Journal of Theoretical and Applied Information Technology
31st December 2011. Vol. 34 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ONTOLOGY-BASED CONTEXT MODELING FOR VEHICLE CONTEXT-AWARE SERVICES

[1]MOHCINE MADKOUR, [2]ABDELILAH MAACH

[1]PhD candidate, Department of computer science, Mohamed 5 University, Rabat, Morocco

[2]Assoc. Prof., Department of computer science, Mohamed 5 University, Rabat, Morocco

E-mail: [1]mouhcine.madkour@gmail.com, [2]maach@emi.ac.ma

ABSTRACT

Computing becomes increasingly mobile and pervasive today; these changes imply that applications and services must be aware of and adapt to their changing contexts in highly dynamic environments. Now a days, vehicles have become an increasingly important and exciting test bed for ubiquitous computing (UbiComp), however, context-aware vehicle service to enable high adaptation of service to driver, vehicle or even road in ubiquitous environment is still little addressed in the literature. The context management in pervasive computing environments must reflect the specific characteristics of these environments, e.g. mobility, resource-constrained devices, or heterogeneity of context sources. Although a number of context models have been presented in the literature, none of them supports all of these requirements to a sufficient extent at the same time. This paper focuses on building an ontology modeling approach for Context Management in Intelligent Pervasive Middleware for Context-Aware Vehicle Services. To support context-awareness, we embed capabilities of context modeling and context reasoning in an ontology-based context system, which focuses on management of the context and generates a consistent model which promises the common information representation and facilitates a development of context-aware services.

Keywords: *Intelligent Vehicle, Intelligent Transportation Systems, Ontology modeling, Ubiquitous Computing, Context-aware service*

1. INTRODUCTION

Pervasive or ubiquitous computing technology is particularly flourishing in the automotive domain, excelling the ”smart car”, embodying intelligent control mechanics, intelligent driver assistance, safety and comfort systems, navigation, tolling, fleet management and car-to-car interaction systems, as one of the outstanding success stories of pervasive computing. Ubiquitous computing aims at building the next-generation computing environments with information and communication technology anywhere, for anyone, at anytime[1]; for achieving this goal, it needs a strong tool to manage the giant amount of different inputs data sensed directly or indirectly from the environment.

On the other hand the first task of designing a pervasive computing application consists of understanding context, establishing its components and modeling it in a precise and concise manner. This will enhance a better usability, more flexible share between devices and ease the adaptation task.

In that context, many methods were proposed for context modeling having particularities related to the techniques used, and the most interesting ones are those based on ontology, though ontology-based models still suffer from the problem of being specific and not generic which keep their use limited to some particular applications, however the use of ontology to model contextual information has many advantages compared to other methods(see later), the ontology is considered as a promising tool for the future of context-aware systems in particular and pervasive computing systems in general.

Although, specifically intensive are current developments for the Semantic Web [2] that can be seen as an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The Semantic Web describes the relationships between things (like A is a part of B and Y is a member of Z) and the properties of things (like size, weight, age, and speech). Our solution uses the semantic web to build dynamic

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

context models, as a user moves from one environment to another we can achieve dynamic building of contexts by sharing knowledge and context information between local pervasive environments through the semantic web.

The complexity of developing context-aware services makes middleware an essential requirement. The current state-of-the-art of context-aware middleware explores quite different approaches to support pervasive and mobile computing based on context information. All of the middleware systems need for collecting context and adapting to its change, indeed in order to enable context-aware adaptation, context information must be gathered, processed, modeled and eventually presented to the application performing the adaptation. Nowadays, context-aware systems and middleware must have a promising way to automatically model and represent a user's context to help end-users obtain their desired services.

Due to the diversity of user's environment, the available contexts may change over time. It is challenging to anticipate a complete set of contexts while we design a context aware system. In this paper, we propose a context modeling approach which can dynamically handle various context types and values. More specifically, our middleware platform [26] is context-aware embedded system designed and implemented to deploy context management system for providing services to vehicles depending to their contextual information. We use ontologies to enhance the meaning of a user's context values and automatically identify the relations among different context values. Based on the relations among context values, we capture in adaptable way the potential services which the user (i.e. driver) might need.

The remainder of this paper is organized as follow: chapter II explains the related work in the context modeling and ontology. Chapter III gives an overview of our ontology-based model. Chapter IV presents the architecture of the platform supporting the ontology model and Chapter V provides a short conclusion and future work.

2. RELATED WORKS

2.1 Ontology of context

In order to discuss the strengths and weaknesses of context models it is important to understand what context is and what role context and context modeling play in current systems. A widely accepted definition of context is [4]: Context is any information that can be used to characterize the status of an entity, where an entity may be a person, a place, or an object that is considered relevant to the interaction between a user and an application, including the user and application themselves. Commonly used contexts consist of location, identity, time, temperature, activity....Briefly we consider that the involved objects in environment are all contexts.

A system is then seen to be context-aware if it uses context to provide relevant information and services to the user, where relevancy depends on the users' task. A widely accepted definition of context- awareness [4]: "Context awareness is the ability to sense and use different contexts, any application that takes advantage of context is a context-aware application". Context-aware computing [5] is the ability of computing devices to detect, interpret, and respond to changes of environment and system.

The semantic information of contexts is essential for dealing with the complex tasks in ubiquitous computing environments. However, representing sharing and reasoning of the contexts is very difficult. In smart spaces, context is hard to represent and use due to its complexity, it requires an approach with strong expression and easy sharing capabilities. In this paper, we attempt to use the ontology technique and the web ontology language (OWL) from the field of semantic web to solve the problem.

2.2 Similar works

Context modeling is a fundamental step for every adaptation application and the development of context-aware systems. It consists of the analyze and design of contextual information contained in the system in an abstract form on the level of data structure as well as the semantic level. Several approaches were proposed for the representation of context. A survey made by Strang et al. [27] contains an interesting comparative study of different modeling methods. They conclude that ontology makes the best description of context compared to the surveyed methods because it provides a good sharing of information with common semantics. A detailed discussion of these methods is out of the scope of this paper but the authors distinguished basically the following models for context representation: a) Key-value models, b) Mark-up scheme models, c) Graphical models, d) Object oriented models, e) Logic based models, and f) Ontology based models.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Key-value model is easy to manipulate but it is not convenient for complicated structure and does not permit a good reasoning on context. It is also specific to localization applications. Mark-up scheme models are most of the time specific to a particular field and are limited to some aspects of context (localization, environment...). Graphical models are simple but less formal than the other methods. Object oriented models require low level execution agreements between applications to ensure interoperability. They are not adapted for knowledge sharing in open and dynamic environments. Logic based models are often based on a centralized context management, a solution which is not adapted to the principle of context distribution in a Pervasive Computing Services. The relations between continuous data cannot be easily described. The ontology seems to be the most suitable tool for context modeling.

2.3 middleware context modeling

Our aim is to propose an approach to context modeling which takes into account the others aspects in context aware service (adaptation, discovering, prediction…). This will be done by using a service oriented definition of context which is enough abstract and helps a lot to limit the set of contextual information and a learning machine approach to predict all possible and meaningful context configurations.

This paper addresses the middleware platform in vehicle for service providing. We developed a semantic and adaptive middleware platform, where entities have intelligent characters and can adjust themselves adaptively according to their changes to provide high-quality services. The middleware [27] is based on semantic information; the features of our middleware Ontology-based model context consist of the following:

1) Autonomy: the platform can dynamically monitor the core application; thus, it can interact autonomously, without any user intervention, with the gathered information and service provider.

2) Adaptability: Both the inner running status information and the outer environment context changes can be discerned during the run time. In terms of these changes, the platform can adaptively modify its inner structure composition and functional behaviors.

3) Scalability: The platform is a component-based middleware and its components can be dynamically added, replaced, and removed, since a different component composition provides a different service, the platform enables its structures and functions to be scalable.

4) Semantic integration: The semantic information is employed to smoothly manage both presentations and interactions of all entities for good understanding and collaboration with each other.

In the previous systems, user location contexts are widely used for guiding the decision making process of context-aware applications [29][30] [31]. However, none of them have taken advantage of the semantics of spatial relations in reasoning about contexts (i.e., information that describes the whole physical space that surrounds a particular location and its relationship to other locations).

Our work is closely related to other pervasive and context-ware computing research such as Intelligent Room [30], Context Toolkit [29]and Cooltown [32], One World [33] and Centaurus [31]. In comparison to the previous systems, our novel design of the context attempts to address challenging issues such as developing explicit ontology representations of contexts, supporting context reasoning and maintenance through logic inferences and providing a concrete and consistent architecture (also see discussions in Section 4).

3. AN ONTOLOGY-BASED MODEL

In this section, we will describe our design considerations and modeling concepts of our context-aware vehicle scenario.

3.1 Context-Aware Vehicle Scenario

A context-aware vehicle is a smart vehicle environment which is equipped with the context-aware pervasive middleware and various sensors/actuators relied to it, such as engine sensors, microphones, RFID, location- based sensors, etc [7]. In the follow, to facilitate the presentation of the paper, let us consider a context scenario as an illustrative example throughout this paper..

Mohcine MADKOUR carrying a cell phone has started his vehicle; the fingerprint system senses his presence and his location information get updated. When Mohcine drives personal communication agent interprets his current status by using the contexts acquired from various sensors and decides to forward all phone calls to his voice car phone.

Mohcine stops to Nadia Abdou who comes back from shopping with her baby girl. She settles down her baby in the baby’s place, and then gets on the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

board of the driver. An audio/visual communication channel can be established between Nadia's place and her baby's one. When she moves around the city, the communication channel is able to automatically switch to and remain alive between Nadia and her friend in job. Thus, Nadia is able to have a face-to-face talk with her friend using the embedded video conferencing panel.

Mohcine wants to have a fish lunch outdoor. He consults his meal arrangement agent, quickly the agent contact the restoration service for available food items based on him preferences and queries on an external fleet management and logistic service, after a while the car pulled up "near" the restaurant…, after Mohcine place orders using the WLAN without needing to queue up at the service-through lane. After lunch, when Nadia get on her place in the car, a air-condition agent automatically queries on an external weather service for the weather condition, later the temperature begins to degrades.

Mohcine wants to host his brother at his arrival to the airport; as he enters the premises of the public hot-spots, he can discover the resources of the convention center, obtain directions to the designated parking lot, and make associated payments using the WLAN infrastructure. He can also navigate to the consulting page of the arrivals trips and perform self check-in even before stepping out of the car! Another attractive application would allow Mohcine to navigate to the registration page of the convention trip and perform self check-in and even make a reservation.

3.2 Design Considerations

A context-aware system requires context information to be exchanged and used between different entities such as driver, vehicle and services in a same semantic understanding. In other word, an appropriate context model should support semantic interoperability which enables the common schemas to be shared between different entities. For example, in the above scenario, the representation of Mohcine's location should be understood between his personal communication agent and his GPS sensor.

Context information exhibits a number of characteristics in intelligent environments. First, context information has a great variety; it includes any information that describes physical objects, applications and users in any domain. Second, context information varies in different sub-domains for example, we are more concerned about device context such as engine and electronic units in a diagnostic environment whereas GPS and WLAN connection in related service environment. Third, context information is interrelated, for example in our scenario, Mohcine's intent (Pay a parking) are closely related to where he is located (located at the parking lot), where the WLAN connection is available (connected to parking' WLAN), and what's the parking current status (pushON). Fourth, context information is inconsistent for example, in our scenario, Mohcine's location context may quickly become out-of-date when he is changing his intent or/and his place.

3.3 Context Ontology

The vision of the semantic web [2] is to enable automatic interoperation between entities on the Web. Ontologies [13] are formal and explicit specifications of certain domains and are shared between large groups of stakeholders.

The ontology form the backbone of the semantic web and it is the key to enable automated interoperation and collaboration. Ontology typically consists of a number of classes, a number of relations (sometimes called properties) between these classes and a number of instances and axioms. These elements are all expressed using some logical language.

In ontology knowledge about a domain is modeled using a knowledge representation language with a reasoning mechanism [14]. The knowledge representation language is used to create a set of terms and assumptions (axioms) about the meanings of the terms as well as to specify classes, properties and relationships between classes and objects in the domain. Reasoning engines available to infer the meanings of the properties and relationships between specified classes and individuals. Reasoning algorithms check the models for consistency and build taxonomic structures.

3.4 Modeling Context

We have defined ontology-model to model the context structure. We have chosen ontology based models because they constitute a standard representation, reasoning and inferring scheme of knowledge. OWL language is chosen for representing information about objects and relationships relevant for the Context-aware service task. Figure 1 illustrates the major entities of the generic ontology definition model. Essentially, ontology contains entities, such as classes, individuals, properties and relations.

Class is an abstract description of a group of

Journal of Theoretical and Applied Information Technology
31st December 2011. Vol. 34 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

concepts with similar characteristics. A class has a name and a set of properties that describe the characteristics of the class, Class is also called "concept", "type", "category" and "kind" in some ontology specification languages. The class can be instantiated by crating Individuals. Property describes an attribute of a class or an individual. A property can also be composed by other properties. "Driver" is located in "Location", we define a property "LocatedIn" for Driver and assign it to the class "Location". Property is also referred to as "aspect", "attribute", "feature" or "characteristic".

Relation: defines ways in which classes or individuals can be associated with each other. In our ontology definition model, the types of relations are predefined. To express specific relations (e.g., MotherOf), we need to use properties. Four types of relations are defined to connect classes and individuals: 1) Subclass: extends an abstract class to convey more concrete knowledge; 2) PartOf: means a class or an individual is a part of another class or individual. 3) Complement: expresses that the members of a class do not belong to another class; and the two classes together contain all the members in a given domain; and 4) Equivalence: means that two classes, individuals or properties are exactly the same.

Our ontology is divided into a generic ontology (the generic level context model) and domain specific ontologies (the specific context model) as proposed [28]. The generic ontology defines the common context concepts necessary to make an application context-aware. The specific concepts inherit from the generic ones adding properties dedicated to the considered situation domain. Existing ontologies may be extended and refined to answer the needs of almost every domain. Such ontologies can be imported in our environment using the OWL-DL [29].

In the generic ontology, each concept represents a context entity. The six principal concepts of the generic ontology are: User Driver, Location, Vehicle, Activity, Network and Physical Environment.

The end-user (i.e. driver) is described by the concept User that specifies her/his basic profile (name, login, etc) and her/his preferences (language, favourite data formats, etc). The concept Location (GPS coordinates) describes the physical space and spatial relations of involved vehicles in the system. The concept Vehicle describes the overall vehicle's components hardware that can enter in the pervasive system (like oil pressure sensor, engine noise sensor or the brake sensor) and the software (like the operating system and the installed applications) capabilities of the vehicle. The concept Activity specifies some properties of the user's activity in progress as well as its starting and ending dates. The concept Network describes the characteristics of the network like the transport protocol, the connectivity and the QoS of the available connections. Finally, the concept Physical Environement provides physical properties of the driver space (like the ambient temperature) and her/his relation with her/his external environment.

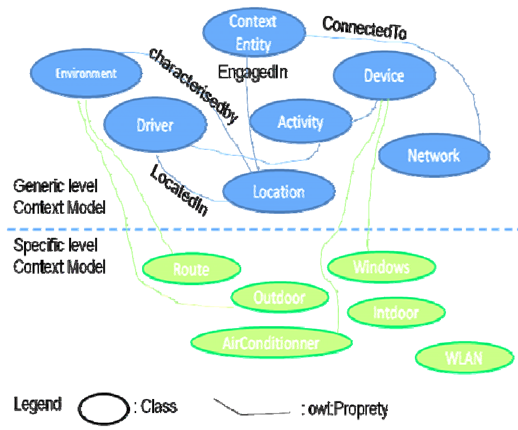

Fig 1: The context ontology structure

For this work, we decided to use Protégé [10]. Protégé is an ontology editor, a knowledge-base editor, as well as an open-source that provides an extensible architecture for the creation of customized knowledge-based applications. Protégé was chosen due to its strong user community, its ability to support the OWL language (discussed below), its ease of use (as determined by previous experience), and its ability to be extended with plug-ins such as visualization tools.

We have used the Protégé tool to build this ontology and create some instances in the vehicle context, using this tool, we can export files in OWL format for further reasoning mechanism. One screenshot is shown in Figure 2.

Journal of Theoretical and Applied Information Technology
31st December 2011. Vol. 34 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

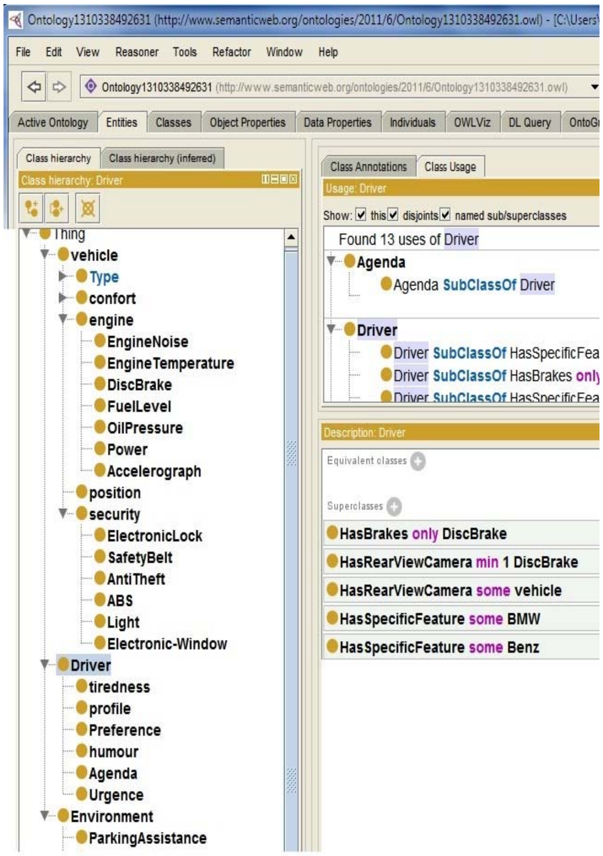

Fig.2. One screenshot of Ontology modeling with Protégé3.4

3.5 Context Reasoning

When taking a formal approach to model context, context can be processed with logical reasoning mechanisms. The use of context reasoning has two folds: checking the consistency of context, and deducing high-level, implicit context from low-level, explicit context.

In this section, we will describe context reasoning based on our platform to demonstrate the key feature of the ontology based context model. We choose to implement context reasoning by using Pellet Model [21].

Pellet is a complete open Source OWL-DL reasoner with acceptable to very good performance, extensive middleware, and a number of unique features. Pellet is the first sound and complete OWL-DL reasoner with extensive support for reasoning with individuals (including nominal support and conjunctive query), user-defined data types, and debugging support for ontologies. It implements several extensions to OWL-DL including combination formalism for OWL-DL ontologies, a non-monotonic operator, and preliminary support for OWL/Rule hybrid reasoning.

4. ARCHITECTURE OVERVIEW

The overview of our proposed architecture is shown in figure 2. The inputs are taken from different sensors which are physically distributed on the vehicle like the oil pressure sensor, engine noise sensor or the brake sensor, Inputs information can come also from the driver itself, like the fingerprint system which is pasted on the steering wheel, the driver's profile and preferences, or even the driver's humors. Other rich source of inputs is the external environment, thus the position information is obtained from a Global Navigation Satellite System (GNSS), and a RFID receivers can detect if there a near oil stations in a area, also the increasing popularity of IEEE 802.11-based wireless LANs (WLANs) in public hot-spots, such as hotels, airports and shopping centers will allow to have plenty of information about their services.

The inputs are given afterwards to the system. The inputs are collected and fused according to the context. The acquired input data is assimilated according to their respective contexts and sent to context reasoning system. The architectural diagram of the Intelligent Pervasive Middleware for Context-Aware Vehicle Services is given below in figure 3.

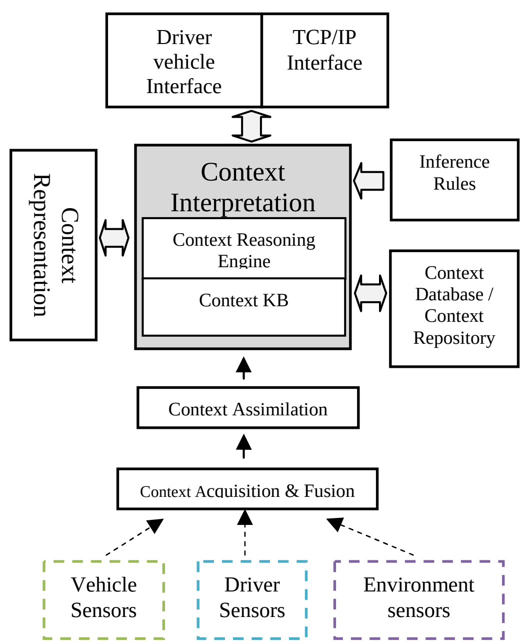

Fig.3. Overview of the platform architecture

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Context Acquisition & fusion: are context providers, they abstract contexts from different sources and convert them to OWL representation so that contexts can be integrated and reused by other components of the platform.

Context Assimilation: To assimilate the context we make it more abstract because abstract structures are studied not only in logic and mathematics but in the fields that apply them.

Context Database: stores some of the pre-history data about the consumed services and taken choices, Also it can be accessed with others specific database for nourishment.

Inference rules: A set of rules can be reacting according to the context for inferring valid information from direct context (Sensed context from physical sensors).

Context Interpreter: consists of Context Reasoning Engines and Context KB (Knowledge Base). The Context Reasoning Engines provide the context reasoning services including inferring deduced contexts, resolving context conflicts and maintaining the consistency of context KB. Different inference rules can be specified and input into the reasoning engines. The context KB provides the service that other components can query, add, delete or modify context knowledge stored in the Context Database.

Context representation: in order to enable interaction between different applications in the same domain we have to make a unified representation of the context, so that the context representation has a direct relation with the Context Interpreter to represent any valid context.

Driver Vehicle Interface: gives the GUI of road scenario and permits the interacting between the driver and the platform system, and generates alerts and warnings to driver.

The following steps are followed for the detailed Architecture of the platform proposed in Figure 2.

Step 1: Collect sensor data from physical and virtual sensors as xml form file.

Step 2: Parse input and feed to the Context Acquisition & Fusion.

Step 3: Assimilate the context and input to context interpretation phase

Step 4&5: Look up the Context Database and repository for such situation has occurred previously and retrieve relevant data according to the context and return to context interpretation

Step 6 Look up the Inference Rules for rules which takes premises existed in the context and returns a conclusion, if it is newly sensed data; learn it and update newly learned data in the database.

Step 7: Context Reasoning Engine query's a request to context representation.

Step 8: Response from Context Representation is given to Context Reasoning Engine.

Step 9: Context Engine perform the action and inputs to API/ Agent.

Step 10: Agent/API send inputs to the Service server side and generates the alerts/warning signals to the Human Vehicle Interface.

Step 11: The action performed from the Service server side is stored in the log database.

5. CONCLUSION AND FUTURE WORK

Ontologies are used to capture and specify the domain knowledge, whose semantics are expressed through consensual terminologies and formal axioms and constraints. Ontologies are key components for building open and dynamic context-aware pervasive computing systems. In this paper, we have described our use of the OWL language and other tools for building an automotive ontology-based model in a new pervasive context-aware architecture. With an explicit representation of context ontologies, our architecture will allow independently domains context and services to interoperate and help them to share and reason about contexts. By describing the semantic relations among context entities, our model can be used to identify a user's needs hiding in the context values and help for making decision for service discovery.

Further work includes 1) extension of a ontology-based modeling to the context-aware application; and 2) working on more sophisticated mechanism for context reasoning ; and 3) development of applications for ITS in smart car using our model.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFRENCES:

[1] Zhaohui Wu, Qing Wu, Hong Cheng, Gang Pan, Minde Zhao, Jie Sun "ScudWare: A Semantic and Adaptive Middleware Platform for Smart Vehicle Space", IEEE Transactions on Intelligent Transportation Systems - TITS , vol. 8, no. 1, pp. 121-132, 2007

[2] WenYing Guo "Using Semantic Web technologies for ubiquitous computing" Ubi-Media Computing, 2008 First IEEE International Conference on page(s): 377 - 381,2008, Lanzhou

[3] Berbers-Lee ET al., T.,Hendlr,J., and Lassila " The semantic web" Scientific American ,2001 284(5) : 34-43

[4] Anind K. Dey " Understanding and Using Context" Personal and Ubiquitous Computing - PUC , vol. 5, no. 1, pp. 4-7, 2001

[5] Daniel Salber, Anind K. Dey, Gregory D. Abowd "Designing and Building Context-Aware Applications" http://www.cc.gatech.edu/fce/contexttoolkit

[6] Tao Gu, Xiao Hang Wang , Hung Keng Pung, Da Qing Zhang"An Ontology-based Context Model in Intelligent Environments" Computer and Information Science

[7] G. Marsden, M. McDonald, and M. Brackstone. "Towards an understanding of Adaptive Cruise control ". Transportation Research, Part C, Emerging Technologies , volume 9,pages 33–51, 2001.

[8] Y. Bouzid, H. Harroud, A. Berrado and M. Boulmalf, "A Context Aware Platform to Support Mobile Users with Personalized Services", IEEE International Conference on Wireless Information Networks and Systems (WINSYS 2009), Milan, Italy, July 7-10, 2009

[9] C. Bisdikian, I. Boamah, P. Castro, A. Misra, J. Rubas, N. Villoutreix, D. Yeh., "Intelligent pervasive middleware for context-based and localized telematics services," ACM MobiCommerce 2002, September 2002, Atlanta, USA.

[10] http://protege.stanford.edu/ Protégé Software

[11] Beckett, D.: RDF/XML Syntax Specification (Revised). http://www.w3.org/TR/rdf-syntax-grammar/ (2003)

[12] McGuinness, D.L., van Harmelen, "OWL Web Ontology Language Overview" http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1.1 (2004)

[13] Fensel, D. plateforme Ontologies plateforme: Silver Bullet for Knowledge Management and Electronic Commerce, 2003, 2nd edition. plateforme Springer-Verlag plateforme, Berlin.

[14] Feruzan, "Context Modeling and Reasoning using plateforme Ontologies plateforme", University of Technology Berlin, July2007.

[15] N. Baumgartner, W. Retschitzegger, and W. Schwinger. Lost in Time, Space and Meaning – An Ontology-Based Approach to Road Trafic Situation Awareness. In Third Workshop on Context-Awareness for Proactive Systems (CAPS'07), 2007.

[16] &A. Benmimoun. "The Driver as Archetype for Driver Assistance Systems? A Driver Model Based Approachfor the Development of Situation-Adaptive DAS". In 13. Aachener Kolloquium Fahrzeug- und Motorentechnik, 2004.

[17] A. Lattner, J. Gehrke, I. Timm, and O. Herzog. "A Knowledge-based Approach to Behavior Decision in Intelligent Vehicles". In Proceedings of the IntelligentVehicles Symposium, pages 466-471, 2005.

[18] J. Michon. "Generic Intelligent Driver Support – A comprehensive report on GIDS". Taylor & Francis Ltd, 1993.

[19] A. Miene, A. Lattner, U. Visser, and O. Herzog. "Dynamic-preserving Qualitative Motion Description for Intelligent Vehicles". In Proceedings of the IEEE Intelligent Vehicles Symposium (IV'04), pages 642-646, 2004.

[20] Saravanan Kannan, Arunkumar Thangavelu, RameshBabu Kalivaradhan "An Intelligent Driver Assistance System (I-DAS) for Vehicle Safety Modelling using Ontology Approach" International Journal Of UbiComp (IJU), Vol.1, No.3, July 2010

[21] Pellet. Pellet - OWL DL Reasoner, 2003. http://www.mindswap.org/2003/pellet

[22] Harry Chen and Tim Finin, "An Ontology for a Context Aware Pervasive Computing Environment", IJCAI workshop on ontologies and distributed systems, Acapulco MX, August 2003.

[23] [8] Anand Ranganathan and Roy H. Campbell, "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments", Proceedings of ACM/IFIP/USENIX International Middleware Conference, Rio de Janeiro, Brazil, June 2003.

[24] Ian Horrocks, "DAML+OIL: a Reason-able Web Ontology Language", Proceedings of the 8th International Conference on Extending

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Database Technology (EDBT), Prague, March 2002.
[25] Leonid Kalinichenko, Michele Missikoff, Federica Schiappelli, Nikolay Skvortsov " Ontological Modeling ", Proceedings of the 5th Russian Conference on Digital Libraries RCDL2003, St.-Petersburg, Russia, 2003
[26] *M. Madkour, A. Maach* "intelligent pervasive middleware for context-aware vehicle services" Journal of communication and computing engineering 2011 in press: http://www.m-sciences.com/index.php?journal=jcce&page=article&op=view&path%5B%5D=94
[27] T. Strang and C. Linnhoff-Popien "A Context Modeling survey", First International Workshop on Advanced context modeling, Reasoning and management, UbiComp 2004.
[28] Gu, T., Pung, H. K., and Zhang, D. Q. A "Middleware for Building Context-Aware mobile Services". In Proceedings of IEEE Vehicular Technology Conference (VTC2004), Milan, Italy,(2004).
[29] Salber et al., Daniel Salber, Anind K. Dey, and Gregory D. Abowd. "The context toolkit: Aiding the development of context-enabled applications" CHI, pages 434– 441, 1999.
[30] Coen, Michael H. Coen. "Design principles or intelligent environments". AAAI/IAAI, pages 547–554, 1998.
[31] Lalana Kagal, Vlad Korolev, Harry Chen, Anupam Joshi, and Timothy Finin. Centaurus "A framework for intelligent services in a mobile environment" Proceedings of the International Workshop on Smart Appliances and Wearable Computing (IWSAWC), 2001.
[32] Tim Kindberg, John Barton. "A Web-based nomadic computing system". Computer Networks (Amsterdam, Netherlands: 1999), 35(4):443–456, 2001.
[33] Robert Grimm, Tom Anderson, Brian Bershad, and David Wetherall. "A system architecture for pervasive computing". In Proceedings of the 9th ACM SIGOPS European Workshop, pages 177–182, 2000. the 9th ACM SIGOPS European Workshop, pages 177–182, 2000.

AUTHOR PROFILES:

Mr. Mohcine MADKOUR : PhD candidate at the Department of Computer Science at University Mohamed 5 Agdal of Rabat, Morocco.
His research interests: the intelligent transportation systems and Inteligent Vehicle include pervasive systems, context awareness, service adaptation and provisioning... He has some published articles and lot of participation in international conferences.

Dr. Abdelilah MAACH : He got his PhD degree in computer science from the University of Ottawa, He obtained "Habilitation à Diriger les Recherches" from the University Mohamed 5 of Rabat Morocco. He's full Professor in Computer Science at the Mohammadia School of Engineering of the University Mohamed 5 Agdal, Morocco. He's leading the Network and Intelligent systems Group and he has a lot of contributions researches.