Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 119 (2015) 1366 – 1374

Procedia Engineering

www.elsevier.com/locate/procedia

13th Computer Control for Water Industry Conference, CCWI 2015

Implementation of OGC Compliant Framework for Data Integration in Water Distribution System

Ambreen Hussain[a], Wenyan Wu[a]*, Gabriel Anzaldi[b], Andreas Abecker[c]

[a]*Staffordshire University, Beaconside Road, Stafford, ST18 0AD, United Kingdom*
[b]*Barcelona Growth Centre building, C/ Roc Boronat, 117 planta 508018 Barcelona Spain*
[c]*Disy Informationssysteme GmbH, Ludwig-Erhard-Allee 6, Karlsruhe, Germany*

Abstract

This paper focuses on implementation of a generic OGC compliant framework to integrate data in Water Supply Distribution Systems using interoperable standards. The architecture of this framework was generic to merge clients' data irrespective of the data format. The Integration Manager (IM) of the architecture processes the data and transforms it into WaterML 2.0 format and exchange observations using a standard web service (OGC SOS).The contribution of this paper in water industry is the use of OGC Standards. Then, an implementation of a generic framework for real case studies to integrate water data.

Peer-review under responsibility of the Scientific Committee of CCWI 2015

Keywords: Web Services; Data Integration; Ontology; Service Oriented Architecture; Hydraulic Information System

1. Introduction

A water supply distribution is a system of engineered hydrologic and hydraulic components to supply water to consumers. A successful water supply system meets the water demand, quality and distribution system requirements

* Corresponding author. Tel.:+44-1785-353324.
E-mail address: w.wu@staffs.ac.uk

Peer-review under responsibility of the Scientific Committee of CCWI 2015
doi:10.1016/j.proeng.2015.08.984

such as maintaining pressure to ensure the durability of supply sources. Information collection, data exchange and system monitoring takes place in each part of the water supply distribution system architecture to keep track of water flow and volume, ensure proper operations and detect system abnormalities. Water supply distribution managers need information from different parts of the system in order to perform a decision making and planning procedures. So, integrating all these segments of the system in an interoperable platform will help managers to make effective decisions.

Information in the water supply distribution chain can be collected by in-situ observations, by reuse of geospatial data and by the use of remote sensors. Moreover, observations and multiple software elements such as decision support systems (DSS) [16], demand management systems (DMS) or hydrological forecast systems provide water managers more information to be analyzed. Regardless the information availability, it consumes time to integrate this data from different resources because of the heterogeneity between systems and data. Also, the information consumes time to be analyzed and manipulated in order to make effective decisions. For example, information analysis to generate proper decisions which can support the match between water supply with water demand while minimizing the energy consumption. Today, water supply distribution managers in Europe use separate tools to gather data from many distributed resources and use this data in the decisional systems. All these tools use different ways of communication. These communication mechanisms are neither standardized nor interlinked.

To overcome these problems, a framework was proposed in this research to allow organizations in water industry to integrate data to accomplish decisional tasks overcoming drawbacks presented by current interoperability approaches taken elsewhere such as CUAHSI. The architecture of this framework was generic so that no matter what form client's data is in, it will be processed and integrated with other clients' data. For example, the data can be in the form of Excel files, database or a web service. The Integration Manager (IM) of the architecture processes the data and transforms it into the form to bring consistency such as a standard web service to exchange observations (OGC SOS) into WaterML 2.0 format. This framework took into account the standards for information exchanged, integrate data and facilitate the interrelation between building blocks thus helping the water managers. The main problem during the integration of disparate data from different resources was the data heterogeneity which can be semantic or syntactic. Syntactic heterogeneity refers to different formats of data representation and storage i.e. use of text files, databases or spreadsheets. Semantic heterogeneity refers to difference in perception and expression of similar terminology. Semantic heterogeneity can be further classified into 1) structural – e.g. language used to describe the names of the observation attributes; and 2) contextual – e.g. the language to encode observation attributes values [14] [15].

Syntactic heterogeneity can be overcome by using standardized languages such as WaterML, EML etc. and semantic heterogeneity can be overcome by building a common knowledge base for the systems that need to be integrated. This can be achieved by using Ontologies which was also included in this research. Ontology is the building of vocabulary to represent knowledge of a particular domain that can be used by people, databases and applications that want to share knowledge [13] [17].

OGC standards were used in building this framework such as SOS (Sensor Observation Service) and WaterML 2.0. A SOS [10] provides an API for managing deployed sensors and retrieving sensor data specifically "Observation Data". The primary focus of SOS is to provide access to observations from sensors and sensor systems in a standard way that is consistent for all sensor systems including remote, in-situ, fixed and mobile sensors. The approach that has been taken in the development of SOS is to carefully model sensors, sensor systems, and observations in such a way that the model covers all variety of sensors and supports the requirements of all users' sensor data. SOS has three mandatory core methods 1) GetCapabilities 2) GetObservation 3) and DescribeSensor. There are two transactional methods RegisterSensor and InsertObservation and six enhanced methods including GetResult, GetFeatureOfInterest, GetFeatureOfInterestTime, DescribeFeatureOfInterest, DescribeObservationType, and DescribeResultModel. Out of these, RegisterSensor and InsertObservation methods are used in the implementation phase of this framework and will be explained in later sections.

WaterML 2.0 [12] is a standard information model for the representation of water observations data with a specific focus on time series structures, with the intent of allowing the exchange of such data sets across information systems. WaterML2 is implemented as an application schema of the GML 3.2.1, making use of the OGC Observations & Measurements standards.

Section 2 describes the general methodology of the generic framework design and how that methodology applied to develop the framework; section 3 implements the framework using two case studies, Section 4 demonstrates discussion and future work, and Section 5 concludes the paper.

2. Methodology of the Generic Framework

The data in water supply distribution systems is mostly time series and observational. The measured data and its storage for each client are in different formats. At this stage, three steps are considered to integrate existing data from clients namely (1) the data access, (2) the data mapping, and (3) the data export. In this section these steps are briefly described.

2.1. Data Access

The first step is to identify the data source for any particular client. At this point, it is required to consider three main objectives: 1) Access time; 2) standard interface; 3) stability of existing infrastructure. Access time is one critical matter as it is required to provide the observation results as close to real-time as possible. It is possible to use a standard interface to access client's data. The integration manager should use it allowing reusability for the implementation of future pilots. The integration should not compromise the stability of the existing infrastructure. For the design of the integration manager it had become apparent that it will have to allow a highly client specific implementation of the data access. For a different client integration might require:

- Reading the data from a single or multiple databases;
- File parsing;
- Calling web services where format and protocol can highly differ;
- Screen scraping or embedding legacy code into the integration module;
- A mix of the access mechanisms above.

Consequently, the general integration module design must enable the data-access layer to be highly flexible. There is a possibility that interface implementation steps for one client can be used for other clients if they share same infrastructure to make their observation results available.

2.2. Data Mapping

The second step is to map client's data with OGC-compliant documents such as SOS, WaterML 2.0 standards. To perform this task the following strategies can be considered:

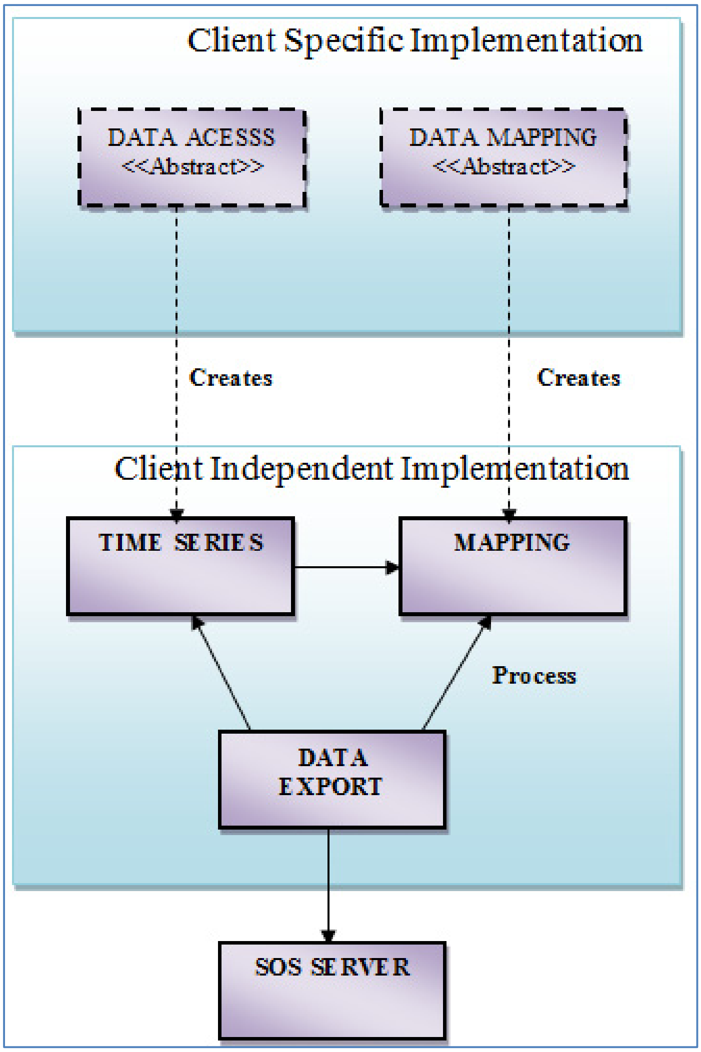
- The data required to map WaterML 2.0 document might be available from client data source but it will have to be transformed through integration manager ;
- If not all the data is provided from client's data sources then integration manager should provide this data.

The implementation of data mapping depends on data supplied by the data access module. In consequence, for general integration manager design same level of flexibility is required for both data access and data mapping modules. After mapping the data is transformed into the required standard document

2.3. Data Export

In this final step the data is exported to the SOS server for sensor registration and observation storage. At this point the results of data access and data mapping consist of same set of information for each client. The generation of WaterML 2.0 as it is based on data mapping and interaction with SOS server is identical for each client. Unlike data access and mapping which are highly client dependent modules data export is independent of client's specific implementation. To interact the data export module with data access and mapping modules, classes are implemented. These classes contain functionality to map time series to generate standardized documents. These documents are transferred over to data export layer from where they are exported to SOS server to perform the required operation.

Fig. 1 General Design Methodology shows the general module design. Data access and data mapping are abstract but have to populate the time series and mapping information using a predefined structure. The data export which consumes the data is independent of the actual client context as the client specifics have no influence on the



export processing.

2.4. Framework Development

Of Many software engineering methodologies are available and being used such as Agile [9], Object-Oriented [3], Component-Based [2], Model-Driven [1], Agent-Based [7] [8] etc. The current design of the framework is using Component-Based software development architecture because of its loose coupled nature. The implementation of designed methodology of framework is shown in Error! Reference source not found.

The clients provide their data in the form of a Database, Excel file or Web Service etc. Client Integration

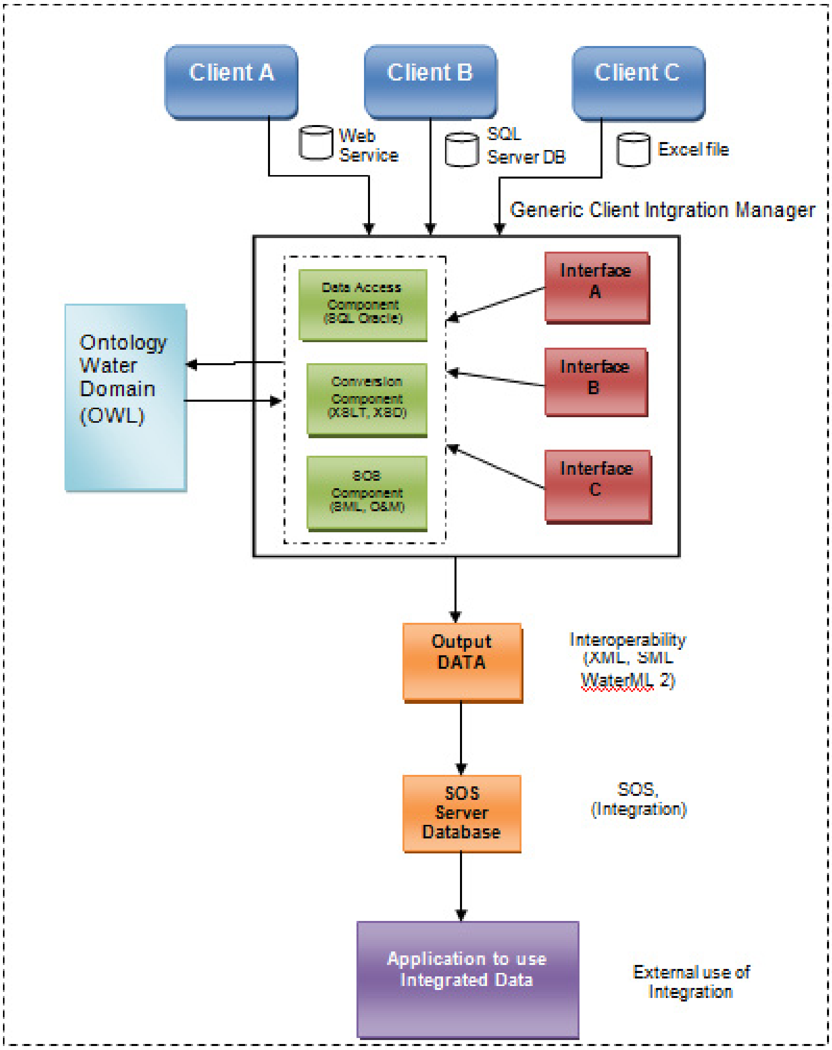


Fig. 2 Generic Integration Framwork

Manager is supported by ontology which is built [16] for each client data using software such as Protégé [11] to map client's terminology and develop the knowledge base. The framework consists of a generic Client Integration Manager Application which is heart of the framework and consists of many components. Each component performs a specific task. The Database component deals with the reading and manipulating of data from the data source, XSLT and XSD components are responsible for providing stylesheets and schema files respectively to conversion components to transform data into required format. The SOS component generates files to register sensors and insert observations into the SOS server. These components are based on some common reusable code base, but also contain own processes. Each client has an interface to use these components according to the requirements. Each interface has its own data access and conversion components. Therefore, this way in case of update in data format or data resource at any client side will need to be updated in that client's interface only. The integration manager uses the ontology in conversion component to overcome semantic heterogeneity by using common knowledge base. For each client XML data files are generated in the form of WaterML 2, SML and O&M which are used to communicate with SOS server database. These provide means of interoperability and can be used by any application in the water distribution chain such as a decision support system, demand management system, or by a large database to store data from all clients at one site. The software is built using Eclipse with Java programming language. The most challenging task was to build the Integration Manager as generic as possible. Many interoperability issues such as data heterogeneity has been considered and resolved by using Ontology and OGC standards while building the framework. Moreover, the framework should be easy to maintain in order to include new procedures or new concepts when a client's new needs emerge or new client participates in the framework. A change at one client site should not affect other client sites. Ontologies should be clarified and understandable by all clients. The viability analysis is performed to evaluate the cost involved building and implementing this framework and the benefits this project can bring in long term [4]. The dependence on network connections for data transference i.e. to/from web server, can be huge and consume considerable amount of time, is also considered [6].

3. Case Studies

In this section the implementation of the generic integration framework for two case studies is presented. For data storage PostgreSQL database is used because it allows PostGIS extension which is used to store spatial data. Both clients use different resources to store and expose data. With the help of the newly build framework these clients will be able to publish the data in a standardized way allowing a homogeneous data access by third parties such as Water Data Warehouse (WDW) and DSS.

3.1. Case Study 1

The first case study represents the lower part of water distribution chain in German city. The interface for this client is a desktop application built using Java programming language and can be installed on a computer by running executable jar files. The Generic framework has been implemented for this client and is currently under testing phase **Error! Reference source not found.**.

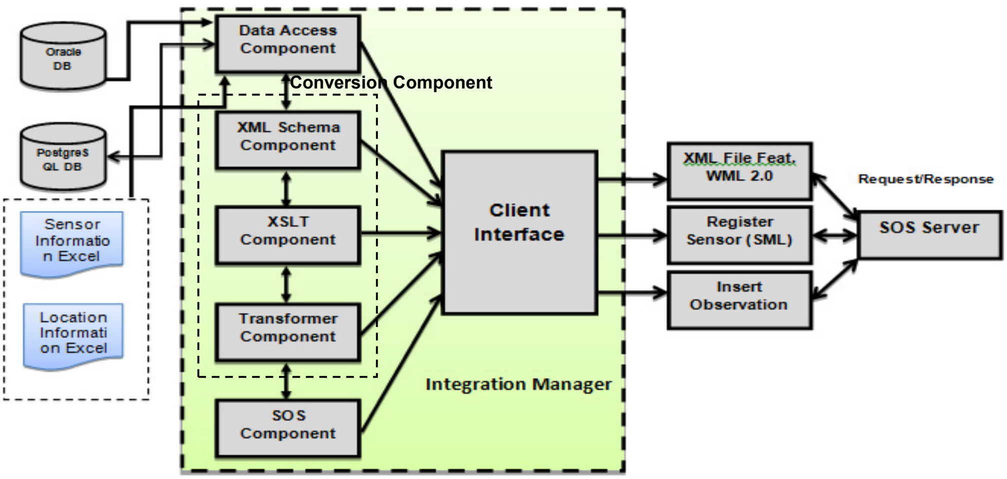


Fig. 3 Client's Integration using new framework

3.1.1. Data Access

The client provides time series data in an Oracle Database and sensor information and location data in Excel sheets. The client interface instructs the data access component of the integration manager to take the sensor and location data from the excel sheets and stores it inside PostGreSQL database. In next step data access component retrieves the sensor data from PostGreSQL database and matches it with time series data from Oracle database.

3.1.2. Data Mapping

The XML schema Component works with the data access component through the client interface for data mapping. When data is mapped the XSLT wrapper works with the Transformer component through an interface layer to transform the data into required standardized document to perform functions through SOS i.e. Register Sensor and Insert Observation

3.1.3. Data Export

In the last step, the interface layer uses the SOS component to export data mainly in three forms: 1) WaterML 2.0 document to insert time series data; 2) Register Sensor request in the form of SML to register sensor; 3) Insert Observation document in the form of O&M to insert observational data. The exported data is transferred through interface layer with the help of SOS component to SOS server. This sensor is registered if not already present in the SOS and observation is inserted if not present and updated if present on SOS server.

3.2. Case Study 2

The second case study represents the upper part of the water distribution chain in Spain region. This client installed a WaterOneFlow server as a central instance to obtain sensor-observation results- WaterOneFlow was developed by the Consortium of the Advanced of Hydrological Sciences Inc. (CUAHSI) [4] for use in U.S. as a standard mechanism for the transfer of hydrologic data. WaterOneFlow web service uses WaterML1 for encoding hydrological observations. WaterML2 is significantly different from WaterML1 as WaterML2 is based on OGC standards Sensor Observation Service (SOS), Observations and Measurements (O&M) and Sensor Model Language (SensorML). The framework has been implemented for this client too and currently undergoing testing phase. The data was in the form of WaterML1 and published on the CUAHSI web service WaterOneFlow Fig. 4. The application for this client is web based and can be deployed on the same server as SOS. For current implementation it is deployed on Tomcat server. The application starts on server startup and triggers data access component to

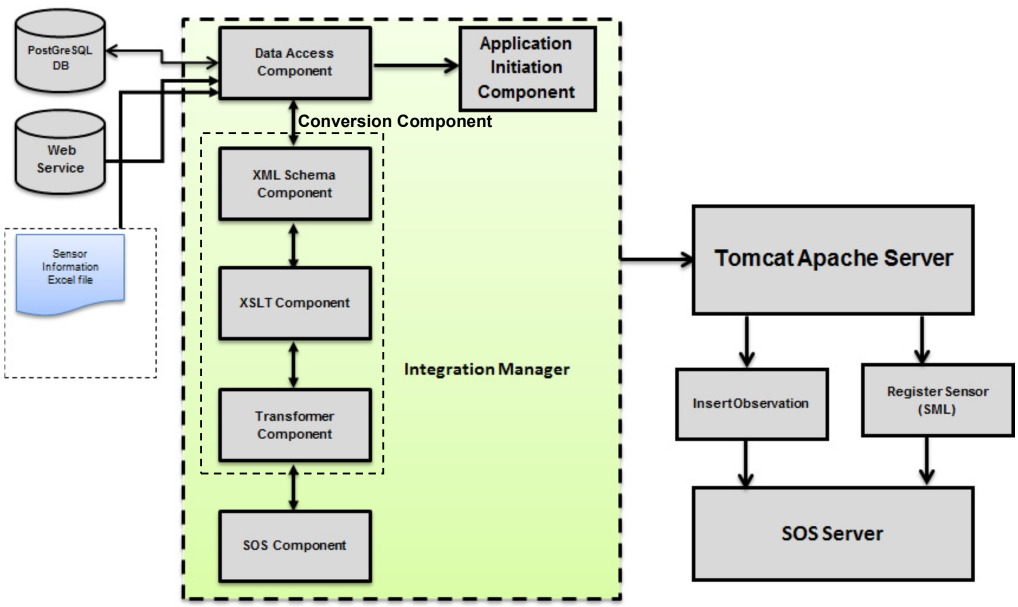


Fig. 4 Second Client's data integration using new Framework

perform the next step.

3.2.1. Data Access

This client provides some sensor data in excel sheets, other sensor data and time series data by exposing WaterOneFlow web service. The Application Initiation Component instructs the data access component of integration manager to transfers the sensor data from excel sheets to PostGreSQL database. In next step the database component retrieves sensors' specific information from database one by one and query WaterOneFlow web service to get time series and other related sensor data.

3.2.2. Data Mapping

In the next step data mapping is triggered and works similarly as in previous client with the help of Xml schema and XSLT components. The data is mapped to transform it to Register Sensor and Insert Observation request documents to send them to SOS server. The Register Sensor request is encoded in SML and Insert Observation in O&M.

3.2.3. Data Export

In this step the SOS component is triggered which exports the documents through sensor observation service to request SOS server to register the sensor if not registered and insert the observations if not already present. This component after sending the request retrieves the response from SOS server as well. The component checks if the observation is already entered in the SOS for a particular sensor. If the observation is present in the SOS it updates the observation if not it inserts the observation.

4. Discussion and Future work

Both case studies are under testing phase at the moment. The framework is generic because the client's data is in different formats and stored in different resources but merges at one place and output application for one client is desktop and for the other is web application by using the same framework. In this regard factory design pattern will be used in future so that changes will not be required to make in components for each client. The implementation of the framework for these case studies has shown that SOS transaction operations have made possible integrating data at one large data warehouse (WDW) for both clients through this framework. The integrated data can be exposed through a web service to DSS, DMS or other applications instead of large database which will be part of future work. Investigation will be performed on Model-Driven Software Architecture to see if it can be applied to build the integration manager in next iteration.

5. Conclusion

Regardless of the provision of data and system heterogeneity, the developed generic framework integrates water supply distribution system by using a common knowledge base driven by Ontology and OGC standards to enable interoperability. The framework is composed of an Integration Manager (IM) which collects the data from clients no matter what form and type of storage data is in. This data is mapped and processed by the IM to generate standardized documents which are transferred to SOS server database to integrate data at one place. The integrated data can be used by water authorities in applications in water supply distribution chain systems to analyze the aggregated data and perform decisional tasks. The novelty in water industry brought by the project is the building of generic Integration manager and use of OGC standards WaterML 2.0 and SOS.

Acknowledgements

The authors acknowledge the financial support of the European Commission's Seventh Framework Program (FP7) under the grants WatERP (318603) and FP7 Marie Curie Actions SmartWater (PIRSES-GA-2012-318985).

References

[1] A. Achilleos, G. M. Kapitsaki, and G. A. Papadopoulos, A Model-Driven Framework for Developing Web Service Oriented Application, *Model-Driven Web Engineering Workshop*, ICWE, 2011.
[2] M. Aoyama, New Age of Software Development: How Component-based Software Engineering Changes the Way of Software Development?, International Workshop on Component-Based Software Engineering, 1998.
[3] G. Booch, Object-Oriented Development, IEEE Transactions of Software Engineering, Vol. SE-12, No. 2, 1998, pp. 211-221.
[4] C. Campos, I. Martí, R. Grangel, A. Mascherpa, R. Chalmeta, A methodological proposal for the development of an interoperability framework. In: Model Driven Interoperability for Sustainable Information System, Proceedings of MDISIS. CEUR-WS, Vol. 340, 2008, pp. 47-57
[5] CUAHSI-HIS, "Hydrologic Information System", Consortium of Universities for the Advancement of Hydrologic Science Inc., 2010, http://his.cuahsi.org/
[6] J. Goodall, J. Horsburgh, T. Whiteaker, D. Maidment and I. Zaslavsky, A First Approach to Web Services for the National Water Information System, Environmental Modelling & Software, Vol. 23, 2008, pp. 404-411
[7] M. Huhns, M. Singh, M. Burstein, K. Decker, E. Durfee, Research Directions for Service-Oriented Multiagent Systems, IEE Internet Computing, Volume 9, Issue 6, 2005 pp. 65-70
[8] N. Jennings, On Agent-Based Software Engineering. Artificial Intelligence, 2000, 117, 277-296
[9] J. Highsmith, A. Cockburn, Agile Software Development: The Business of Innovation, Computer, v.34 n.9, 2001, p.120-122
[10] A. Na, M. Priest, OGC Implementation Specification: Sensor Observation Service (SOS), Version 1.0.0 (06-009R6), Wayland, MA, USA, Open Geospatial Consortium Inc. 2007.
[11] Protégé, protégé.stanford.edu, Stanford University, 2014.
[12] P. Taylor, "OGC Implementation Specification: WaterML 2.0: Part 1- Timeseries", (10-126r3), Wayland, MA, USA, Open Geospatial Consortium Inc. 2012.
[13] J. Heflin, OWL Web Ontology Language Use Cases and Requirements, W3C Proposed Recommendations 15 December, available at http://liris.cnrs.fr/amille/enseignements/Ecole_Centrale/projets_2004/owl.pdf. 2003
[14] J. Horsburgh, D. Tarboton, M. Piasecki, D. Maidment, I. Zaslavsky, D. Valentine, T. Whitenack, An Integrated system for publishing environment observation data, Environment Modelling & Software, Vol. 24, 2009, pp 879-888.
[15] A. Sheth and J. Larson, Federated database systems for managing distributed heterogeneous, and autonomous databases, ACM Computing Surveys, Vol. 22, No. 3, 1990, pp. 186-187.
[16] G. Anzaldi , E. Rubion, A. Corchero, R. Sanfeliu, X. Domingo, J. Pijuan and F. Tersa, Towards an enhanced knowledge-based Decision Support System (DSS) for Integrated Water Resource Management (IWRM), 16th conference on Water Distribution System Analysis (WDSA), Procedia Engineering 89, 2014, pp. 1097-1104.
[17] G. Anzaldi, C. Chomat, E. Rubion, A. Corchero, A. Moya, C. Moya, J. Ciancio, J. Helmbrecht, A holistic ICT solution to improve matching between supply and demand over the water supply distribution chain. In: Proc .of 8th conference on sustainable development of energy, water and environment systems 2013, pp. 0997-1 - 0997-12.