Profiling and Internet Connectivity in Automotive Environments

M. Cilia, P. Hasselmeyer 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 demo combines active DB technology in open, heterogeneous environments with the Web presence requirements of nomadic users. It illustrates these through profiling of users and Internet-enabled vehicles. A scenario is developed in which useful functionality is provided, such as instrument adjustments, maintenance and diagnostic information handling with the corresponding workflows, and convenience features, such as position-dependent language translation support and traffic information. The customization mechanism relies on an active functionality service.

1 Introduction

The demonstration presents a case study of using an active functionality service for customizing user experience. This service expands active database functionality to cover open, distributed and heterogeneous environments. The demo is realized on the basis of active database concepts, loosely-coupled distribution principles and data integration strategies.

Similar to other pervasive computing environments, cars will see a convergence of Internet, multimedia, wireless connectivity, consumer devices, and automotive electronics [6]. 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.

Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002

Consider for instance an automobile scenario, where vehicles, persons and devices have a web presence (see Figure 1). Within this scenario new possibilities emerge, for example the adjustment of instruments according to personal settings stored in the portals. This 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.

Under these circumstances, a frequent traveller can 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, "find and set the route to the next gas station", or "book an appointment to change oil" can take into account car manufacturer's, company's, and/or driver's preferences.

2 Technology Used

It is well-known that data from different sources can only be properly interpreted when sufficient context information about its intended meaning is known. In general this information is left implicit and as a consequence it is lost when data is exchanged across institutional or system boundaries. For this reason, to exchange and process events from independent sources in a semantically meaningful way explicit information about its semantics in the form of additional metadata is required.

The architecture adopted here is based on the use of shared concepts (ontologies) expressed through common vocabularies as a basis for interpretation of data and metadata. In particular, we represent events, or event content to be precise, using a self-describing data model, called MIX [1].

On the foundations of an ontology-based infrastructure, an active functionality service was developed providing the following benefits: services interact using an appropriate vocabulary at a semantic level,

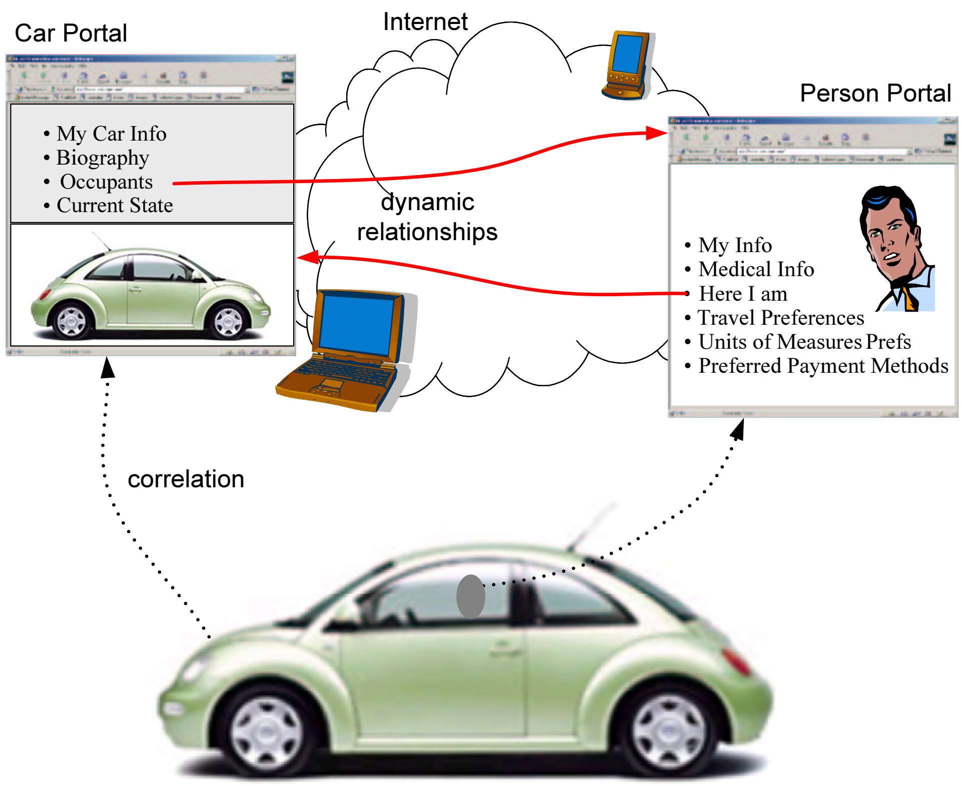

Figure 1: Car scenario

events from different sources are signaled using common terms and additional contextual information, and rule definition languages can be tailored for different domains using a conceptual representation, providing end-users the most appropriate way to define rules. This is accomplished by using a three layered rule representation: external, conceptual and internal. Here the conceptual layer provides independence between the implementation of the underlying mechanism (internal) and end-user's rule definition (external). This conceptual representation enables the use of a "generic" active functionality service for different domains, making the underlying service independent from the rule specification. For instance, this service is also used in the context of online auctions [3].

By means of an integral use of ontologies as part of the infrastructure, the definition of rules can benefit from the use of contexts. Contexts can be attached to conditions and actions in order to evaluate them under the defined contextual information. For instance, a condition predicate that verifies distances can define "metric system" as context. This way, incoming events from heterogeneous sources are automatically first converted by the infrastructure to the metric system (if necessary) before they are used for evaluation. Consequently, conditions and actions are always specified at a domain-specific level, and are independent from source-specific representations. This provides a very useful and powerful mechanism for combining events from heterogeneous sources.

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 means of a notification service that is based on a concept-based publish/subscribe mechanism.

In this work traditional processing of Event-Condition-Action rules (ECA-rules) [7, 8, 9] is decomposed into its elementary and autonomous parts. These parts 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.

Figure 2 shows an abstract overview of the overall active functionality service. Details about this approach can be found in [2].

Rule-based personalization uses specific information about individuals to react in a certain situation

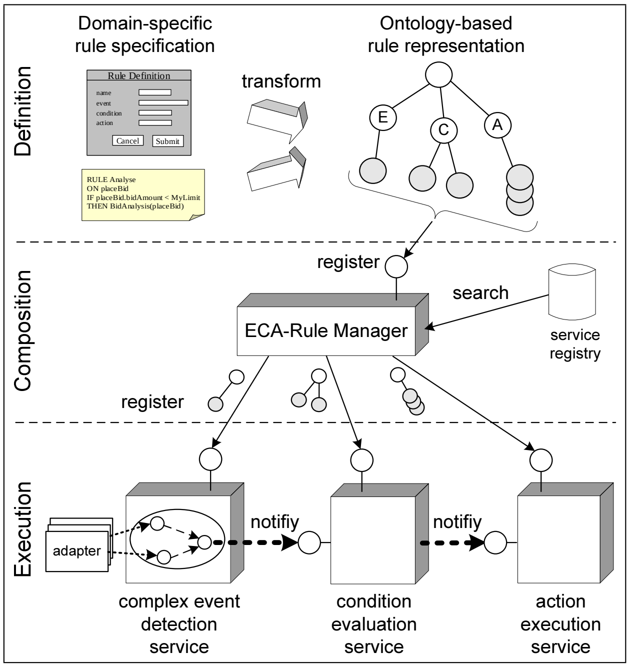

Figure 2: Overview of the active functionality service

according to their preferences. ECA-Rules are an appropriated technology for implementing personalization [4].

2.1 Scenario-related Technology

Under the CoolTown model [5], 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.

Consider for instance the situation depicted in Figure 1 where a car and a person have their own portals. Here the correlation of real-world objects and their portals are depicted with dotted arrows. Dynamic relationships among real-world objects (e.g. car occupants) can be also maintained in the portal (shown in the figure with solid arrows). Considering that information and status maintained in portals reflect the current real-world status, why not personalize a user's experience according to his or her preferences. Assuming that changes on portals are carried out immediately, reactions to some of these changes can play an important role. In this way, portals can be enhanced with the active functionality service in order to provide reactions according to happenings of interest and user preferences.

3 Overview of the Demonstration

For the purpose of this demonstration, 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 (like interfaces J1850, ODB-II) and it is responsible for announcing status changes 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.

Preferences are stored and managed by the portal manager. They can run at any server in the Internet or simply at a small device like a PDA.

This demonstration tries to show the reaction of a personalized car to different situations according to a set of user-defined rules. A description of a partial set of rules that are used for this demonstration follows.

Part failure: The car detects a problem or reports a warning of a likely failure. This is communicated to the portal manager. As a reaction, a rule is fired that starts a workflow in order 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 and the severity of the problem.

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 format of date/time, the units used for temperature, distance and velocity, radio stations or music preferences, etc.) are automatically loaded. This is achieved because the driver uses his/her identification (e.g. badge, key) when getting into the car and the the car's box communicates this happening to its portal. As a reaction, a rule is fired which in turn reads driver's instrument preferences and contacts the box to set/load them into the car's instruments.

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-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 can be read. Because drivers should concentrate on driving, all this information can be read out by using a text-to-speech service.

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.

Location-dependent personal preferences: Similar to the well-known profiling service provided by portals like Yahoo!, preferences can be adjusted as a function of location, personal calendar, etc.

For this particular scenario, there is a set of rule templates, such as those mentioned above, that can be configured through a web page.

In this demonstration, car diagnostics and other car status variables are simulated through a control panel. For instance, by means of this panel, the level of fuel and geographical position of the car can be changed or failures can be simulated in order to see the reactions of the vehicle according to different drivers' profiles.

3.1 Prototype Implementation

The active functionality service and its elementary services were developed using Java and they run on top of HP's Core Services Framework (CSF). An adapted version of the CoolTown Web Presence Manager was used to manage portals extended to collaborate with the active functionality service. Additionally, the HP Application Server and the HP Web Services platform were also used. Complex rule reactions which involve several services are carried out using a workflow engine (HP Process Manager). The notification service was built on top of TIB/Rendezvous (for historical reasons). Ontology concepts are implemented using Java.

Acknowledgements

The authors would like to thank Christof Bornhövd, Fabio Casati, Umesh Dayal, Patrick Goddi, Li-Jie Jin, Memhet Sayal, and Ming-Chien Shan 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. Springer.

[2] M. Cilia, C. Bornhövd, and A. Buchmann. Moving Active Functionality from Centralized to Open Distributed Heterogeneous Environments. In *Proceesings of the 9th IFCIS Conference on Cooperative Information Systems (CoopIS'01)*, volume 2172 of *LNCS*, pages 195–210, Trento, Italy, September 2001. Springer.

[3] M. Cilia and A. Buchmann. An Active Functionality Service for E-Business Applications. *ACM SIGMOD Record*, 31(1), March 2002. Special Section on Data Management Issues in Electronic Commerce.

[4] J. Lopes de Oliveira, C. Bauzer Medeiros, and M. Cilia. Active Customization of GIS User Interfaces. In *Proceesings of the International Conference on Data Engineering (ICDE '97)*, pages 487–496, Birmingham, U.K., April 1997. IEEE Computer Society Press.

[5] 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.

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

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

[8] The Act-Net Consortium. The Active Database Management System Manifesto: A Rulebase of ADBMS Features. *SIGMOD Record*, 25(3):40–49, September 1996. `www.acm.org/sigmod/sigmod\_record/9609/adbms.ps`.

[9] J. Widom and S. Ceri, editors. *Active Database Systems: Triggers and Rules for Advanced Database Processing*. Data Management Systems. Morgan Kaufmann Publishers, 1996.