

A Knowledge-Based Framework for the Support of Sensor Web Deployment Using Multi-Agent Geo-Simulation

Mehdi Mekni, Normand Bergeron
 Institut National de Recherche Scientifique
 Environnement Terre Eau, Quebec, Canada
mmekni@gmail.com
Normand.Bergeron@ete.inrs.ca

Hedi Haddad, Bernard Moulin
 Depart. of Computer Science and Software Engineering
 Laval University, Quebec, Canada
Hedi.Haddad.1@ulaval.ca
Bernard.Moulin@ift.laval.ca

Abstract— Sensor Web deployment is by nature a spatial problem since nodes are highly constrained by the geographic characteristics of the environment. Therefore, there is a need for an efficient modelling paradigm to address the issue of SW deployment taking into consideration the constraints of the geographic space and knowledge it provides to support their autonomous decision making capabilities. In this paper we propose a knowledge-based framework to support the simulation of SW deployments in Informed Virtual Geographic Environments (VGE) using multi-agent geo-simulation techniques. This framework builds on our previous works on Informed Virtual Geographic Environments, on spatially reasoning agents and on qualitative reasoning about geo-simulation results. The framework is illustrated with a scenario of a sensor web deployment for weather monitoring purposes.

Keywords: *Sensor Webs, Multi-Agent Geo-Simulation, Informed Virtual Geographic Environments, Knowledge Representation, Spatio-temporal Reasoning.*

I. INTRODUCTION

Recent advances in wireless communications and sensing technologies have enabled the development of low-cost, low-power, multi-functional sensor nodes that are small in size and which communicate over short distances [1]. Sensor Webs (SW) are distributed network systems composed of hundreds of such sensor nodes [1]. New capabilities such as micro-sensing and in-situ sensing as well as the wireless connection of these nodes open new possibilities for applications in various domains such as military, environment and disaster relief [2]. The low per-node cost and the shrinking size of microprocessors in addition to the enhancement of their computation capacities, while decreasing their energy consumption, will allow dense distribution of these wireless networks of sensors and actuators [1]. SW can be thought of as a macro-instrument concept that allows for the spatio-temporal understanding of phenomena which take place in geographic environments through the coordinated efforts of a large number of sensing nodes of different types [2]. However, once SW are deployed, the management of such complex systems is a real challenge because of their limited energy, communication, and processing capabilities [2]. Sensor web management, as defined in [3], is “*to manage, co-ordinate and integrate sensor nodes to accomplish*

specific and often dynamic sensing mission objectives”. The term “*manage*” means the control over the sensors. The term “*co-ordinate*” outlines the load balancing of tasks assigned to sensors with respect to their limited capabilities. Finally, the term “*integrate*” highlights the organization of the sensors into a coherent and structured network deployed in a geographic environment. Thus, sensor web deployment is by nature a spatial problem since nodes are highly constrained by the geographic characteristics of the environment.

The majority of currently deployed sensor webs are mainly used for prototyping purposes [1]. In many cases, it is impractical to experiment on real sensor web systems for several reasons. First, a particular hardware platform, while theoretically possible, may not yet be manufactured because its fabrication may be constrained by technical or design limits [2]. Second, even if the hardware platform exists, it may be prohibitively expensive for experimentation [1]. For example, applications developed for research purposes may require hundreds or thousands of nodes in order to accurately monitor a natural phenomenon [2]. With current sensor nodes costing up to hundreds of dollars, evaluating such research could cost tens of thousands dollars [2]. Third, even if it is practical to evaluate research on the real hardware platform, it may not be practical to experiment in an appropriate environment [2]. An example of this are sensor webs which operate on glaciers, remote wildlife habitats, volcanos, and other environments where in-situ sensing techniques are required and with which it is expensive or dangerous to experiment [1]. Therefore, there is a need for an efficient modelling paradigm to address the issue of sensor webs deployment using actors representing sensor nodes evolving in and interacting with a representation of their geographic environment.

One solution to this problem is Multi-Agent Geo-Simulation (MAGS). MAGS is a modelling and simulation paradigm which aims to study geographic phenomena in a variety of domains involving a large number of heterogeneous actors (implemented as software agents) evolving in, and interacting with, a Virtual representation of the Geographic Environment (VGE) [4]. Most of the current SW simulation platforms that we analyze and assessed suffer from several limits. In fact, these platforms lack an explicit representation of the geographic environment [5]. A SW is deployed in a spatial environment, and ignoring the characteristics of this

environment would greatly decrease the quality of SW simulations. Consequently, sensor models are usually over simplified and do not support autonomous spatial reasoning and decision making capabilities that take into account the characteristics of the geographic environment. Moreover, SW simulations generate a huge volume of data which is usually analyzed using mathematical and statistical models [6]. However, since human reasoning is mainly qualitative and not quantitative, qualitative data analysis models are more suitable for decision support purposes.

In order to remedy to the above mentioned limits, a critical step towards the geo-simulation of SW deployment is the creation of appropriate representations of the geographic space (VGE) and of the sensors evolving in it, in order to efficiently support the sensors' spatio-temporal reasoning capabilities [5]. Moreover, a VGE should provide sensor agents with knowledge about the virtual environment in which they evolve and with which they interact. A number of challenges arise when creating knowledge about the environment, among which we mention: 1) to represent knowledge using a standard formalism; 2) to provide agents with tools and mechanisms to allow them acquire knowledge about the environment; and 3) to infer and to predict based on premises and facts that characterise the geographic environment in order to support spatial agents' decision-making.

In order to address the above mentioned challenges, we propose a knowledge-based multi-agent geo-simulation framework to support the simulation of SW deployments in VGEs. This framework builds on our previous works on Informed Virtual Geographic Environments [5], on spatially reasoning agents [2] and on qualitative reasoning about geo-simulation results [7].

The rest of the paper is organised as follows. Section 1 presents related works on agent-based simulation tools for SW. Section 2 presents our framework and details its underlying components. Section 3 illustrates our framework through a SW deployment scenario for weather monitoring purposes. Section 4 discusses the results and concludes with our future works.

II. AGENT-BASED APPROCHES FOR SENSOR WEBS

According to our literature review, architectures for the management of sensor webs involving the Multi-Agent Geosimulation paradigm do not exist. However, a few research projects have attempted to integrate the agent paradigm into sensor web architectures such as IrisNet [8], Abacus [9], Biswas and Phoha's architecture [10], and SWAP [11].

Most of these architectures identify the need for distributed data collection and processing, and propose layered architectures to achieve this. In Abacus different agents in the processing layer detect and report alert conditions to a higher layer interacting with users [9]. IrisNet uses agents such as Sensor Agents (SA) and Sensor Organisers (SO) to collect and analyze data from sensors to answer specific classes of queries [8]. Biswas and Phoha's

approach uses agents in the service layer to analyze data from sensors and transfer it to the application layer [10].

All these approaches deal with data collection by providing a distributed infrastructure for publishing, discovering and accessing sensor resources. They also address the challenge of data fusion, to some extent, and aim to provide end-users with the information they need. These approaches share a common objective through the use of the agent-paradigm which is the distribution of tasks. However, these applications do not take complete advantage of the multi-agent systems approach. Indeed, they use reactive agents which are efficient for alerting purposes, but are neither able to perform situated behaviors nor autonomous decision-making. On the one hand, situated behaviors include performing spatial reasoning and taking advantage of the virtual environment's description where sensor agents are located. On the other hand, autonomous decision-making includes managing sensor nodes in order to efficiently cover the area of interest while taking into account their limited capabilities as well as local spatial characteristics. We think that, in order to achieve intelligent and autonomous, it is essential to use a multi-agent geo-simulation approach in which agents are endowed with advanced capabilities such as perception, navigation, memory, and knowledge management. As the abovementioned architectures do not address the more challenging sensor web management issues, we propose the knowledge-based multi-agent geo-simulation framework for the intelligent management of sensor webs.

III. A KNOWLEDGE-BASED MULTI-AGENT GEO-SIMULATION FRAMEWORK

As we mentioned in the introduction, in this paper we propose a knowledge-based multi-agent geo-simulation framework to support the simulation of SW deployments in VGEs.

Figure 1 illustrates the main components of the framework. Multi-Agent Geo-Simulation is used to simulate the behaviour of a SW in a dynamic virtual geographic environment. Sensors are modeled as intelligent agents embedded in a virtual space where dynamic phenomena can occur. Sensor agents have reasoning capabilities allowing them to reason about the virtual space and to react to its dynamic phenomena. Spatio-Temporal knowledge is used for two main purposes.

First, it is used during the geo-simulation to support agents reasoning capabilities. Second, it is used to analyze the results of the geo-simulation and to offer decision support to users. Finally, the results of the geo-simulation (which are inserted as facts in the *Result Facts Base*) are analyzed in order to offer decision support. In the following we present these components.

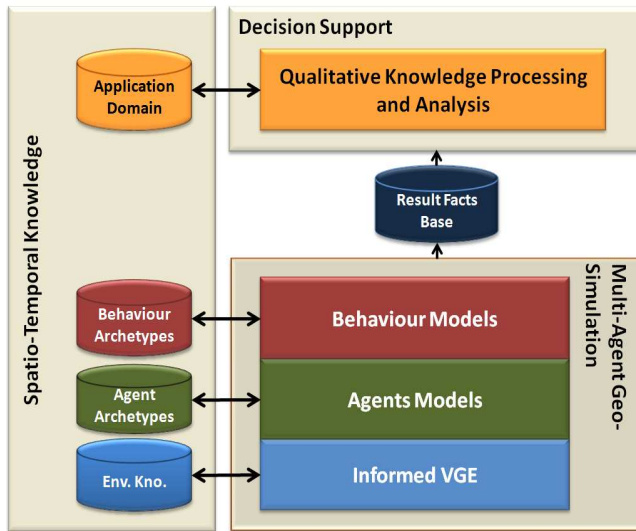


Figure 1: The proposed framework

A. Multi-Agent Geo-Simulation

The idea behind a multi-agent geo-simulation approach is to move the most intensive processing out of the Physical Sensor Web (PSW) into a parallel Virtual Sensor Web (VSW) operating on a base station or a remote server. The objective is to reproduce, in a realistic manner, the real world in a virtual environment. Indeed, in this virtual environment, which imposes no limits on data processing, energy consumption and communication capabilities, it is possible to create a system for the deployment of the physical sensor web. In order to faithfully mimic the physical sensor web deployed in the area of interest, we need to simulate, in a realistic way, the physical sensor nodes as well as the geographic environment where they are located. Physical sensor web are represented in the virtual environment using software agents. An agent is a program with domain knowledge, goals and actions [2]. An agent can observe and sense its environment as well as affect it. Agents' capabilities may include (quasi-) autonomy, perception, reasoning, assessing, understanding, learning, goal processing, and goal-directed knowledge processing [7]. The reproduction of the geographic environment in which physical sensor nodes are deployed should be based on reliable data obtained from Geographic Information Systems (GIS). The concept of Multi-Agent Geo-Simulation (MAGS) evolves from such type of simulations involving software agents immersed in a virtual geographic environment.

MAGS has attracted a growing interest from researchers and practitioners to simulate various phenomena in a variety of domains including traffic simulation, crowd simulation, urban dynamics, and changes of land use and cover, to name a few [6]. Such approaches are used to study various phenomena (i.e. car traffic, crowd behaviours, etc.) involving a large number of simulated actors (implemented

as software agents) evolving in, and interacting with, an explicit description of the geographic environment called Virtual Geographic Environment (VGE) [4].

MAGS is a useful approach to integrate the spatial dimension in our sensor web models. From this perspective, the Geographic Information System (GIS) plays an important role in the development of sensor deployment geo-simulation models. MAGS can be thought of as a coupling of two technologies: the Multi-Agent Systems (MAS) and the Geographic Information Systems [2]. Based on the MAS technology, the simulated entities are represented by software agents that can behave and make decisions autonomously. They can interact with other agents and with a virtual representation of the actual geographic environment. They may be reactive, proactive, stationary or mobile, social or cognitive [2]. These agents evolve and interact with their VGE.

Although several research works have addressed the issue of modeling and representing agents' characteristics using formal and standard formalisms [12, 13], only a few works have attempted to adopt a standard formalism in order to represent virtual environments' characteristics [14]. The main reason why virtual environments have received less interest from practitioners is that geographic environments may be complex, large-scale, and densely populated with geographic features of various extents. As a consequence, formally representing knowledge about geographic environments is usually complex and time and effort consuming. Another issue which needs to be addressed is the way to allow spatial agents to acquire this knowledge in order to autonomously make decisions with respect to their environment's characteristics. There is a need for a knowledge management approach: (1) to represent knowledge about geographic environments using standard formalisms; (2) to allow spatial agents to acquire knowledge about the environment; (3) to allow agents to reason and to make decisions while taking into account knowledge about geographic environments.

B. Spatio-Temporal Knowledge

As we mentioned so far, spatio-temporal knowledge is used in the framework 1) to support agents decision making during the geo-simulation and 2) to analyze the results of the geo-simulation in order to offer decision support. In the following we respectively present the representation formalism and the categories of spatio-temporal knowledge used in the framework.

1) Representation formalism

We use Conceptual Graphs (CGS) to represent spatio-temporal knowledge and to support spatio-temporal reasoning. CGs were introduced by Sowa [16] as a system of logic based on Peirce's existential graphs and semantic networks of artificial intelligence. They provide extensible means to capture and represent the semantic of real-world knowledge and have been implemented in a variety of projects for information retrieval, database design, expert

systems, qualitative simulations, and natural language processing. However, their application to dynamic geographic spaces modeling and analyzing is an innovative issue [15]. More details about CGs and their theoretical foundations can be found in [16], among others.

Syntactically, a conceptual graph is a network of concept nodes linked by relation nodes. Concept nodes are represented by the notation [Concept Type: Concept Instance] and relation nodes by (Relationship-Name). The formalism can be represented in either graphical or character-based notations. In the graphical notation, concepts are represented by rectangles, relations by circles and the links between concepts and relation nodes by arrows. The character-based notation (or linear form) is more compact than the graphical one and uses square brackets instead of boxes and parentheses instead of circles. Some examples are presented in the following sub-section.

2) Knowledge Categories

We distinguish three levels of spatio-temporal knowledge: 1) Knowledge about the environment, 2) Knowledge about actors and their behaviours and 3) knowledge about the application domain (Figure 1)

a) Knowledge about the environment

We define the notion of knowledge about the environment (Environment Knowledge (EK) for short) as “*a specification of a conceptualization of the environment characteristics: the objects, agents, and other entities that are assumed to exist in the informed virtual geographic environment and the relationships that hold among them*”. Hence, EK is a description (like a formal specification of a program) of the spatial concepts (geographic features) and relationships (topologic, semantic) that may exist in a geographic environment. What is more important is what environment knowledge is *for*. In multi-agent geo-simulation, EK is a specification used for enabling knowledge exploitation for spatial agents. Practically, EK is an agreement to use spatial concepts (i.e., ask queries and make assertions), spatial relationships (i.e., describe actions and behaviors), in a way that is consistent so we can share knowledge with and among spatial agents. Our aim is to improve the perception-decision-action loop on which relies most of the existing agents’ models.

b) Knowledge about Actors and Behaviours Archetypes

When dealing with MAGS simulating Sensor Webs involving a large number of sensors of various extents, the specification of their attributes and associated spatial behaviors may be complex and time and effort consuming. Agents’ characterization aims to specify: (1) the *agent archetype*, its super-types and sub-types according to the semantic type hierarchy; and (3) the behavior archetype that an agent archetype is allowed to perform including moving within the informed VGE, perception of the geographic features and other spatial agents. Figure 2 shows an example of a semantic type hierarchy of agent archetypes. *Entity* is

an abstract node and *Storm*, *River*, *Building* and *Sensor* are instance nodes (leaves) of this agent archetype lattice. A key characteristic of agent archetype is *inheritance*. Agents belonging to one or several agent archetypes inherit the characteristics associates with these agent archetypes. For example, let us consider two agent archetypes *Temp_sensor* and *Press_sensor* respectively sensing temperature and pressure. The *Temp_sensor* is characterised by a measurement frequency f . On the other hand, *Press_sensor* is characterised by a one meter circular sensing field. Consider now *TP_sensor* a multi-functional sensor which inherits from *Temp_sensor* and *Press_sensor*. Thanks to the inheritance property provided by agent archetypes, this agent performs measurements at a frequency f within a circular sensing area of one meter.

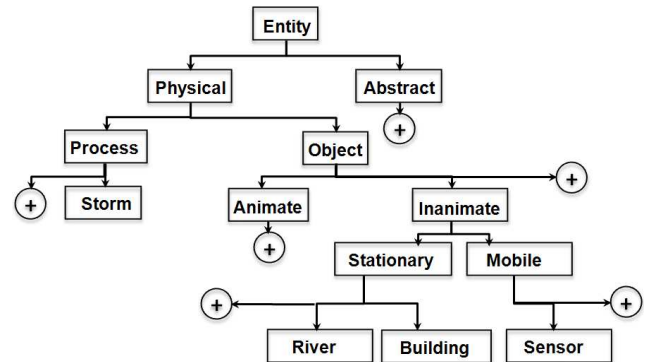


Figure 2: Semantic Type Hierarchy of Agent Archetypes

Since our research addresses the simulation of spatial behaviors, it has been influenced by some basic tenets of *active theory* [17]. In particular, our approach to manage environment knowledge rests on the commitments in active theory that: (1) activities are directed toward objects, zones, or actors; (2) activities are hierarchically structure; and (3) activities capture some context-dependence of the meaning of information [17]. Theoretically, the common philosophy between our knowledge-based approach and activity theory is a view of the geographic environment from the perspective of an agent interacting with it [18]. Practically, the most important borrowings from activity theory are the idea that: (1) the semantic of behaviors and objects are inseparable; and (2) behaviors, objects, as well as agents are hierarchically structured [18].

Let us define the following behavior archetypes that we associate with the *Sensor* agent archetype as follows (Figure 3):

1. “an agent $*m$ which is a sensor measures an object $*c$ which is a phenomenon with a frequency $*f$ ”
2. “an agent $*m$ which is a sensor measures an object $*c$ which is a measurement of unit $*u$ ”

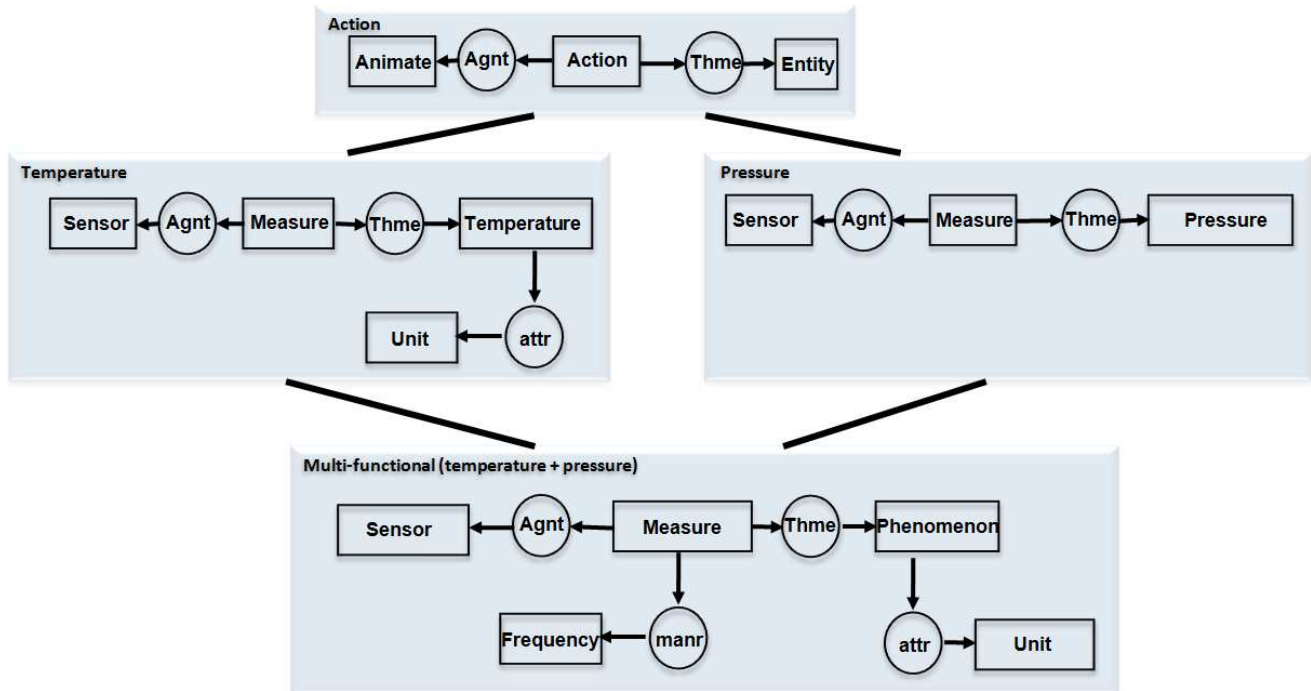


Figure 3: An example of behaviour archetypes

These behaviour archetypes are expressed using linear notation of CGs respectively as follows:

```
[MEASURE:*d]
-(agnt)->[SENSOR:*m]
-(obj)->[Phenomenon: *c]
-(manr)->[Frequency:*f]
```

And

```
[MEASURE:*d]
-(agnt)->[SENSOR:*m]
-(obj)->[MEASUREMENT: *c]
-(attr)->[Unit:*u]
```

Since the above description is equal or more specific than the antecedent of the following behaviour, it can be inferred, by deduction, that:

```
[MEASURE:*d]
-(agnt)->[SENSOR:*m]
-(obj)->[MEASUREMENT: *c]
-(manr)->[Frequency:*f]
-(attr)->[Unit:*u]
```

c) Knowledge about the Application Domain

The above mentioned levels of knowledge are used during the geo-simulation to support agents in their decision making. In contrast, knowledge about the application domain is mainly used to qualitatively analyze the results of the geo-simulation and is thus more linked to decision support. In the context of SW deployment, nodes are aimed to collect measurements about phenomena of interest which vary according to the application domain (military, environmental surveillance, etc.). Knowledge about the

application domain defines phenomena of interest in a particular application domain.

In our framework, we use the concept of *spatio-temporal situations* [19] to model and reason about phenomena of interest. A spatio-temporal situation represents a state, an event or a process situated in space and time and involving various objects of the world. Examples of spatio-temporal situations can be a sensor which brake down for a certain period of time at a certain spatial area (state), the start of rain in certain spatial area (punctual event) or a durative heavy rain in a given area (process). A spatio-temporal situation has a semantic type (rain, network breakdown, etc.), a start and end times and is located in space. Knowledge about the application domain defines spatio-temporal situations of interest according to their temporal (state, punctual event or durative process) and semantic characteristics. For example, the semantic punctual event “Start of rain” can be defined as the fact of water level exceeding a given threshold. Relationships between spatio-temporal situations (temporal and spatial) are also specified in the application domain knowledge, which enables defining complex phenomena. For example, a situation of storm can be defined as a situation of heavy rain *followed by* / *accompanied with* a situation of strong wind.

3) Decision support

The decision support component analyzes the result of the geo-simulation using application domain knowledge in order to identify situations of interest to the user. This data analysis process is implemented using the approach proposed in [19]. Details of this approach are beyond the

scope of this paper. We only illustrate the principle using the simple example showed in Figure 4.

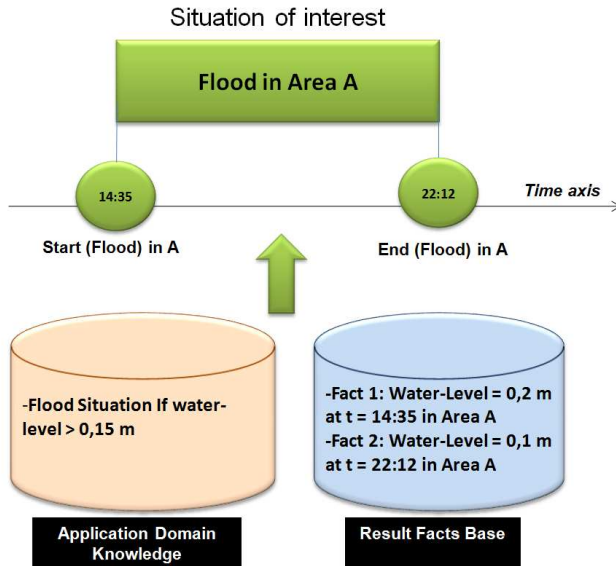


Figure 4: An example of simulation result processing.

In this example, the situation of interest is Flood. The application domain knowledge specifies that there is a flood situation if the water-level exceeds 0,15 meter. Otherwise, there is no flood situation. The decision support component uses this knowledge in order to analyze the facts collected by agents during the simulation (Result Facts Base). Particularly, the two facts illustrated in Figure 4 are respectively interpreted as the start of flood in (Area A, $t = 14:35$) and the end of flood in (Area A, $22:12$) (punctual events). Finally, start and end flood situations are used to identify the flood situation itself as a process located in Area A during the time interval $[14:35, 22:12]$. Obviously,

detection of real complex situations requires taking into consideration other aspects (measurement errors, conflict of measurements between several sensors, etc.) that are beyond the scope of this paper.

IV. RESULTS AND SCENARIO

In order to illustrate our knowledge-based multi-agent geo-simulation framework, we propose to simulate a sensor web deployed in an IVGE representing the experimental forest of *Montmorency* (Quebec, Canada) for weather monitoring purposes (Figure 5 and Figure 6). This scenario shows how agents adapt their spatial behaviors with respect to knowledge they acquire from the IVGE using our environment knowledge management along with their perception capabilities. The objective of the simulated sensor web is to identify and monitor a simulated storm evolving in the IVGE. In order to reason about knowledge, we used a platform to deal with CGs manipulation called *Amine* [20]. Amine platform provides a pattern-matching and rule based programming paradigm embedded in Prolog+CG language which is basically an object-oriented and conceptual graphs-based extension of Prolog language. A few agent archetypes representing different kinds of sensors are involved in this scenario. These sensors are first randomly deployed in the IVGE. Then, each sensor computes a path in order to reach its deployment position while taking into account the geographic environment characteristics. When sensors reach their final destinations, some of them stay active while other switch to idle in order to preserve the overall energy of the sensor web. Active sensors make measurements at a frequency f in order to monitor weather conditions. Active and idle sensors as well as the measurement frequency are specified in the simulation scenario created by the MAGS's user.

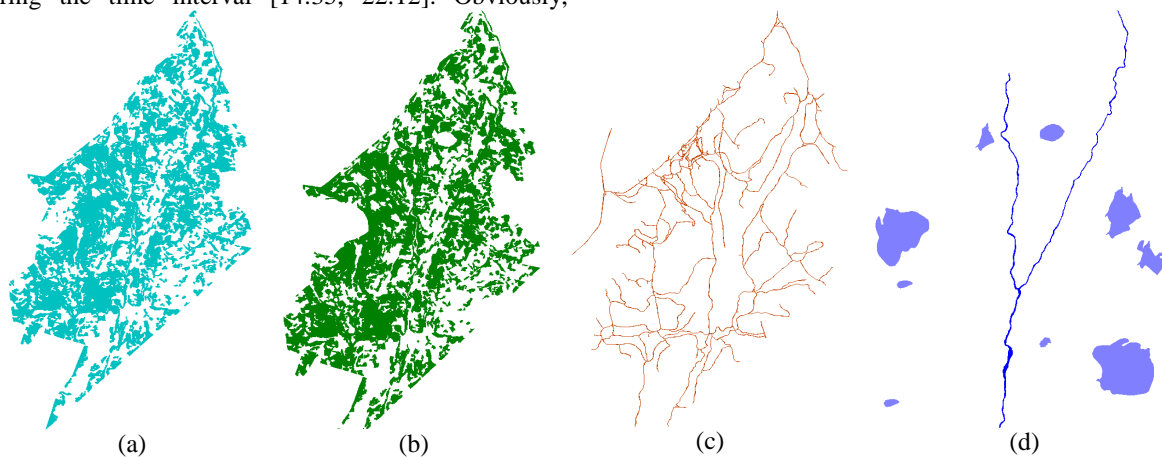


Figure 5: Various geometric and semantic layers related to the Montmorency experimental forest, St.Lawrence Region, Quebec, Canada: (a) and (b) two types of vegetation characterizing the land cover; (c) water resources including rivers and lakes; and (d) road network.

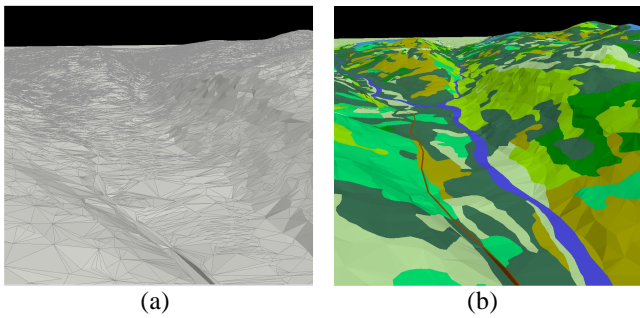


Figure 6: IVGE representing the Montmorency experimental forest; (a) the 2.5D triangulated elevation map and (b) the 3D unified map.

The simulated storm appears after a time frame t from the beginning of the simulation. If an active sensor perceives the storm agent, it directly accesses to its properties and extracts the information that it monitors depending on its kind of sensor, i.e. temperature, pressure, wind speed and direction, or humidity. If a difference above a certain threshold Δ is observed, the sensor proceeds as follows: (1) it accelerates its measurements frequency, (2) it adds a new fact that keeps track of the event with its timestamp in the Result Facts Base; and (3) it sends a message to wake up idle all sensors of the same kind which are situated in a certain estimated distance.

As the simulation time goes by and the storm agent evolves in the IVGE, most of initially idle sensors become active to sense the observed phenomenon. When the storm agent is out of the perception field of the sensor, this latter senses a new difference between the past and the current measurement. It notifies the Result Facts Base by adding a new fact that keeps track of the new event with its timestamp; Idle and active sensors switch states in order to preserve their energy.

In order to model the simulation described above, let us first consider the two agent archetypes *ZONE* and *SENSOR*. In contrast with the *ZONE* agent archetype which represents a geographic area, the *SENSOR* agent archetype is associated with individual sensors deployed in the informed virtual geographic environment.

Let us consider the two agent sub-types *WEATHERZONE* and *STORMZONE*. Agents of type *WEATHERZONE* are stationary and represent meteorological conditions within the geographic area they cover. 5 instances of *WEATHERZONE* are created in order to approximately cover the monitored area (Figure 7). In addition to their geometric characteristics, these agents encompass attributes which characterize the meteorological conditions such as temperature = 18° C, pressure = 1010 hPa, humidity = 30%, and wind = 3km/h-NW. A single instance of type *STORMZONE* is created to represent the weather storm phenomenon. Obviously, this agent is mobile and also encompasses attributes characterizing its meteorological conditions.

When a sensor detects a difference above the threshold, it adds a fact in the Results Facts Base using the CGs formalism. Consider the following example involving the *sensortemp1* adding a fact describing an observed difference of temperature measurement of value 18 at cell 367 at 15h : 34 : 22.

```
[MEASURE:*temperature]
-(agnt)->[SENSOR:*sensortemp1]
-(obj)->[TEMPERATURE: 18]
-(time)->[TIMESTAMP: 15:34:22]
-(loc)->[CELL: 367]
```

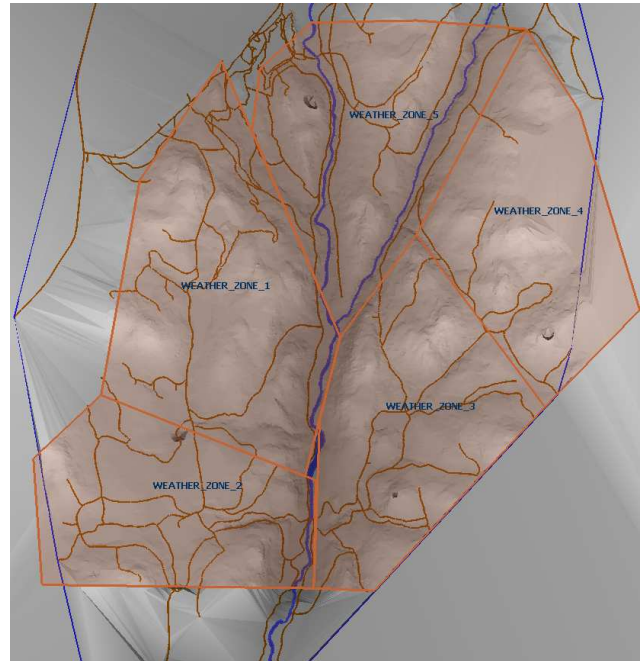


Figure 7: 5 *WeatherZone* agents covering the monitored geographic area.

Let us now define the following sub-types of *SENSOR* archetype: *TEMPSENSOR* for temperature measurement, *PRESSENSOR* for atmospheric pressure measurement, *WINDSENSOR* for wind speed and orientation measurement, and *HUMISENSOR* for humidity measurement (Figure 8).

In this scenario, the situation of interest is the storm. Let us suppose that we need to describe the evolution of the storm during the geo-simulation for decision support purposes. Knowledge about the application domain allows specifying how the presence of a storm phenomenon can be detected in a certain area.

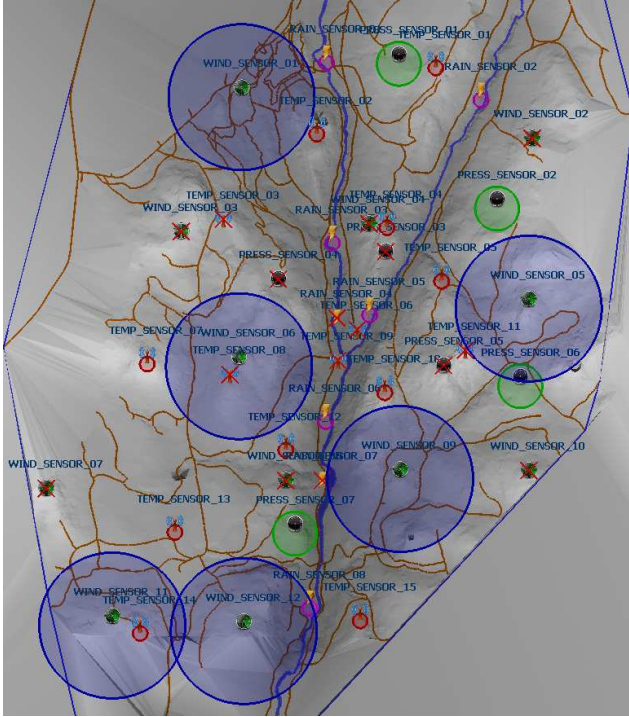


Figure 8: The simulated sensors autonomously reaching their deployment position in the IVGE. The blue circle corresponds to the perception field for active sensors.

For simplification, let us consider the following (Prolog+CG) rule specifying that a storm is detected at time T in area A if there are (in the results facts base) facts describing that temperature exceeds 20°C and wind speed exceeds 30km/h at the same time T and the same area A :

```
[MEASURE:*temperature]
-(agnt)->[SENSOR: x]
-(obj)->[TEMPERATURE: v > 20]
-(time)->[TIMESTAMP: T]
-(loc)->[AREA: Y]
```

And

```
[MEASURE:*windspeed]
-(agnt)->[SENSOR: y]
-(obj)->[SPEED: s > 30]
-(time)->[TIMESTAMP: T]
-(loc)->[AREA: Y]
```

A storm detection event is triggered as soon as the conjunction of these two facts exists within the result facts base (Figure 9). The analysis of the simulation results conceptualize this event using the knowledge associated with the situation description as follows:

```
[STORM:*storm]
-(loc)->[AREA: A]
-(time)->[TIMESTAMP: T]
```

This conceptual graph may be interpreted as follows: “a storm is detected at area A at time T ”.

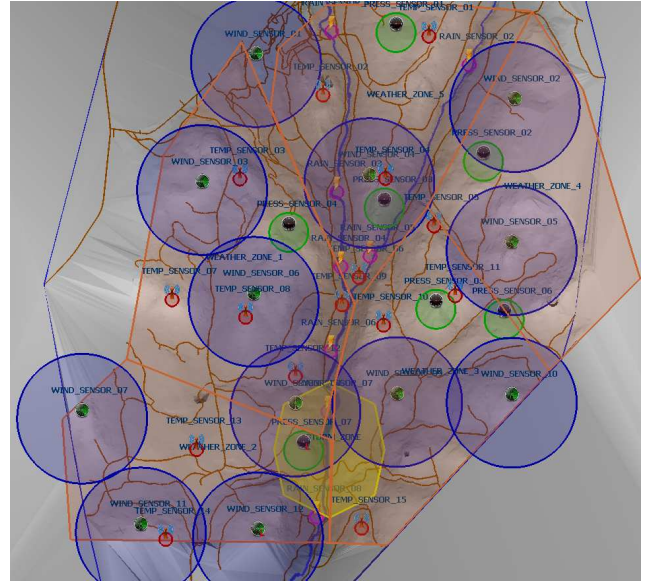


Figure 9: Idle sensors are waken up by those who perceives the storm agent (in yellow at the bottom of the figure).

V. DISCUSSION AND CONCLUSION

Our environment knowledge management approach is original at various aspects. First, a multi-agent geo-simulation model which integrates an informed virtual geographic environment populated with spatial agents capable of acquiring and reasoning about environment knowledge does not exist. Second, a formal representation of knowledge about the environment using CGs which leverages a semantically-enriched description of the virtual geographic environment has not yet been proposed. Third, providing agent with the capability to reason about a contextualized description of their virtual environment and phenomenon occurring within it during the simulation is also an innovation that characterizes our approach.

We are currently working on the automated assessment of different simulation scenarios f sensor web deployment using our qualitative knowledge processing and analysis module. Indeed, users usually need to analyze and compare various scenarios in order to make informed decisions. This task may be complex and effort and time consuming. However, since our framework already supports the analysis of the multi-agent simulation results, it is easy to extend it to automatically assess scenarios.

REFERENCES

- [1] I. F. Akyildiz, M. C. Vuran, and O. B. Akan. On exploiting spatial and temporal correlation in wireless sensor networks. In *WiOpt'04: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 71–80, 2004.
- [2] Mehdi Mekni and Bernard Moulin. Using Multi-Agent Geo-Simulation Techniques for Intelligent Sensor Web Management. In *Proceedings of the Fifth International Conference on Soft Computing*

- as Transdisciplinary Science and Technology (CSTST'08), 27-31 Septembre, Paris, France
- [3] G. W. Ng and K. H. Ng, "Sensor management - what, why and how," *Information Fusion*, vol. 1, pp. 67-75, 2000.
 - [4] Sébastien Paris, Mehdi Mekni, and Bernard Moulin. Informed virtual geographic environments: an accurate topological approach. In *GEOW'09: Proceedings of The International Conference on Advanced Geographic Information Systems & Web Services (GEOWS)*, Cancun, Mexico, February 1-7, 2009, pages 1 – 6. IEEE Computer Society Press, 2009
 - [5] Mehdi Mekni and Bernard Moulin. A survey on sensor webs simulation tools. In *International Conference on Sensor Technologies and Applications SensorComm'08*, volume 0, pages 574–579, Agay, France, 2008. IEEE Computer Society
 - [6] Ali, W.; Moulin, B.; Bédard, Y.; Proulx, M.J. & Rivest, S. (2007). Coupling Multiagent Geosimulation and Spatial OLAP for Better Geosimulation Data Analysis. *URISA Journal*, Vol. 19, No. 2, pp. 5-14
 - [7] Haddad, H. and Moulin, B. (2008). An Agent-Based Geosimulation Multidisciplinary Approach to Support Scenarios Evaluation in Dynamic Virtual Geographic Environments, *Proceedings of the Agent-Directed Simulation Symposium - ADS 2008*, pp. 53 – 60, April 14 - 17, 2008, Ottawa, Canada.
 - [8] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Srinivasan, "IrisNet: an architecture for a worldwide sensor Web," *Pervasive Computing*, IEEE, vol. 2, pp. 22-33, 2003.
 - [9] N. Athanasiadis, M. Milis, M. P.A., and S. C. Michaelides, "Abacus: A multi-agent system for meteorological radar data management and decision support," presented at Sixth Int'l Symposium on Environmental Software Systems (ISESS-05), Sesimbra, Portugal, 2005.
 - [10] P. K. Biswas and S. Phoha, "A middleware-driven architecture for information dissemination in distributed sensor networks," presented at Sensor Networks and Information Processing Conference, 2004.
 - [11] D. Moodley and I. Simonis, "New architecture for the sensor web: the SWAP framework," presented at ISWC 2006: 5th International semantic web conference, Athens, GA, USA, 2006.
 - [12] John Thangarajah, Lin Padgham, and James Harland. Representation and reasoning for goals in bdi agents. volume 24, pages 259–265, Los Alamitos, CA, USA, 2002. IEEE Computer Society Press
 - [13] Guy Mineau. Representing and enforcing interaction protocols in multi-agent systems: an approach based on conceptual graphs. In *IEEE/WIC International Conference on Intelligent Agent Technology*, pages 261 – 267, 13-16 Oct 2003.
 - [14] Michael Wooldridge and Nick Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2), 1995.
 - [15] H. Haddad and B. Moulin, "Using Cognitive Archetypes and Conceptual Graphs to Model Dynamic Phenomena in Spatial Environments", in *Conceptual Structures: Knowledge Architectures for Smart Applications*, LNAI 4604, eds. U. Priss, S. Polovina, and R. Hill, Berlin Heidelberg: Springer-Verlag, pp. 69–82, 2007.
 - [16] Sowa, J.F. (1984), *Conceptual Structures: Information Processing in Mind and Machine*, Massachusetts: Adison-Wesley
 - [17] V. Kaptelinin, B. Nardi, and C. Macaulay. *Methods & tools: The activity checklist: a tool for representing the "space" of context. interactions*, 6(4):27–39, 1999.
 - [18] Rachel Bellamy, Susanne Bødker, Ellen Christiansen, Yrjö Engeström, Virginia Escalante, Dorothy Holland, Victor Kaptelinin, Kari Kuutti, Bonnie A. Nardi, Arne Raeithel, James Reeves, Boris Velichkovsky, and Vladimir P. Zinchenko. *Context and Consciousness: Activity Theory and Human-Computer Interaction*. The MIT Press, first edition, November 1996.
 - [19] H. Haddad, "An Approach to Support Qualitative Analysis of COAs in a Virtual Geographic Environment, "What if" Analysis as an Example" (in French), PhD dissertation, March 2009, Department of Computer Science and Software Engineering, Laval University, Québec, Canada.
 - [20] Adil Kabbaj. Development of intelligent systems and multi-agents systems with amine platform. In Henrik Schärfe, Pascal Hitzler, and Peter Øhrstrøm, editors, *14th International Conference on Conceptual Structures: ICCS'06*, volume 4068 of *Lecture Notes in Computer Science*, pages 286–299. Springer, 2006.