Performance Evaluation on OpenGIS Consortium for Sensor Web Enablement Services

Thiago C. Tavares*†, Regina H. C. Santana*, Marcos J. Santana*, Julio C. Estrella*
*Institute of Mathematics and Computer Science, ICMC/USP
{thiagocp,rcs,mjs,jcezar}@icmc.usp.br
†Federal Institute of Education, Science and Technology of Southern of Minas Gerais, IFSULDEMINAS
{thiago.tavares}@ifsuldeminas.edu.br

***Abstract*—The aim of this paper is to describe a performance evaluation of the interface model of Sensor Web Enablement, especially highlighting the Sensor Observation Service, Sensor Event Service and Sensor Instance Registry. These standards provide a transparent and interoperable way to access data measured by sensors. Studies found in the literature do not treat a performance evaluation on highlighted services in a detailed way. So, the performance evaluation in our study considers several factors that can influence the access time on these services. The results show an important influence of different filter types in the service response times. The result analysis demonstrated that the implementation of application that uses these services should be careful on use of these filters, as, due their definition, the performance of these applications can decrease.**

***Keywords*—Sensor Networks; Service-Oriented Architecture; Web services.**

I. Introduction

A sensor network is composed of sensors that monitor one or a combination of physical data in which the results are sent to an application or final user. It is used in a wide range of monitoring and tracking applications. Furthermore, the breakthrough of their applications has been possible due to the improvement and feasibility of the sensor platforms' cost [1][2]. However, a major challenge in the use of these sensor networks is the feasibility of managing them and providing the necessary information for the use in different applications. On the one hand, there is the infrastructure composed by the sensors and usage strategies of them, as well as the information obtained by them. On the other hand, there are applications or observers who should receive the information and process them. Besides, the sensor networks must also have a communication infrastructure to provide data exchange, between sensors, as well as between network and the observers.

In order to enable the use of sensor networks, it is possible to develop a middleware that provides the tools needed to manage them. Therefore, the literature presents a number of proposals and implementations of middleware used to facilitate the information access provided by these networks regarding the installation, maintenance and execution of applications [3].

One approach that has been proposed in the literature considers the sensor network as a Web Service, i.e., some specifications and languages are used to make an abstraction of the complexity of the sensor system [4]. The abstraction mechanisms provide a standardized interface to access the information following an approach of the Service-Oriented Architectures (SOA). Middlewares that use the SOA concepts have been widely discussed in the literature [5][6]. The OpenGIS Consortium (OGC), a consortium of over 400 companies and academic institutions, has been working on the definition of standards, specifications and programming frameworks in order to use them in the development of sensor networks available as services [7]. In this context, it has been proposed the SWE (Sensor Web Enablement), which is composed of a set of standards, protocols and interfaces that enable the information obtained by the sensor networks to be available through Web Services, following the principles of service-oriented architectures.

Therefore, it is possible to highlight the SOS (Sensor Obervation Service), SES (Sensor Event Service) and SIR (Sensor Instance Registry) services, among the set of interfaces proposed by the SWE. They perform the functions of obtaining observations, alerting and search of sensors, respectively. The SOS is one of the most studied service in the literature, regarding the studies that focus on qualitative and quantitative evaluations on context of SWE service interfaces [8][9][10]. However, there is a gap in relation to a more complete performance evaluation that takes into account other important services, such as the SES and SIR. Thus, this paper presents a performance evaluation that analyzes in detail the main interfaces, defined by SWE, for the access to sensor systems.

This paper is organized as follows: Section II discusses the standards defined in the SWE. Section III presents some works that are related to the one proposed in this paper as well as the gap in the area. Section IV discusses the results of the performance evaluation of SWE services presenting the design of experiments and the evaluation scenario used to perform them. Finally, Section V presents the conclusions and future works that could be developed from the study discussed in this paper.

II. Background

As shown in Section I, the OGC is the creator and maintainer of SWE. Since 2003, some work groups have developed and discussed a set of standards that enable the use of sensors exposed through the Web. In this context, sensors are defined as devices that are discovered and accessed through a standardization of protocols and interfaces. They are infrastructures that enable the integration of sensing resources where applications or users can discover, access, modify and register services of alert and sensing, in a standardized way.

 ISBN: 978-1-61208-305-6

Therefore, the WWW provides an infrastructure that enables the sharing of data measured into a sensor system in a well-defined way, abstracting the complexities of the lower layers of the sensing platforms. For example, the standards defined by SWE abstract the details of communication protocols, the hardware architecture and programming languages used in sensor platforms. So, this abstraction facilitates the development of applications. Besides, it allows the developer to concentrate on the logic of its application, not in the details of communication and programming of sensing platforms.

SWE standards are under development and some updates were published in 2012. Bröring et al. [11] presents an overview of these standards and their recent advances and updates. According to the authors, the SWE standards are divided into two informal subgroups: information model and interface model. The former includes data models and encodings used for data representation standards, while the latter comprises different interface specifications of Web services.

Moreover, the information model includes a set of standards that define data models to be used to code the observations of the sensors as well as their metadata. Aiming this, the SWE contains two main specifications: Observation & Measurements (O&M) and the Sensor Model Language (SensorML). The latter specifies a model and a XML codification for describing sensors. In this language, it is mainly defined the location, input and output data, and the phenomena that are observed by sensors. On the other hand, the standard Observation & Measurements defines a framework for the description of the observations made by the sensors. In addition to the standards, other patterns were also defined: the data model (SWE Common) that provides a low-level model for data exchange related to sensors and it is used by several other patterns of SWE. The SWE Common was previously inserted into the SensorML specification, and nowadays, it is available separately as SWE Common 2.0 specification [7].

In turn, the interface model is used to provide a data access mechanism and measurements performed by sensors via a Web service. Several services were defined in the SWE standards, among them it is possible to highlight the SOS, SES and SIR.

A. SOS

The SOS allows obtaining the measured data by the sensors. Besides, it is important to mention that the observations returned by SOS are encoded within the standard O&M. The SOS standard provides an interface to manage and obtain metadata and observations of heterogeneous sensor systems. Thus, this interface defines how the descriptions and observations of sensors are accessed through an interoperable manner. Among the several possible operations by the SOS interface, the following stand out [12]:

- **GetCapabilities**: gets information about the service.
- **DescribeSensor**: gets the description of a sensor or sensor system.
- **GetObservation**: gets a set of observations that may have different filters (time, location, etc.).
- **RegisterSensor**: allows adding new sensors or sensor system in the service.
- **InsertObservation**: allows the addition of new observations for a particular sensor.

B. SES

The SES allows the users registration and/or applications in an alert system. In this case, the user and/or application make the register in the service and receive notifications of it when the criteria for triggering these notifications are met. The SES clients register filters that are used to define the criteria of triggering alerts in a sensor network. Thus, the SES service operates as a Broker of information that carries the mediation between sensor networks and their clients. In general, the notifications made by SES are encoded in the O&M standard. Three levels of filters can be defined in the SES [13]:

- **Level 1**: allows the registration of a filter that sends alerts via an XPath expression.
- **Level 2**: allows the registration of temporal filters, of location and comparison through FES specification (Filter Encoding Specification).
- **Level 3**: allows the determination of filters with multiple patterns. In this case, it is possible to determine a composition of various filters in the emission of alerts.

C. SIR

The SIR provides an interface for managing metadata of sensors. These metadata are encoded through SensorML language. Furthermore, several types of search requests can be submitted to the SIR service. For example, searches can be performed using criteria such as type of service (SOS or SES), types of observed phenomena, location, description, etc. Additionally, it is possible to update sensor information and insert status information of a sensor characteristic as the battery status [14]. The SWE also provides an interface called the SOR for the management of the semantics of the phenomena observed by the sensors. However, this service is not addressed in the study presented in this paper. Section III presents some related works and the gaps identified in these studies.

III. RELATED WORK

This section aims to present some works related to qualitative and quantitative evaluations in the context of the SWE standards. The work presented by McFerren et al. [8] discusses implementations of the Observation Service Sensor highlighting features such as easy installation, documentation quality, and completeness of implementation in relation to the standard definitions. The authors consider four types of implementations: 52°North Initiative, PySOS, MapServer and Deegree SOS and they do not consider any quantitative analysis such as a performance evaluation of the implementations concerning the functionalities provided by them.

Moreover, Poorazizi et al. [10] presents a complementary study of the work found in [8]. The performance of several implementations of SOS services. The authors present a review of SOS considering different filters of data acquisition such as number of sensors, location, and time. The study has considered three of the four implementations discussed [8] (52°North, Deegree, and MapServer). Furthermore, the performance analysis took into account two characteristics: response time and size of documents returned by the service.

In turn, Tamayo et al. [15] presents a performance evaluation of SWE standards in a mobile computing environment. In this study, the authors evaluated the performance of different Smartphone in the document processing with sensor observations obtained through SOS. Besides the processing, the authors also considered the size of these documents and their transmission through different types of networks such as Wi-Fi and 3G, as well as different XML processing APIs for the Android platform.

Finally, Tamayo et al. [9] presents an empirical study of current instances of SOS providers. The authors conducted an investigative work raising tens of SOS services available on the Web. These services have undergone several tests to check, for example, which parts of the specification are more frequent in SOS service implementations. Besides, the authors also found that many of the implemented providers have validation problems with the documents of observations returned by these servers, i.e., many of the documents returned by these servers could not be validated with the XML Schema that defines them.

As shown in this section, several studies in the literature analyze the SOS service, although many other services of SWE interfaces model are not considered. For example, SES is an important service within the interfaces model and it has not been treated by the literature in studies of performance evaluation. Alert services are important tools for developing applications of critical systems, which the delays in the delivery of alerts can hinder the effectiveness of these applications. Additionally, the registry service (SIR) is not considered in others SWE performance evaluation studies. The SIR is an important discovery service of sensor systems, although it is not a pattern of SWE yet. Currently, the SIR is treated as a “discussion paper”. However, it is already possible to find available implementations of this service as the one available on the website of 52° North [16]. Thus, Section IV aims to present and discuss the methodology and results of a performance evaluation of the SWE interfaces model, especially regarding the services SOS, SES and SIR.

IV. Performance Evaluation

This section aims to present a performance evaluation of SOS, SES and SIR services that compose the model of the SWE interfaces. Therefore, the purpose of this evaluation is to verify distinctions in performance using different types of filters in requests submitted to these services. Additionally, the evaluation proposed in this section considers a full factorial experiment design with three factors and two levels: **Amount Of Clients**, **Submitting Rate** and **Filter Types** (2^3, 8 Experiments). This design is applied to each of the evaluated services and it is defined in Table I.

TABLE I. Experiment Design

	Amount Of Clients	Submitting Rate	Filter Types
SOS	50/100	120/240	1Obs/288Obs
SES	50/100	120/240	Level1/Level2
SIR	50/100	120/240	Phenomenon/ID

The Amount of Clients and Submitting Rate factors possess the same levels for all services evaluated. The variation in the number of clients is performed by creating multiple threads that mimic the behavior of multiple clients accessing the services. In turn, the Submitting Rate factor simulates the submission of requests rate following an exponential function with averages of 120 and 240 requests per minute. Besides, it is important to know that each client (thread) submits 10 requests to the service using the exponential function highlighted.

The Filter Types factor has different levels, respecting the specificity of each service. In the SOS service case, are tested two variations of the GetObservation requests. The SOS services configured on the machines contain a database with the observations of sensors that measure the level of water concentration. The insertion of the observations in the database mimics the behavior of a sensor network by sending an observation every 5 minutes to the SOS service during a month. This behavior generates a total of 8640 observations registered in the server of the service provider. Therefore, in the context of the SOS experiments, the variations in the request messages are in relation to the periods of time to obtain the observations. The first experiment of SOS service concerns a period of time, which only one observation is returned, while the second type takes into account a period that the observations of a day are returned, totaling 288 observations.

In the case of the SES service, Level 1 and Level 2 that define the criteria for triggering alerts are used. As mentioned in Section II-C, the Level 1 considers a XPath expression that checks the value of the element *om:procedure*, while the Level 2 takes into account criteria such as sensor location, value observation, etc. In the case of the experiments performed in this performance evaluation, it is considered a criterion for location shooting, i.e., there will be an alert triggering when the SES receives sensors data that are located in a certain area. Finally, the experiments performed by the SIR consider two types of search criteria: the name of an observed phenomenon and the ID of the sensor in the service registry. The configuration of the SIR for this evaluation has 12 registered sensor systems that offer the same sensing information. Thus, experiments using a filter for the name of the phenomenon return 12 sensors descriptions (SensorML). However, the use of the ID in the search filter returns only one description. Section IV-A presents the infrastructure and the scenario implemented to perform the experiments.

A. Evaluation Scenario

The evaluation scenario uses an infrastructure composed of two virtualized machines (KVM) on different physical nodes. The physical nodes used for virtualization of these two machines have the following characteristics:

- Processor: Intel(R) Core(TM)2 Quad CPU Q9400 of 2.66GHz.
- Memory: 8 GB RAM DDR 3.
- Size disk: 500 GB. 7200 RPM.

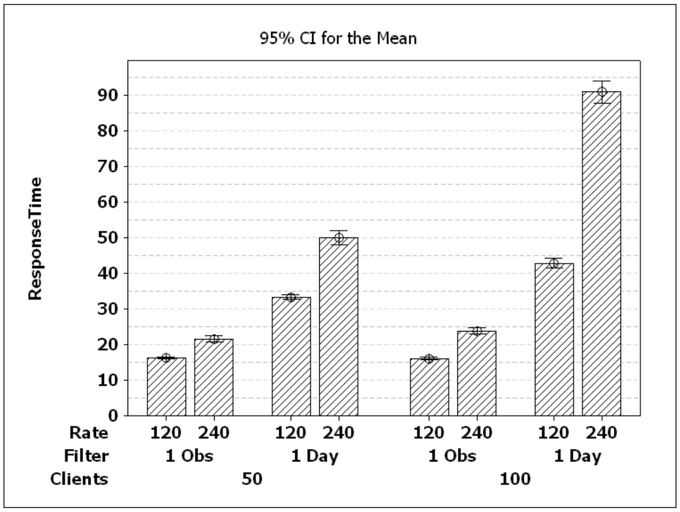

Figure 1. SOS: Response Times

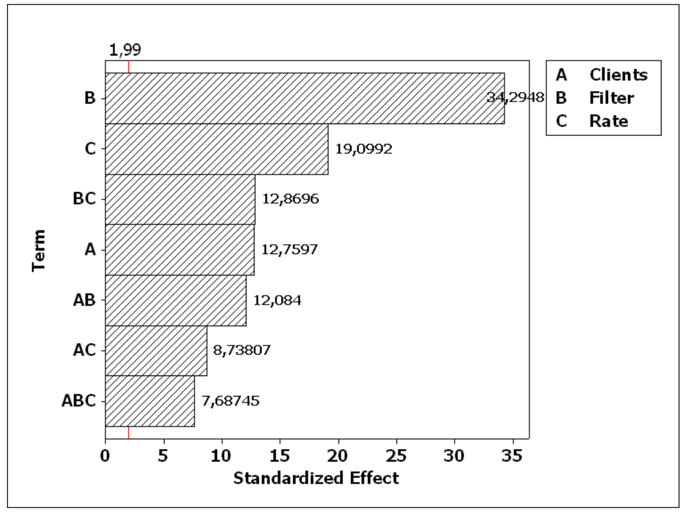

Figure 2. SOS: Factor Influence

In turn, the two virtual machines instantiated for the experiments have different settings, following the characteristics defined in Table II.

TABLE II. VIRTUAL MACHINE SETTINGS

Machine	Processors	Memory	Disk Size	Operating System
Server	4	4GB	15 GB	Ubuntu 12.04 (64-bits)
Client	2	2GB	15 GB	Ubuntu 12.04 (64-bits)

Regarding software, it was used the implementations provided by 52° North Initiative. It was used the versions as follows [16]:

- SOS: 3.5.0 version;
- SES: 1.0.0 version;
- SIR: 0.4 version;

B. Results

The results of the design of experiments presented in this section are shown in two types of charts:

- **Charts of the response times**: in these charts are presented the variations of the average response times in relation to variation in the levels of the factors. The confidence intervals calculated use a 0.05 alpha (95% of confidence). Furthermore, the averages are obtained by performing 30 replicates for each experiment.
- **Pareto Charts**: these charts show the influences of each of the factors in the tests. They use a vertical line that indicates the point where the factors start to have an influence in the experiments. In other words, the factors that lie above that line influence the response time. Additionally, the calculation of the influence percentage of each factor can be achieved through of calculating of each value of the factors in the Pareto chart divided by the sum of all of them.

As mentioned in Section III, several works performed studies of SOS services performance. However, the experiments conducted about this service in the study presented in this paper differ from those found in the literature. The performance evaluations on the SOS presented here use different evaluation factors. Besides, the analysis considers the behavior of the SOS service in relation to the variation of the number of clients accessing the service and the request rate submitted by each of them, in addition to the filters that determine different amounts of returned values. Thus, the chart in Figure 1 shows that the largest increase in response times occurs on changing the filter that returns only one observation for a filter that returns 288 observations (1 day of observation). In other words, significant increases in response times, considering the increase of clients, occur to the filter of 1 day. Response times are close in relation to the increase of clients for experiments with requests that return only 1 observation. The Pareto chart in Figure 2 shows that all factors influence the response time in the experiments, including the interactions between factors. In summary, the Filter factor has 31.9% of influence followed by Submitting Rate with 17.8% and Amount of Clients with 11.9%. Although the type of filter used has a greater impact on the response times, it is important to consider the number of customers and the rate of submission of requests, mostly for filters that return many observations.

The results obtained for the SES are shown in Figures 3 and 4. In the Figure 3, it is possible to observe that the large difference in response times occurs when the amount of clients are differents. Additionally, levels of filters also influence on the response times, especially for experiments with the average of 240 requests per minute and the experiments with 100 clients. In such cases, the experiments that consider the Level 2 have response times considerably higher than those obtained by the Level 1. The Pareto chart shown in Figure 4 shows that the number of customers is the most prominent factor in the experiments, followed by the factors of filters level and rate of requests. Therefore, the Amount of clients factor has an influence of 24.5% approximately, whereas the filter and rate factors have an influence of 21.9% and 18%, respectively. You can also verify that the interaction between these factors also represents significant influences. One of the

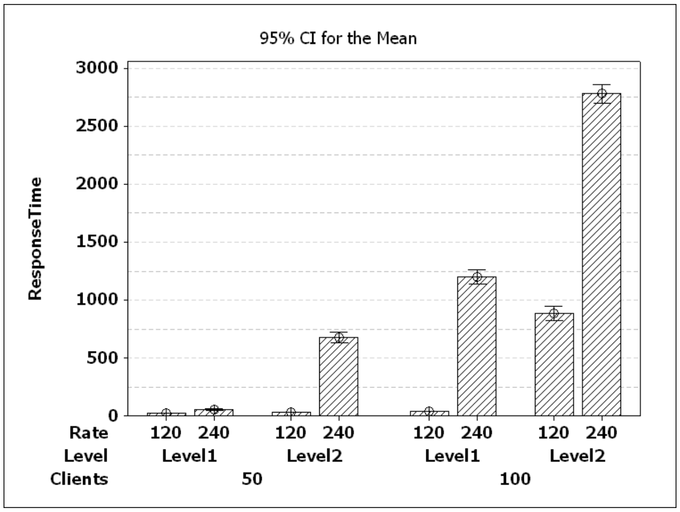

Figure 3. SES: Response Times

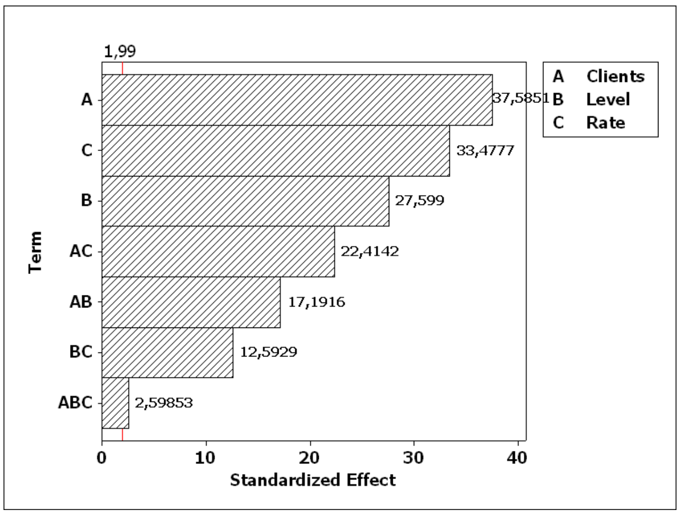

Figure 4. SES: Factor Influence

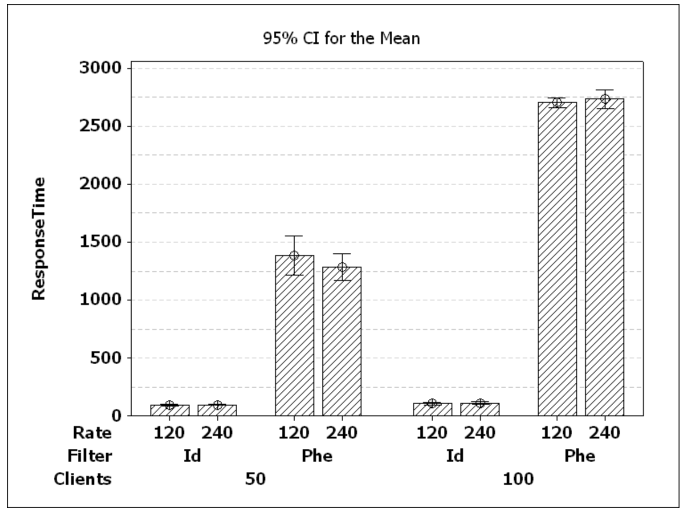

Figure 5. SIR: Response Times

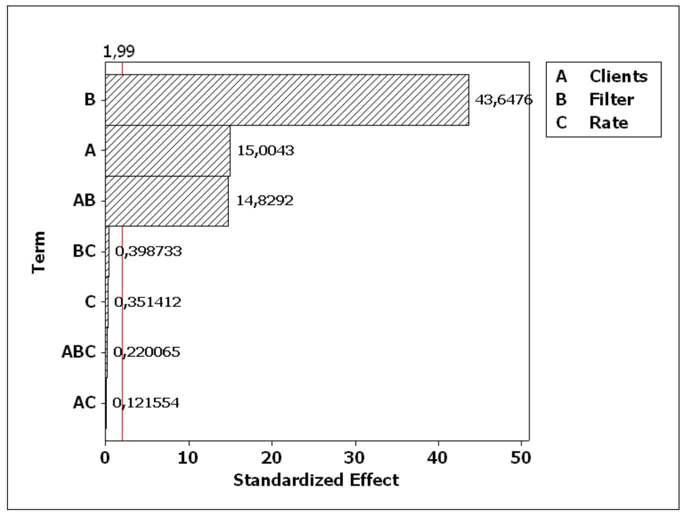

Figure 6. SIR: Factor Influence

main findings obtained in the execution of the experiments is related to the influence of the filters levels used in SES service. Applications that use SES can employ the results obtained in the experiments of this service to mark the usage of filter types in a more rigorous way. For example, certain applications that receive data from sensors networks and which react to alert messages may opt to computationally lighter filters as in the case of Level 1, when possible. Thus, as shown in the experiments, the proper definition of the filters can improve performance in the process of alerting.

Finally, the experiments related to the SIR are shown in Figures 5 and 6. The chart in Figure 5 shows that response times have significant differences in the Filter factor. Besides, the search for information of sensors using its ID in the service is much more efficient, since there is only one description of the sensor. However, it is impossible to know the ID of the sensor without performing a more generic search, such as the name of the observed phenomenon. Thus, if the application needs to check frequently possible updates in the sensor description, it firstly uses a search for the observed phenomenon and the subsequent searches by the ID obtained in the first interaction. Another mechanism that may be used to optimize the search of sensor systems in the SIR is the insertion of a broker that makes a cache of the search messages sent to the SIR. In this case, the broker can relate the search messages with the sensor Ids returned by SIR. Thus, the Broker can use the identifiers through search messages stored in the cache. For example, a client does a search for sensors that measure the wind speed and submits this search to the Broker. Then, the Broker receives this search message and forwards it to the SIR. The SIR response is stored in a tuple with the search message and sensor ID (*find_msg,sensor_id*) in the cache Broker. Therefore, when other clients submit the same search message, the broker replaces this message by a search message through the sensor ID, reducing the access time to the service registry.

Furthermore, searches performed by the ID of the sensor have no significant changes in time with the increase of clients' number and the rate of submitting requests. In such cases, it is possible to observe that the averages are statistically equal.

 ISBN: 978-1-61208-305-6

This behavior is reflected in the Pareto chart, demonstrated in Figure 6. The chart shows an influence of 58.5% in the filter factor.

The results presented in this paper demonstrate a performance differences on distinct types of filters in the considered services. So, the appropriate choice of these filters can benefit the performance of applications that use the SWE standards. For example, the developers of SWE applications have a better option in filters choosing that return less data. In high workload situations, the response time on changing a filter that returns only one observation for a filter that returns one day of observation can increase almost three times. In turn, the levels of filters on SES services also influence the performance of applications that use this service. Applications most rigorous regarding response time should choose level 1 filter that have better results and do some value comparison on own logic. Finally, searches on SIR hold improved performance using filter by ID. However, it is impossible to know an ID without using another filter type. So, it is indicated the use a phenomena filter, for example, in first search and a search for ID for the other searches. This type of interaction is indicated to application that send several searches for same ID to verify changes on descriptions of the sensor systems.

The results also show important influences in factors as amount of clients and submitting rate. They impact the response time in several tests. A solution to improve the performance of applications respecting these factors should be a cloud infrastructure. In this case, it is interesting to have an infrastructure where is possible to increase the computational capacity that offers the service. The OGC mentions the use of a cloud infrastructure in a white paper published in its official site [17]. Section V presents the performance evaluation conclusions and it discusses future works that can be developed from this study.

V. Conclusion and Future Work

This paper presented a performance evaluation of the interface models of SWE, especially highlighting the SOS, SES and SIR services. This evaluation considered the amount of clients, type of filters and submission rate as influencing factors in response times when accessing the services highlighted. Therefore, the results demonstrated an important influence of the filters type in the service response times. The influence of different filters in the requests was 24.5%, 31.9% and 58.5% for the SES, SOS and SIR services, respectively. The analyzes showed that the implementation of applications that use these services should carefully use the filters of these services, since the definition of them can significantly impact the performance of these applications.

Future studies should be developed to consider other services of SWE as SPS, and also improve the performance evaluation by increasing the variation of these filters. In the case of SIR Service, a Broker that manages the search messages to optimize the performance in accessing this service can be developed. Moreover, it is possible to develop mechanisms in relation to the provision of quality of service in the access of SWE interfaces model services, once the patterns specified do not consider this type of problem.

Acknowledgements

The authors would like to thank the financial support of FAPESP (São Paulo Research Foundation), FAPEMIG (Minas Gerais Research Foundation), CAPES (Coordination for the Improvement of Higher Education Personnel) and IFSULDEMINAS (Federal Institute of Education, Science and Technology of Southern of Minas Gerais).

References

[1] I. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. New York, NY, USA: John Wiley & Sons, Inc., 2010.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] M. Wang, J. Cao, J. Li, and S. K. Das, "Middleware for wireless sensor networks: A survey," *J. Comput. Sci. Technol.*, vol. 23, no. 3, pp. 305–326, 2008.

[4] T. Laukkarinen, J. Suhonen, and M. Hännikäinen, "A survey of wireless sensor network abstraction for application development." *IJDSN*, vol. 2012, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/ijdsn/ijdsn2012.html#LaukkarinenSH12

[5] F. C. Neto and C. M. F. A. Ribeiro, "Dynamic change of services in wireless sensor network middleware based on semantic technologies," *Autonomic and Autonomous Systems, International Conference on*, vol. 0, pp. 58–63, 2010.

[6] N. Mohamed and J. Al-Jaroodi, "Service-oriented middleware approaches for wireless sensor networks," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, 2011, pp. 1–9.

[7] OGC, "Ogc standards and specifications," December 2013, available in: http://www.opengeospatial.org/standards. Last Access: 06/05/2013.

[8] G. McFerren, D. Hohls, G. Fleming, and T. Sutton, "Evaluating sensor observation service implementations," in *Geoscience and Remote Sensing Symposium,2009 IEEE International,IGARSS 2009*, vol. 5, 2009, pp. V–363–V–366.

[9] A. Tamayo, P. Viciano, C. Granell, and J. Huerta, "Empirical study of sensor observation services server instances," in *Advancing Geoinformation Science for a Changing World*, ser. Lecture Notes in Geoinformation and Cartography, S. Geertman, W. Reinhardt, and F. Toppen, Eds. Springer Berlin Heidelberg, 2011, pp. 185–209. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-19789-5_10

[10] M. E. Poorazizi, S. H. L. Liang, and A. J. S. Hunter, "Testing of sensor observation services: a performance evaluation," in *Proceedings of the First ACM SIGSPATIAL Workshop on Sensor Web Enablement*, ser. SWE '12. New York, NY, USA: ACM, 2012, pp. 32–38. [Online]. Available: http://doi.acm.org/10.1145/2451716.2451721

[11] A. Bröring, J. Echterhoff, S. Jirka, I. Simonis, T. Everding, C. Stasch, S. Liang, and R. Lemmens, "New generation sensor web enablement," *Sensors*, vol. 11, no. 3, pp. 2652–2699, 2011. [Online]. Available: http://www.mdpi.com/1424-8220/11/3/2652

[12] SOS, "Ogc - sensor observation service interface standard," 2012, available in: http://www.opengeospatial.org/standards/sos. Last Access: 3/06/2013.

[13] SES, "Opengis - sensor event service interface specification," 2008, available in: http://portal.opengeospatial.org/files/?artifact_id=29576. Last Access: 12/06/2013.

[14] SIR, "Sensor instance registry discussion paper," 2010, available in: http://portal.opengeospatial.org/files/?artifact_id=40609. Last Access: 25/05/2013.

[15] A. Tamayo, C. Granell, and J. Huerta, "Using swe standards for ubiquitous environmental sensing: A performance analysis," *Sensors*, vol. 12, no. 9, pp. 12 026–12 051, 2012. [Online]. Available: http://www.mdpi.com/1424-8220/12/9/12026

[16] 52North, "Softwares," 2013, available in: http://52north.org/downloads/sensor-web. Last Access: 20/08/2013.

[17] SOS, "Ogc standards and cloud computing," 2011, available in: https://portal.opengeospatial.org/files/?artifact_id=43743. Last Access: 20/08/2013.

ISBN: 978-1-61208-305-6