# Sharing In-Vehicle HMI observations into the Semantic Sensor Web

Álvaro Sigüenza, David D. Pardo, Vasile Vancea,
José Luis Blanco, Luis Hernández-Gómez
ETSI Telecomunicación - UPM
Madrid 28040, SPAIN
e-mail: alvaro.siguenza@gaps.ssr.upm.es

Jesús Bernat Vercher, David Conejero
Telefónica I+D
28043 Madrid, SPAIN
e-mail: bernat@tid.es

***Abstract*—Current evolving concepts such as the "Internet of Things" points toward a future in which many connected objects acquire meaningful information about their environment and communicate it to other objects and to people. This paper discusses how Human Machine Interaction (HMI) systems embedded in connected cars can be designed to make drivers' observations about the traffic or their environment available as shareable human-generated observations across the Semantic Sensor Web. An experimental implementation is presented integrating spoken dialogue HMI capabilities and Semantic Web technologies into an on-board OSGi architecture.**

***Keywords: connected car; human-generated observations; Human-Machine Interaction; Semantic Sensor Web***

## I. INTRODUCTION

The current evolution of ubiquitous computing and information networks is rapidly merging the physical and the digital worlds enabling the ideation and development of a new generation of intelligent applications as eHealth, Logistics, Intelligent Transportation, Environmental Monitoring, Smart Grids, Smart Metering or Home Automation. This scenario, seminal in Mark Weiser's Ubiquitous Computing work [1] and now evolving into the "Internet of Things" [2] concept, points toward a future in which many objects around us will be able to acquire meaningful information about their environment and communicate it to other objects and to people.

Among this universe of interconnected objects, those embedding Human-Machine Interaction (HMI) technologies, such as mobile phones, connected vehicles, home appliances, smart buildings, interactive urban infrastructures, etc., can play an important role as they can be aware of real-world information and, at the same time, provide enriched information to other users, objects or applications. Data could come from human (social networks, monitor systems, actions, gestures), or from machine input (e.g. different Sensor Networks), and the HMI is the connection between these two sources.

Sensor and Actuator Networks (SANs) are becoming an inexhaustible source of real world information, so the Sensor Web term is being used to describe a middleware between sensors and applications: "Web accessible sensor networks and archived sensor data that can be discovered and accessed using standard protocols and application programming interfaces" [3]. An emerging number of Sensor Web portals, such as Sensorpedia[1], SensorMap[2], SensorBase[3] or Pachube[4], are currently being developed to enable users to upload and share sensor data. One of the most influential Sensor Web initiatives is the Sensor Web Enablement (SWE) initiative of the Open Geospatial Consortium (OGC). The SWE [4] is defining a set of standards to develop "an infrastructure which enables an interoperable usage of sensor resources by enabling their discovery, access, tasking, as well as eventing and alerting within the Sensor Web in a standardized way".

Further efforts to improve interoperability of a world of heterogeneous and geographically dispersed interconnected SANs include the proposal of a Semantic Sensor Web [5], [6]. The Semantic Sensor Web brings Semantic Web technologies to annotate sensor data making it easier for different applications to extract homogeneous interpretations of them.

To progress towards a full harmonization between HMI systems and the Sensor Web, advancements are needed in two fundamental areas: the integration and accessibility of a growing number of heterogeneous sensor data into HMI systems, and new mechanisms that allow sharing real-world information provided by users of connected objects into the Sensor Web or the Semantic Sensor Web.

In our previous research [7] we have presented some contributions to the first issue, so in this paper we will try to contribute to the second one. In particular we will present our results based on our activities in the MARTA[5] project (Mobility for Advanced Transport Networks), a Spanish public-funded project where several context-aware interactive services were designed and implemented for In-Vehicle Information Systems (IVIS) and Advanced Driver Assistance Systems (ADAS).

In this paper we will be focused on a scenario where a driver of a connected car provides, through the interaction with an in-vehicle HMI system, context information that can be valuable for other applications. For example, the driver detects some potential dangerous observations on the road (ice-patches, pedestrians on the road, etc.), traffic conditions (accidents or congestions) or environmental context (a dense foggy or heavy rain area) and, through interaction with an in-vehicle HMI system, makes this information available to

[1] www.sensorpedia.com
[2] atom.research.microsoft.com/sensewebv3/sensormap/
[3] sensorbase.org
[4] www.pachube.com
[5] www.cenitmarta.org

other interested applications (e.g. a Road Safety Authority or other HMI systems in surrounding connected vehicles).

Future in-car interaction scenarios must be considered not as simple "local" driver-system interfaces, but, as Fig. 1 illustrates, as complex systems. HMI systems for connected cars have to manage, not only different driver's interaction modalities (speech–microphones and loudspeakers; vision–displays; haptic–knobs, buttons, touch screen; etc.), but also local and remote sensor information. As shown in Fig. 1, context-aware HMI systems can be regarded as systems that use sensor data and user inputs to interact with applications, but at the same time HMIs may be regarded as sensing systems capable of producing real-world information to the Sensor Web. This capability of using HMI systems embedded in a connected object to publish information into the Sensor Web could be related, either to measurements from its local sensors (attached to the object), or to data directly provided by her user. In [9] we discussed some of the main issues when using HMI systems to process and publish local sensors data, in this paper we will address those related to the publication of user-generated observations.

In this work we will also rely on the design principles proposed by the W3C's Multimodal Architecture and Interfaces (MMI) [8]. Following these principles we will discuss the design of in-vehicle context-aware multimodal HMI systems able to collect driver's information reporting observations on different road, traffic or environmental situations, and generate semantic representations of them.

The rest of the paper is organized as follows: Section II presents some related research. Section III describes the design of in-vehicle HMI systems to collect driver-generated observations following the principles of the W3C's MMI architecture instantiated on an OSGi framework. The semantic annotation of driver-generated observations and their publication in the Semantic Sensor Web are discussed in Section IV. Section V presents our experimental set-up, implemented on an on-board unit of a connected car. Finally, conclusions and future work are discussed in Section VI.

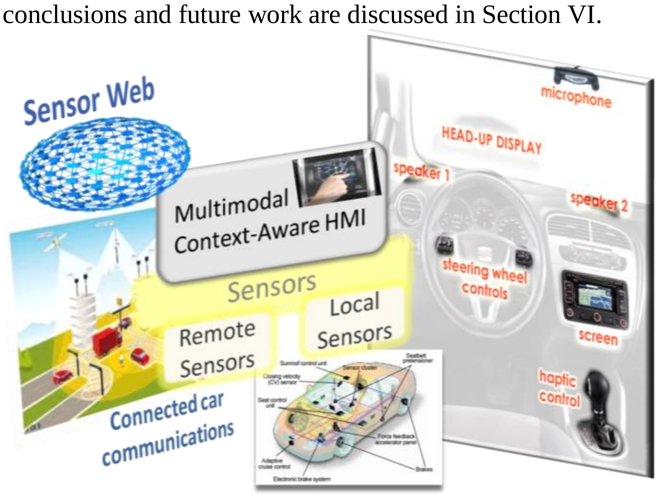


Figure 1. In-vehicle HMI system for connected cars.

## II. Related Work

In-vehicle context-aware HMI systems and the more recent conceptions of user-generated sensors or the Human Sensor Web are research areas closely related to the work in this paper.

Recent research on in-vehicle context-aware HMI systems is directed to ensure that they can deal with the heterogeneity adapting to all kinds of situations and contexts, always giving a right and safe feedback to their users ([10], [11], [12]). Integrating both multimodal interaction and context for in-vehicle applications has also been addressed, and a common approach [12] is to consider three independents domains: driver, vehicle and environment. However most of the research in context-aware multimodal HMI systems in vehicles has been more oriented on how to manage high-level representations of context than on the integration with the underlying infrastructures providing sensor data. Only few approaches, as the work presented in [13] for an in-car OSGi framework, have addressed the design of HMI including the management of different car components. Nevertheless these studies only take into account data from local sensors (attached to the car) and do not consider the access or the sharing (publication) of sensor data through the Internet.

The research presented in this paper can be also related to emergent concepts of user-generated sensors or human observations (descriptions of real-world phenomena), that are different from those of human sensor observations (particular sensors carried by or attached to humans). The seminal work in [14] presents the Human Sensor Web as "an effort for creating and sharing human observations as well as sensor observations on the Web", and presents an example of establishing a noise mapping community. According to this vision, future systems will use different types of observations: conventional sensor data, human sensed observations (e.g. vocal, image or text) and human collected data (sensors carried by humans, like smart phones or other personal devices), and will integrate them into the Human Sensor Web [15]. The work in [14] also identifies some challenges to tackle the Human Sensor Web. The most persistent challenges are the accuracy of data, the personal privacy issues and how collective intelligence can improve current conventional methods [16]. In a similar direction, in our work we will discuss some preliminary approximations on using semantic representations, already used to represent sensor data, to describe human observations, and we will explore the use of the Semantic Sensor Web principles for publishing and accessing them.

## III. HMI Systems to Collect Driver's Observations

As we stated before, developing in-vehicle HMI systems requires not only the integration of driver's input/output information (e.g. speech, touch, graphic displays, etc.), but also the proper management of data provided from different sensor sources: from the car (e.g., speed, wheel traction), the driver (e.g. mood, fatigue), and environment (road, traffic, weather, etc.) [17]. The HMI designer typically needs to interpret the sensed data in order to identify situations that

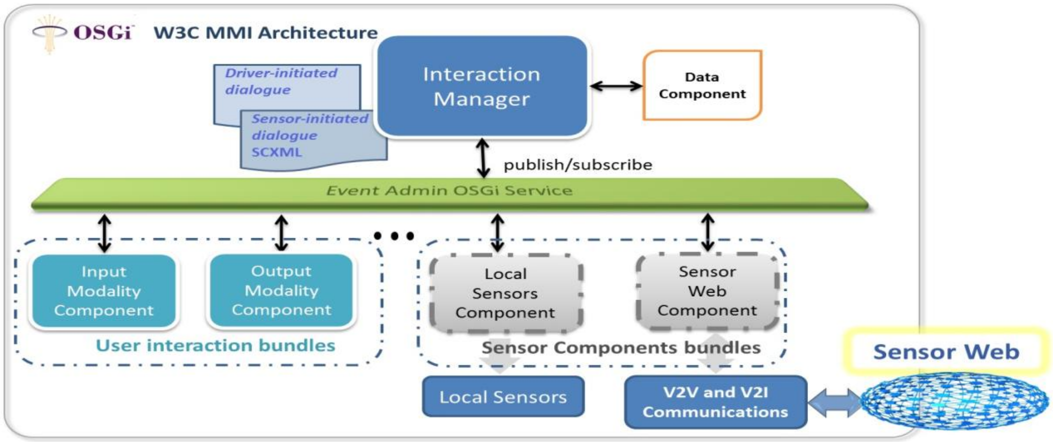


Figure 2. In-vehicle HMI system to collect driver's observations following the W3C MMI Architecture.

are either, of direct interest to the driver, or which will help shape communication strategies that are appropriate for each situation.

The W3C is in the process of defining an architecture recommendation for the design of multimodal interfaces, the MMI Reference Architecture [8]. Main components in the W3C MMI architecture, represented in Fig. 2, are the Input and Output Modality Components, which handle the information coming in from and out to the human user, and the Interaction Manager, which coordinates the flow of the communication in the different modalities and decides the overall communication strategy in response to the successive inputs from the user. The MMI architecture also considers two important elements: 1) a data component which harbours the data that the Interaction Manager needs to perform its functions; and 2) an event-based communication layer to carry events between the modality components and the Interaction Manager.

This standardized reference architecture represents a very attractive framework for dealing with the high complexity of designing in-vehicle HMI systems. In our experimental implementation, that will be detailed in Section V, an OSGi framework [18] was used to instantiate the W3C MMI architecture. OSGi is a Java-based service platform that allows applications to be developed from small, reusable and collaborative components called bundles. So far, the main components in the MMI Architecture (i.e. Interaction Manager, and Input/Output Modality Components) can be implemented as OSGi bundles. The platform also provides an EventAdmin OSGi Service bundle as a standard way of dealing with events in the OSGi Environment using the publish/subscribe model. Therefore this event management capability in OSGi can represent the event-based communication layer in the W3C MMI architecture. The mapping between these OSGi capabilities and the MMI architecture is illustrated in Fig. 2.

Also, extending this basic HMI architecture to include information from different sensor sources is rather straight. Information from both local sensors (attached to the car) and remote sensors (e.g. from the Sensor Web) can be directly accessed by developing specific bundles acting as "Sensor Components" between the sensor providers and the HMI Interaction Manager (see the Local Sensor Component and the Sensor Web Component included in Fig. 2).

Obviously, to have access to remote sensors (i.e. the Sensor Web Component) the OSGi framework must also include a connected car infrastructure, for example supporting V2V (vehicle to vehicle) and V2I (vehicle to infrastructure) communications, or just communication capabilities through in-car nomadic devices, such as the driver's mobile phone (OSGi is also a technology suitable to be integrated into mobile phones).

Inside the W3C MMI architecture, as in any HMI system, a key component is the Interaction Manager. The Interaction Manager receives ordered sequences of events and data from the different Components (both from the user and sensor sources) and decides what to do with them. Events may be for the Interaction Manager's own consumption, they may be forwarded to other Components or they may result in the generation of new events or data by the Interaction Manager. For the purpose of designing flexible and easily configurable Interaction Managers the W3C is developing the SCXML language (State Chart eXtensible Markup Language) [19], a generic event-based state-machine execution environment based on Harel statecharts. Statecharts are extensions of conventional finite state machines, with additional properties that lend themselves to describing complex control mechanisms in reactive systems in which it is necessary to coordinate components of diverse nature. SCXML is being proposed by the W3C as a major candidate language to control interaction flow in Human-Machine Interactive systems (HMIs). It is being considered for future interactive speech systems, in W3C VoiceXML 3.0 [20], as well as for multimodal systems [8]. As we have presented in a previous research [21], SCXML can be also very useful to combine both user's input information and sensor information.

In this work we have implemented an SCXML-based Interaction Manager controlling the exchanges with the driver (see the details in Section V). Driver's input information is obtained using speech recognition controlled with a push-to-talk button in the steering wheel, and output information is provided through text-to-speech synthesis and a visual display. Two different spoken dialogue interactions have been implemented for collecting driver's observations that we refer to as: driver-initiated and sensor-initiated. In a driver-initiated dialogue, the dialogue is started by the driver, using a specific button in the steering wheel, when she observes what she believes a relevant situation; for example when entering a dense foggy area. In the sensor-initiative, a dialogue is automatically initiated once a sensor detects a possible relevant situation. This later case, the sensor-initiated dialogue, represents a rather simple dialogue as the HMI system has only to request the driver to confirm (using yes/no expressions) the particular sensor-detected situation or observation. However, the first case (driver-initiated dialogue) represents a more challenging situation due to a probably high number of observations a driver can report and the spontaneous language she can use; requiring Natural Language Processing capabilities not implemented in our OSGi framework. To address this issue, in our implementation, the driver-initiated interaction has been restricted to follow a menu-based dialogue. So once the driver decides to report an observation, she has to follow a system-directed dialogue offering a limited set of possible observations. In order to avoid speech recognition errors, which can lead to unsafe driving situations [22] the number of different observations has been limited to 16, arranged into two sub-menu levels. In the first level the driver has to choose the category of her observation between road, traffic or environment, and in the second level she has to select the particular observation. Some results from a preliminary usability evaluation of these two strategies are discussed in Section VI.

Finally, it is important to point out that, apart from the difficulties in designing robust and safe spoken dialogue strategies to collect human-generated observations, an important challenge, not addressed in our work, is how to provide a confidence level on the quality of the information the driver is reporting. Some strategies already in use in social networks could be explored: as ratings of particular reliable users or matching for coincident observations.

## IV. Publishing Driver's Observations into the Semantic Sensor Web

Once a driver-generated observation has been collected through a driver-initiated or a sensor-initiated dialogue, the Interaction Manager has to start a procedure to make it available into the Sensor Web. Two major steps are required: 1) to provide a homogeneous representation to the human-generated observation; and 2) to drive a mechanism for publishing it into the Sensor Web.

As we discussed in the Introduction (Section I), there is currently an emerging number of Sensor Web portals (i.e Pachube, Sensorpedia, etc.) that could be considered to publish this human-generated observations. Even, the use of Social Network infrastructures, such as text-based posts (e.g. Twitter), could be explored.

In this work we will explore the OGC SWE (Sensor web Enablement) principles [4], as they represent one of the most mature and active proposals in the field. Nevertheless, the requirements to publish driver-generated observations using current SWE standards are far from obvious.

- Firstly, it should be necessary to describe the in-vehicle HMI system as a sensing system using SensorML [4]. SensorML is the OGC SWE language used to describe different types of sensors and sensor systems, from simple sensors to complex sensor systems, as, for example, earth observing satellites (or in our case a driver-observer).
- Then, this observation entity should be registered into an OGC Catalog Service (CS-W) [4] so its observation could be discovered by other applications.
- Driver-observations should be represented using the O&M (Observation & Measurement) language [4]. O&M defines a domain independent conceptual model for the representation of –spatiotemporal– sensed data.
- Finally, through a set of basic Web Services, as the Sensor Observation Service (SOS) [4] (SWE only standardizes their interfaces), the human-generated sensor resources could be registered, discovered and accessed.

Due to the difficulty of addressing the above points in our in-vehicle environment, in this work we have tried to approach the recent initiative of blending the Sensor Web with Semantic Web technologies into what is referred to as the Semantic Sensor Web [5], [6]. Although, as stated in the excellent position paper presented in [24], it can be hard to measure how successful are these recent initiatives, we will explore the use of URI-based descriptions of human-generated observations encoded using the Resource Description Framework (RDF), as it is an accepted Semantic Web standard [25]. This will empower building many applications such as Web mashups and, as we will discuss and illustrate in Section V, if adopting Linked Data principles (Linked Sensor Data [24]), "to use URIs as reference for look-up as well as RDF and SPARQL for storage, access, and querying".

So far, adopting what we can call Semantic Sensor Web principles, next sub-sections will discuss how to describe, store and access the HMI-collected driver-generated observation.

### A. Semantic description of driver-generated observations

In order to provide a semantic representation for the driver's observations we have followed the approach proposed by Henson et al. [23] based on the encoding of the

Observations and Measurements language in OWL (the Web Ontology Language [26]). In O&M-OWL an ontology covers a subset of concepts in O&M, and, in a similar way than proposed in [23] for general sensor observations, we think it can also offer interesting possibilities for managing human-generated observations. Annotating human-generated observations using O&M-OWL provides us with many benefits over other schemes: it offers the ability to reason over observations using semantic technologies: it gives access to the wider set of applications that makes use of the Semantic Web; it enables the straightforward use of querying mechanisms, such as SPARQL, to discover new information; and it provides the possibility of integrating new observations with the great amount of information enabled through RDF and OWL in the Semantic Web. Fig. 3 shows the translation of O&M into OWL, adapted to our driver-generated observations scenario.

In O&M-OWL, relations between concepts are described using RDF triples, which correspond to a subject-predicate-object structure. Thus, for example, the O&M-OWL representation of a driver-generated observation detecting a dense fog in a road would be as follows:

```
om:obs_1     rdf:type    om:Observation .
om:obs_1    om:featureOfInterest   om:road_1 .
om:road_1     rdf:type     e:Road .
e: Road          rdfs:subClassOf   om:Feature .
om:obs_1    om:observedProperty   e:denseFog .
e:denseFog   rdf:type   om:Property .
om:obs_1    om:samplingTime    om:time_1 .
om:time_1    rdf:type    owl-time:Instant
om:time_1    owl-time:date-time  "20110610T08:55:00" .
om:obs_1   om:observationLocation   om:location_1 .
om:location_1  dbpedia-owl:location dbpedia:Pirineos .
om:obs_1  om:procedure  om:human_1 .
om:human_1    rdf:type   e:Driver .
```

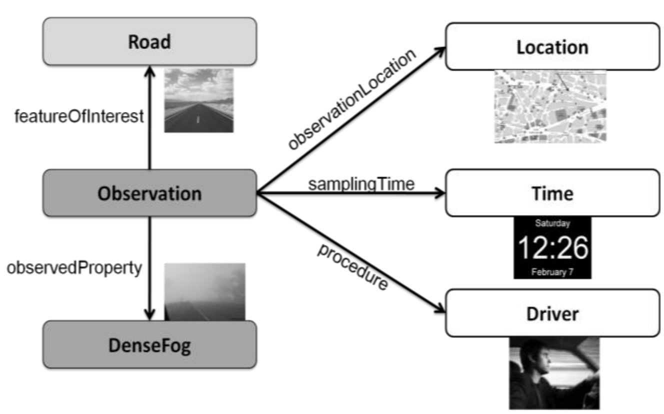


Figure 3. O&M-OWL model (adapted from [23]) applied to driver-generated observations

From this example, it is important to point out that, as it was discussed in Section III, the in-vehicle HMI system only collects (using a driver-initiated or sensor-initiated dialogue) the driver's description of the observed phenomenon (i.e. the om:featureOfInterest). Consequently the HMI architecture should automatically provide all the remaining data to be included in the O&M-OWL representation, such as the particular situation of the car in the road (om:observationLocation) or the observation time (om:samplingTime).

It is also important to note, that, as shown in the example, the observation entity (om:procedure) can be linked to a particular driver or to an anonymous (or nickname) driver. This can be very useful when addressing the relevant issue of managing privacy in human-generated observations.

### B. *Publishing into the Semantic Sensor Web*

Together with the use of O&M-OWL to generate driver-generated observations encoded as a set of RDF triples, it is important to address how these semantically annotated observations can be accessed for inference or query.

In our work, differently from the use of semantically enabled OGC services proposed in [23] (in particular the extension of SOS into SemSOS), we explored a preliminary step towards making human-generated observations accessible using the existing information space of the Web. So we stored RDF driver-observations into public repositories (i.e. SPARQL Endpoints [27]). In that way O&M-OWL observations encoded in RDF and linked to specific ontologies can be shared with other systems and other applications through the Semantic Sensor Web (SSW). The information published in the SSW can then be used for a wide variety of purposes: it can be further mashed up with other information to acquire yet higher levels of knowledge, it can be pooled to analyses patterns of use of applications (for example for a Road Safety Authority), or it can be fed back to applications (i.e. other connected car HMI systems) thus closing an information loop.

To illustrate with an example, applications could have access to the human-generated observations stored as RDF Graphs, which can be retrieved via SPARQL queries. Through these queries it will be possible to filter the RDF triples in the repository that fulfill a set of desired conditions. The following example shows a SPARQL query searching for driver-generated observations from roads in a specific mountain area "Pirineos" and with "denseFog" as the observed property.

```
PREFIX e:<http://www.sensor.gaps.upm.es/e/>
PREFIX dbpedia:<http://dbpedia.org/resource/>
PREFIX om:<http://www.opengis.net/om/1.0>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbpedia-owl:<http://dbpedia.org/ontology/>
PREFIX owl-time:<http://www.w3.org/2002/07/owl-time#>
```

```
SELECT DISTINCT ?obs WHERE {
  ?c rdf:type e:Road .
  ?obs om:featureOfInterest ?c ;
      om:observedProperty e:denseFog ;
      om:observationLocation ?loc .
 ?loc dbpedia-owl:location dbpedia:Pirineos.
}
```

In this example the query works by matching the triples RDF in the "WHERE" clause against the triples in the RDF graph stored in the repository. Our RDF example in Subsection A matches this clause, so the query result will include the values corresponding to that particular observation (along with all others published in the repository that fulfill the query requirements); in this case the value <om:obs_1>, that represents an observation of a specific road segment (<om:road_1>), would be assigned to the variable ?obs.

Additional knowledge from semantically annotated driver-observations could be obtained by using rule-based reasoning to deduce new ontological assertions from known instances and class descriptions. For example, the driver-generated observation of a road under dense foggy conditions in the previous subsection could be used by a Road Safety Authority monitoring application to warn other drivers entering into that area. So, a driver planning a trip over this area using a navigator connected to the Semantic Sensor Web (in her car or mobile phone) could be alerted on the dense fog situation and be suggested to follow and alternative route.

## V. EXPERIMENTAL SETUP

We have built an experimental setup to perform an exploratory analysis of the different approaches and technologies we have considered for sharing driver-generated observations, collected through in-vehicle HMI systems, into the Semantic Sensor Web. Our testing scenario corresponds to a realistic connected car environment, developed into the MARTA [6] (Mobility for Advanced Transport Networks) research project, where several HMI systems were developed for different In-Vehicle Information Systems (IVIS) and Advanced Driver Assistance Systems (ADAS) applications.

The instantiation of the W3C MMI architecture described in Section III, including mechanisms to annotate and publish driver-observations in Section IV, was implemented on an On-Board Unit (OBU) in charge of managing the Human-Machine Interaction. This OBU was integrated with the new technologies (GRPRS, UMTS, HDSPA, CALM -Continuous Air interface for Long and Medium distance-, etc.) developed in MARTA to gave support to the V2V (vehicle to vehicle) and V2I (vehicle to infrastructure) communications.

The final implementation was integrated in a CarPC, which is a computer designed to be specifically installed and run in vehicles. The CarPC was set up with a Linux OS, a Java Virtual machine and the 3.4 release of the OSGi platform [18].

Fig. 4 represents the main components we implemented. As already described in Section III, the Interaction Manager of our MMI architecture was implemented using SCXML, so a specific bundle was developed including the SCXML engine provided by Apache Commons SCXML[7]. Both the driver-initiated and the sensor-initiated dialogues were implemented using SCXML documents invoking proprietary Telefónica R&D speech technologies (ASR and TTS) accessed through a Speech Server bundle. Dialogue management also included interaction with events coming from buttons in the steering wheel (for example to receive an order to start a driver-initiated dialogue).

Also as discussed in Section III, the event-based communication layer (an important element in the W3C MMI architecture) was supported by the EventAdmin OSGi Service bundle; a standard way of dealing with events using the publish/subscribe model. It is through this service that the SCXML-based interaction manager interacts with the Speech Server bundle (ASR/TTS) as well as with several bundles receiving sensor data (i.e. Sensor Components). Data from Local Sensors Components (car-sensors) were received through specific wrapping components accessing the CAN bus, while a specific bundle, including Internet access through GPRS, was developed to access the Semantic Sensor Web (i.e. to query RDF repositories).

Another important experimental development was how to integrate several technologies to reach our final goal of making the driver-generated observations, collected by the SCXML dialogues, available into the Semantic Sensor Web. To this end we followed two major development steps:

- First, a specific bundle (the Semantic Annotation bundle in Fig. 4) was implemented. This bundle receives events from the Interaction Manager and generates RDF annotations using the O&M-OWL model. As it was discussed in Section IV, to complete the data in all the generated RDF triples, this bundle was connected to other in-car information systems, in our case to the navigation system, to obtain the road name (om:Observation) and current time (om:samplingTime) and position (as the specific km in the specific road, om:observationLocation).
- Second, each time the Semantic Annotation bundle generates a RDF annotated driver-observation, a Semantic Sensor Web publication bundle (SSWP, see Fig. 4) is used to publish it into a RDF repository. To this purpose we have made use of the features provided by Sesame [28], an open source Java Framework for the storage and querying of RDF data. Specifically, we have used the Sesame workbench to create an offline repository. When the SSWP bundle is invoked the Sesame Java is used to add new RDF triples to this repository.

[6] www.cenitmarta.org

[7] http://commons.apache.org/scxml/

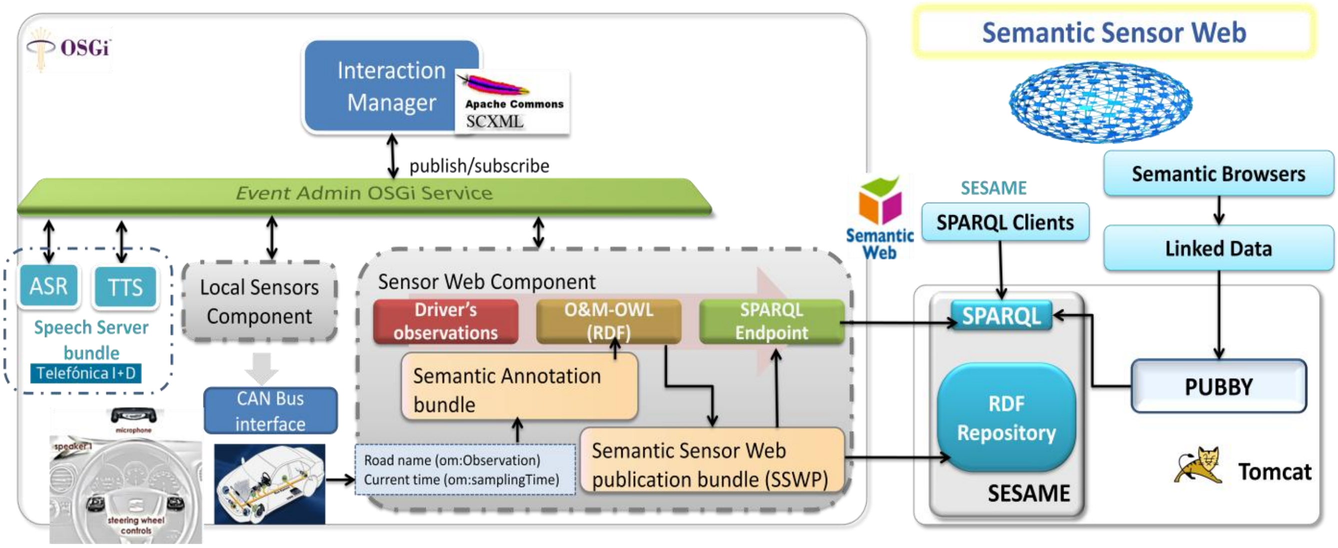


Figure 4. Experimental setup for publishing driver-generated observations into the Semantic Sensor Web

However, it is important to notice that driver-observations stored as RDF triples in repositories can be only accessed by sending SPARQL queries to a SPARQL endpoint. In RDF, the resources are identified by means of URIs. These URIs used in the SPARQL repositories are not deferenceable, meaning that they cannot be accessed from a Semantic Web Browser and therefore by a growing variety of Linked Data applications and clients. For example, in our particular car-related scenario, the resources in the namespace "e" (used in the example in Section IV) can be found following the URI http://www.sensor.gaps.upm.es/e/. However, the SPARQL endpoint is accessible through the local address http://www.sensor.gaps.upm.es/openrdf-sesame/repositories/e. Therefore, the RDF in this repository only will be accessible locally by the SPARQL clients, making it necessary to perform a mapping that allows access through semantic browsers and linked data clients.

To tackle this difficulty, Pubby [29], a Linked Data Front End for SPARQL Endpoints was integrated with our initial Sesame repository. Pubby also provides a server (only requiring a servlet container such as Apache Tomcat) that is in charge of mapping the URIs retrieved by SPARQL endpoints to deferenceable URIs. Pubby handles requests from semantic browsers by connecting to the SPARQL endpoint, requesting from it information regarding the original URI, and returning the results to the client through an access point. Thus, with the Pubby server configured to run at http://www.sensor.gaps.upm.es/e/ , when the semantic browser or linked data client decides to access a particular URI, such as http://www.sensor.gaps.upm.es/e/Road, it accesses the Pubby server, which then collects the information regarding the resource in question from the SPARQL endpoint (http://www.sensor.gaps.upm.es/openrdf-sesame/repositories/e). The resource information is then returned to the client in machine-readable format.

This, in sum, is how we are able to make the new driver-generated observations collected through in-vehicle HMI systems shareable in the Semantic Sensor Web.

## VI. Conclusion and Further Research

The central theme presented in this paper has been that interconnected objects, embedding Human-Machine Interaction (HMI) technologies, can play an important role to obtain relevant human-generated real-world information that can be shared with other users, connected objects or applications.

It has been discussed how the design of in-vehicle HMI systems can make driver-generated observations shareable into the Semantic Sensor Web. Our approach has been based on collecting driver's observations using an HMI system following the W3C MMI architecture, providing them with semantic annotation using O&M-OWL, and making the RDF-generated data available into SPARQL Endpoints and Linked Data Front-Ends. An experimental setup, integrating different HMI and Semantic Web technologies, has also been presented, implemented over an OSGi platform for a connected car On-Board Unit.

The experimental framework, we have presented for the particular scenario of a connected car, has shown the possibilities of integrating HMI systems into emerging Sensor Web initiatives. However, it has also uncovered important future challenges needed to be tackle, some of them related to the HMI systems development while others to the future evolution of the Sensor Web.

Preliminary usability tests we are conducting on driver-initiated and sensor-initiated dialogues, are showing that both inaccuracies in sensors and limitations on the number of allowed reported observations can generate important frustration to drivers, leading to unsafe interaction patterns. So far, although this can be considered an HMI specific issue, we believe that this field of study should have its own place of research under the HMI community.

The continuous emergence of new terms such as Sensor Web, Real-World Internet, Semantic Sensor Web, Semantic Sensor Internet or Human Sensor Web, reflects a situation where more fundamental research effort is required to advance on how to effectively articulated and provide access to human-generated observations, human sensors, sensor observations and the Internet. We believe that the use of Semantic Web principles and technologies can assist in this task, but as the amount of human-generated and sensor-generated data might become critical their scalability and efficiency for intensive distributed computing will become critical factors. Our future research will also address mechanisms for the proper management of privacy and quality of information; two key components for successfully sharing human-generated observations.

## ACKNOWLEDGMENT

The activities in this paper were funded by the Spanish Ministry of Science and Technology TEC2009-14719-C02-02 project, and MARTA project, CDTI, 3rd CENIT Program, INGENIO 2010 of the Spanish Government.

## REFERENCES


[1] Weiser, M., “The computer for the 21st century,” Scientific American, vol. 265, no. 3, 1991, pp. 66-75.

[2] Sundmaeker, H., Guillemin, P., Friess, P., Woelfflé, S., Vision and Challenges for Realising the Internet of Things, Cluster of European Research Projects on the Internet of Things (CERP- IoT), March 2010.

[3] Arne, B., Echterhoff, J., Jirka, J., Simonis, I., Everding, T., Stasch, C., Steve Liang, S., Lemmens, R., "New Generation Sensor Web Enablement,” Sensors, Volume 11, Number 3, 2011, pp. 2652-2699.

[4] OGC® Open Geospatial Consortium, Inc. : Sensor Web Enablement (SWE), www.opengeospatial.org/projects/groups/sensorweb (last accessed on 06-11-2011).

[5] Knoesis Center, Semantic Sensor Web Project, http://wiki.knoesis.org/index.php/SSW (last accessed on 06-11-2011)

[6] Amit Sheth, Cory Henson, and Satya Sahoo, "Semantic Sensor Web," IEEE Internet Computing, July/August 2008, p. 78-83.

[7] Sigüenza, A., Blanco, J.L., Bernat, J. and Hernández, L., “Using SCXML for Semantic Sensor Networks,” *Proceedings of the Intl. Workshop on Semantic Sensor Networks, at the 9th International Semantic Web Conference (ISWC)*, 2010, Shanghai, China.

[8] W3C, “Multimodal Architecture and Interfaces,” W3C Working Draft 25 January 2011 http://www.w3.org/TR/mmi-arch/ (last accessed on 06-11-2011).

[9] Sigüenza, A., Pardo, D., Blanco, J.L., Bernat, J., Garijo, M. and Hernández, L., “Bridging the Semantic Sensor Web and Multimodal Human-Machine Interaction using SCXML,” International Journal of Sensors, Wireless Communications and Control, Special Issue on Semantic Sensor Networks, in press.

[10] McGlaun G., Althoff F., Lang M., and Rigoll G., “Towards Multimodal Error Management: Experimental Evaluation of User Strategies in Event of Faulty Application Behavior in Automotive Environments,” 7th World Multi-Conference on Systems, Cybernetics, and Informatics, Orlando, FL, USA, 2003.

[11] Pieraccini R., Krishna Dayanidhi, K., Bloom, J., Dahan, J., Phillips, M., “A Multimodal Conversational Interface for a Concept Vehicle,” Comunications of the ACM archive. Special Issue: Multimodal interfaces that flex, adapt, and persist table of contents, vol. 47, Issue 1, 2004, pp- 47-49.

[12] Amditis, A., Kussmann, H., Polynchronopoulos, A., Engström, J., and Andreone, L., “System Architecture for integrated adaptive HMI solutions. In: Intelligent Vehicles Symposium,” Tokyo, Japan, 2006.

[13] Nelson E., “Defining Interfaces as Services in Embedded Vehicle Software, Research and Advanced Engineering,” Ford Motor Company, Automotive Software Workshop San Diego, Jabuary 2004.

[14] Foerster, T., Jirka, S., Stasch, C., Pross, B., Everding, T., Bröring, A., Juerrens, E.H., “Integrating Human Observations and Sensor Observations—the Example of a Noise Mapping Community,” in Proc. of Towards Digital Earth Workshop at FI Symposium, Berlin, Germany; CEUR-WS: Aachen, Germany, Volume 640, 2010.

[15] Jirka, S., Bröring, A., Foerster, T., “Handling the Semantics of Sensor Observables within SWE Discovery Solutions,” in Proceedings of Intl. Symposium on CollaborativeTechnologies and Systems, Workshop on Sensor Web Enablement (SWE 2010), Chicago, IL, USA, May 2010; IEEE: New York, NY, USA, 2010; pp. 322-329.

[16] Goodchild, M.F. (2007a): "Citizens as sensors: The world of volunteered geography", GeoJournal, 69, pp. 211-221.

[17] Hoch, S., Schweigert, M., Althoff, F. and Rigoll G., “The BMW SURF Project: A Contribution to the Research on Cognitive Vehicles,” in Proceedings of the IEEE Intelligent Vehicles Symposium. Istanbul, Turkey, 2007.

[18] OSGi Alliance, OSGi Service Platform Release 3, The Netherlands, http://www.osgi.org/Download/File?url=/download/r3/r3.book.pdf March 2003 (last accessed on 06-11-2011).

[19] W3C: State Chart XML (SCXML): State Machine Notation for Control Abstraction, http://www.w3.org/TR/scxml/, W3C Working Draft 26 April (2011)

[20] W3C: Voice Extensible Markup Language (VoiceXML) 3.0, W3C Working Draft 17 June 2010 http://www.w3.org/TR/2010/WD-voicexml30-20100617/ (last accessed on 06-11-2011).

[21] Sigüenza, A., Blanco, J.L., Bernat, J. and Hernández, L., “Using SCXML to Integrate Semantic Sensor Information into Context-Aware User Interfaces,” *In Proceedings of the International Workshop on Semantic Sensor Web*, in conjunction with the 2nd International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, pp 47-59. Valencia , 2010.

[22] Vollrath, M. 2007. “Speech and driving-solution or problem?,” Intelligent Transport Systems, vol. 1, pp. 89–94

[23] Henson, C. A., Pschorr, J. K., Sheth, A. P., and Thuirunarayan, K. 2009. SemSOS: Semantic Sensor Observation Service. In Proceedings of the 2009 International Symposium on Collaborative Technologies and Systems (CTS 2009), Baltimore, MD, May 18-22.

[24] Carsten Keßler and Krzysztof Janowicz, “Linking Sensor Data – Why, to What, and How?” In Kerry Taylor, Arun Ayyagari, David De Roure (Eds.): Proceedings of the 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010), 7–11 November 2010, Shanghai, China. CEUR-WS vol. 668.

[25] W3C: “Resource Description Framework (RDF),” http://www.w3.org/RDF/ (last accessed on 06-11-2011).

[26] W3C: “OWL Web Ontology Language,” http://www.w3.org/2004/OWL/ (last accessed on 06-11-2011).

[27] Prud’hommeaux, E. and Seaborne, A. SPARQL Query Language for RDF. W3C Working Draft. http://www.w3.org/TR/rdf-sparql-query (2004). (last accesed on 06-11-2011).

[28] Sesame: RDF Schema Querying and Storage, http://www.openrdf.org (last accessed on 06-11-2011).

[29] Cyganiak, R. and Bizer, C. Pubby – A Linked Data Frontend for SPARQL Endpoints, 2007. Available at: http://www4.wiwiss.fu-berlin.de/pubby (last accessed on 06-11-2011).

[30] Henricksen, K., and Robinson, R. 2006. A survey of middleware for sensor networks: State-of-the-art and future directions. In Proc. the Int. Workshop on Middleware for Sensor Networks. Melbourne, Australia, pp.60-65.