A Software Architecture to Integrate Sensor Data and Volunteered Geographic Information for Flood Risk Management

Raul E. Simoni Castanhari
ICMC, University of São Paulo, Brazil
raul@icmc.usp.br

Roberto dos Santos Rocha
ICMC, University of São Paulo, Brazil
ERCIS, University of Münster, Germany
rsrocha@usp.br

Sidgley Camargo de Andrade
ICMC, University of São Paulo, Brazil
UFTPR, Federal Technological University of Paraná, Brazil
sidgleyandrade@usp.br

João Porto de Albuquerque
Centre for Interdisciplinary Methodologies, University of Warwick, UK
ICMC, University of São Paulo, Brazil
J.Porto @warwick.ac.uk

ABSTRACT

Natural disasters are phenomena that can cause great damage to people in urban and rural areas, and thus require preventive and reactive measures. If they involve multiple sources of information, these measures can be more useful and effective. However, the integration of heterogeneous data still poses challenges due to the differences in their structures and contents. To overcome this difficulty, this paper outlines a service-oriented architecture, as part of the AGORA platform, which aims to support the integration of sensor data and Volunteered Geographic Information (VGI) related to floods. The composition of the architectural components enables sensor data to be integrated with VGI by using several algorithms in a flexible and automated manner. The architecture was implemented by means of a prototype as a proof of concept and the results were used to generate thematic maps. These maps can improve flood risk awareness and support decision-making in flood risk management.

Keywords

Service-Oriented Architecture, Integration, Geospatial Data, Volunteered Geographic Information, Flood Management.

INTRODUCTION

Natural disasters, such as, floods, storms, fires, and hurricanes can cause serious damage to people if they occur in places of human activity, since they lead to loss of food, natural resources, urban infrastructure, and, most

importantly, human lives. In 2014, natural disasters incurred financial losses close to US$110 billion worldwide and caused 7,700 fatalities (Munich, 2014). With regard to the type of natural disasters that occurred in 2014, 73% were caused by extreme events of a hydrometeorological origin, *i.e.*, floods, flash floods, thunderstorms, avalanches, as well as landslides caused by rainfall (Munich, 2014).

There has been increasing concern to be prepared for oncoming events and to mitigate their consequences. This has encouraged the establishment of more resilient communities, especially those located in risk areas, such as landscapes or near rivers. Resilience can be defined as the capacity of communities to withstand extreme events (Mediondo, 2010), and involves social, political, economic and environmental aspects. Using environmental monitoring systems is one alternative way of achieving better preparedness for such events (Mansourian, Rajabifard, Zoej and Williamson, 2006). A requirement for this alternative is the use of timely, accurate and continuous information. This could greatly assist the emergency agencies involved in flood risk management, as discussed by Horita, Albuquerque, Degrossi, Mendiondo and Ueyama (2015). However, the continuous monitoring of flood risks requires local information such as observational data on rainfall and river levels to evaluate the risk of potential flooding. This can be a considerable challenge in regions where there is a vast territorial extension.

In this context, Volunteered Geographic Information (VGI) has been of great practical value as a means of complementing existing geospatial datasets (Goodchild, 2007) owing to the large number of volunteers who act as potential 'sensors' (Poser and Dransch, 2010). This raises several research challenges to make VGI an effective data source for flood risk management. One of them is how to integrate VGI with existing data sets rather than considering it as a parallel source of information (Albuquerque, Herfort, Brenning and Zipf, 2015). In particular, research about how to integrate VGI with sensor data still face challenges for the following reasons: (i) the sensor data is structured and standardized, while volunteered information is almost entirely non-standardized; (ii) data from sensors are precise, while those from volunteers depends on their perceptions; and (iii) the volunteered data are generally less reliable than that obtained from sensors. However, integrating these data is important, because the use of several data sources can provide a pool of disaster information that is more complete and updated than just relying on one source (Goodchild, 2007; Horita *et al.*, 2015).

Moreover, the nature of VGI usually requires a degree of manual processing (*e.g.*, the observation of a video by a human to extract information). This limits the use of VGI since this kind of task has very specific goals and makes the process inflexible. In addition, manual activities are very time consuming, which can be critical in a disaster management situation, where emergency agencies have to act rapidly.

This research tackles these issues by proposing "AGORA Information Fusion and Management" (AGORA-IFM): a service-oriented architecture to support the integration of sensor and volunteered data, and assist this process by: (i) automating the integration, speeding up the process and making it applicable to different situations; (ii) making the process flexible, by allowing the employment of different spatiotemporal methods; and (iii) making the process adaptable, so that it can be modified through Web service chaining.

The remainder of this paper is structured as follows: the "Background" section introduces the basic concepts that will be used throughout the paper; the "A Software Architecture for Information Fusion and Management" section describes the proposed architecture; the "Architecture evaluation" section explains how the architecture was qualitatively evaluated; the "Proof of concept" section discusses the results obtained from its instantiation and how it is used to produce thematic maps; the "Related works" section makes a comparison with previous related works; and the "Conclusion" summarizes the findings of the research.

BACKGROUND

Geographic data integration is the process of making different data sets compatible with each other, so they can be clearly displayed on the same map so that their relationships can be analyzed (Flowerdew, 1991). In this respect, some research projects are being carried out to obtain integrated heterogeneous geographic data. For example, Schnebele, Oxendine, Cervone, Ferreira and Waters (2015) proposed an application to fill the gaps in remote sensing data during disasters by making use of data collected from non-authoritative sources, like VGI, or collected for purposes other than disaster assessment, such as that from traffic cameras. However, this study does not allow users to produce a set with all the integrated data. In a previous work, Schnebele, Cervone and Waters (2014) devised a methodology for estimating flood damage by integrating authoritative and non-authoritative data. They integrated the information of photographs classified by crowdsourcing and videos that

were manually classified to determine the severity of the flood damage, but their methodology does not allow automatic integrations.

Kotsev, Pantisano, Schade and Jirka (2015) proposed an approach to enable the interoperability of environmental sensors that combine on-board storage with Web service capabilities. Through their approach, observation data can easily be combined and reused. However, despite combining sensor data (even that produced by a citizen's mobile devices), their approach fails to take account of data provided by citizen perceptions (like filling out a form).

Meissen and Fuchs-Kittowski (2014) proposed a reference architecture of crowdsourcing integration in early warning systems, which integrates the components of monitoring systems with those of crowdsourcing systems components. In this way, their architecture has a capacity for participant management to improve the system for issuing and spreading warnings, but it cannot be implemented, since it requires further specifications before being developed in real contexts.

A SOFTWARE ARCHITECTURE FORE INFORMATION FUSION AND MANAGEMENT

In this section, we examine some previous works, and provide an overview of the proposed software architecture, together with an explanation of how it works.

Preliminary Work

Floods cause most damage in Brazil, with approximately 48 million people affected between 1980 and 2010 (UNISDR, 2014). In this context, the AGORA research project[1] was proposed aiming to create a platform to bring together the perspectives of science, the general public, and the government to tackle social challenges, with an initial focus on floods (Albuquerque and Zipf, 2012).

The AGORA project comprises three parts (Figure 1): (*i*) acquisition, responsible for obtaining and storing geosensor data and VGI; (*ii*) integration, where the goal is to integrate data from the two previous types; and (*iii*) application, responsible for producing indicators to facilitate the decision-making process and heighten community awareness. As highlighted in Figure 1, this study takes place at the integration part of the project.

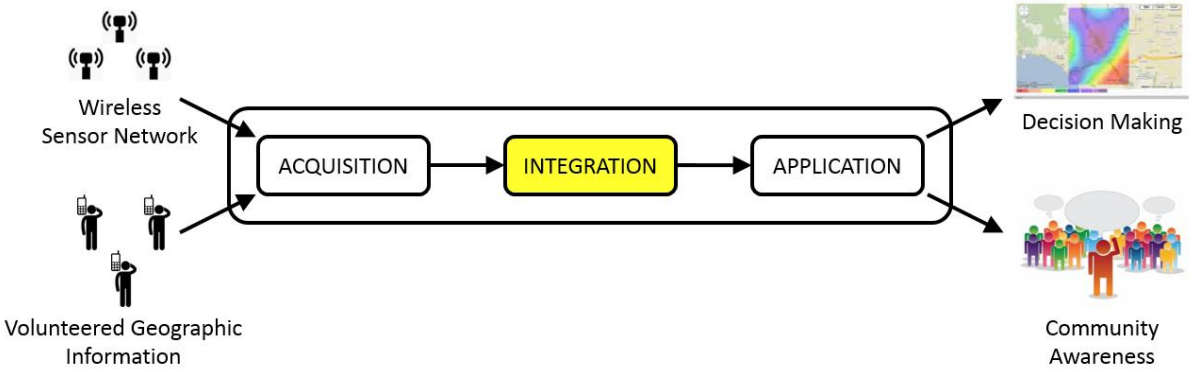

Figure 1. AGORA Project Overview

Overview of the software architecture

The service-oriented architecture called AGORA Information Fusion and Management (AGORA-IFM) was proposed to support the integration of sensor data and VGI by different spatial-temporal methods in an easily updatable and automatic way; the purpose of this was to speed up the integration process and make it more flexible. Its results allow a greater concentration of the spatial and temporal information that is required for the management of natural disasters.

[1] http://www.agora.icmc.usp.br, Access on 15 February 2016.

This architecture uses the Dynamic Sensor Management (AGORA-DSM), a system that dynamically manages heterogeneous geosensors, to obtain sensor data (Assis, Behnck, Doering, Freitas, Pereira, Horita, Ueyama and Albuquerque, 2016). These data are coded in O&M. This architecture obtains volunteered data by means of the Volunteered Observation Service (AGORA-VOS), a system accessible via Web site or smartphone application and managed by a customized instance of the Ushahidi[2] crowdsourcing platform (Degrossi, Albuquerque, Fava and Mendiondo, 2014). In general, VGI can comprise unstructured texts, photos, videos and audios, which represent different data structures. It is hard to extract relevant information from these data and this is usually done manually. The interface of this system guides the volunteers on how to convey their information, and structures it as an O&M, although it also stores other unstructured data. AGORA-DSM and AGORA-VOS embed their data in a SOS server so that they can be made available. They correspond to the acquisition part of the project (Figure 1) and are not included in this study. AGORA-IFM integrates sensor data and VGI through these systems, together with an engine of spatial-temporal geostatistical functions that isolates the integration algorithm. This makes it easily changeable and allows an automatic integration of the two previous data types. Finally, the results are stored in a geographic database and exported via WMS.

Workflow

The Figure 2 shows the workflow based on the Business Process Model and Notation (BPMN) (OMG, 2011). As can be observed, there are 3 pools (SOS, IFM and R Script Engine), 8 lanes (Scheduler, Orchestrator, Sensor Data Handler, VGI Handler, Data Binder, Categorizer, R Data Handler and Data Exporter) and 12 activities (Call Handle Sensor Data, Handle Sensor Data, Call Handle VGI, Handle VGI, Call Bind Data, Bind Data, Call Categorize Data, Categorize Data, Call Handle R Data, Handle R Data, Call Export Data and Export Data). The activities are arranged by an orchestrator that is responsible for executing them in the right order.

This workflow is triggered periodically by a scheduler and, in each execution, it processes the data related to a time defined by user, for example data arrived since its previous execution. The scheduler activation period is programmable - for example, the workflow can be triggered every ten minutes. As shown in Figure 3, we have defined that each lane of the Figure 2 represents a component of the Unified Modeling Language component diagram (OMG, 2004). A component diagram depicts how components are wired together to form larger components and or software system.

The two first activities, ‘Handle sensor data’ and ‘Handle VGI’, are responsible for handling data that was previously collected by the acquisition part and stored separately in a SOS server. ‘Handle sensor data’ uses AGORA-DSM to achieve its goal, what includes making a data selection. AGORA-DSM can store several kinds of sensor data, like water level, wind speed or air temperature, as well as storing historical data. Hence, this activity supports the selection of the data property and period - for example, by selecting the water level from the previous five minutes. ‘Handle VGI’ uses AGORA-VOS to obtain volunteered data, from another selection. AGORA-VOS can store some predefined kinds of data, which are chosen by volunteers while the report is being sent. For example, it might be a free text, the state of the street around a river (whether it is passable or not), or a water level based on the volunteer's perception (normal, high, etc.). It also stores historical reports. This activity supports the selection of the data type and period, but currently only selects data stored with the type of water level to allow an automatic categorization of the activities that follow. These two first activities obtain data in O&M but, due to this format allows a variable amount of its fields and values, the variations in these data require separated processing. For this reason, these activities also transform data from both sources into a single format, which allows them to be categorized and parsed faster. The services that implement these activities correspond to the following components: ‘Sensor data handler’ and ‘VGI handler’.

[2] http://www.ushahidi.com/

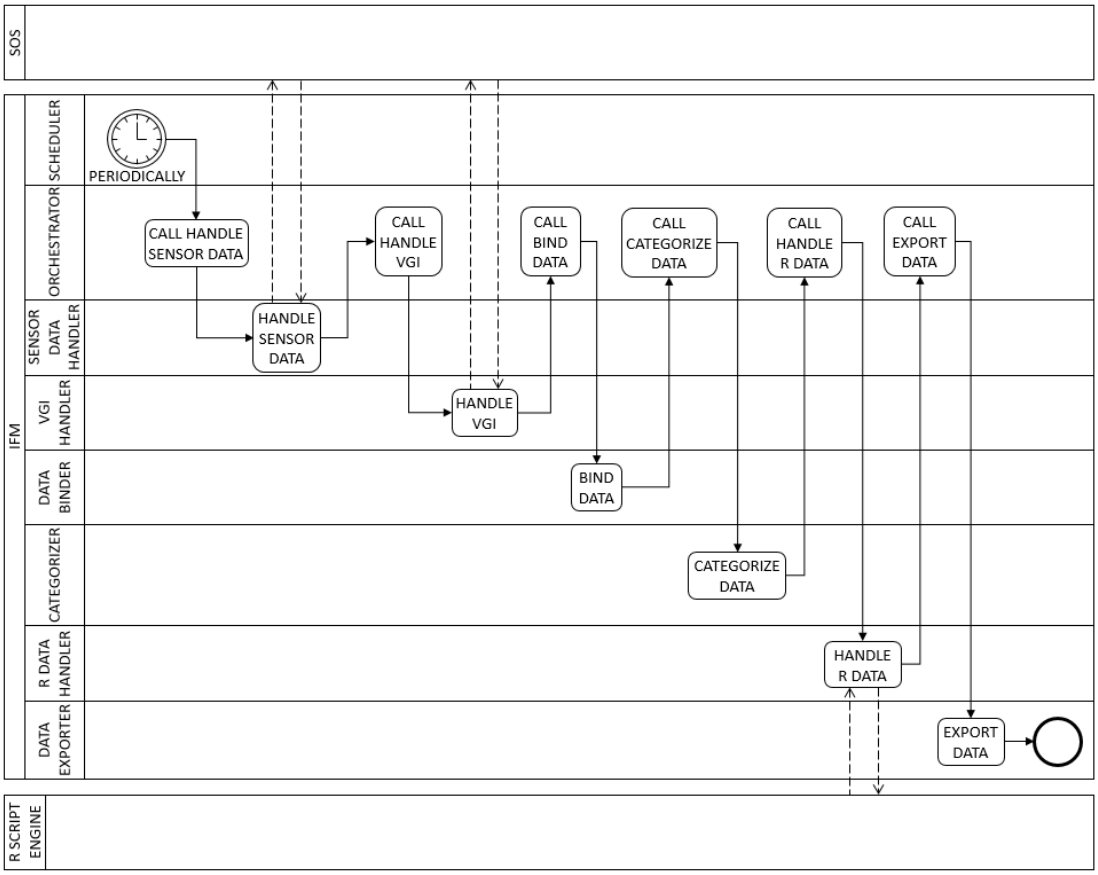

Figure 2. Workflow diagram for the AGORA-IFM architecture

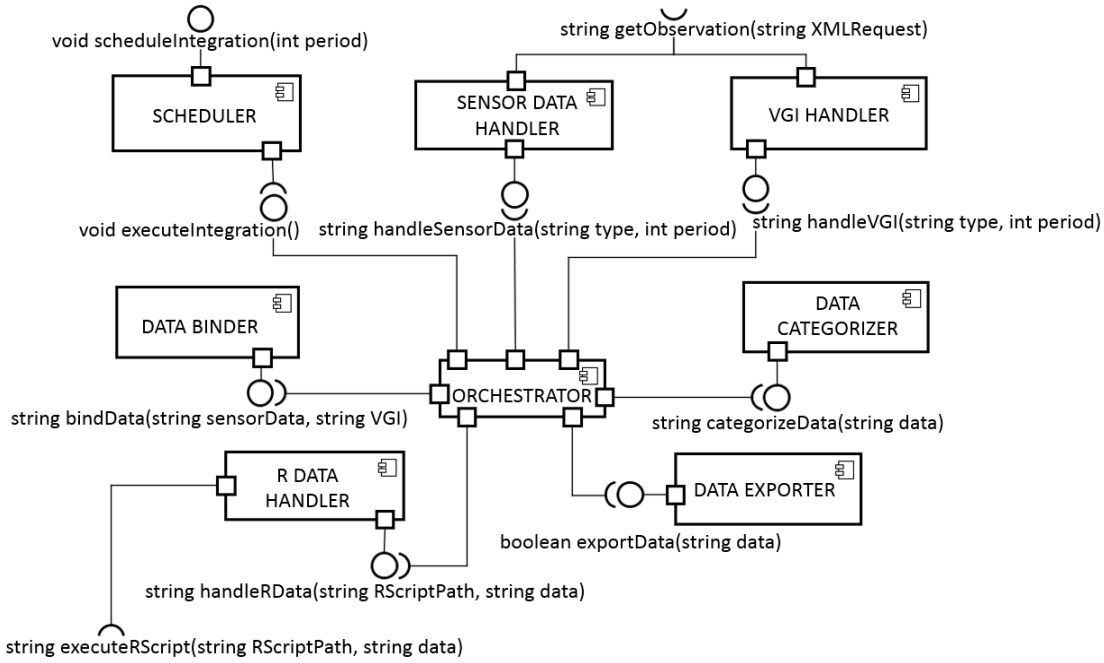

Figure 3. Components diagram of the AGORA-IFM architecture

The third activity, 'Bind data', just bind sensor and volunteered data in a single source. The service that implements it, which corresponds to the 'Data binder' component, is also used internally by the first activity to bind data from all sensors, and by the second activity to bind all data obtained from the volunteers.

The fourth activity, 'Categorize data', involves categorizing all the data, which is based on the water level values and the metadata of the VGI. The categorization consists of assigning one of the categories ('low', 'normal', 'high' or 'overflowing') to each observation, so that the situation of the river can be easily understood. The thresholds that determine the category for a given water level value depend on the river and vary at each river point; for this reason a previous analysis of all the rivers in question is required. This research is based on the assumption that the thresholds have been previously established. 'Data categorizer' is the component related to the services that carries out this categorization.

The fifth activity, 'Handle R data', performs the integration of sensor data and VGI in an easily updatable and automatic way. It consists of a service, which corresponds to the 'R data handler', that interacts with an engine of algorithms written in the R language. R is initially used due to the large amount of geographical methods for it, however the correspondent interface is being generalized to support other languages and engines. The parameters of the service interface include the path to the program script that has to be executed and the categorized data that have to be integrated. This activity is carried out as follow: first, the component transforms the data passed from the orchestrator to specific data types, then, it sends these data to an external R engine. Next, the engine reads the script file that is specified by the path argument and executes it line by line. Lastly, the engine returns the resulting data to the component, which transforms the data back to common data types and returns them to the orchestrator. As this process isolates the integration algorithm, the effort to adapt it, for example by changing its parameters or even changing the integration method for another statistical model, only involves updating the script, in the cases that the input and output data of the algorithms are of the same type. It makes the process quite generic and easy to update. Adapting the algorithm in the cases that the inputs and outputs are of different types is also possible through a transformation (see the Implementation section).

Finally, the last activity, 'Export data', makes the results available for external applications. This is performed by arranging the integrated data in the way expected by a geographic database (generally at geometric points with the values of their associated water level), and then connecting to it and storing each resulting point with its geographic position format. In this way, an external server accesses these data and exports them via WMS. The service that carries out such activity corresponds to the 'Data exporter' component.

All the integration process (as represented in Figure 2 by the clock as an initial event) is initialized by a component that schedules it so that it can be executed periodically, which is possible since none of the activities require manual activities. In this way, new incoming data can be processed to update the results, which increases the temporal data coverage and provides a situational view of the considered area that is updated along the time. The 'Scheduler' component is responsible for this behavior.

EVALUATION OF THE ARCHITECTURE

An adaptation of the Architecture Tradeoff Analysis Method (ATAM) was employed to evaluate the AGORA-IFM. The architectural evaluation based on scenarios is the type of method most used by industry. Among these method types, ATAM is the most used in the evaluation of system of systems (SoS) (Santos, Oliveira, Guessi, Oquendo, Delamaro and Nakagawa, 2014). SoS are systems that aggregate independent and heterogeneous systems to perform new and emerging capabilities. In this sense, AGORA is a SOS and ATAM an adequate architectural evaluation method for our case.

ATAM

This method assesses the architecture by comparing the decisions made by the architect with regard to the architectural requirements and scenarios to determine if they conform to the relevant quality attributes. It involves people with knowledge in software engineering and domain application (project team), software architecture (evaluation team), and other stakeholders including the end users and customers. The method defines two phases, that involve a meeting with selected teams and a set of steps. The first phase has six steps and involves the evaluation and the project team. The second phase has the previous six steps from phase one

and further three steps, involving all the stakeholders. Along the application of the method, we identify, refine, and analyze architectural decisions, priority requirements and scenarios by verifying risks (potentially problematic decisions) and non-risks (good decisions). See Clements, Kazman and Klein (2002) for details.

AGORA-IFM Architecture Evaluation and Results

We employed an adapted ATAM because some artifacts used in it had been designed previously rather than design them during the evaluation. The purpose of this was to save time, and then we just refined these artifacts during the meetings. Table 1 shows the architectural decisions made by the architect.

Id.	Architectural decision	Quality attributes	Description
D1	Service-oriented architecture.	Modularity Interoperability	The features of the integration process were arranged into independent services.
D2	Public availability of the main services.	Reusability	The main services were made publicly available to allow their individual use in different contexts.
D3	Separation of data integration script and other services.	Modifiability	The integration script is executed from an independent file regarding to the implementation code. This means that, if the script inputs and outputs are equal to the ones required by the service that executes it, changing the script will require little effort.
D4	Cloud hosting.	Availability/ Time Behavior	The architecture is deployed in cloud to ensure high hardware availability and tolerance to hardware faults.
D5	Message formatting encapsulated in service.	Interoperability	The message formatting encapsulation in a service facilitates a change of the formatting standard.

Table 1. List of Architectural Decisions

As a result, we were able to determine the points where the architecture satisfies its requirements and where it has limitations. Table 2 summarizes these points. Requirement R8 was not met because, even facilitating the formatting standard change, it was found that it would have to determine what specific standard would be used in each message exchanging. This has not been carried out yet because there are many standards in this domain and different kinds of messages that have to be considered, and this is one of our current tasks. For the same reason, requirement R4 was not met because, although the components are distributed as web services, the format of the data exchanged by the new services needs to be defined. However, new services that conform to the formats of the services already defined can be seamless added or replaced. Requirements R6, R7 and R13 were not met due to the lack of decisions. This was because these requirements were identified during the evaluation, after the AGORA-IFM had been developed. R7 involves reducing the integration costs in cases where software faults to the system remain active; R13 consists of providing previously categorized and stored data to clients be able to directly do several integration requests in parallel. We are also currently considering them. The other requirements were successfully met, including the most important, which is to change the integration script. R2 involves allowing the workflow to be customized, like using just sensor data as the source; and allowing services to be used individually like just obtaining volunteered data. The integration process was demonstrated by the implementation of AGORA-IFM, which is outlined in detail in the next section.

Requirement	Quality Attribute	Architectural Decision	Risk	Nonrisk
R2. Customized use of the workflow and individual use of the services.	Reusability	D2. Public availability of the main services.		X
R3. Changing the integration script.	Modifiability	D3. Separation of the data integration script and other services.		X
R4. Replacing and addition of new services to the workflow.	Modifiability	No decision.	X	
R6. High software availability.	Availability, Recoverability	No decision.	X	
R7. Providing a lightweight workflow.	Recoverability	No decision.	X	

R8. Using recognized standards.	Interoperability	D5. Message formatting encapsulated in service.	X	
R10. The integration response time should be low.	Time Behavior	D4. Cloud hosting.		X
R13. Integration by several models in parallel.	Time Behavior	No decision.	X	

Table 2. Summary of the Evaluation Result of AGORA-IFM

PROOF OF CONCEPT

A proof of concept is demonstrated through the implementation of AGORA-IFM and an experiment carried out in Monjolinho River in the center of the city of São Carlos, São Paulo, Brazil, where flooding often affects the city´s infrastructure and citizens.

Implementation

The described process was implemented in accordance with this architecture to determine its feasibility and results. Architectural components and a client were developed as cloud services in the Azure[3] platform using the Java programming language, and their communication is operated via SOAP with the support of the Axis[4] framework. All the loaded data is stored as text to maintain interoperability between different systems, which contain information like the observed value, position, date and time.

The categorization service calculates the category so that the numbers 1, 2, 3 and 4 correspond to the low, normal, high and overflowing categories respectively, and stores them in the previous text data set. This is necessary to enable processing over VGI that reports a water level such as 'normal water level', which cannot be integrated with numerical values. The integration process uses these numbers to make predictions in regions where there are no observations but which have sensor or volunteered observations close to them.

The rJava[5] package was used to execute R spatial-temporal statistics from Java. This is responsible for the interface between both languages and sending the script commands to the R processor, in this case, GNU's R 3.1.2. The only points of contact between the Java and R programs are the input and output data types transformations, which make all the rest independent.

The statistical method to obtain the water level estimates was instantiated with a spatial interpolation called Kriging through the geoR[6] library. Using other methods with inputs and outputs of types different from Kriging is also possible, but this requires the transformation of these types. This can be carried out either inside the script, by transforming the R types generated by the already implemented services in the new R types, or creating new web services, by transforming the Java code in the new R types. It is also possible to use methods from different R libraries apart from the geoR, since it just requires the corresponding libraries be installed on the server that is hosting the R engine. It should be pointed out that the contribution of this research is not the integration algorithm itself, but the support of it, *i.e.,* the possibility that the architecture provides to use different integration methods to integrate sensor and volunteered data automatically.

The data resulting from the integration process is stored in a PostgreSQL database with the PostGIS[7] extension, which adds support for geographic objects and allows SQL location queries. These data consist of points with water level categories for the entire region among the collected observations, and have geographic metadata to allow spatial-temporal manipulations and visualization.

We have employed the architecture implementation to conduct an experiment with the goal of testing its capacity to support automatic integration processes and to assess the flexibility that it provides. On the basis of the data resulting from the experiment, we have built thematic maps to show water level along the whole river and the points where there is a probability of flooding.

[3] https://azure.microsoft.com/
[4] http://axis.apache.org/
[5] https://www.rforge.net/rJava/
[6] https://cran.r-project.org/web/packages/geoR/
[7] http://postgis.net/

Experiment to Make Estimates of Flooding

The experiment uses a set of sensors installed at strategic points of the Monjolinho River analyzed by hydrologists and volunteer reports located at points where there is a great movement of people. Figure 4 shows the scenario with the spatial distribution of data samples (S_i and V_i) and places with frequent floods (A and B). The samples were based on measurements from real sensors (deployed on the banks of the River Monjolinho), volunteers reports, and simulated VGI (due to an insufficient number of volunteers and reports). We collected water level measurements from the last measure of each sensor from AGORA-DSM and data with the water level type from the past one hour from AGORA-VOS. Although the temporal scale can affect to closer relationship between virtual and real world, it was not dealt. Table 3 shows the input data used in the integration process. In determining the thresholds used in the categorization, we made use of the measurements calculated by hydrologists about the depth and water normal level at one point with frequent floods (Point A in Figure 4) and applied them to the whole river.

Id.	Value	Coordinates (lat, long)	Id.	Value	Coordinates (lat, long)	Id.	Value	Coordinates (lat, long)
S1	127.2cm	-22.00051, -47.89978	V5	N	-22.00941, -47.90503	V15	O	-22.01712, -47.91180
S2	124.7cm	-22.00241, -47.90037	V6	O	-22.00800, -47.90356	V16	L	-22.01510, -47.91026
S3	109.6cm	-22.00687, -47.90531	V7	L	-22.00580, -47.90386	V17	O	-22.00858, -47.90400
S4	98.5cm	-22.01010, -47.90625	V8	H	-22.00463, -47.90146	V18	L	-22.01341, -47.90827
S5	109.7cm	-22.01557, -47.91104	V9	L	-22.01290, -47.90737	V19	O	-22.00678, -47.90456
S6	115.4cm	-22.01775, -47.91279	V10	H	-22.00380, -47.90145	V20	N	-22.00082, -47.89980
V1	O	-22.01672, -47.91224	V11	O	-22.00358, -47.90029	V21	O	-22.00383, -47.90036
V2	L	-22.01443, -47.90861	V12	H	-22.00120, -47.90042	V22	L	-22.01982, -47.91340
V3	L	-22.00536, -47.90256	V13	N	-22.01883, -47.91306	V23	L	-22.02061, -47.91350
V4	L	-22.01139, -47.90637	V14	O	-22.01816, -47.91340	V24	L	-22.00919, -47.90490

Table 3. Input Data for the Integration Process. The values in centimeters are the water levels measured by sensors. 'L', 'N', 'H', and 'O' are the water levels reported by volunteers, standing respectively for 'Low', 'Normal', 'High', and 'Overflowing'. 'lat' and 'long' in coordinates stands for 'latitude' and 'longitude'.

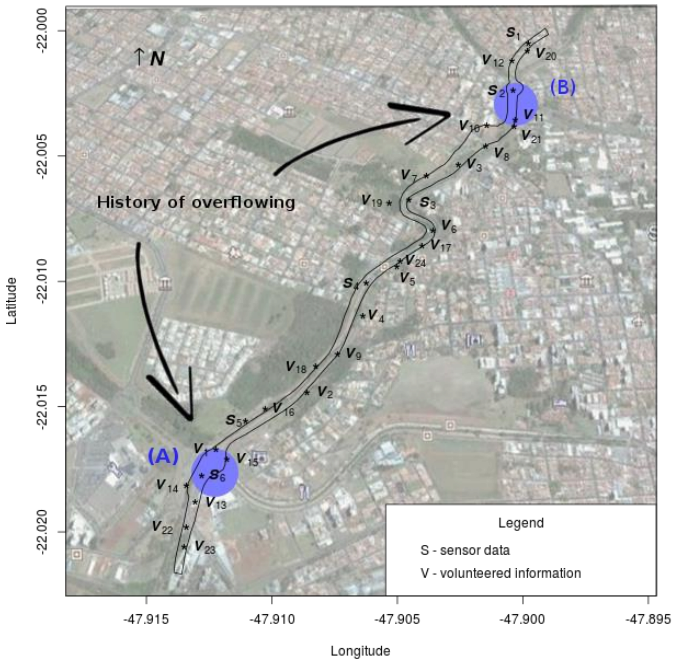

Figure 4. Reference scenario - Monjolinho River, São Carlos, Brazil

We used Kriging interpolation as the statistical method to estimate the water level along the whole river (Figure 5) and areas likely to flood (Figure 6). The interpolation was performed with a spatial resolution of

approximately one square meter. The Gaussian model was chosen by the Weighted Least Squares (WLS) estimation method as the model fit that best represents the gathered data. Both the estimation method and model fit were selected after making a preliminary review and set as parameters of Kriging.

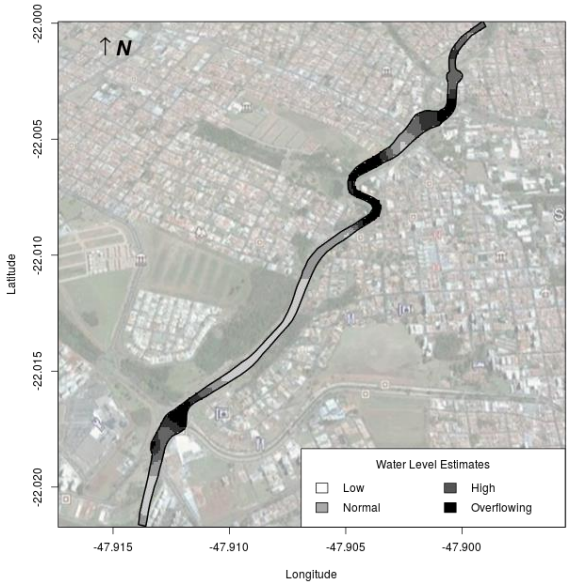

Figure 5. Result of the integration process using Kriging to estimate water levels

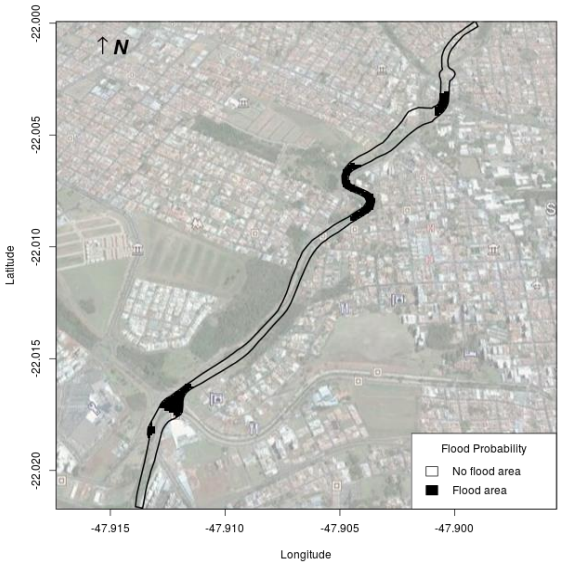

Figure 6. Result of the integration process using Kriging to predict flood areas

The interpolated data have been dichotomized to predict the flood areas. The values that were in the overflowing category, *i.e.*, that exceeded the river height, were associated with the number 1, and the rest were associated with 0. This stage was conducted manually, although it is possible to customize the R script so that it can be done automatically.

Thematic maps were designed to show the results yielded by AGORA-IFM (Figure 5 and Figure 6). However, it should be stressed that this stage is not the focal point of this research since it is in the application part of the global project (Figure 1). Thus, the maps were designed manually in R from the results of AGORA-IFM.

Discussion of the Results

During the experiment we were able to execute the entire process several times because it is fully automatic, and lasted less than ten minutes for each execution. Every time we changed the integration parameters or even the whole method. This was done by simply editing the file corresponding to the script, which did not influence the rest of the process. This demonstrates the flexibility and the benefits of the automation provided by the architecture. In addition, we executed the process by using just the services about the sensor data and that about VGI, which demonstrates that the process itself is adaptable, since it can be defined by selecting the services we need.

The estimation of the water level using both sensor data and VGI provides benefits by determining water level values for areas that do not have sensor data but where there is a movement of people. It can be seen that the prediction of flood areas matches the flood areas of the reference scenario (Figure 4), and it is possible to visualize the situation of each region of the river (Figure 5). This can heighten community awareness and improve the decision-making by responsible authorities. With regard to decision-making, Figure 6 shows the areas more likely to be flooded, which can support public policies in undertaking a flood prevention more accurately, especially through urban development plans and by spending money on control measures against flooding (*e.g.*, through infrastructure schemes and vegetation). In view of this, both thematic maps can be used to prevent human and material losses, economic disruption and social problems (*e.g.*, heavy traffic and power outage) caused by floods.

This result is a simulation and cannot be generalized because there are some limitations involved. First, the thresholds must be calculated at several points of the Monjolinho River. Second, a temporal resolution must be used for collection of water level measurements of the proof of concept. Third, VGI must be collected through AGORA-VOS without simulation, which can be done when there are enough volunteers reports. In this case, as the number of observations increase, the result tends to be improved. In addition, by putting into effect a continuous production of data, such as sensors and volunteers who are continuously sending data, the information concentration can be improved also over time. These limitations, however, do not disqualify the results of this proof of concept.

Finally, although essential for the quality of the results of integration process, semantic inconsistency was not discussed in this paper. There is one ongoing research related to the VGI quality being carried out in our group. Thus, these results will be incorporated to our architecture and discussed in future works.

CONCLUSION

This paper has presented a software architecture to support the integration of sensor data and VGI with the aim of increasing the spatial-temporal coverage of information on floods that is required to improve disaster risk management.

The architectural evaluation showed that most of found risks occurred because the associated requirements were not identified before the architectural development. We now are working to include these requirements. In addition, we were able to verify that the main requirements were met and the architecture is suitable for its purposes, which was analyzed by its implementation.

AGORA-IFM can be implemented in real contexts to heighten the awareness of decision makers and communities. It also involves volunteers who submit reports, enable clients to choose the integration method that is most suitable for their needs, and perform data integration automatically and online. Thus, three main research contributions of this architecture can be summarized as follows: first, it executes its workflow automatically, and produces results faster than using manual activities, which allows it to be used in several situations. Second, it supports the flexible integration of heterogeneous data, by enabling the use of different spatiotemporal methods. Finally, it makes the integration process adaptable, by the selection of the required components.

In future studies, we intend to consider the new requirements identified and extend the integration process to make it more robust. For example, this can be achieved by adding a quality factor to differentiate the reliability of sensor data in comparison with the volunteered data. In addition, it is possible to use predicted water levels to generate forecasted estimates. Furthermore, an optimization can be carried out by comparing the data of two consecutive process executions to analyze the need to update the previous results, and thus avoid unnecessary usage of resources when possible.

ACKNOWLEDGMENTS

The authors would like to thank CAPES (grant no. 88887.091744/2014-01) for providing partial support to this research, as well as Microsoft Research for providing access to computational resources within the Microsoft Azure for Research Award program. Parts of the research leading to these results has received funding from the European Community's Seventh Framework Programme (Marie Curie Actions/Initial Training Networks) under Grant Agreement n° 317382. The authors would like to thank the support and collaboration of Lívia C. Degrossi, Werner Leyh, Lucas B. R. de Oliveira, Milena Guessi, Lina M. G. Rodríguez, Brauner R. do N. Oliveira, Narumi Abe, Camilo E. R. Estrada and, especially as evaluation leader, Daniel S. Santos in the evaluation of the architecture described in this paper. Many of the ideas of this paper result from the intense and productive collaboration with Prof. Eduardo Mario Mendiondo and Prof. Jó Ueyama, as well as with their respective research teams.

REFERENCES

1. de Albuquerque, J. P., Herfort, B., Brenning, A. and Zipf, A. (2015) A Geographic Approach for Combining Social Media and Authoritative Data towards Identifying Useful Information for Disaster Management. *International Journal of Geographical Information Science*, 29(4): 667-689.

2. de Albuquerque, J. P. and Zipf, A. (2012) Collaborative information systems for disaster management: Building resilience against disasters by combining participatory environmental monitoring and vulnerability communication. *In Proc. of the Alumni Seminar Natural Hazards*, Teresópolis, RJ.
3. Assis, L. F. F. G. d., Behnck, L. P., Doering, D., Freitas, E. P., Pereira, C. E., Horita, F. E. A., Ueyama, J., de Albuquerque, J. P. (2016) Dynamic sensor management: Extending sensor web for near real-time mobile sensor integration in dynamic scenarios. *30th IEEE International Conference on Advanced Information Networking and Applications* (AINA), [S.l.: s.n.].
4. Clements, P., Kazman, R., Klein, M. (2002) Evaluating Software Architectures: Methods and Case Studies. [S.l.]: *Addison-Wesley*.
5. Degrossi, L. C., de Albuquerque, J. P., Fava, M. C., and Mendiondo, E. M. (2014) Flood citizen observatory: a crowdsourcing-based approach for flood risk management in Brazil. In *The 26th International Conference on Software Engineering and Knowledge Engineering, Vancouver, Canada*, pages 570–575.
6. Flowerdew, R. (1991) Spatial data integration. Geographical Information Systems, v. 1, p. 375–387.
7. Goodchild, M. F. (2007) Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221.
8. Horita, F. E. A., de Albuquerque, J. P., Degrossi, L. C., Mendiondo, E. M., and Ueyama, J. (2015). Development of a spatial decision support system for flood risk management in brazil that combines volunteered geographic information with wireless sensor networks. *Computers & Geosciences*, 80:84 – 94.
9. Kotsev, A., Pantisano, F., Schade, S., and Jirka, S. (2015) Architecture of a service-enabled sensing platform for the environment. *Sensors*, 15(2):4470–4495.
10. Mansourian, A., Rajabifard, A., Zoej, M. V., and Williamson, I. (2006) Using SDI and web-based system to facilitate disaster management. *Computers & Geosciences*, 32(3):303–315.
11. Mendiondo, M. (2010) *Integrated Urban Water Management: Humid Tropics*, chapter Reducing vulnerability to water-related disasters in urban areas of the humid tropics, pages 109–127. CRC Press.
12. Meissen, U. and Fuchs-Kittowski, F. (2014) Towards a reference architecture of crowd-sourcing integration in early warning systems. In *Proceedings of the 11th International ISCRAM Conference*.
13. Munich RE (2014) Annual statistics 2014. Retrieved from http://www.munichre.com/en/reinsurance/business/non-life/natcatservice/annual-statistics/.
14. OMG (2004) UML Superstructure Specification, v2.0. Retrieved from http://www.omg.org/spec/UML/2.0/
15. OMG (2011) Business Process Model and Notation (BPMN). Retrieved from http://www.omg.org/spec/BPMN/2.0/
16. Poser, K. and Dransch, D. (2010) Volunteered geographic information for disaster management with application to rapid flood damage estimation.*Geomatica*, 64(1):89–98.
17. Santos, D. S.; Oliveira, B.; Guessi, M.; Oquendo, F.; Delamaro, M.; Nakagawa, E. Y. (2014) Towards the evaluation of system of systems softaware architecture. *VIII Workshop em Desenvolvimento Distribuido de Software, Ecossistemas de Software e Sistema de Sistemas (WDES)*, Maceió, Brazil, p. 53–57, 2014.
18. Schnebele, E., Cervone, G., and Waters, N. (2014) Road assessment after flood events using non-authoritative data. *Natural Hazards and Earth System Science*, 14(4):1007–1015.
19. Schnebele, E., Oxendine, C., Cervone, G., Ferreira, C., and Waters, N. (2015) Using non-authoritative sources during emergencies in urban areas. In Helbich, M., Jokar Arsanjani, J., and Leitner, M., editors, *Computational Approaches for Urban Environments, volume 13 of Geotechnologies and the Environment*, pages 337–361. Springer International Publishing.
20. UNISDR, United Nations Office for Disaster Risk Reduction (2014) Brazil – disaster statistics. Retrieved from http://www.preventionweb.net/english/countries/statistics/?cid=24.