Location Models for Pervasive Road Networks

Ana Roxin, Maxime Wack

Location models for pervasive road networks

A. Roxin and M. Wack

Laboratoire Systèmes et Transports (SeT)
Université de Technologie de Belfort-Montbéliard (UTBM)
90010 Belfort, France

***Abstract*—**Context-aware mobile computing aims at designing applications that automatically adapt their behaviors to the available location information and the available nearby sensors and devices. This is done in order to fulfill tasks in a way that suits users' current context best. To achieve this, context representation and manipulation are important issues, so as to establish formal context models. In this paper, basic elements of context-aware systems are described with an emphasis on location information representations. Space models for location-based applications are presented. Considered realistic applications concern intelligent vehicles and pervasive road networks.

***Index Terms*—**Context-aware systems, pervasive computing, ubiquitous computing, location information modeling, location-aware applications, intelligent vehicle, road network model.

I. INTRODUCTION

Context-aware pervasive computing has enjoyed remarkable attention from researchers in diverse areas such as distributed computing and human–computer interaction. Identified as a key issue when dealing with ubiquitous and pervasive computing, context-aware computing challenges involve formal modeling and representation of the real world and the design of adaptive reasoning mechanisms for high-level contexts. More precisely, to develop ubiquitous and pervasive applications, context-aware systems and new interaction paradigms are required to cope with dynamically changing context environments. According to Gaber's classification, interaction paradigms can be classified into three categories: the traditional client to server paradigm (CSP) and two alternative paradigms, the Adaptive Services to Client Paradigm (SCP) and the Spontaneous Service Emergence Paradigm (SEP) that suits more ubiquitous and pervasive environments and requirements respectively [1, 2, 3]. Recall that the main objective of Ubiquitous Computing (UC) is to provide users information and service accesses anytime and irrespective to their location, while in Pervasive Computing (PC), the main objective is to provide spontaneous emergent services created on the fly by mobiles that interact by ad hoc connections [1, 2, 3]. In other words, PC may be viewed as a collective use and cooperation of mobile computers and wireless devices that exist and are available in users' physical environment.

Context-aware software should be able to automatically adapt its behavior to the available location information and the available nearby sensors and devices, to fulfill tasks in a way that suits user's current context best. In this paper, space models for location-based applications are presented. Realistic applications considered concern intelligent transportation systems and infrastructures investigated by many research works recently [10, 11].

The paper is organized as follows: in section 2, basic elements of context-aware systems are described. In section 3, location information modeling is investigated and space models are presented. In section 4, three spatial models for pervasive road networks are presented. Conclusion and future works are given in section 5.

II. ELEMENTS OF CONTEXT-AWARE PERVASIVE SYSTEMS

A context-aware pervasive system has three basic functionalities [13]:

- **Sensing** – finding and presenting information and services to a user;
- **Reasoning** – tagging of context to information to support later retrieval;
- **Acting** – executing a service for a user.

Systems can vary in sophistication in each of these functionalities. Also, these functionalities can be realized in a centralized or a distributed architecture over one or more physical devices. The three functionalities are described below.

Sensing

Sensors provide means to acquire information about the surrounding physical world. The more sensors are used, the more comprehensive the view of the physical world is. The kind of information that can be sensed is determined by a large variety of sensors: light sensors, temperature sensors, motion sensors, touch sensors, smoke sensors, etc.

Reasoning

Once data is obtained using a collection of sensors, the task is to utilize such data and to make sense of it. Various techniques have been used for this:

- Physical mathematical models, such as Kalman filtering
- Feature-based inference techniques, such as cluster algorithms, correlation measures, pattern recognition and neural networks,
- Cognitive-based models, such as logical templates,

knowledge bases, and fuzzy logic.

Apart from data analysis techniques, knowledge-based approaches have been used to represent and manipulate context information acquired from sensors. Such approaches attempt a rich, explicit model of context to facilitate more sophisticated reasoning.

Acting

Once context information has been gathered and situations identified, actions are triggered. Effectors and the actions to be taken are application specific, and the action itself might be to perform further sensing. Two considerations must be taken into account:

- **Performance** – actions might need to be performed in time for it to be of use to the users, and before the situation that triggered the action, changes.
- **Control** – the user should retain control and be able to override actions, cancel actions, stop actions, or reverse the effect of actions. However, only some of these are possible, depending on the nature of the action and the application considered.

The previously mentioned phases of sensing, thinking, and acting, determine design considerations for context-aware system building. These considerations include situations to be recognized, available context information and sensors to be used. An appropriate reasoning technique is then chosen, ranging from simple event-condition rules to sophisticated AI techniques. Finally, appropriate effectors, hardware, and software are employed. The distribution of each of these components would depend on the application (all these components might be situated on the same machine or in a distributed infrastructure).

The sensing subsystem, the thinking subsystem, and the acting subsystem need to be connected. Each subsystem might be complex and decoupled from each other, or tightly integrated into one device. Each subsystem itself might be a collection of distributed components.

Several general architectures are given in the literature for context-aware systems. The following figure shows an abstract layered architecture, labeled with subsystem divisions:

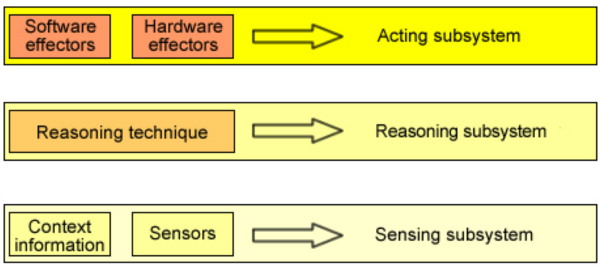

Fig. 1. General architecture of a context-aware system.

An application may have its own thinking and sensing subsystem, rather than use shared one. Between subsystems, generic interfaces may exist. This way each subsystem may interact with another subsystem without knowing the underlying details.

III. Location-Based Applications

Context information may be divided into the following categories:

- **The primary context information** consists in spatial information (identity, time and location). This kind of context information may be used in order to index or to identify entities.
- **The secondary context information** consists of further aspects of a given entity, for example its current state, its activity.

Therefore, spatial information can be used as an index for context information. In order to correctly use this kind of information, a unified model must be defined. Based on this approach, we will try to derive some requirements for general location modeling.

A. Location information considerations

Location-based applications can be indoor, outdoor, or supported both within buildings and outdoors. Important considerations are the positioning technology available and the network coverage, as well as device support for networking technologies.

Depending on the positioning technology used, location information can be represented in different ways — including coordinates in a coordinate system, human-understandable labels, in relative terms and in different granularities.

The granularity of location information might be within centimeters, meters, or greater. For example, the global positioning system (GPS) uses satellites to determine the position of a GPS receiver within an accuracy of several meters [4]. Mobile phone networks can determine the location of an individual within a suburb or town (around 150 to 300m accuracy) in the case of an emergency call.

The variety of positioning technologies and short-range networking technologies will surely continue to improve in their reliability and accuracy as time passes. An extensive review of location systems for ubiquitous computing is given in [5].

B. Location information modeling

We have defined location information as a type of context, moreover as primary context information, which can be used to index entities in a given environment.

A variety of different location models exist, which can be classified as topographical, topological, or hybrid models.

- **Topographic models** use geometry to model space. They model the spatial entities as geometric shapes that are placed within a coordinate system. The relations between the entities, e.g. which entities are next to each other, are implicitly defined by their location.
- **Topological models** describe the relations between

spatial objects explicitly without localizing them in a coordinate system.

- **Hybrid models** combine the localization of the spatial entities within a coordinate system with the explicit modeling of their relations.

The complexity of the models differs widely, e.g. regarding the level of detail and the available functionality. Common tasks are querying for objects within a distinct area or determining the nearest object with respect to a given position. Hence, the context model requires a spatial structure. Since the basis for context models is the spatial structure of the real world, this context models can also be called world models.

Location-aware applications need to be aware of the user's position. Therefore, a location-based application requires a spatial model, in order to allow the assignment of information to a given user position. Three different approaches have been proposed in literature: the Nexus approach, the CANU approach and the EasyLiving approach.

The Nexus approach aims at the definition of world models for mobile and ubiquitous computing applications. It deals with communication, information management, methods for model representation and sensor data integration. The Nexus approach defines a location service that collects and manages location information of mobile entities, without depending on a given application or sensor system. The Location Service uses a distributed architecture to store location information and utilizes efficient dead-reckoning protocols to transmit the location information. A specialized indoor Spatial Model Server for the Nexus platform to support location based indoor applications is described in [6, 7].

The CANU approach aims at location management in mobile ad hoc networks (MANETs) [8]. As for the Nexus approach, CANU applications depend on how the data about their environment is modeled. The CANU approach defines a topologic spatial model, especially designed to answer to the MANETs special requirements. The major fields of research for this project are:

- Defining new routing schemes for dynamic MANETs, that exploit the user's movement characteristics (notably by defining a user profile representing the user's movement pattern [12])
- Defining solid application architecture taking into account the specificities of MANETs.

The EasyLiving approach is a ubiquitous computing project of the Vision Group at Microsoft Research, EasyLiving [9]. This project is mainly focused on an indoor environment – specifically on a living room. EasyLiving uses an explicit spatial model to store its spatial information. This model is stored within a standard relational database system without any spatial extensions. The spatial model of EasyLiving concerns a limited-scale space (e.g., a living room).

IV. A SPATIAL MODEL FOR A ROAD NETWORK

In this work, we focus on spatial models for road networks, in particular for Vehicle ad hoc networks (VANETs).

A. Vehicle ad hoc networks (VANETs)

Before defining the characteristics of a vehicular network, we will first define the characteristics of a single intelligent vehicle. A vehicle becomes intelligent when it is equipped with sensors, computing and communication systems. A so-called vehicle will have the following characteristics:

- Environment sensing with various sensors (mechanical, acoustic or thermal sensors)
- Powerful computing capability with onboard processors, in order to handle real-time electronic signals from the above mentioned sensors.
- Location information awareness with a GPS receiver providing real-time geographic position and global time information.
- Wireless communications with RFID, GPRS or WLAN 802.11 systems allowing the vehicle to communicate with other vehicles.

With these characteristics, an intelligent vehicle can now be compared to a computer node in a MANET. Several intelligent vehicles form a vehicular network, VANET which is a type of MANET, but has its own properties, notably:

- Constrained node spatial distribution – indeed, vehicles usually run on roads, so the distribution of vehicles inside a given area is non-uniform, but constrained by the road network.
- Dynamic topology – since vehicles can move at high speed, the topology of a vehicular network will have to suffer frequent changes.

B. Road network spatial model

Given the above mentioned characteristics of a vehicular network, spatial information of road networks can help vehicles (that are mobile nodes of a MANET) understand their environment. In order to access the spatial information, each node, or vehicle, includes a spatial model to accommodate all known spatial information. The representation of spatial models varies according to the complexity of spatial information. Here we introduce three types of spatial models: a flat graph model, a hierarchical graph model and a real-time graph model.

Flat graph model

The flat graph model only stores the road network topology and the geographic coordinates of each node. Therefore, the road network may be represented through a 2D graph, as shown in the following figure. The crossroads are represented by the graph's vertices, and road segments are represented by the graph's edges.

This model allows the following operations on spatial information:

- Point location – given the point's coordinates, the model returns the graph's edge where the point is located;
- Shortest path computation – given the coordinates of the

start and the end points, the model returns the shortest path between these two points, as well as its length.

Fig.2. Illustration of the flat graph model

This first road network spatial model can be compared to the EasyLiving approach, we have presented above. It does not support mechanisms for extension of the spatial model. The model is a topographic model and is based on geometric relations between the objects.

Hierarchical graph model

The second proposed model is based on an hierarchical graph. Indeed, the first model offered limited operations on spatial information (retrieving the relation between two road segments). Additional semantic information is needed, notably the roads' name, level of importance, quality or transportation capacity. This hierarchical model represents the road network through a graph with vertices and edges that have extended properties, comparing to the flat graph model.
The graph's edges will include two types of information:

- Semantic information – Road name, section name, segment name, road status, road's level of importance.
- Geographic information – Road length.

Semantic information implies a hierarchy between the different information it includes. Therefore, the graph's edges name must take into account this hierarchy.

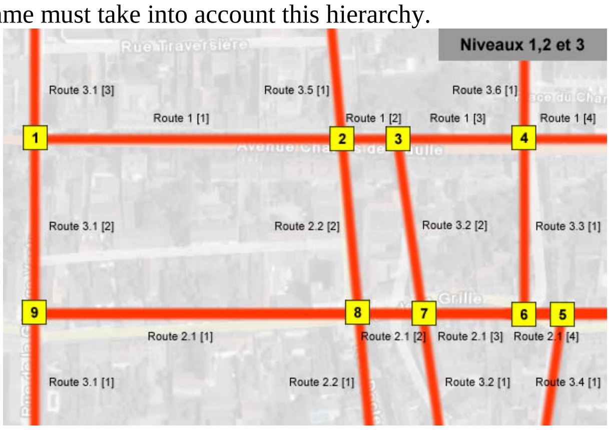

Fig.3. Illustration of the hierarchical graph model – levels 1, 2 and 3.

The spatial model must provide functions to map both semantic and geographic information. For example, if semantic information includes road's level of importance, the spatial model will provide different views of the same road network, as showed in the following figure.

Fig.4. Illustration of the hierarchical graph model – levels 1 and 2.

This hierarchical graph model may be compared to the CANU approach, since it makes route computation easier, and could easily be applied to MANETs.

Real-time graph model

The last proposed model takes into account the time parameter, in order to reflect the real-time characteristics of a road network. This model integrates more complex spatial information, notably:

- Density of vehicles on the road (N)
- Road condition (Q)

A 2D graph may also be used to represent this model. Its edges and vertices have attributes that include real-time data. A timer must be defined, in order to trigger the refreshment or the deletion of real-time data. The following figure shows a representation of this real-time graph model.

Fig.5. Illustration of the real-time graph model.

Operations allowed by this real-time model are:

- Data update
- Data local aggregation – the spatial model allows combining data from several edges.

This model has much in common with the Nexus approach we have mentioned earlier in this article. Indeed, as the Nexus model which defines a standardized but extensible world model, this third model allows creating an infrastructure framework to manage the information of the world model. This real-time model takes time into account, so the established framework allows data update.

V. Conclusion

In this article, main definitions and concepts for context-aware pervasive computing are presented with an emphasis on location-aware applications. In particular, this kind of application needs to model the real environment, in order to perform context-dependent tasks. Three spatial models for VANETs are presented. Each model allows different operations and therefore, they each suit a different application. For further work, rules and appropriate actions have to be defined for each type of application. When these rules are triggered, notifications are sent to the application reasoning and acting components. In addition, performance and control issues have to be considered in particular for VANETs requirements.

VI. References

[1] Bakhouya, M., Gaber, J., Ubiquitous and Pervasive Application Design. Encyclopedia of Mobile Computing & Commerce, Eds. D. Taniar, Idea Group Pub, Fb. 2007.

[2] Gaber, J., New paradigms for ubiquitous and pervasive computing, White paper, Université de Technologie de Belfort-Montbéliard (UTBM), France, 2000.

[3] Gaber, J., New paradigms for ubiquitous and pervasive applications, Proceeding of First Workshop on Software Engineering Challenges for Ubiquitous Computing, Lancaster, UK, 2006.

[4] Andersson, C., GPRS and 3G Wireless Applications: Professional Developer's Guide, John Wiley & Sons, U.S.A., 2001.

[5] Roxin, A., Survey of location systems for ubiquitous and pervasive environments, Research Report RR-11-06, Université de Technologie de Belfort-Montbéliard (UTBM), France, 2006.

[6] Grzan, S., Enabling technology for an indoor location aware information system, Diploma Thesis No. 1958, 2002.

[7] Nicklas, D., Großmann, M., Schwarz, T., Volz, S., Architecture and Data Model of Nexus, GIS Geo-Informations-Systeme, vol. 9/2001, Heidelberg, 2001.

[8] CANU project is available online at: http://www.ipvs.uni-stuttgart.de/abteilungen/vs/forschung/projekte/Communication_in_Ad-hoc_Networks_for_Ubiquitous_Computing/de

[9] EasyLiving project is available online at : http://research.microsoft.com/easyliving/

[10] Dridi M., Mesghouni K., Borne P.: Public transport regulation using evolutionnary algorithms. The 10th World Congress on Intelligent Transport Systems, ITS'2003, Madrid, November 2003.

[11] Dridi M., Kacem I., Mesghouni K., Control of an urban transportation system: optimisation and vehicle scheduling. Combinatorial Optimisation 2004, Lancaster University, Lancaster, March 2004.

[12] Bauer M., Becker C., Rothermel K., Location Models from the Perspective of Context-Aware Applications and Mobile Ad Hoc Networks. Personal and Ubiquitous Computing Volume 6, Issue 5-6 (December 2002) Pages: 322 – 328, 2002.

[13] Loke S., Context-Aware Pervasive Systems - Architectures for a New Breed of Applications, Auerbach Publications, 2007.

VII. Biography

Ana Roxin is a PhD candidate in the Systems and Transports Department at the University of Technology of Belfort-Montbéliard. She received her engineer diploma in 2006, in Mobile Networks and Embedded Systems. Her current research interests lie in the area of geolocation, localization and positioning techniques, mobile networks and ubiquitous and pervasive computing.

Maxime Wack received his PhD degree in 1981 in computer science from the Université de Technologie de Compiègne (UTC), France. Currently, he is associate professor-HDR (Habilité à diriger les recherches) at the laboratory Systems and Transportation at Université de Technologie de Belfort Montbéliard (UTBM). He is the head of the research group Geopositioning, Embedded systems and Mobility (GSEM). His research interests include information systems, security, digital signature and certification, ubiquitous and pervasive computing, location based services and distributed systems.