QuPreSS: A Service-Oriented Framework for Predictive Services Quality Assessment

Silverio Martínez-Fernández[1], Jesús Bisbal[2] and Xavier Franch[1]

1. GESSI, Universitat Politècnica de Catalunya (UPC), C/Jordi Girona 1-3, 08034, Barcelona, Spain, email: {smartinez, franch}@essi.upc.edu

2. Dept. ICT, Universitat Pompeu Fabra (UPF), C/Tánger 122-140, 08018, Barcelona, Spain, email: jesus.bisbal@upf.edu

Abstract Nowadays there are lots of predictive services for several domains such as stock market and bookmakers. The value delivered by these services relies on the quality of their predictions. This paper presents QuPreSS, a general framework which measures predictive service quality and guides the selection of the most accurate predictive service. To do so, services are monitored and their predictions are compared over time by means of forecast verification with observations. A systematic literature review was performed to design a service-oriented framework architecture that fits into the current body of knowledge. The service-oriented nature of the framework makes it extensible and interoperable, being able to integrate existing services regardless their heterogeneity of platforms and languages. Finally, we also present an instantiation of the generic framework architecture for the weather forecast domain, freely available at http://gessi.lsi.upc.edu/qupress/.

1 Introduction

A service system is a dynamic configuration of people, technologies, organisations and shared information that create and deliver value to customers and other stakeholders [6]. Examples of customers receiving a service are: a person taking a bus to go somewhere or going to a restaurant to have a meal, or an information technology (IT) company subcontracting the information backup process to another company in order to save costs and time.

Service-oriented architecture (SOA) has become in the last years the most usual strategy to implement service systems. Basically, this emerging development paradigm allows service providers to offer loosely coupled services. These services are normally owned only by the providers. As a result, the service customer (or client or user) does not need to worry about the development, maintenance, infrastructure, or any other issue related to the service operation. Instead, the customer just has to find and choose the most appropriate service.

Due to the success of service technology, there is a big offer of services covering many domains. Therefore it is needed to assess which service is the most appropriate for the customer's needs. Each customer has different needs, which require to be satisfied by the provider: quality, reputation, cost, security, personalisation, locality and so on.

Among all kind of services, we concentrate on predictive (or forecasting) services. We define *predictive services* as those services whose main functionality is to show in advance a condition or occurrence about the future. Predictive services emerge in a lot of domains to predict: stock market prices, results in bookmakers, election polls, sales forecasting and so on. We are quite used to see this functionality offered by websites that provide forecasts over specific data [15], but they are not so often exposed as web services (WS).

An example that is really familiar to all of us is weather forecast. Weather conditions affect our decisions in daily routines such as deciding what to wear first thing in the morning or planning a trip. To make these decisions, different services like the weather forecast section on TV news or specialised web sites (such as forecastadvisor.com and forecastwatch.com) are consulted. However, sometimes their predictions do not agree and therefore a fundamental question arises:

> Given a portfolio of candidate predictive services, which one is expected to be the most accurate to satisfy some given user needs?

In order to answer the question above, two related activities become necessary. On the one hand, *service monitoring* by using a monitoring infrastructure that observes the behaviour of such predictive services [3]. On the other hand, *forecast verification*, defined as the process of assessing the quality of a forecast by comparing it with its corresponding observation [14]. Both activities need to exist together: forecast verification cannot be implemented without monitoring real data, and it makes no sense to establish a monitoring infrastructure if quality analysis is not done. Therefore, a general framework that performs both activities is needed in order to properly answer the question above.

This paper presents an approach to evaluate the quality of predictive services. It has four objectives, which are respectively covered in the following four sections, that are: (1) to understand the state-of-the-art in predictive services quality assessment; (2) to identify gaps in current research about this problem; (3) to propose a generic framework software architecture to support predictive service monitoring, verification and selection; and (4) to develop an instantiation of the generic framework software architecture as a proof-of-concept in the weather forecast domain.

2 State of the Art in Predictive Services

To make a thorough and unbiased state-of-the-art about forecast verification for predictive services, we have performed a Systematic Literature Review (SLR) following the guidelines of [9]. In Section 2.1, an excerpt of the systematic review protocol is presented. A summary of the results is reported in Section 2.2. To see all the details and the rest of the SLR, the reader is referred to [11].

2.1 Systematic Literature Review Protocol

The protocol consists of three main stages: identifying the research questions that the review is intended to answer, formulating the strategy to search for primary studies, and determining the study selection criteria and procedures.

Research questions (RQ). The aim of this review consists of identifying the existing research on predictive services in order to measure how accurate their predictions are. It addresses the following questions:

RQ1. Are there existing SOA frameworks which measure predictive service quality?
RQ2. In which domains such frameworks are being applied?
RQ3. Which are the main criteria used to evaluate predictive services?

Search strategy. Our primary interest is finding service-oriented frameworks for forecasting. We started then with a search string with two parts: different synonymous for “forecast” looking for the functionality, and three different keywords for service orientation (the general term “service-oriented”, the architecture “SOA” and the technology “web service”) covering the target platform:

1. (forecast OR forecasting OR foresight OR foretell OR foretelling OR forethought OR predict OR prediction OR predictive OR predictability OR predicting OR prognosis OR prognosticate OR prognostication OR prevision OR anticipation OR outlook) AND ("service-oriented" OR SOA OR "web service")

However, the systematic search with this string was not promising. We understood that the search could be too specific so we decided to complement this search string with another one more general, looking for state-of-the-art papers on service orientation that can include thus forecasting as part of the results:

2. (SOA OR "service-oriented") AND ("systematic review" OR survey OR "state-of-the-art")

Study selection. We rigorously identified 213 studies from the literature with the search strategy, of which 18 passed the selection criteria (which is based on three stages processing title, abstract and full text, respectively) and quality assessment. The included studies, which are presented in Table 1, covered three topics: SOA, monitoring and prediction in SOA systems. The column document type indicates if the paper is a journal article (JA), a conference paper (CP), a technical report (TR), or a master thesis (MT).

Table 1 Primary studies considered in the systematic literature review

Authors	Year	Type	Doc. Type	Ref.
L.B.R. de Oliveira et al.	2010	SOA	JA	[16]
M.H. Valipour et al.	2009	SOA	CP	[21]
M.P. Papazoglou et al.	2007	SOA	JA	[19]
U. Zdun et al.	2006	SOA	JA	[23]
G. Canfora et al.	2009	Monitoring	CP	[2]
M. Oriol	2009	Monitoring	MT	[17]
A.T. Endo et al.	2010	Monitoring	TR	[5]
A. Bertolino	2007	Monitoring	CP	[1]
N. Delgado et al.	2004	Monitoring	JA	[3]
R. Guha	2008	Prediction	JA	[7]
S. Punitha et al.	2008	Prediction	CP	[20]
M. Marzolla et al.	2007	Prediction	CP	[12]
H.N. Meng et al.	2007	Prediction	CP	[13]
D.M. Han et al.	2006	Prediction	CP	[8]
C.B.C. Latha et al.	2010	Prediction	JA	[10]
N. Xiao et al.	2008	Prediction	JA	[22]
A.H. Murphy	1993	Prediction	JA	[14]
B. Domenico	2007	Prediction	TR	[4]

2.2 Review Results

The data extraction and synthesis of knowledge arisen from the SLR are discussed below in relation to the three research questions.

RQ1: SOA Frameworks which measure predictive service quality. The literature review found only one service-oriented framework which measures predictive service quality [4]. In this work, geosciences web services are used to integrate sources of data. Once the data is collected, they perform forecast verification with observations.

Nevertheless, the SLR located other service-oriented frameworks that monitor and/or predict but do not consider forecast quality.

There are numerous monitoring frameworks to get quality of services (QoS) and to alert when a service level agreement (SLA) between a provider and a customer is predicted to be violated (for details, the reader is referred to the related work section of [18]). Some of these monitoring frameworks have their own prediction models to predict the performance of services in order to balance the workload in a composite SOA. Other studies predict the performance of services either to help users during the selection process of a provider [12] or to improve the reliability of services using these performance predictions (e.g., to detect bottlenecks at design time [20], to promote efficiency [22] or to avoid a service crash because of exceeding memory usage [13]).

On the other hand, predictions models which require huge amounts of data from different sources and vast computing power are starting to use SOA due to its benefits. We found examples in the weather forecast domain [10], macroeconomic domain [8] and drug design domain [7].

RQ2: Predictive domains. The only domain in which we have found existing SOA frameworks that measure predictive services quality is weather forecast [4].

Moreover, we found other composite SOA frameworks and simple WS which do not measure how accurate predictive (web) services are, but perform predictions. Their domains were: weather forecast [10], macroeconomic analysis [8], drug design [7] and QoS prediction with the final goal of improving composition of services and service's reliability [22]. It is worth to note other relevant predictive domains that we found in the excluded studies but whose systems are not service-oriented [11]: business and economic forecasting, automotive forecasting, prediction of flight delays, prediction of protein's issues in medicine, sales forecasting for the calculation of safety stock and results in betting shops.

RQ3: Criteria to evaluate predictive services. There are three criteria that determine if a forecast is good (or bad), namely consistency, quality and value [14]. Consistency is the correspondence between judgments and forecasts. There exists a difference between judgments (forecaster's internal assessment which is only recorded in his/her mind) and forecasts (forecaster's external spoken/written statements regarding future conditions). Quality is the correspondence between forecasts and observations (i.e., the accuracy of the forecasts). A forecast has high quality if it predicts the observed conditions well according to some aspect like accuracy and skill. Value refers to the incremental benefits of forecasts to users. A forecast has value if it helps the user to make a better decision.

In this paper, we focus on quality because of three reasons. First, it is useful in the forecast verification process. Second, consistency could not be entirely projected onto a software prototype, since it contains an element of uncertainty and subjectivity by definition. Third, value requires knowledge about the impact of predictions which is out of the IT scope of this paper.

SLR Conclusion. As a main result, we can conclude that forecast verification for predictive services has not received sufficient attention and it justifies the focus of further research. More effort is necessary to integrate methods for forecast verification in SOA monitoring frameworks. This has motivated our work.

3 Predictive Services Quality

As we mentioned in Section 1, given a portfolio of candidate predictive services, we aim to find the most accurate to satisfy some given user needs. To this end, we need the following four inputs, which are summarised in Fig. 1:

- Predictive services. The set of sources that give predictions.
- Ground truth service. It refers to the information that is collected on location (i.e., the real observations gathered by sensors once they happen). This source is trusted and reliable, hence only one is needed.
- Predictive context. It consists of relevant context conditions for predictions (like current date, location, etc).
- Customer query. The desired predictions in a specific domain.

Among all available predictive services, the one with the highest likelihood to make right predictions should be used. By performing forecast verification with the above inputs, the predictive service with the highest quality is selected.

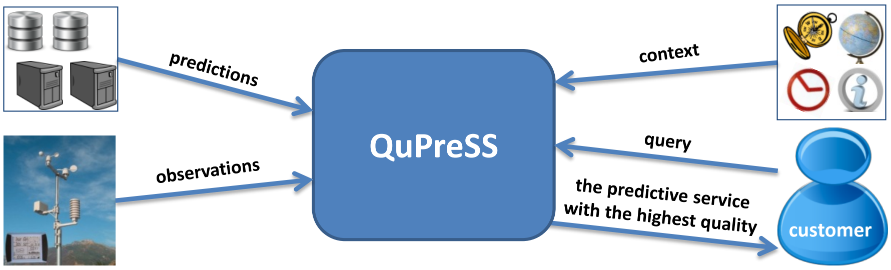

Fig. 1 Inputs and outputs of the QuPreSS (QUality for PREdictive Service Selection) framework

More formally, the problem targeted by the QuPreSS framework can be announced as follows.

The prediction problem. Let $PS = \{PS_1, PS_2, \ldots, PS_n\}$ be a set of predictive services in a given domain, GT the ground truth for this domain, C the context of the prediction, and Q the customer query as follows:

- $PS_i : Time \text{ x } D_0 \rightarrow D_1$
- $GT : Time \text{ x } D_0 \rightarrow D_1$
- $C \subseteq \langle Y_1, Y_2, \ldots Y_m \rangle$
- $Q \subseteq \langle Time \text{ x } D_0 \rangle$

where *Time* is the moment for which a prediction or observation is wanted, D_0 the parameters to every available query dimension that define the conditions or occurrences to be predicted or observed, D_1 the prediction or observation, and Y_i the value of every relevant context condition.

Then the prediction problem can be defined as: to find a prediction function $pred(PS, GT, C, Q) \rightarrow PS$ such that:

$$\forall i: 1 \leq i \leq n: quality(pred(PS, GT, C, Q), GT, C, Q) \geq quality(PS_i, GT, C, Q)$$

where *quality* is the function that measures the forecast quality of a predictive service PS_i for the query Q, given the context C and the data gathered from the ground truth GT.

Examples of quality measures appear in Section 5. There are two ways to calculate quality: measured-oriented calculation focuses on one or two overall aspects of forecast quality (e.g., accuracy and skill) whereas distribution-oriented calculation constitutes forecast quality in its fullest sense. For example, the distribution-oriented approach allows comparing two or more sets of forecasts [14].

Forecast quality is inherently multifaceted in nature. Various aspects of forecast quality are: bias, association, accuracy, skill, reliability, resolution, sharpness, discrimination and uncertainty. To see their definitions and relevant distributions, the reader is referred to [14].

4 Framework Architecture

In this section we present a software architecture that implements the prediction problem as announced in the previous sections. The most important functional requirement of our architecture is to select the most appropriate predictive service to satisfy a customer's request according to some evolvable forecasting model. Subordinated to this main requirement, we have identified others: show the prediction of the highest ranked service, monitor registered predictive services, save their predictions, and get and save the real behaviour according to some trusted ground truth service. Besides, it should offer data to external systems and services. Among the non-functional requirements, the platform shall be: extensible to add new predictive services, adhered to standards to facilitate interoperability, able to process continuous data flows, compatible with existing components when possible, aligned with legal statements of external sources, and robust.

In order to fulfil the previous requirements, and following Section 3 above, the architecture presented in Fig. 2 was designed. Its components are:

- External sources: the Ground Truth service and a portfolio of predictive services. They are external and heterogeneous. They can be implemented by WS or other technologies. To integrate all these heterogeneous sources, we chose an SOA. If an external source is not exposed as a WS, it is wrapped into a WS proxy. Predictive services have to provide forecasts following a pre-defined document format to be easily integrated, or otherwise they are also wrapped into WS proxies to homogenise these formats.

- A Monitor service to save in a systematic manner the QoS of each predictive service and the response for every periodical request to the WS. These requests contain the predictions.
- The Forecasting Data Collector service, to collect both the ground truth and the predictions gathered by the Monitor service.
- Two database management systems (DBMS): one in charge of saving observations (Ground Truth Database) and another one to save predictions (Forecast Data Database). In those prediction domains where predictions change very frequently, it would be better to use data stream management systems (DSMS) instead of DBMS to process continuous data flows.
- The QuPreSS web application and the QuPreSS service which show/offer the results of the QuPreSS framework. The QuPreSS web application is designed for human customers whereas external services can use the QuPreSS service. They verify predictions quality by means of the Forecast Verifier component. Both of them serve to support decisions about choosing between different potential predictive services and to directly redirect to the most accurate one (with the help of the Invocator component).

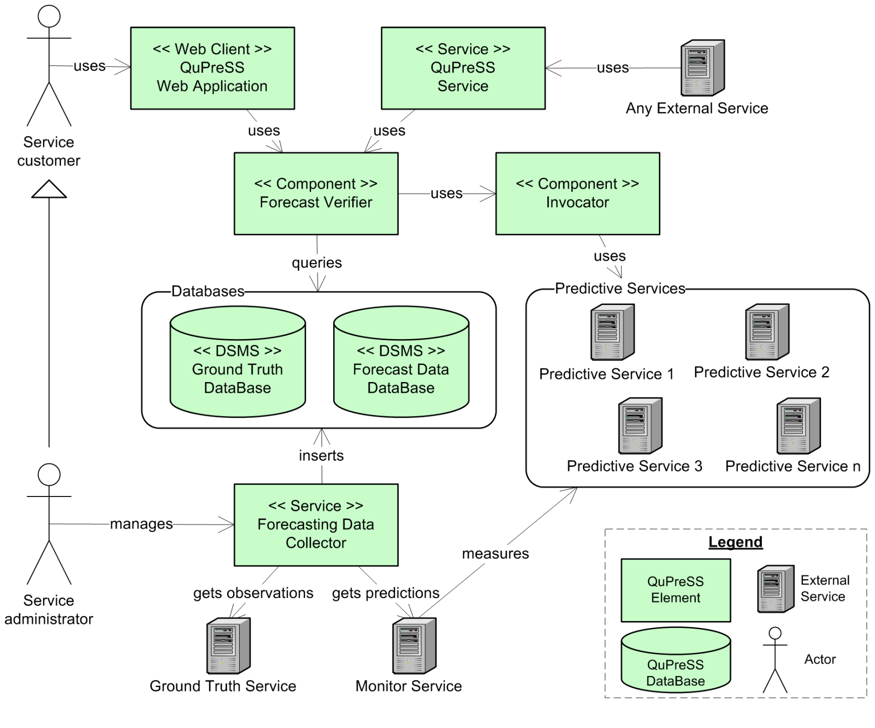

Fig. 2 QuPreSS framework architecture

Fig. 2 shows two human actors: the Service Customer and the Service Administrator. The Service Administrator manages the forecasting data collector service. After configuring this service, the system is set up to read, parse and save prediction data from registered predictive services, as observations from the ground truth service. On the other hand, either a service customer or external services get observations, predictions and services quality assessment from the QuPreSS framework.

5 An Exemplar Implementation for the Weather Forecast Case

In this section, the development of a prototype is discussed as a proof-of-concept. It fulfils the architecture requirements defined in the previous section, since it is an instantiation of the QuPreSS framework with all its elements for the weather forecast domain. These elements have been implemented using WS technologies, MySQL databases and Java Web technologies. Besides, it uses the existing SALMon monitor [18], invoking the services that its SOA offers.

During the selection process of predictive services [11], we found that most of the WS which provide weather forecasts for cities all over the world are not free. However, there are other kinds of sources that without being WS (e.g., XML files), provide free weather forecasts. Thus, we deployed WS proxies which use them as a backend. In order to facilitate the integration of new services, the format of the response exposed by these proxies is the same for all of them. As a result, with this approach we resolved both the lack of free WS and the integration of heterogeneous sources.

The ground truth is obtained from AEMET (the Spanish Meteorological Agency) weather stations[1]. On the other hand, the SALMon monitor service saves in systematic manner predictions for four Catalan cities. These predictions come from the following predictive services: AEMET forecasts[2], open data Meteocat[3] (the Catalan Meteorological Agency), and Yahoo! Weather RSS feed[4].

In this proof-of-concept, the accuracy parameter taken into account was how precise the forecasts of high and low temperatures are. To verify the correctness of predictive services, we used the mean-squared error (MSE) and the approximation error (AE). Both of them quantify the difference between values implied by an estimator and the true values.

The formula to calculate the MSE of a predictive service *PS* is:

[1] ftp://ftpdatos.aemet.es/datos_observacion/resumenes_diarios/

[2] http://www.aemet.es/es/eltiempo/prediccion/municipios

[3] http://dadesobertes.gencat.cat/ca/dades-obertes/prediccions.html

[4] http://developer.yahoo.com/weather/

$$MSE_{PS} = \frac{\sqrt{\sum_{i=1}^{n}\left(\mu_{T_{GT}} - T_{PS_i}\right)^2}}{n} \tag{1}$$

where μ_{TGT} is the average of temperatures (high or low) of the corresponding observations (ground truth), T_{PSi} is a predicted temperature (high or low) from the *PS*, *i* is an index (which refers to a date), and *n* is the total amount of observations from the *GT* and predictions of the *PS*.

The AE measures the discrepancy between an observation and its prediction:

$$AE_{PS} = \frac{\sum_{i=1}^{n} |T_{GT_i} - T_{PS_i}|}{\sum_{i=1}^{n} |T_{GT_i}|} \tag{2}$$

where *T* indicates a high or low temperature that is either a prediction from the *PS* or its corresponding observation from the ground truth (*GT*), *i* is an index (which refers to a date), and *n* is the total amount of observations/predictions compared.

We had a limitation regarding the availability of several different observations that would correspond to a given prediction, since predictions refer to an area (e.g., Barcelona), but observations are taken in very specific locations (e.g., Sagrada Familia). In this case, we used the most centric (in a geographic sense) observation. Another limitation is to find correspondent observations to specific predictions (e.g., if an observation says that it rained just a drop, can we consider right a prediction that said that is was going to rain?).

Results. High and low temperatures of four cities (Barcelona, Girona, Lleida and Tarragona) have been monitored since July 2011. Every day both real observations from AEMET weather stations and forecasts from three predictive services (AEMET, Meteocat and Yahoo!) are saved in our DBMS.

The Forecast Verifier component calculates prediction errors (MSE and AE) of the three predictive services in a period of time (by default the last two weeks) for a given city. Then, it returns either a ranking of predictive services ordered by quality (so that customers may select themselves, usually the first in the ranking), or the actual forecast of the predictive service with most quality. In the second case, the Invocator component is responsible to directly offer forecasts from the most accurate predictive service, hiding any analysis and redundant forecasts.

The presentation layer has been implemented by a web client application for customers and a WS for external services. This tool is freely available on-line at: http://gessi.lsi.upc.edu/qupress/.

6 Conclusions and Future Work

Forecast quality is of obvious importance to users since forecasts guide their decisions, and also to providers because it affects their reputation. Nevertheless, as we have shown in the SLR results, we can conclude that forecast verification for predictive services has not received sufficient attention and it justifies the focus of further research. More effort is necessary to integrate methods for forecast verification in SOA monitoring frameworks.

The main contribution of this paper is the presentation and development of QuPreSS, a generic SOA framework which performs forecast verification with observations in predictive domains. This architecture is scalable and easy to integrate with other systems and services. The goal of the architecture is to determine which predictive service provides better predictions by assessing the forecast quality of all of them. This assessment assists customers in making better decisions and gives them an edge over the competition. Besides, providers of these services can use this evaluation to improve their predictions.

We have identified general issues that need to be solved. Different predictive services from the same domain may provide predictions with different characteristics, which make them not easily comparable. For instance, when a service provides hail probability, other provides wind speed. Another example, when one service provides forecasts two days in advance, and another does so one week in advance. Likewise, it is needed to know the quality aspect or error that behaves better in a specific case.

At present, more important future work relates to: (1) increasing the amount of monitored services and collect more data about predictions, (2) applying this architecture to other domains apart from the weather forecast domain, (3) identifying the current knowledge about parameters that determine predictive service quality, and (4) studying the forecast value criteria of predictions.

Acknowledgments This work has been supported by the Spanish project TIN2010-19130-C02-01. We would also like to thank Marc Oriol and Marc Rodríguez for their friendly cooperation while integrating QuPreSS with SALMon.

References

1. Bertolino, A.: Software testing research: Achievements, challenges, dreams. In: Proc. of the Future of Software Engineering Symposium (FOSE), pp. 85-103, IEEE-CS Press (2007)
2. Canfora, G., Di Penta, M.: Service-Oriented Architectures Testing: A Survey. In: Proc. of the 31st International Spring Seminar on Electronics Technology (ISSSE), pp. 78-105 (2008)
3. Delgado, N., Gates, A.Q., Roach, S.: A taxonomy and catalog of runtime software-fault monitoring tools. IEEE Transactions on software Engineering 30(12), 859-872 (2004)
4. Domenico, B.: Forecast Verification with Observations: Use Case for Geosciences Web Services. (Available at),

http://www.unidata.ucar.edu/projects/THREDDS/GALEON/Phase2Connections/VerificationUseCase.html. Cited 19 Feb 2012

5. Endo, A.T., Simao, A.S.: Formal Testing Approaches for Service-Oriented Architectures and Web Services: a Systematic Review. Technical Report Instituto de Ciências Matemáticas e de Computaçao 348, ISSN-0103-2569 (2010)
6. Gregory, M., Spohrer, J.: Succeeding through service innovation: A service perspective for education, research, business and government. IBM IfM - University of Cambridge Institute for Manufacturing, ISBN: 978-1-902546-65-0 (2008)
7. Guha, R.: Flexible Web Service Infrastructure for the Development and Deployment of Predictive Models. J. Chem. Inf. Model 48(2), 456–464 (2008)
8. Han, D., Huang, H., Cao, H., Cui, C., Jia, C: A Service-Oriented Architecture Based Macroeconomic Analysis & Forecasting System. In: Proc. of Frontiers of WWW Research and Development (APWeb), LNCS 3841, pp. 1107-1117 (2006)
9. Kitchenham, B.: Procedures for Performing Systematic Literature Reviews. Keele University Technical Report, TR/RE-0401 (2004)
10. Latha, C.B.C., Paul, S., Kirubakaran, E., Sathianarayanan: A Service Oriented Architecture for Weather Forecasting Using Data Mining. Int. J. of Advanced Networking and Applications 2(2), 608-613 (2010)
11. Martínez-Fernández, S.: Accuracy Assessment of Forecasting Services. Master Thesis Universitat Politècnica de Catalunya (2011)
12. Marzolla, M., Mirandola, R.: Performance Prediction of Web Service Workflows. In: QoSA, LNCS 4880, pp. 127-144 (2007)
13. Meng, H.N., Qi, Y., Hou, D., Zhang, Y., Liu, L.: Forecasting software aging of service-oriented application server based on wavelet network with adaptive genetic algorithm. In: Proc. of 3rd International Conference on Natural Computation (ICNC), pp. 353-357 (2007)
14. Murphy, A.H.: What is a good forecast? An essay on the nature of goodness in weather forecasting. Weather Forecasting 8, 281-293 (1993)
15. Nikolopoulos, K., Metaxiotis, K., Assimakopoulos, V., Tavanidou, E.: A first approach to e-forecasting: a survey of forecasting Web services. Information Management & Computer Security 11(3), 146-152 (2003)
16. de Oliveira, L.B.R., Romero, K., Feitosa, D., Yumi, E.: Reference Models and Reference Architectures Based on Service-Oriented Architecture: A Systematic Review. In: ECSA, LNCS 6285, pp. 360-367 (2010)
17. Oriol, M.: Quality of Service (QoS) in SOA Systems: A Systematic Review. Master Thesis Universitat Politècnica de Catalunya (2009)
18. Oriol, M., Franch, X., Marco, J.: SALMon: A SOA System for Monitoring Service Level Agreements. Universitat Politècnica de Catalunya Technical Report, LSI-10-18-R (2010)
19. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges. Computer 40(11), 38-45 (2007)
20. Punitha, S., Babu, C.: Performance Prediction Model for Service Oriented Applications. In: Proc. of 10th IEEE International Conference on High Performance Computing and Communications (HPC), pp. 995-1000 (2008)
21. Valipour, M.H., Amirzafari, B., Maleki, K.N., Daneshpour, N.: A brief survey of software architecture concepts and service oriented architecture. In: Proc. of 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT), pp. 34-38 (2009)
22. Xiao, N., Fu, W., Chen, T., Huang, Q.: A service-oriented monitoring system with a forecasting algorithm of a time sequence-based hybrid model. International Journal of Parallel, Emergent and Distributed Systems 23(2), 137-151 (2008)
23. Zdun, U., Hentrich, C., van der Aalst, W.M.P.: A survey of patterns for service-oriented architectures. International Journal of Internet Protocol Technology 1(3), 132-143 (2006)