Dynamic Sensor Management: Extending Sensor Web for Near Real-Time Mobile Sensor Integration in Dynamic Scenarios

Luiz Fernando F. G. de Assis[1]
Image Processing Division
Brazilian National Institute for Space Research (INPE)
São José dos Campos/SP, Brazil
email: luizffga@gmail.com

Lucas P. Behnck, Dionísio Doering,
Edison P. de Freitas and Carlos E. Pereira
Federal University of Rio Grande do Sul
Porto Alegre/RS, Brazil
email:{lucas.pluceno, dionisio.doering}@ufrgs.br,
edison.pignaton@inf.ufrgs.br, cpereira@ece.ufrgs.br

Flávio E. A. Horita and Jó Ueyama
Institute of Mathematics and Computer Science (ICMC)[1]
University of São Paulo (USP)
São Carlos/SP, Brazil
email:{horita, joueyama}@icmc.usp.br

João Porto de Albuquerque[1]
Centre for Interdisciplinary Methodologies
University of Warwick
Coventry, England
email:jporto@gmail.com

***Abstract*—Wireless Sensor Networks (WSN) are often composed of a wide range of sensor nodes, which may vary greatly in their type of hardware platform, as well as their sensing and mobility capabilities. The ability of a sensor to move is a particularly important feature in dynamic scenarios, since mobile sensors can fill the gap caused by the failures of those that are stationary, and thus extend the lifetime and span of a WSN. However, there remains the problems of intersensory communication in the field when integrating mobile sensors into the Sensor Web in dynamic scenarios since it does not have the necessary interoperability for automatically managing the different types of sensor data and activities involved in such scenarios. This paper tackles this problem by adopting an approach consisting of an enhanced messaging protocol and a dynamic sensor management component. In validating the proposal, two different realistic scenarios were simulated to evaluate the achieved results in terms of interoperability and performance. The results provided evidence that the proposal complies with Sensor Web standards as well as being suitable for near real-time data publication, and is thus able to support applications in dynamic scenarios.**

I. Introduction

Wireless sensor networks (WSN) that are entirely composed of stationary sensor nodes, have been found occasionally to be unsuitable for a wide range of applications since they still require great effort to increase their lifetime [1]. Many WSN nodes are unable to cover important regions due to failures and, as a result, the network cannot deliver its desired services [2]. For these reasons, mobile sensors have been used for overlapping the neighborhood of these unstable or problematic nodes when the application needs near real-time data. They can reach places where manned systems are unable to carry out safely, as well as can act either as a data relay node or data provider node, by collecting and forwarding data from isolated groups of nodes to connected parts of the WSN [2].

Although stationary and mobile sensors have a great value when working together, the dynamic features of their integration still pose some challenges. Stationary sensors cover an area until they fail, while mobile sensors cover areas as they move through them and disclose them when they move away [3]. This results in an intermittent connection between these nodes, which is difficult to manage due to the high dynamicity imposed on the data forwarding and the complexity of keeping track of the acquired data updates. In addition, such sensors have distinct protocols and interfaces, which can act as a paramount barrier to their control and access if no interoperability is considered.

Most of the existing approaches only carry out this integration using either proprietary mechanisms or a particular platform that targets specific hardware and software technologies. In order to improve the interoperability of this integration, an intermediary layer called Sensor Web was created to establish a clear and transparent communication between sensors and applications [4]. Open Geospatial Consortium (OGC[1]) has supported this approach by providing a set of standard activities, rules and guidelines for web-based sensors via the Sensor Web Enablement (SWE) initiative. However, one of the SWE limitations is that it cannot easily be modified and adapted to dynamic scenarios, specially problematic for mobile sensors, because a set of enhanced requirements must hold to control and access different sensor data flow rate and format [5], [6], keep the sensor updates available in near real-time [7], [8], and enable the exchange of large volumes of data between

[1] http://www.opengeospatial.org

applications and devices with limited capabilities [9], [10].

Dynamic scenarios are inherent to applications such as disaster management and surveillance. Focusing on disaster management, it is specially challenging and complex area due to several reasons, from which it is possible to highlight the following: (i) it needs flexible and adaptable "solution" techniques to integrate multiple information sources; (ii) it needs to handle requirements that change rapidly so that it can support decision-making in near real-time with accurate guidelines for fast and efficiently operations [11].

Dealing with all these issues is not a trivial task, since account must be taken of the following: (1) the different types of stationary sensors, which supply specific data from risky areas (e.g. the water level on the riverbed or volume of rain); (2) the different types of mobile sensors, which supply data from areas (e.g. hydrological) where appropriate stationary sensors have not been installed or areas with problematic or unstable sensors; (3) the management of all the heterogeneous data that might be provided in multiple formats (e.g. numerical, textual or images) at a different flow rates (e.g. periodically, sporadically or asynchronously); and (4) the management of sensor activities and updates in near real-time during hazard situations [12], [13] .

Since there is a lack of interoperability for automatically integrating an abundance of heterogeneous and near real-time sensors into the Sensor Web involved in dynamic scenarios such as disaster management, this work proposes an approach to support and overcome this need. The Sensor Observation Service (SOS) - an interoperable standard defined by the Open Geospatial Consortium (OGC) - is used to make data available for the Sensor Web in an interoperable fashion.

Thus, the main contributions made by this work are the following:

1) Defining a general purpose architecture to dynamically integrate mobile sensor nodes involved in dynamic scenarios into the Sensor Web in accordance with SWE standards;
2) Learning lessons from the application of the proposed approach in a realistic scenario of environmental monitoring systems for disaster management.

This paper is structured as follows. Section II examines the related works. Section III outlines the problem definition. Section IV describes the proposed approach. Section V provides details about the simulation scenarios. Section VI depicts the acquired results, while Section VII conducts a discussion about them. Finally, Section VIII concludes the paper and makes recommendations for future works.

II. Related Work

A. The use of mobile sensors in WSN

Existing projects aim to integrate mobile sensors to WSN for a variety of purposes. AWARE project [14] aims at integrating a sensor network of resource constrained ground nodes with mobile sensors, carried on the ground by Unmanned Ground Vehicle (UGVs) and in the air by Unmanned Aerial Vehicles (UAVs). This project handles different concerns from interoperability issues to cooperation protocols. The integration problem is handled by selecting certain nodes as gateways that are able to gather data from the stationary sensors. A middleware is responsible for the details involved in the sensor discovery and registration. Compared with our work, this study is not conducted in such general terms since the gateway-based approach is very specific to the types of sensors that are used. Moreover, they fail to carry out an easy integration with web-based applications, as we do by publishing data through the sensor web.

In [15], the concept of delay tolerant networks (DTN) is explored about how to support integration among stationary and mobile sensor nodes. The authors' potential-based routing mechanism gives a higher potential to data providers, an intermediary potential to the mobile sinks and a lower one to the central node. This potential-based mechanism is statically configured before the system runtime, while in our case, this assumption does not apply, since we employ a completely dynamic scenario in which no preconfigured parameter or setup is assumed. Moreover, their work does not provide data publication on the Web, while our proposal does.

PLANET project consists of a platform to enable adaptive deployments and operations of large-scale and complex mobile and stationary sensors cooperation. The platform has been validated in scenarios in which techniques of communication and cooperation pose challenges, as well as are very sensitive to the impact of pollution and with highly automated airfield[2]. EC-SAFEMOBIL project also aims to facilitate the cooperation, coordination and traffic control management of the UAVs deployment, by using methods and technologies of control and motion estimation[3]. A proposal for using UAVs to overlap disjoint segments and isolated partitions of WSN is presented in [16]. However, these proposals focus on the data collection without addressing the data delivery concerns towards publication on the Web.

B. Dynamically Integrating Sensors into the Sensor Web

Traditional WSNs are configured to gather data and transmit them to an uplink point, without any change in their behavior over time. In contrast, Sensor Web involves much more interaction between the sensors and their data since Sensor Web nodes act in line with the data that they and their neighbors have gathered. Most of the approaches aim to integrate sensors into the Sensor Web using their own mechanisms, which requires adaptation efforts since they do not consider interoperability and reuse properties.

Such limitations can be addressed by adopting a generic approach based on existing standard interfaces such as SWE standards and/or Devices Profile for Web Services(DPWS) [17], [18]. Although several applications rely on standard protocols, the communication between sensors and the integration of sensors in highly dynamic scenario in near real-time still is not fully addressed.

[2] http://www.planet-ict.eu/

[3] http://ec-safemobil-project.eu/

A message bus architecture containing a common communication infrastructure, a set of adaptable interfaces and a well-defined message protocol was built to ensure a semantically-enabled sensor plug-and-play via an automatic mediation between sensors and SWE standards [19]. However, this architecture, also known as Sensor Bus, does not take into account the following issues: (1) an automatically control and access of different sensor data flow rate and format [5], [6]; (2) keep the sensor status available on the Web in near real-time [7], [8]; and (3) the use of lightweight standard protocols to exchange information between devices and sensor applications when both large volumes of data are exchanged, and devices have limited bandwidth and processing capabilities [9], [10] .

Specifically for mobile sensors, existing approaches consider their management by improving SOS operations that act in a layer between the Sensor Web and Application layer [20], [21]. Since mobile sensors can send different types of data streams, the approach in this paper was designed to handle this kind of data although they require a great deal of data throughput and compression. The goal in this project is not to synchronize or transform mobile sensor data streams [22], but to carry out mobile sensor integration into the Sensor Web in a fast and interoperable way considering scenarios where sensors: (1) create and retransmit heterogeneous data flow rate and format, and (2) are inserted and removed all the time inside a network.

III. Problem Definition

The major goal of this paper is to present a contribution on the dynamically integration of mobile sensors into the Sensor Web. This is performed by using the ability of these nodes to move throughout uncovered areas of a sensor field and to make their heterogeneous data available on the Web in near real-time. The sensor field in which these sensors are inserted and removed at any time can be seen in Figure 1.

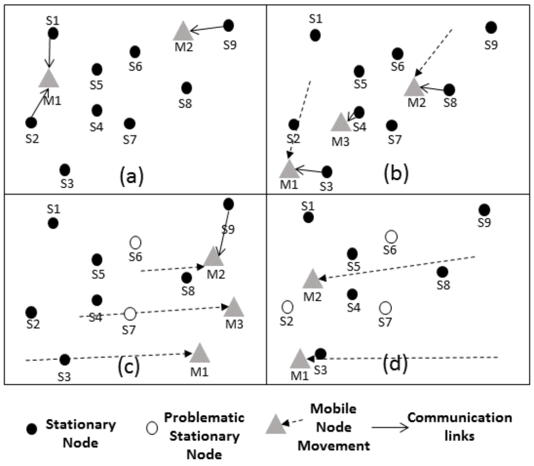

Fig. 1. Example of four subsequent scenarios of the sensor field.

The sensor field is modeled as a time-series of graph $G = \{G^0,...,G^n\}$. $G^t = (S^t,C^t)$, where $S^t = \{s_0,...,s_n,m_0,...,m_p\}$ is a set of stationary sensors s_i and mobile sensors m_j working in the sensor field at a time t, while C^t is a set of communication links between the stationary sensors $s_i \in S$ and a mobile sensors $m_j \in S$ at a time t.

Figure 1 depicts four scenarios (a), (b), (c) and (d) representing four subsequent instants $T1$, $T2$, $T3$ and $T4$ ($G = \{G^{T1}, G^{T2}, G^{T3}, G^{T4}\}$). In scenario (a), the mobile sensor m_1 is connected to the stationary sensors s_1 and s_2, while m_2 is connected to s_9; in scenario (b), a mobile sensor m_3 starts working and is connected to s_4, m_1 is connected to s_3, while m_2 is connected to s_8; in scenario (c), two stationary sensors (s_6 and s_7) stop working, while m_2 is connected to s_9 again; in scenario (d), s_2 and m_3 stop working.

The communication links between the sensors will depend on the mobile sensor planning path. For each mobile sensor m_j, its path consists of an array $m_j.a$ containing Points of Interests (POIs) o_k in a visiting order ($m_j.a = [o_1,...,o_q]$, where $o \in R^2$, q relates to the amount of POIs, o_1 is the first point to be visited and o_q is the last one). After a mobile sensor m_j has ensured the coverage its planning path $m_j.a$, the graph G^t should contain all the communication links expected when the path was modeled considering operating range of the stationary sensors.

The sensor field containing sensors with different trajectories and configurations over time, several types of hardware that hamper their effective management and unpredictable occurrence of events (POIs) represent a dynamic scenario. This dynamic scenario is represented as an ad hoc network monitoring the occurrence of events, in which the involved sensors act both as a data providers and as data forwarders.

IV. Proposed Approach

An automated set of tasks is required to integrate mobile sensors into the Sensor Web. These tasks vary from access to control of the wide range of data and activity diversity that have to be handled. One of the reasons for that is due to the fact that before a sensor starts sending updates and observations, it needs to be registered on the Sensor Web in accordance with its specification. For this, it is necessary to get the capabilities of the Sensor Web service, and create a sensor description standard encoding depending on the sensor specification. This encoding has to be sent embedded with a sensor registration into the service. After this sensor registration, the sensor is able to insert both its observation and updates, by initially creating an observation standard encoding and updating its status inside the sensor description respectively. Then, again it is necessary to get the capabilities of the service to either insert the new observation or update the sensor description. Figure 2 aims to represent the required tasks to integrate mobile sensors acting both as a data providers and as data forwarders into the Sensor Web.

However, it is difficult to accomplish the above mentioned tasks in dynamic scenarios since a large amount of heterogeneous data format and flow rate is exchanged by different sensor specifications, specially those that both have limitations and change their states very frequently. In this sense, this section aims at describing the proposed approach to integrate mobile sensors into the Sensor Web in dynamic scenarios, by

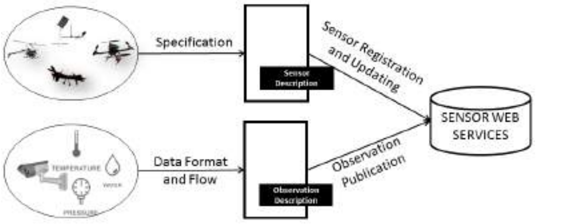

Fig. 2. Registering, Updating and Publicating Sensor Information into the Sensor Web.

presenting an enhanced message protocol (subsection IV-A) and a dynamic sensor management component (subsection IV-B). Figure 3 presents an overview of this whole process.

A. Message Protocol

The message protocol extends an existing protocol [23] by seeking to improve communication between sensors, keep the sensor status available in near real-time and enable sensors with limited capabilities to exchange a large number of data. The message protocol contains seven different messages that represent the following activities within the Sensor Web: 1) registering new sensors, 2) publish new observations, 3) start monitoring, 4) stop monitoring, 5) sleep monitoring, 6) wake up monitoring and 7) updating the current sensor position. Each message has a syntax, which is detailed in Table I.

TABLE I
MESSAGE PROTOCOL.

Message	Syntax
Register Sensor	SensorRegister*<sensorID >
Publish Observation	PublishObservation*<sensorID>* <timeStamp>*<location>*<value>
Start Monitoring	StartMonitoring*<sensorID>*<timeStamp>*<location>
Stop Monitoring	StopMonitoring*<sensorID>*<timeStamp>
Sleep Monitoring	SleepMonitoring*<sensorID>*<timeStamp>
Wake up Monitoring	WakeUpMonitoring*<sensorID>*<timeStamp>*<location>
Current Position	CurrentPosition*<sensorID>*<timeStamp>*<location>

The first message contained in the Table I is about registering new sensors on the Web. This message is responsible for carrying out important information to create the description of the sensor to be inserted into the Sensor Web. This description is a *SensorML*[4], a standard model that was designed to discover sensors and observations. Instead of providing a description of the hardware, *SensorML* provides a functional model of these sensors by handling each of its components as a process. Without *SensorML*, the observations from sensors that have not been registered are not published. Since this message includes only the sensor identifier, it enables a fast and efficiently sensor integration due to the lightness of the message.

The service responsible for storing the sensor data into the Sensor Web is the Sensor Observation Service (SOS). SOS is an API to manage deployed sensors and recover sensor data related to their observations that contains a repository and a set of operations. Its main goal is to enable the access to the sensor observations in a standard way. For example, to insert a new sensor into the SOS, it is necessary to use a *InsertSensor* operation using *SensorML.*

The second message of the protocol aims at publishing new observations from sensors already registered. An observation is an act of observing a property such as temperature, pressure, light, represented by numeric, text or binary values depending on the property. This observation is encoded as a *Observations & Measurements* (O&M[5]), which aims to provide a very general model that allows the packaging and linking of observations of different data format and structures from a wide variety of sensors types. It is important to emphasize that the dynamic sensor management component aid in the treatment of several type of observations. In case, mobile sensors provide images, the dynamic sensor management component can enable the conversion of the image into a binary chain and thus making them available on the Web. To make the observations from registered sensors available into the SOS, it is necessary to use a *InsertObservation* operation using *O&M*.

All the other messages are about updating the sensor status. For example, sensors can have different mobilities (stationary or mobile), and in case they are mobile, they can have different positions (latitude, longitude) and states (active or not active) over time (timestamp). In order to make such sensor updates available on the Web in near real-time, every time a sensor changes, the update needs to be published. This is normally performed by updating the description of an existing sensor using a *UpdateSensorDescription* operation by means of a *SensorML*. This is the key to keep the sensor status (mobility, activity, timestamp and location) up-to-date.

B. Dynamic Sensor Management

In a real-world scenario, when either a stationary sensor is deployed or a mobile sensor flies for the first time, these sensors need to be registered in the Sensor Web. For this, they need to send a *RegisterSensor* message and retransmit to either another sensor nearby or directly to the dynamic sensor management component, which interprets and converts the data into the respective sensor activity.

In addition, sensors may fail e.g., due to the lack of battery, then they need to be recharged (in cases in which the recharging is possible). In order to overcome this problem, the sensors need to send a command indicating that they are in a sleeping mode (*SleepMonitoring* message). When the problem of sleeping is solved, the sensors are able to start monitoring again (*WakeUpMonitoring* message). In case, a sensor has just been registered or is beginning a new mission, it starts monitoring an area (*StartMonitoring* message). Conversely, as

[4] http://www.opengeospatial.org/standards/sensorml

[5] http://www.ogcnetwork.net/OM

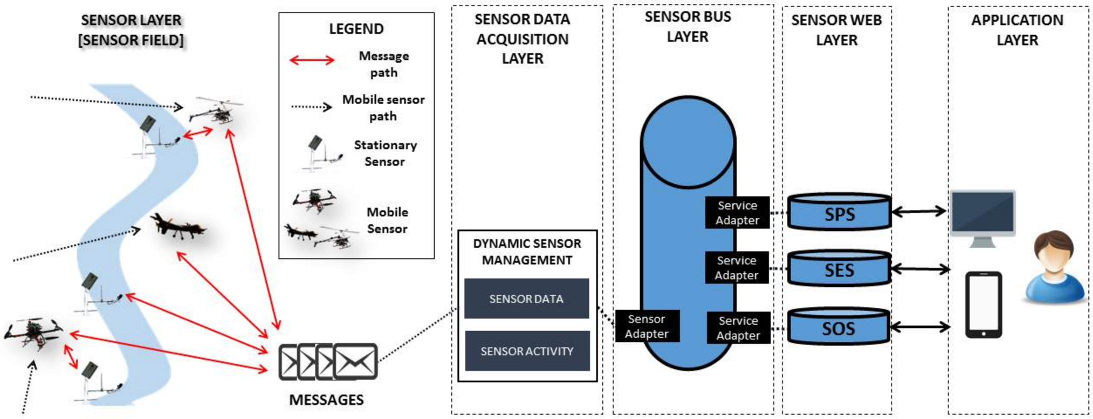

Fig. 3. A diagram of the approach overview.

it stops monitoring, when it completes the mission, it sends a *StopMonitoring* message. Lastly, mobile sensors can fly over an area, thus keeping their track available on the Web can help stakeholders to quickly make decisions about new tracks, for example. These sensors use the *CurrentPosition* message to make that possible.

For these reasons, controlling and accessing heterogeneous sensor data, keeping the sensor updates available in near real-time, and exchange large volumes of data are not only interesting features but also necessary, since critical applications require an abundance of on-the-fly sensor data. These components and how they relate to each other are depicted in the dynamic sensor management represented in Figure 4.

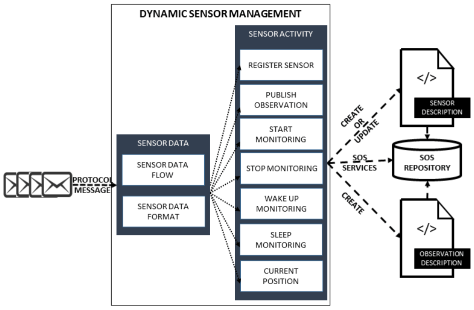

Fig. 4. Dynamic Sensor Management Architecture of the approach.

V. Simulation Setup Scenarios

Due to the high cost and complexity of deploying real sensors, simulation was the chosen method to evaluate the proposal. The performed simulations are based on 2 scenarios that consider a real-world application of a wireless stationary sensor network to monitor a river water level [24] and mobile sensors that are able to perform cooperative coordinated missions [25]. The stationary sensors have a small physical size and a low-cost long-range communication device, while the mobile sensors are small autonomous aircrafts (UAVs) containing a Mission and Vision Control Module composed of additional hardware as shown in Figure 5.

The performed simulations take into account important factors of supporting disaster management systems. They difficult the collection of updated information about the current state of rivers, as well as hamper the effectiveness of the risk management and resilience.

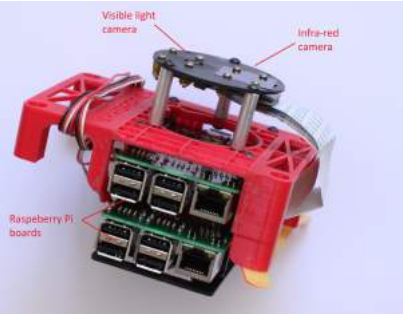

Fig. 5. A Mission and Vision Control Module composed of two Raspberry Pi boards with infrared and visible light cameras.

The UAVs used in the simulations are based on the APM platform[6], which uses an open source operating system and supports different types of aircraft (either with rotary or fixed wing). The platform supports communication with external devices through the Mavlink[7], which is a communication protocol used for communications among UAVs with a Vision and Mission Control module. APM platform also contains an open source flight simulator software that was used in this project to evaluate the path planning algorithm executed in the Mission Control Module.

A typical flight mission corresponds to the inspection of critical POIs by one or more UAVs. Each aircraft can carry different types of sensor devices (e.g., either an IR or a visible light camera) which are better suited to a given type of POI. The shortest paths for the UAVs, which takes into account their specific inspection capabilities, are calculated by executing a Simulated Annealing (SA) algorithm in a single Mission and

[6] http://ardupilot.com/
[7] http://qgroundcontrol.org/mavlink/

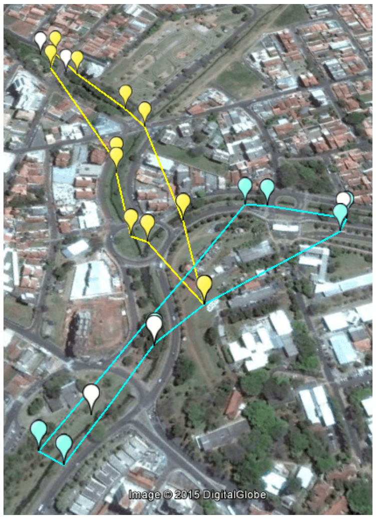

Fig. 6. Simulation Setup Scenario One.

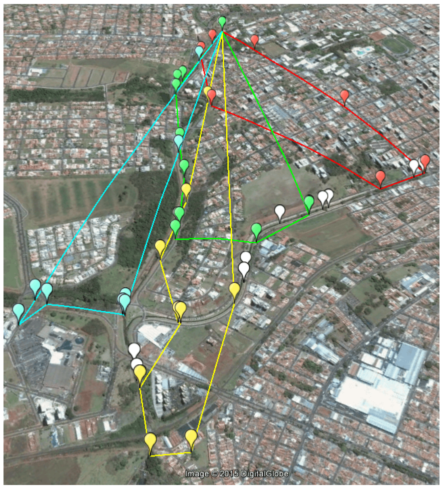

Fig. 7. Simulation Setup Scenario Two

Vision Control Module. This procedure considers the path planning optimization as a multiple problem for the Traveling Salesman Problem (TSP) based on [26]. The algorithm considers a penalty factor, which is calculated for paths in which the corresponding UAV visits an unsuitable POI, i.e., a POI for which its sensor device is not the most suitable. This procedure takes the solutions of the SA algorithm to regions where the UAV specialty and POI type are likely to match.

The first simulated scenario contained two mobile and five stationary sensor nodes, which provided data and changed their status and position over time (Figure 6). In the second scenario the number of mobile and stationary sensor nodes were doubled (Figure 7). The white markers in the Figures represent simulated stationary sensors, while the colored markers represent each POI of UAVs path planning. The details of the simulations are depicted in Table II.

TABLE II
PARAMETERS IN SIMULATION.

Scenario 1	
Parameters	**Values**
# mobile sensors	2
# stationary sensors	5
simulation area (km^2)	0.1
mobile sensor speed (km/h)	6.5
# POIs	17

Scenario 2	
Parameters	**Values**
# mobile sensors	4
# stationary sensors	10
simulation area (km^2)	1.98
mobile sensor speed (km/h)	6.5
# POIs	36

Both scenarios contained different number of stationary and mobile sensors to facilitate the evaluation of both the latency and scalability of the proposal. During the simulations, mobile sensors of both scenarios published observations at every POI inspection, while stationary sensors publish at random time intervals (average mean of 120 seconds and standard deviation of 60 seconds). This helps to represent the dynamic scenario found in disaster situations.

In this work, SOS framework[8] was considered as an instance of Sensor Web standards. Its advantages are that it has an open source code (easily adapted) and it provides all of the required SOS operations, unlike other existing implementations.

VI. RESULTS

The conducted simulation involves: (1) the deployment of sensors in a dynamic scenario to simulate realistic conditions such as those involved in disaster management; (2) the performance (latency and scalability) analysis of the messages sent by the sensors to check near real-time data publication; and (3) the adoption of SOS standards to achieve genericity in the approach by handling sensor data in an interoperable way.

One of the means to evaluate interoperability is through the syntactical interpretation and use of a message. Since the proposed message protocol is easily to be interpreted due to its simplicity and the low cost of a lexical analysis, when a message either misses a field or is incorrect, the dynamic sensor management component notices this as soon as it arrives. Then, the response to this request will point out that there is a mistake. However, if the message has a consistent structure it will be transmitted to the dynamic sensor management component. In addition, the fact of using an established set of standards such SOS services help to improve the interoperability analysis of the proposed approach. Considering the Sensor Layer, to exchange a large number of data using the message protocol, sensors need less complexity and size compared to SWE standards (Table III).

TABLE III
SIZE COMPARISON.

message protocol	size (bytes)
register sensor	25
publish observation	60
stop monitoring	37
start monitoring	49
sleep monitoring	38
wake up monitoring	50
current position	49

SWE standard	size (bytes)
insert sensor	3,431
insert observation	1,498
update sensor description	2,955

[8] http://52north.org/communities/sensorweb/sos/

To evaluate the performance, the elapsed time to receive, interpret and convert the sensor data contained within the message into a Sensor Web standard and publish them on the Web was calculated. In this evaluation, the SOS elapsed time ΔT_{SOS} (SOS latency) was omitted since the interest was in evaluating only the time latency of the components implemented in this work to process the data, and not the one related to the existing SOS 2.0 implementations that are involved to perform the required operations.

The SOS latency is defined as the elapsed time between the request sent to the SOS (SOS_0) and the publication of the sensor data into the SOS repository (SOS_f). This time depends on the implementation of the SOS that is used, which is why it is not being considered in the performed evaluation. Therefore, the following mathematical formulation explains the performed evaluation: 1 represents the SOS time that is not being considered, while 2 represents the initial time when the dynamic sensor management component received the message (DSM_0) and the final time when the dynamic sensor management published the sensor data into the Sensor Web (DSM_f).

$$\triangle T_{SOS} = SOS_f - SOS_0, \tag{1}$$

$$\triangle T_{Approach} = (DSM_f - SOS_f) + (SOS_0 - DSM_0), \tag{2}$$

While the tests were being conducted, the simulated sensors were sending all kinds of messages as a realistic timeline of their operational behaviour in a dynamic scenario (as can be observed in Figures 8 and 9). Once the sensors are in time-of-flight or deployed for the first time, they send a *RegisterSensor* message. After that, they start sending *StartMonitoring*, *SleepMonitoring*, *WakeUpMonitoring* and *StopMonitoring* messages in an interchangeable way in accordance with a plausible sequence. The performed analysis also helped to observe no great impact of the different amount of sensors on the acquired results in both scenarios. However, as has been observed, since *PublishObservation* message requires further processing, it takes more time to be available, while the others are able to perform better near real-time data publication.

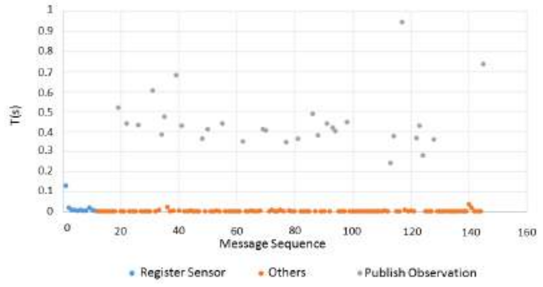

Fig. 8. Scenario One - Sent Messages

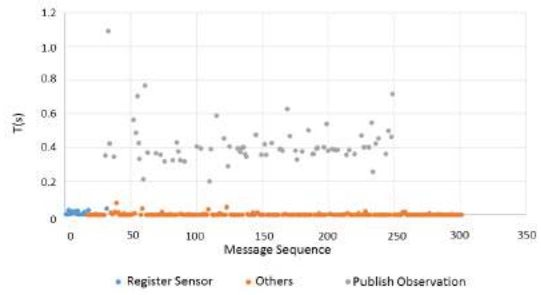

Fig. 9. Scenario Two - Sent Messages

VII. Discussions

In the above described simulations, stationary and mobile sensors deployed in a given sensor field can send messages to each other so that their data can be published. Whenever a sensor message is received by the dynamic sensor management component, a standard is interpreted and used as a scalable container interface within the execution program. Although this standard template must be initially saved and operations of output/input are required, the proposed approach achieves a low latency since no entire format standard should be created every time during the execution program.

Instead of having intensive work on the programming to control a unit software of a wide variety of sensor devices [5], [6], this project consider a generic mechanism to overcome this need. In addition, a small size and less complex message protocol helps to improve the performance of the sensor network when mobile sensors with processing capabilities limitations are deployed avoiding dependence on a stable network, with no Internet connection.

This approach is not concerned with any enhancement of SOS operations in the interaction between the Sensor Web and Application layer [20], [21]. Instead, it acts at a lower level of the Sensor Web layer and seeks to create mechanisms to improve the interoperability by an on-the-fly integration of the sensor data into the Sensor Web. The evaluated scenarios are more complex and dynamic than those that existing approaches can handle [23]. This is based on an assumption that all sensors can operate without any pre-existing infrastructure.

Unlike [14], [15], this approach automatically makes sensor data available on the Web and enables access to the current sensor updates in an interoperable way, publishing them on the Web. Moreover, this approach was designed to handle heterogeneous data without need for any complex preprocessing techniques. Although mobile sensor data streams are complex to manage [22], a fast and interoperable way in this project enables its management in scenarios in which sensors are inserted and removed all the time.

VIII. Conclusion

The main contributions made by this work are the support for near real-time mobile sensor data integration into the Sensor Web, considering the use of interoperable and reusable mechanisms, as well as the ability of sensors to move in dynamic scenarios and act either as a data relay or data provider. Real-world application scenarios of environmental monitoring system for disaster management were considered for the proposal evaluation.

Since the protocol is easy to interpret and convert, it helps to provide a standardized way of interoperability, and consequently a low latency to publish data on the Web, which is crucial for applications demanding near real-time data publication such as disaster management. The dynamic sensor management component generally takes less than one second to publish sensor data, which means the proposed approach can manage the heterogeneous sensor data flow rate and format, as well as can keep and publish sensor updates in dynamic scenarios with a very short delay.

The scalability of the proposal could also be evaluated, which was done by varying the number of sensors in the evaluated scenarios. Future works aim to integrate mobile devices such as smart phones acting as sensors, exploring people roaming. This could increase the diversity of the data sources besides exploring unexpected patterns according to people spontaneous movements. Moreover, other algorithms can be used by the mobile sensors to overlap the stationary sensors within a network.

Acknowledgment

We would like to thank São Paulo Research Foundation (FAPESP) by the grant #13/16202-1 and #14/08398-6. We also would like to thank Rio Grande do Sul Research Agency (FAPERGS) by the grant #2240-2551/14-2 and the National Council for Scientific and Technological Development (CNPq) by the grant #477499/2012-0. Jó Ueyama would also like to thank Office of Naval Research Global.

References

[1] R. Patel and B. Kaushik, "Rifmas: River flow management system using wireless sensing agents," in *Advance Computing Conference, 2009. IACC 2009. IEEE International*. IEEE, 2009, pp. 853–857.

[2] E. Pignaton de Freitas, T. Heimfarth, I. F. Netto, C. E. Pereira, A. M. Ferreira, F. R. Wagner, and T. Larsson, "Handling failures of static sensor nodes in wireless sensor network by use of mobile sensors," in *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on*. IEEE, 2011, pp. 127–134.

[3] B. Liu, O. Dousse, P. Nain, and D. Towsley, "Dynamic coverage of mobile sensor networks," in *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 2. IEEE, 2013, pp. 301–311.

[4] A. Bröring, J. Echterhoff, S. Jirka, I. Simonis, T. Everding, C. Stasch, S. Liang, and R. Lemmens, "New generation sensor web enablement," in *Sensors*, vol. 11, no. 3. Molecular Diversity Preservation International, 2011, pp. 2652–2699.

[5] J. Geipel, M. Jackenkroll, M. Weis, and W. Claupein, "A sensor web-enabled infrastructure for precision farming," in *ISPRS International Journal of Geo-Information*, vol. 4, no. 1. Multidisciplinary Digital Publishing Institute, 2015, pp. 385–399.

[6] N. Chen, X. Zhang, Z. Chen, and S. Yan, "Integrated geospatial sensor web for agricultural soil moisture monitoring," in *Agro-Geoinformatics (Agro-geoinformatics), 2015 Fourth International Conference on*. IEEE, 2015, pp. 28–32.

[7] C.-H. Park, J. Cho, and D.-H. Kim, "Sensor web for supporting mobility in sensor networks," in *Computer Applications for Web, Human Computer Interaction, Signal and Image Processing, and Pattern Recognition*. Springer, 2012, pp. 203–208.

[8] M. Bakillah, S. H. Liang, A. Zipf, and M. A. Mostafavi, "A dynamic and context-aware semantic mediation service for discovering and fusion of heterogeneous sensor data," in *Journal of Spatial Information Science*, no. 6, 2015, pp. 155–185.

[9] A. Tamayo, C. Granell, and J. Huerta, "Using swe standards for ubiquitous environmental sensing: a performance analysis," in *Sensors*, vol. 12, no. 9. Molecular Diversity Preservation International, 2012, pp. 12 026–12 051.

[10] V. Arnaboldi, M. Conti, F. Delmastro, G. Minutiello, and L. Ricci, "Sensor mobile enablement (sme): A light-weight standard for opportunistic sensing services," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*. IEEE, 2013, pp. 236–241.

[11] M. Tropmann-Frick and T. Ziebermayr, "Generic approach for dynamic disaster management system component," in *Database and Expert Systems Applications (DEXA), 2014 25th International Workshop on*. IEEE, 2014, pp. 160–164.

[12] S. Jirka, A. Bröring, and C. Stasch, "Applying ogc sensor web enablement to risk monitoring and disaster management," in *GSDI 11 World Conference, Rotterdam, Netherlands*, 2009.

[13] N. Chen, K. Wang, C. Xiao, and J. Gong, "A heterogeneous sensor web node meta-model for the management of a flood monitoring system," in *Environmental Modelling & Software*, vol. 54. Elsevier, 2014, pp. 222–237.

[14] A. Tuysuz Erman and P. Havinga, "Data dissemination of emergency messages in mobile multi-sink wireless sensor networks," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2010 The 9th IFIP Annual Mediterranean*, June 2010, pp. 1–8.

[15] H. Ochiai, H. Ishizuka, Y. Kawakami, and H. Esaki, "A field experience on dtn-based sensor data gathering in agricultural scenarios," in *Sensors, 2010 IEEE*, Nov 2010, pp. 955–958.

[16] T. Heimfarth and J. P. De Araujo, "Using unmanned aerial vehicle to connect disjoint segments of wireless sensor network," in *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*. IEEE, 2014, pp. 907–914.

[17] D. M. Beder, J. Ueyama, J. P. de Albuquerque, and M. L. Chaim, "Flexft: A generic framework for developing fault-tolerant applications in the sensor web," in *International Journal of Distributed Sensor Networks*, vol. 2013. Hindawi Publishing Corporation, 2013.

[18] S. H. Liang and C.-Y. Huang, "Geocens: A geospatial cyberinfrastructure for the world-wide sensor web," in *Sensors*, vol. 13, no. 10. Multidisciplinary Digital Publishing Institute, 2013, pp. 13 402–13 424.

[19] C. Malewski, A. Broring, P. Maué, and K. Janowicz, "Semantic matchmaking & mediation for sensors on the sensor web," in *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 7, no. 3. IEEE, 2014, pp. 929–934.

[20] C. Stasch, A. Bröring, and A. C. Walkowski, "Providing mobile sensor data in a standardized way-the sosmobile web service interface," in *11th AGILE International Conference on Geographic Information Science*, 2008.

[21] R. Müller, M. Fabritius, and M. Mock, "An ogc compliant sensor observation service for mobile sensors," in *Advancing Geoinformation Science for a Changing World*. Springer, 2011, pp. 163–184.

[22] M. Rieke, T. Foerster, and A. Bröring, "Unmanned aerial vehicles as mobilemulti-sensor platforms," in *The 14th AGILE International Conference on Geographic Information Science.*, April 2011.

[23] A. Bröring, T. Foerster, S. Jirka, and C. Priess, "Sensor bus: An intermediary layer for linking geosensor networks and the sensor web," in *COM.Geo 2010: 1st International Conference on Computing for Geospatial Research and Applications*, Washington DC, USA.

[24] F. E. Horita, J. P. de Albuquerque, L. C. Degrossi, E. M. Mendiondo, and J. Ueyama, "Development of a spatial decision support system for flood risk management in brazil that combines volunteered geographic information with wireless sensor networks," in *Computers & Geosciences*, vol. 80. Elsevier, 2015, pp. 84–94.

[25] D. Doering, A. Benemann, R. Lerm, E. P. Freitas, I. Mller, J. M. Winter, and C. E. Pereira, "Design and optimization of a heterogeneous platform for multiple uav use in precision agriculture applications," in *World Congress*, vol. 19, no. 1, 2014, pp. 12 272–12 277.

[26] C.-H. Song, K. Lee, and W. D. Lee, "Extended simulated annealing for augmented tsp and multi-salesmen tsp," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 3. IEEE, 2003, pp. 2340–2343.