer, 1999.