Future Generation Computer Systems 28 (2012) 593–604

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

# A service oriented architecture for decision support systems in environmental crisis management

Vassilios Vescoukis, Nikolaos Doulamis *, Sofia Karagiorgou

*National Technical University of Athens, 9, Heroon Polytechiou Str. Zografou, Athens 157 80, Greece*

ARTICLE INFO

*Article history:*
Received 14 May 2010
Received in revised form
2 March 2011
Accepted 11 March 2011
Available online 22 March 2011

*Keywords:*
Environmental modeling
Computer applications
Service oriented architectures
Decision support
Crisis management

ABSTRACT

Efficient management of natural disasters impose great research challenges to the current environmental crisis management systems in terms of both architecture and services. This is mainly due to the fact that a large amount of geospatial content is usually distributed, non-compliant to standards, and needs to be transmitted under a QoS guaranteed framework to support effective decision making either in case of an emergency or in advance planning. Incorporating real time capabilities in Web services, both in terms of dynamic configuration and service selection, is an open research agenda. The things get worst in geospatial context due to the huge amount of data transmitted from distributed sensors under heterogeneous platforms, making the need of synchronization an important issue. In this paper, we propose a flexible service oriented architecture for planning and decision support in environmental crisis management. The suggested architecture uses real time geospatial data sets and 3D presentation tools, integrated with added-value services, such as simulation models for assisting decision making in case of emergency. The proposed architectural framework goes beyond integration and presentation of static spatial data, to include real time middleware that is responsible for selecting the most appropriate method of the available geospatial content and service in order to satisfy the QoS requirements of users and/or application. A case study of a complete, real world implementation of the suggested framework dealing with forest fire crisis management system is also presented.



## 1. Introduction

The great increase of natural disasters (e.g., forest fires, flooding, landslides) has stimulated a great research interest in developing smart and intelligent Environmental Information Management (EIM) Systems able to collect, process, visualize and interpret geospatial data and workflows of added-value applications so as to support decision making in case of emergency [1]. Natural disasters pose a great threat to people's lives and their properties while they present a negative impact to the economies. However, in order to efficiently handle, forecast, mitigate and prevent such disasters, new open scalable and distributed service platforms need to be created [2]. Such Service Oriented Architectures enable integration of heterogeneous geospatial data of different types and format, real time geospatial data streaming and filtering, as well as incorporation of a plethora of new added-value services that allow, not only mash-ups of geospatial data (useful towards an event-based presentation and prediction), but also simulation of natural phenomena and decision making mechanisms. All these issues are addressed in this paper by proposing a novel Service Oriented Information System proper for Environmental Information Management, as well as for planning and decision support in case of emergency. The proposed architecture is in close collaboration with real world stakeholders in civil protection and environmental crisis management, and has been implemented as a real system, currently into production, use for planning and decision support in forest fire crisis management.

In general, environmental modeling is a time consuming task of high complexity. This task usually requires long iterative processes of mathematical computations along with analysis, reformatting and integration of the heterogeneous geospatial data, generally an arduous task due to the existence of multiple models, formats and protocols [3]. Typical examples of environmental modeling include wild fire dispersion simulation, gas flow simulations, oil spills detection and so on. Such tasks are performed by appropriate models, whose choice is based not only on the performance requirements of each specific case, but also on data processing considerations. The traditional monolithic centralized approaches present a series of disadvantages such as limited scalability, isolation and high cost to integrate geospatial datasets coming from diverse sources and independent entities; emphasis is often given on integrating the disparate information instead of supporting the main objective which is decision making and

* Corresponding author. Tel.: +30 210 7722678; fax: +30 210 772 2677.
*E-mail addresses:* v.vescoukis@cs.ntua.gr (V. Vescoukis), ndoulam@cs.ntua.gr (N. Doulamis), karageor@mail.ntua.gr (S. Karagiorgou).

0167-739X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2011.03.010

disaster handling [4]. In other words, when client–server is chosen as the architectural style, quality attributes like performance or maintainability are typically the primary architectural concerns; in the context of SOA, the scope of service pools, the runtime behavior of services, the cooperation with external stakeholders and governance issues become major concerns that need to be dealt with [5], which are key issues in decision making. For this reason, heterogeneous, distributed and scalable service platforms are introduced for environmental modeling [6,7].

The distributed geospatial services and components are integrated using the open framework of Service Oriented Architecture (SOA), or Web services in case of an TCP/IP Internet implementation. Service Oriented Architecture is a flexible set of design principles that enable the development of distributed applications and loosely-coupled solutions whereby service providers advertise the services they offer, and solution providers and software developers access the service repositories to search for suitable services to invoke for a given purpose. Web services is one of the currently available options for the development of a scalable, open, distributed and platform-independent environmental information management system, since it allows for consumption and composition of ready-to-use services, coming from different providers, that they have to be integrated to implement real world applications. Such systems, in their turn, can be available as new services for invocation by end-users or further integration.

Geospatial services are different from the "traditional" computing services, because of the nature of the geospatial information (data) on which they operate [8]. Geospatial data and services are much more complex to integrate due to the variety of existing data models, data formats, data semantics and spatial relationships, which in practice are limiting factors to the interoperability. For this reason, the Open Geospatial Consortium (OGG) [9] has introduced an interface for implementing web services using geospatial data. This interface includes several specifications the most important of which are the Geography Markup Language (GML), the Web Feature Service (WFS) and the Web Map Service (WMS). The Geography Markup Language (GML) is "an XML grammar written in XML Schema for the modeling, transport, and storage of geographic information including both the spatial and non-spatial properties of geographic features" [10]. It is developed as an implementation specification by the Open GIS Consortium to foster data interoperability and exchange between different systems. The WFS is an OpenGIS implementation specification [11] that allows a client to retrieve geospatial data encoded in GML from multiple Web Feature Services. Finally, the Web Map Service Interface Standard (WMS) provides a simple HTTP interface for requesting georegistered map images from one or more distributed geospatial databases [12]. Using these specifications, one can create geospatial service oriented architectures, which retain the main principles of web services using, however, different data description languages able to handle geospatial properties, which are a key element in environmental information management systems.

Another main aspect for an effective environmental information management system is *its real time properties*. In many cases, geospatial information data such as environmental measurements should be transmitted in real time for an efficient decision making. Real time services are data processing or analysis applications exposed as Web Services and connected with each other via a publish/subscribe messaging substrate. Real time data and messages *collected from distributed environmental sensors* are processed using these services. It is clear that real time access in a streaming fashion requires software architectures that are able to fulfill all the performance, availability and reliability requirements of an information management application.

Considering current technology advances, real time sensor measurements are becoming the most popular type of data sources; high spatial density of such sensors which can be achieved as their cost decreases, presents us the requirement to ensure that the capacity to produce and transfer tremendous amount of measurements is available. Such measurements may require more than the processing capacity that "reasonably budgeted" systems can handle. This imposes new challenges in the design principles of a real time geospatial service oriented architecture, since methods for data filtering, visualization and presentation should be incorporated. Processing of huge amounts of data sets requires new architectures due to the large computing resources, network bandwidth, and storage challenges.

Finally, computational methods are used to transform the acquired data to new geospatial datasets that contain new attributes useful for the information management system, such as simulation results. The models can be retrieved from a wide set of available models, using intelligent service selection mechanisms that exploit the current contextual information provided by the real time data sets, together with performance requirements. To do this, context adaptation methodologies are needed not only for the efficient data filtering and simulation model selection, but also for efficient presentation of the geospatial data sets. Although available options are many, integration of all current possibilities in a complete, reliable system with a cost-effective architecture, that can provide added-value services as decision support, remains a challenging research issue.

### 1.1. Previous works

In geospatial domain Alameh was one of the first attempts at addressing the problem of geospatial service chaining or composition [13], following by other methods, such as the work of [14], where distributed components can be created using Common Object Broker Architecture (CORBA) [15], Remote Method Invocation (RMI), or Web Services technologies, which enable components at different geographic locations to communicate. CORBA based technology for geographical distributed applications has been reported in [16]. However, the CORBA model presents a series of limitations such as difficult implementation. For this reason, OGC Web mapping specifications are currently used in the design and implementation of a environmental crisis management system [9].

Chang and Park present an XML web service model for distributed GIS systems [17]. In this paper, the distributed geospatial service are described as XML web services and integrated by client when necessary. OGC specifications are used for interoperably describing geospatial data. However, still many open interoperability issues arise when different tools from different providers are to integrate together [18]. Risk management issues using service oriented architectures have been reported in [19]. The use of distributed geospatial services in *Digital Earth* have been reported in [20]. In particular this paper uses geospatial information processing (DGIP) methods that enable easy integration of widely distributed geospatial resources. In the same frame [21] proposes a Spatial Data Infrastructure (SDI) framework for creating geospatial services by concentrating all the required functionalities in a single, publicity accessible geospatial component. Finally, [22] presents an evaluation of the OGC web processing services as well as a road map for a future integration. Reusability issues are presented in [3].

The main drawback of all the aforementioned approaches is that they concentrate on an integrating framework for collecting distributed geospatial services but not in a real time framework for geographical information systems by exploiting the principles of the service oriented architecture. These issues are addressed in [23]. A publish/subscribe framework has been introduced in [24] for handling real time geospatial data. Publish/Subscribe is a protocol that produces/declares the topics of interest and subscribes registers for the topics of interest. A sensor network framework for environmental information management has been reported in [25]. The paper discusses the sensor infrastructure studied in the framework of the ORCHESTRA EU funded project [1]. A service oriented architecture for supporting real time requirements in service engineering has been reported in the Integrated EU funded Project of IRMOS [26–28]. The IRMOS design, develops and validates Cloud Solutions which will allow the adoption of interactive real time applications with Quality of Service guarantees [29,27]. However, the IRMOS project does not handle geospatial services and component. In addition, emphasis is given in the infrastructure and resource allocation for achieving the real time QoS requirements instead of the Web service interface.

However, the main focus of the works mentioned above is on the integration of geospatial data, collaboration and presentation services over the web. Therefore, they can be seen as the essential framework upon which an application-targeted SOA for environmental crisis management can be built, as will be discussed in the sequel.

### 1.2. Contribution

In this paper, we propose a flexible service oriented architecture for planning and decision support for an environmental information management. The architecture uses real time geospatial data sets and 3D presentation tools, integrated with added-value services for environmental modeling and operational logistics and support decision making in case of emergency. This enables the creation of enriched geospatial data that can be used to support early planning and crisis management decisions under a proper context. This paper aims to contribute a new architectural template for heterogeneous, distributed information systems useful in planning and decision support for environmental information management, that go beyond integration and presentation of static spatial data, to include real time data flows from sensor networks and added-value services for environmental modeling integrated with operational logistics.

## 2. Design requirements and main architectural components

In this section, we discuss the main design requirements of the proposed architecture and present the main architectural components of the proposed environmental information management system.

### 2.1. Requirements

The proposed architecture uses the service oriented architecture (SOA) principles. It also addresses interoperability issues in a way that geospatial data of different type and format to be able to incorporated in the proposed architecture, and service reusability and composability. In addition, SOA provides an efficient framework for distributed service communication regardless of the machine and languages used for service execution and description. However, additional requirements are needed to be included in order to provide efficient handling of a crisis. In particular, we need to incorporate (i) real time service delivery (e.g., in a QoS guaranteed framework), (ii) efficient 3D and contextual presentation of the geospatial content, (iii) simulation capabilities that offer added-value services in the environmental information management architecture and (iv) data filtering methods for efficiently delivery, handling and presenting the huge amount of the real time geospatial data.

Fig. 1 presents the main requirements of the proposed service oriented architectures for an efficient management of a nature disaster.

Real time service capabilities are one of the most imminent design requirements for efficient environmental information management architecture. It is clear that geospatial information and

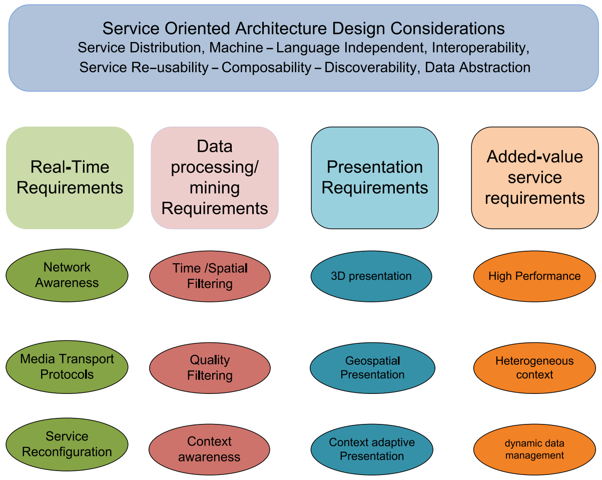


**Fig. 1.** The main design requirements of the proposed service oriented architecture used for efficient crisis handling.

services should be delivered under a QoS guaranteed framework so as to allow for experts to take decisions that minimize the effects of the natural disaster. For example, in case of a forest fire, real time delivery of geospatial information coming from a distributed wireless fire sensor network, camera or other source is of great importance; otherwise the danger for a fire expansion will increase. However, timing and OoS guarantees issues, for guaranteeing pre-defined constraints in real time service execution, are not thoroughly studied in the service oriented architecture framework. To provide real time capabilities in Web services, we need *network awareness* that refers to the monitoring of the current conditions of the network. Network monitoring allows implementation of network control mechanisms through *media transport protocols* (i.e., TCP/IP, UDP or DCCP) able to automatically adapt media delivered streams to the current network capabilities maximizing, however, users'/service QoS requirements as much as possible. In addition, we need dynamic *service reconfiguration* in order to adjust the service itself according to the dynamic nature of the networks, and ensure that the best possible performance can be achieved.

Another important design is data processing and mining. Real time measurements from sensors networks can produce tremendous amounts of information, which might be more than what current WAN networks and processing systems of a reasonable budget can handle. This presents challenges regarding processing, storage and communication aspects. For this reason, *time, space and quality filtering* methods are required to be considered in an environmental crisis management architecture. *Time filtering* can be implemented using algorithms that remove the temporal data redundancy (through, for example, data dropping mechanisms), while *spatial filtering* aims at reducing the spatial one (e.g., spatial data down-sampling). On the other hand, *quality filtering* automatically re-quantizes the geospatial content to reduce the transmitted information. All these types of filtering are framed with *context awareness mechanisms*, which proportionally activates one or all of the above three filtering tools according to the type of disaster and environmental conditions. It is clear that without the data processing/mining requirement principle, it is not able to process, analyze, interpret and transmit the huge amount of the real time geospatial data sets and are needed for an efficient environmental information management.

Another very important design requirement are the presentation ones. Apart from a web based user interface, the future environmental information management systems will incorporate (i) 3D rendering and presentation methods (ii) geospatial presentation and (iii) context aware adaptive visualization tools. *3D*

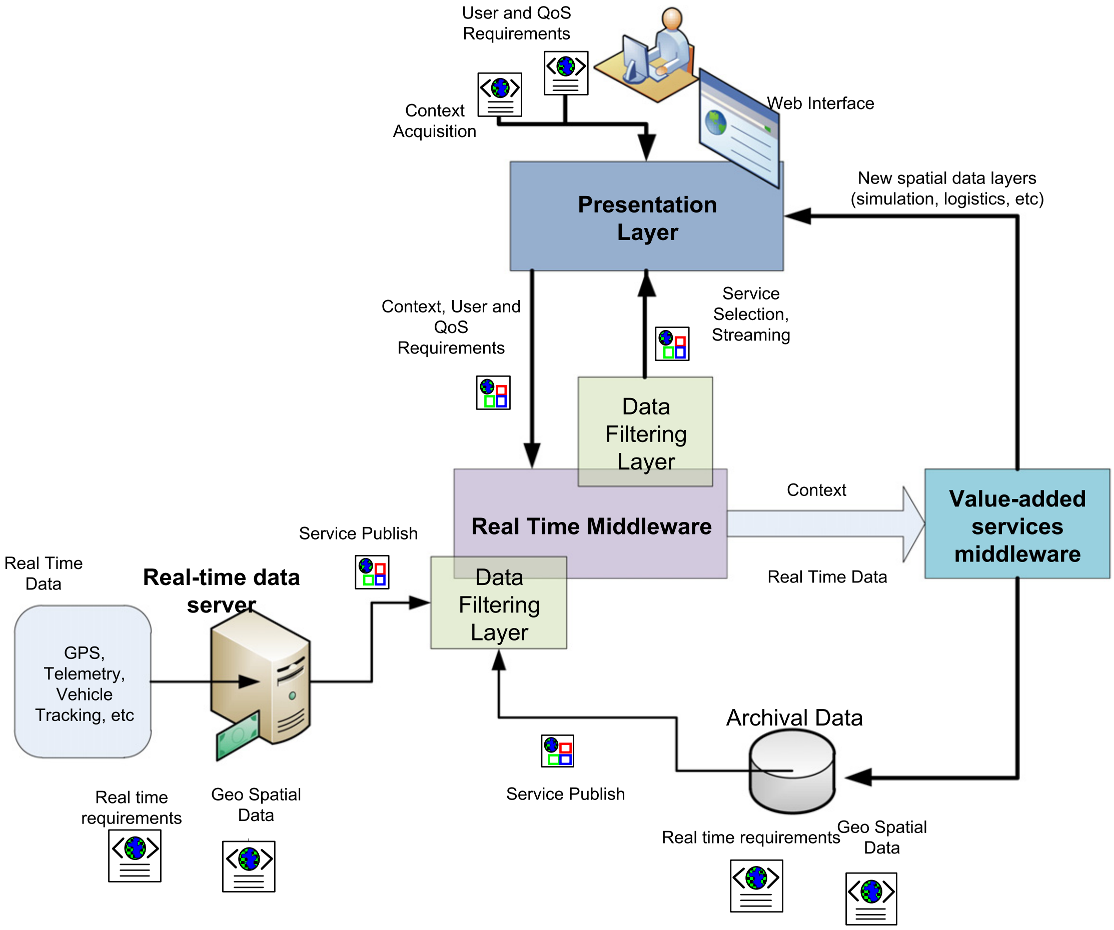


**Fig. 2.** The main architectural components of the proposed environmental information management system.

*presentation* and real time rendering allows for a better visualization of the natural disaster and therefore yields a more efficient environmental management. *Geospatial presentation* includes tools for depicted media overlays of geographic enhancements. Finally, *context adaptive methods* increase the system efficiency, since it permits a differentiated presentation according to either the capabilities of the terminal devices (e.g., PC or PDA) or the requirements of the application scenario. Different natural disasters require different visualization methodologies and the same stands for different simulations and other added-value services. For example, to create a spatial visualization of the probability of the natural eruption of a forest fire, close to real time data from on site sensors is required; the same goes for the results of fire simulation, where data needed to be pre-processed to achieve best 3D performance.

Finally, added-value services such as simulation and integrated operational logistics that enable prediction of the outcome of a natural disaster or future evolution, are of a great importance. Simulation increases the added value of the geospatial services; integrated with traditional logistics services, such as vehicle routing and producing new geospatial content layers. The possibilities are ever increasing, since many different methods from other disciplines can be implemented as added-value services. Some examples are evacuation simulations, clustering, and more. To make things more complex, spatial data produced in any of the aforementioned added-value services, can be used as input to another; for example, results of evacuation simulation can alter the performance of the road network used in routing. All of the above should be offered as *high-performance services*, in *heterogeneous contexts* and by considering *dynamic data management schemes*.

### 2.2. Main architectural components

The main architectural components of the proposed environmental information management system are presented in Fig. 2. Each implementation of this template can provide further refinements as per the nodes that are actually required to provide each service. As can be observed, two different types of geospatial data are supported; *the archival and the real time data*. These data are encoded using the specification of the Geospatial Markup Language (GML), which is actually an XML schema. Apart from the geospatial information, we also encode the real time requirements of the geospatial data. The real time requirements are described using the WS-resource framework [30] which is a family of specification used for providing statefulness appearance of the web services.

Real time and archival data are published to the *Real Time Middleware* using a Web Service interface. The Real Time Middleware is responsible for selecting the most appropriate service and geospatial component according to the current user's and context requirements as well as Quality of Service (QoS) parameters. The Real Time Middleware incorporates advanced data filtering methods for reducing the transmitted information especially for the real time data. It is also responsible for preprocessing the spatial data in order to best support the required 3D visualization performance and to provide the pre-processed data layers required by the added-value services middleware layer. Data filtering is accomplished in the temporal, spatial and quality data direction. In addition, the Real Time Middleware incorporates methods that enable service interaction with the network layer of the OSI architecture. This interaction allows for a real time service streaming of the data.

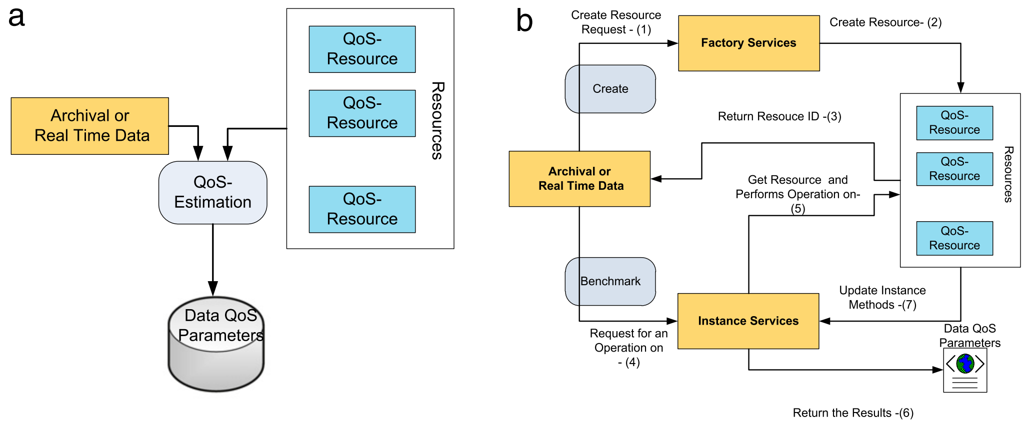


**Fig. 3.** (a) The benchmarking implementation for estimating the QoS parameters for each archival and real time data. (b) The Web service framework for estimating the QoS parameters for the geospatial data and their respective methods.

In the proposed architecture a *added-value middleware* has been introduced. This is a unique feature that allows added-value computational services to be integrated with real time and persistent spatial data to achieve the required functionality. The added-value services layer provides simulation of natural phenomena, simulation of human behavior in spatial contexts and operational logistics and therefore allows for an efficient decision making and handling of a natural disaster either in real time or in advance planning. The developed services take as input the current contextual information and the real time data and create new geospatial data that can be sent to the presentation layer.

*The presentation layer* is responsible for transforming the received geospatial data in a format suitable for Web browser interface. Contextual adaptation methods are also incorporated in order to adjust the results with respect to terminals' capabilities and user's information needs. This adaptation allows the results presentation on different types of devices, like PDA, PC, mobile phones, etc. We also include 3D presentation schemes to better visualize the geospatial content.

## 3. Data encoding layer

As discussed above, two different types of data are supported; the archival and the real time data. The archival data stored in spatial and distributed databases, while the real time data comes from sensors. Both data are encoding using the OGC specifications and particularly the Geography Markup Language (GML). However, apart from the interoperable representation of the geospatial information, the real time requirements of both the archival and real time data are encoded.

### *3.1. Geospatial representation*

Regarding the representation of static geospatial data, the geometric properties of geographic objects are encoded. In case of the forest fire the geographic objects are maps, roads, current fire location and future expansion of the fire using the simulation models. For each of the above geometric objects, several properties are used like for example the polygon over which a fire is expanded or the coordinates of the road geometric object. In addition, a Coordinate Reference System (CRS) is also included to determine the geometry of the geometric element in GML document.

### *3.2. Real time requirements representation*

On the other hand, real time requirements are representing using the WS-Resource framework. The values of the real time requirements are provided through the execution of a benchmarking mechanism on the data site. This benchmarking can be improved either automatic or using users' interaction, meaning that users' indicate degree of data down-sampling, the geographical scales that should be transmitted and the type of presentation (3D or 2D) in order to accelerate data transmission and computational processing demands.

Each geospatial datum is characterized by a set of *real time or QoS requirements*. These *Data QoS parameters* are estimated using a set of available resources with specific QoS requirements. Different QoS-resources are created not only in case of a fixed or mobile network connection, but also in case of different network parameterization. For example, a different allocation of the bit rate creates a different QoS-resource allocation.

Each archival and real time data is associated with a set of QoS resources. It is clear that the same methods present different QoS parameters when they are executed (or delivered) under different resources. At every archival and sensor server, we have included a QoS estimator. This module takes as input the QoS parameters of each resource as well as the respective archival and real time data requirements. The purpose of this module is to estimate the QoS parameters for each method of the geospatial data in respect with the available resources.

The WS resource framework is adopted for estimating the QoS parameters of the geospatial data and their respective methods. The WS resource provides an interface that relates Web services with stateful resources [31,32]. Fig. 3 presents the architecture used for estimating the QoS parameters for the archival and real time data. As we have stated above, a set of available QoS resources exist. Initially, the client sends a resource request to the factory of resources (see step 1 of Fig. 3(b)). Then, a resource is allocated and an ID is returned to the client (see steps 2 and 3). In the following, a request for an operation on a resource is activated as described by the step 4 of Fig. 3(a). This request is triggered from the Real Time Middleware interface as it is discussed in Section 4. The request is sending to the instance service interface where a set of methods are available. The instance interface triggers the resource archive in order to get the resource and performs a specific operation on it (see step 6). The results, which are actually the QoS parameters for a geospatial data or a method are stored in an XML formal of the WS resource framework (see step 7). It is also possible to update the methods of the instance service interface (see step 8).

Using this framework, a set of QoS parameters are estimated. Therefore, each archival and real time server has two different types of data; the geospatial ones (along with the respective methods) and the QoS parameters which are estimated using the WS resource framework. The estimation of the QoS parameters is accomplished in an on-line mode and it is triggered by the Real Time Middleware layer.

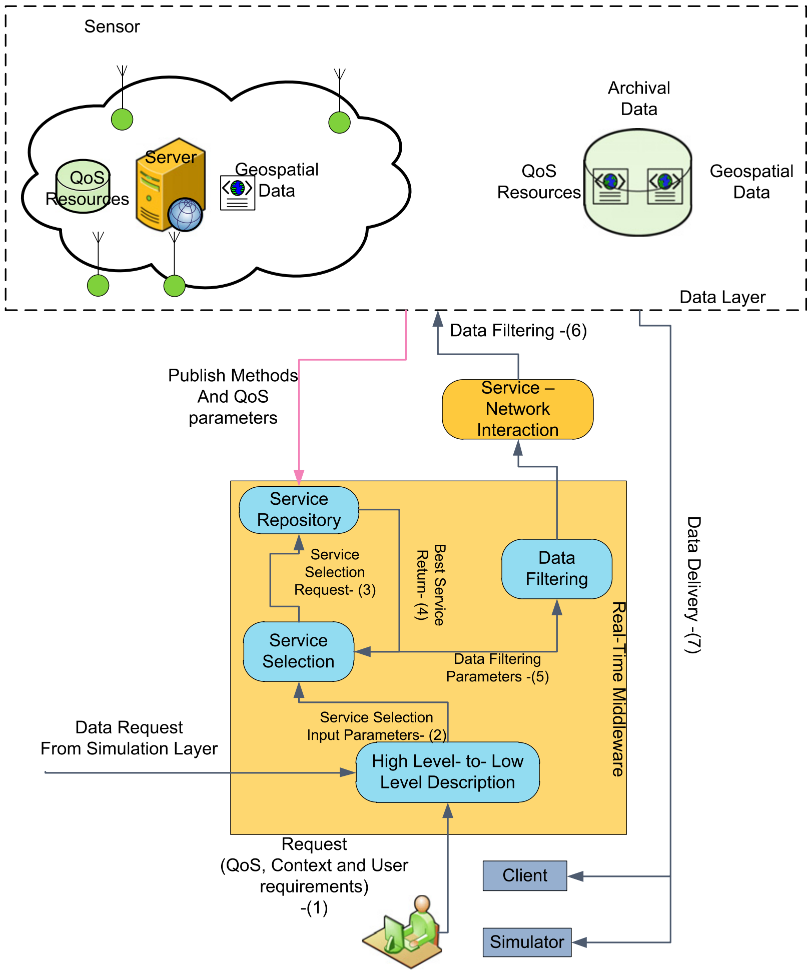


**Fig. 4.** The Real Time Middleware architecture the respective system processes.

The IRMOS architecture [26] also handles real time issues in a SAO framework. However, the IRMOS use case scenarios is oriented to interactive multimedia applications, virtual augmented reality and e-learning. Instead, this paper focuses on geospatial applications which impose new requirements in the architectural design. In particular, in our case, we need to handle huge amount of heterogeneous data resulting to the need of real time data filtering techniques. In addition, It requires synchronization between data and geospatial overlays, which demands modifications of the real time middleware compared to the IRMOS framework. Finally, there is another difference in the presentation layer that is equipped in our case with 3D functionalities and methods of handling data coming from heterogeneous contexts.

To orchestrate the flow activities presented in Fig. 3(b), we could use the Business Process Execution Language (BPEL) [33]. BPEL is a standard executable language for specifying actions within business processes with web services. This defines an interoperable integration model that should facilitate the expansion of automated process integration both within and between businesses. Although in this paper an ad hoc method is adopted for controlling service flows, the design of the architecture is generic and BPEL language can be included.

## 4. Real Time Middleware layer

The Real Time Middleware is responsible for selecting the most appropriate method of the available geospatial content and service in order to satisfy the QoS requirements. Fig. 4 presents the main architectural structure of the Real Time Middleware.

Initially, the user or the environmental information management system itself provides a set of requirements to the Real Time Middleware. These requirements are *the user requirements, the QoS demands as well as the acquired context*. For example, user requirements may be meta-data that refers to the geospatial information needs of the user. In addition, QoS requirements are the high level specifications of the executed geospatial services. Finally, context refers to the terminal capabilities. All these requirements are described using an XML schema and are delivered to the Real-Time Middleware.

The Middleware is responsible for reading the XML information and transform the high level requirements to a set of low level parameters that can be passed to other computational elements. Using the set of low level parameters, we activate the service selection mechanism. This module is responsible for selecting the most appropriate service among all the available. The service selection module interacts with the service registry over which we have included all the available distributed geospatial services and components.

Then, service selection estimates the most appropriate service that matches the QoS requirements. In this case an interaction between the service registry and the distributed archival and/or real time data is activated. Factors that can be considered in the service selection process are the availability of data required by each specific service implementation at the format that it is required, the performance of the service, the network overhead, and more.

The real-time requirement performance is considered as the most prominent criterion regarding the evaluation of the factors

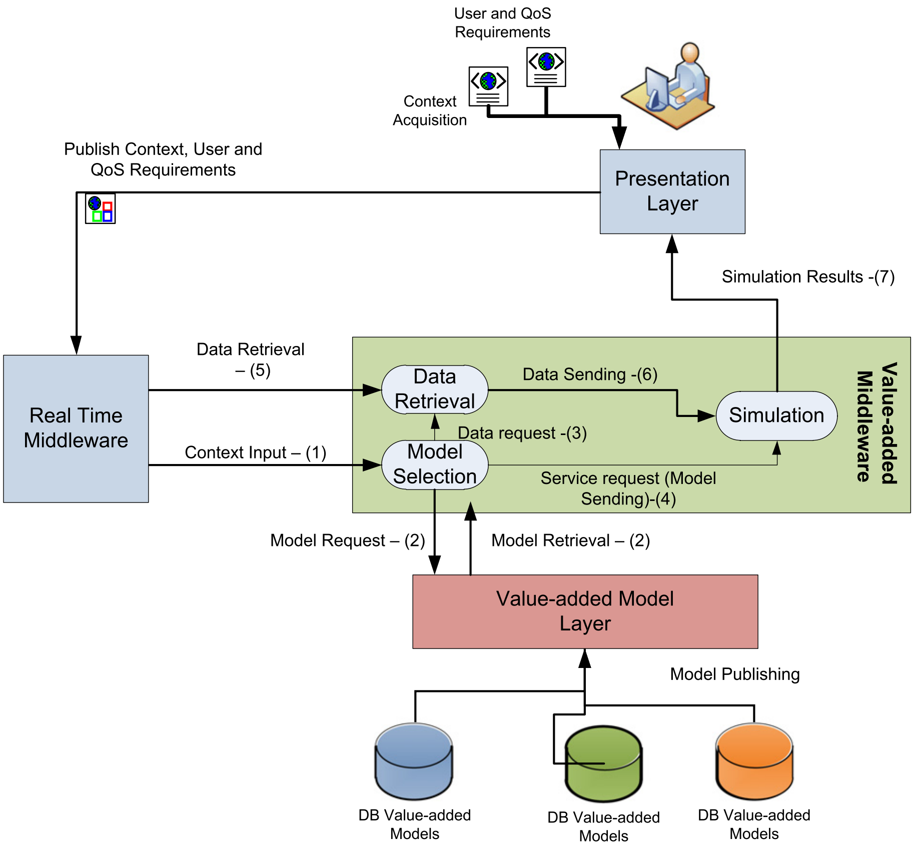


**Fig. 5.** The architecture of the added-value middleware and the respective system processes.

used for service selection. This means that, among a set of suitable services, we select the ones that present minimum transmission rates. The next most significant factor deals with service computational complexity, in combination, however, with the terminal capabilities.

The interaction activates the WS resource framework as it is described in Section 3 (see Fig. 3(a)). This means that for each service a respective QoS value is computed using the parameters of the available resources.

In case that there is no service available, the service selection module estimates the degree of reduction (transformation) that is required for QoS-matching. This can be, for example, a down-sampling estimate of the geospatial data, resulting in a *rate shaping approach*. The degree of rate shaping is given to the Data Filtering Module, which is responsible for applying a temporal and spatial content-based sampling algorithm in order to tailor the communication demands of the geospatial data to the QoS requirements.

The next step is to trigger the Service–Network Interaction module, which is responsible for translating the application-layer parameters of the Data Filtering Module to network parameters. In other worlds, the Service–Network Interface acts as mediators between the application and the network layer. Once the network parameters are configured, geospatial information is delivered to the client, that is to the environmental information management system. As previously, orchestration of the service flow is implemented in an ad hoc way which is suitable to the specific application scenario of disaster handling (e.g., fire). However, our design is generic to support the BPEL language for coordination of service flows [33].

### 4.1. *Data filtering*

The purpose of this module is to adjust the QoS demands of the geospatial data in order to match the QoS requirements. This shaping is achieved in this approach using a temporal and spatial down-sampling technique as well as quality filtering. Temporal and spatial down-sampling selectively discard data either in temporal or spatial direction in order to reduce the amount of data transmitted over the network. For example, the environmental information management system triggers sensor not to capture and send geographical information to accelerate computational processing and data transmission. This is achieved by the usage of optimization strategies which monitor the type and content of the transmitted data and apply algorithms to avoid transmission and capturing of unnecessary information.

## 5. Added-value middleware

One of the main advantages of the proposed system is the added-value services layer, which is responsible for adding value to archived and real time environmental data, by predicting future environmental states, providing operational guidance, and solving combinations of optimization and simulation problems, the boundary conditions of which depend on each other. For example, the contour that describes the predicted bounds of a forest fire, can be used to extract the underlying road network from being considered in a routing algorithm.

Fig. 5 presents the main architecture of the Simulation Middleware. As we have stated in Section 2.1, the information management system provides a set of *user, QoS and context requirements*. These requirements are encoded in an XML schema and are published in the Real Time Middleware. In the following, the Simulation Module is activated by a context delivery request form the Real Time Layer (see step 1 of Fig. 5). This request triggers the *Model Selection* module of the added-value middleware. The Model Selection Module is responsible for estimating the most appropriate component from a set of available ones. The components along with their interface and semantic requirements are stored in distributed database and they are published to the *added-value* framework (see Fig. 5) through the use of a Web service interface. In particular, Web services publish the methods of the simulation models. The added-value models can be of different types regarding the adopted application scenario; several examples of such models as already discussed, are environmental simulation (fires, clouds and other computational fluid dynamics simulations), traffic simulation, evacuation simulation, routing, clustering, dynamic prediction of event probabilities, and more.

The Model Selection is responsible for retrieving the most appropriate added-value service model from the added-value service Model Layer. This is accomplished using a Web service interface. In particular, the Model Selection sends a request to the added-value service Model (see step 2) according to the context input received from the Real Time Middleware. Then, the added-value service Models returns the most appropriate model for this context, i.e., application scenario and geospatial information. In the following, the Model Selection module sends a request to the Data Retrieval Module (see step 3) in order to retrieve the most suitable real time data for the service requested. At the same time, the Model Selection module triggers the added-value service interface for commencing the requested process (see step 4). The Real Time Middleware response to the Data Retrieval request and returns a set of real time geospatial data to the added-value service Middleware (see step 5). Transmission of the real time data is accomplished by activating the processes described in previous section. In this framework, the data filtering framework is activated to tailor the real time QoS demands to the user requirements.

Data Retrieval module sends the real time data to the added-value service interface (see step 6). The added-value service receives the real time data and the respective added-value service model (see step 4) and performs the task. The results are forwarded to the presentation layer. Similarly, the BPEL language can be included for service flow orchestration [33].

## 6. Presentation layer

In this paper, a Web based approach is adopted for visualizing the results of (i) the archival and real time data and (ii) the simulation models. The use of a Web based interface presents several advantages compared to the usage of a standard alone centralized information server. First it allows for a universal access of the application regardless the operation system, the type of terminal device and the network interface (mobile or fixed). Second, the simple and friendly Web interface greatly reduces learning time and thus attracts more non-GIS professionals. Third, Users can potentially pay, at the time of use, only for the services (functions and data) they utilize. This means that users are not requested to purchase a full version GIS software. Finally, the World Wide Web is nearly ubiquitous allowing access from anywhere, anytime and at any device.

However, The Web was originally built on the HTML (Hypertext Markup Language) standard for encoding/decoding web site page descriptions. Current versions of Web browsers can interpret XML schemas and instructions. As we have stated in previous section, data of this architecture are delivered using the interoperable interface of the OCG specification. In particular, the archival, real time and simulation data are encoding using the GML, which is an XML extension for encoding the transport and storage of geographic information that includes geometry and properties of geographic features.

GML adheres to the principle of separating content from presentation, so it does not address the visualization of encoded data. GML can be visualized either in a raster or in a vector image format. The main advantage of the raster implementation is that all Web browsers provide native support for raster images. However, raster representation presents a series of drawbacks, especially in case of geospatial information content where detailed maps and dynamic behavior are usually required. To overcome these difficulties vector formats are introduced. In particular, the W3C consortium has developed the Scalable Vector Graphics (SVC) interface which is a set of XML specifications for describing two-dimensional vector graphics, both static and dynamic (i.e. interactive or animated) [34]. Other vector specifications are the Vector Markup Language (VML) supported by the Microsoft Internet Explorer [35] and the Precision Graphics Markup Language (PGML) supported by the Adobe applications [36]. Google Maps currently uses VML to make vector paths work when running on Internet Explorer 5.5+, and SVG on all other browsers. In other worlds, the GML provides a framework for encoding and storage geospatial content, while vector languages an interface for viewing geospatial content.

In the proposed architecture, GML specifications are transformed to vector languages, for example to the SVG format, using the XSL transformation. XSL transformation is a set of XML languages for transforming an XML document into another XML format. This special XSLT engine is necessary for interpreting a GML file by a Web browser supported for example the SVG standard. For instance in case we are focusing on an SVG style, the XSLT engine is used to locate the GML elements and to transform these elements into a SVG file.

Fig. 6 presents the main architectural components of the presentation layer of the proposed system. In the client site there is two main architectural components; the Web browser and the Context Acquisition Module. We use a Web browser interface of supporting vector image representation in order to provide dynamic and high detailed geospatial content visualization. The user submits to the system a set of user's requirements along with the QoS demands. These parameters are received by the Web browser and transform to appropriate XML files by the application server. In addition, we have introduced a context acquisition module, which is responsible for detecting the terminal capabilities and the display context. In other worlds, small display size is anticipated to request less detailed information than terminals of higher monitor resolution and computational processing capabilities. This information is forwarded to the other layers of the proposed information system. For example, terminal and display context is one of key important element for selecting the most appropriate geospatial service and content as well as for reducing the geospatial information by exploiting the Data Filtering Module. The above mentioned information is forwarded to the Web server through Http request, while in the following the Application server takes place for delivering geospatial input to the other layers.

Once a geospatial content or service is received to the presentation layer, several actions are taken into consideration for transforming the received GML content in a format interpretable by the Web browser. For this reason, the XML file of the geospatial content is forwarded towards an XSLT engine able to transform the GML representation to a vector image format. The XSLT engine takes as input the transformation style, described using the XSLT framework, and the vector language style. Regarding of the version and of the type of the Web browser, we use, different vector markup languages are supported. For example, in case of the Internet Microsoft Explorer, it is preferable to use the VML (Vector Markup Language). Instead in case of the Google Chrome Browser, the Scalable Vector Graphics format is suitable for visualization. The Document Object Model (DOM) is used for parsing an SVG XML file. This combination between DOM and SVG allows for scripting languages, like Java Script, to produce Web maps of high interactivity.

The aforementioned vector languages specifications allows for a 2D geospatial content presentation. In case, we need a 3D

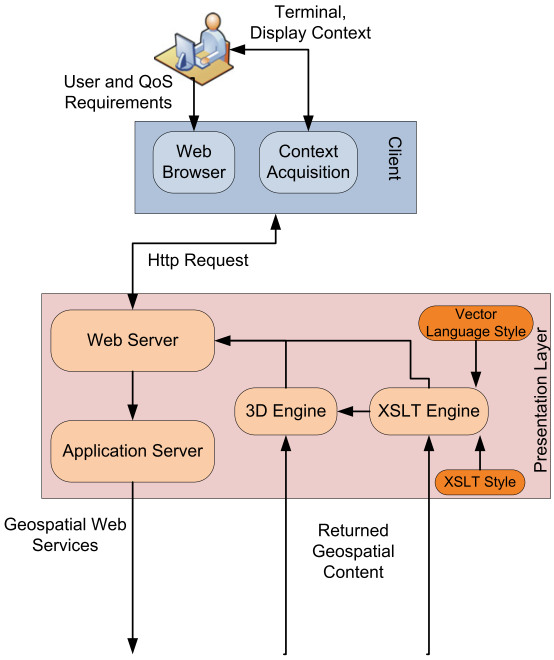


**Fig. 6.** The architecture of the presentation layer.

representation, we need additional 3D vector image languages. In a similar way with the 2D image representation, the 3D content is visualized by dynamically transforming the GML geospatial data into 3D coordinates and a 3D vector language format. For the 3D representation the X3D specification is used [37]. X3D is an ISO XML file format standard that extents the Virtual Reality Modeling Language (VRML) in order to support visualization of 3D content on the Web browsers. Currently, there are exists X3D applets which are OpenGL compatible 3D graphics technology able to display X3D compatible content in several different browsers, like the FireFox and the Internet Explorer.

## 7. Case study: a forest fire crisis environmental information management system

A real world implementation of the service oriented architecture discussed above has been done for the local state authority of Messinia, located about 200 km SW of Athens, Greece. The objective of the project was to implement an integrated information system for forest fire management. This involves early planning and scenario development, as well as real time monitoring and decision support in the event of a real forest fire incident.

Real time data flows are defined for operational vehicles management including fire trucks, water couriers, police cars, ambulances and civil protection authority vehicles. This monitoring is achieved via an Automatic Vehicle Location (AVL) software application along with the required services for real time data processing. Also, weather stations located in the area of interest provide real time meteorological data flows that is stored into the Database. Satellite images and vector data such as road network, PoIs, and administrative divisions, are integrated to provide a high quality 3D depiction of the area.

Added-value services include high-performance forest fire simulation, forest vegetation information, vehicle routing, panic evacuation simulation, as well as a clustering application to provide operational resource allocation for a given set of constraints and requirements. The specific implementation (instance) of the generic architecture presented in Section 2.2 for this particular implementation is shown in Fig. 7.

Real time data comes from vehicle tracking and meteo stations and is received by a communication server (Real time data server). This data is processed so as to be optimized (size) and is stored into the persistent storage database of the system. All the spatial data (vector data, PoIs, satellite images, etc.) is stored in that database. The end-user experiences a Java applet interface through a web browser, which sends service request to the "presentation layer" of the system. The presentation layer is optimized for 3D data visualization and is also responsible for forwarding application requests to the "Real time service management node" of the system.

A web service accepts all the requests and decides where to forward each one of them and what are the parameters that should be passed. If the request is about a GIS-like service, it is handled locally at the Presentation layer, since the spatial data is already pre-processed for optimal GIS presentation in 3D. If the request contains a database query, then it is passed to the data archive, and in the sequence data is sent to the presentation layer. If the request is for a added-value service among the simulation and operational logistics services available, then before actually calling the required module, a "context based service selection" takes place. During this phase, the user request context is examined in order to determine which exactly service instance needs to be invoked, as well as the required parameters. For example, if a request is about vehicle routing from point A to point B, then if no fire simulation exists in context, the router is directly invoked; if a fire simulation is active, then before calling the router, the "context-based service selection" module first executes a spatial query to exclude the road network segments covered by the fire contour.

A similar behavior is exhibited when many-to-many point routing is requested, or when evacuation simulation takes place while a forest fire simulation is active. To optimize the performance of the added-value services, a dynamic spatial data repository is used. Data stored in this repository is pre-processed data for the forest fire simulation and routing algorithms, as well as short-lived parameters of subsequent calls to the added-value services, so as to avoid addressing the large main database of the system. To optimize performance, all the added-value services have been implemented as "windows services" (.NET 3.5, Windows Server 2008), and are called via the XML-RPC network protocol.

This implementation follows the main concept of the generic architecture proposed in this paper, which allows a fully distributed implementation. Indeed, all the communication shown in Fig. 7, can be done over a network which allows for manageability, scalability and flexibility. It is important to emphasize that although the Presentation layer and the Real time service management layer are implementation-specific, the added-value services middleware is not. This means that all the added-value services can be invoked by any network application that implements their service interface.

In Fig. 8(a)–(d), four screen shots of the implemented system are shown. Environmental modeling (forest fire simulation) in 3D presentation is shown in Fig. 8(a), while panic evacuation simulation is shown in Fig. 8(b). In Fig. 8(c), environmental simulation is combined with routing: the predicted forest fire contour is used to exclude part of the road network from being used in routing. Finally, Fig. 8(d) shows many-to-many point resource assignment and routing.

The above implementation has been fully completed and is currently in real world production use for almost one year. Its main considerations are the Web Service implementation overhead, as well as the XML–RPC implementation overhead. The former has a considerably larger time cost than the latter, but it seems like a logical cost considering the flexibility and scalability of the system architecture. In terms of capacity, the system performs very well for about 100 vehicles in a 3000 sq. km. area in which the added-value services discussed can be offered. Should

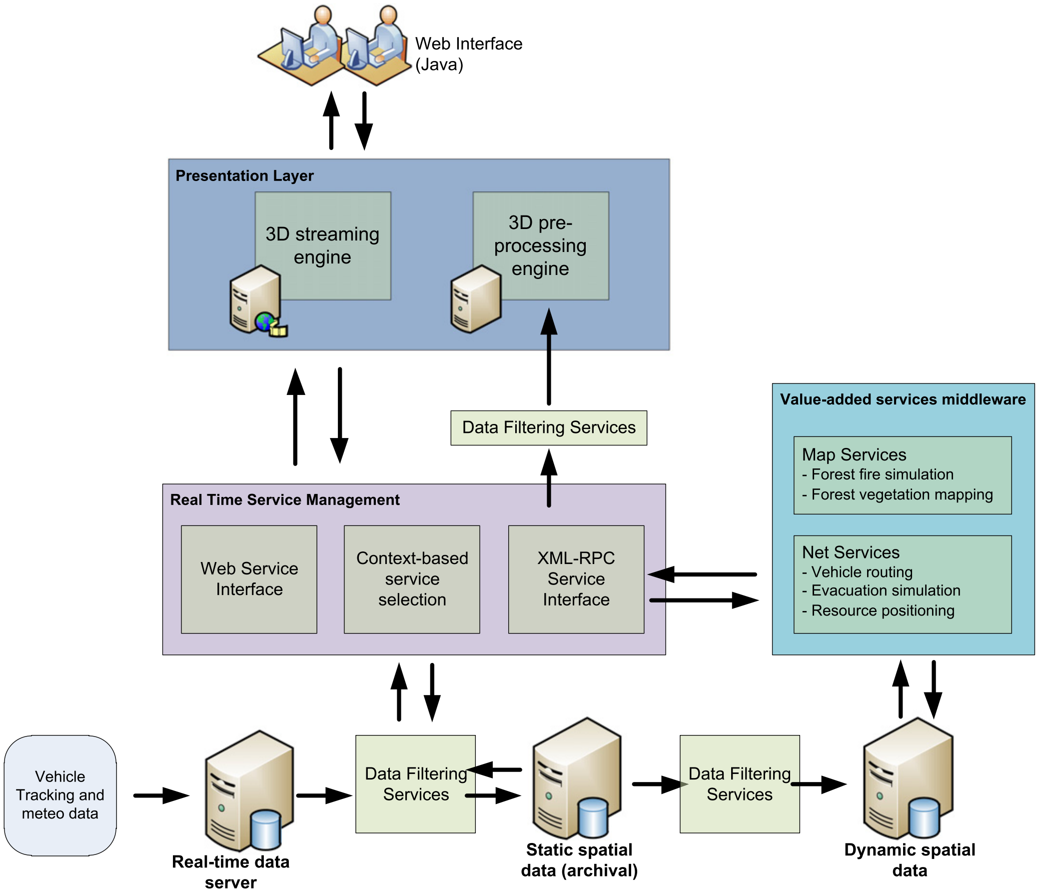


**Fig. 7.** Implementation of the proposed SOA in the Messinia prefecture (2009).

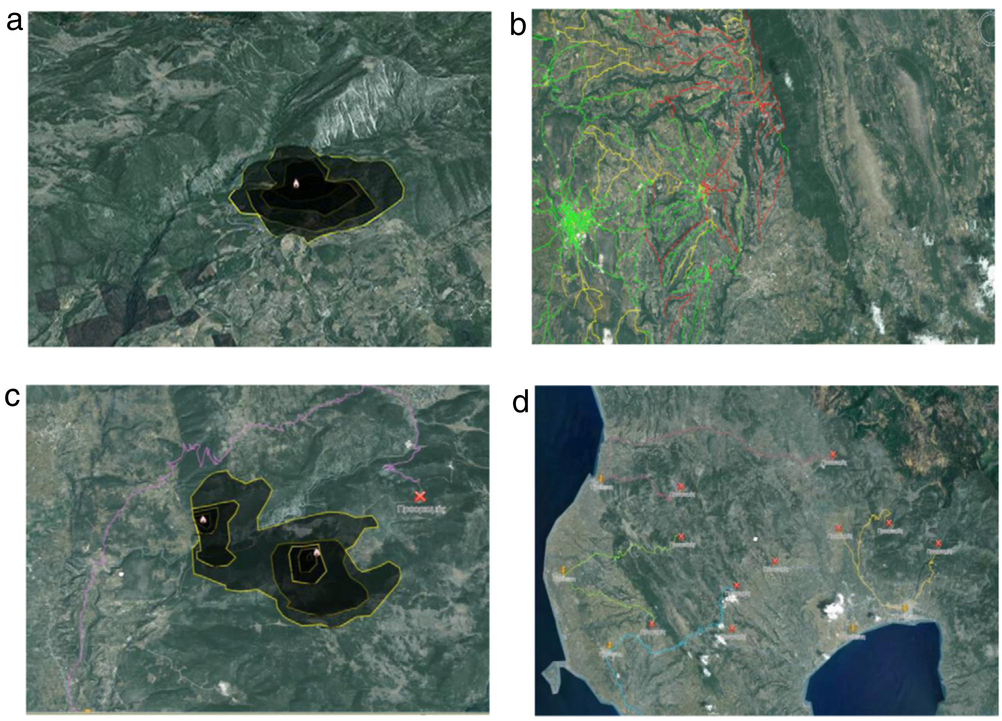


**Fig. 8.** Environmental modeling and operational logistics implementation.

capacity requirements increase, a server farm using load-balancing techniques or even a cloud can be considered for critical parts of the system, without affecting the interfaces and operation of the others.

## 8. Conclusions

Design and implementation of an environmental information management systems is a challenge in a multi-disciplinary area, involving software engineering, environmental modeling, logistics, spatial data, real time services, as well service and data semantics. Distributed, heterogeneous added-value architectures enable users to evaluate the outcome of a natural disaster, a necessary task for effective decision making, and to do early planning, useful in awareness and preparedness. In this paper, we proposed an architectural framework for environmental crisis planning and management systems, incorporating data and presentation services, as well as dynamically selected simulation models able to predict future geospatial states from real time and static spatial data. Although several works have been reported in the literature for efficient prediction of the outcome of a natural disaster, like for example the forest fire growth, such systems remain isolated and monolithic applications, and the same stands for logistics useful in crisis management. Little effort has been done in the area of integrating all the above in a working SOA.

In this paper, we proposed a QoS-aware service oriented architecture, suitable for geospatial information management systems targeted in planning and crisis management. The proposed architecture supports QoS guarantees for delivering geospatial information, which is a very important aspect for decision making and addressing a case of emergency. Furthermore, the proposed architecture exploits added-value service components to go further from simply presenting spatial data, and include services such as environmental simulation models and logistics. Finally, presentation aspects are discussed that enable efficient vector based image representation of GML compatible geospatial content.

The main contribution of this paper is focused on the proposed architectural schema that allows heterogeneous data sources, presentation frameworks and added-value service providers to be integrated in a QoS-aware, coherent system in order to provide planning and decision support in environmental information management. Several considerations of this architecture such as QoS and data filtering have been motivated form a real world implementation that is also discussed.

However, although many possibilities are theoretically possible, there is still a long way to go in order to reach the limits of the systems discussed in this paper. First, although standards do exist, real world spatial data is usually not compliant, nor is it offered through open network interfaces. Second, QoS requirements that are of critical importance to this application domain either cannot be implemented, or, even worse, are rarely a consideration. Third, added-value services such as operational logistics and environmental modeling remain isolated to their originating scientific communities, such as Computational Fluid Dynamics and Operational Research, and only lately begin to be considered as services that can be integrated in such systems.


## Acknowledgements

The authors would like to thank all the anonymous Reviewers for their constructive comments which help us to improve the quality and readability of this paper.


## References


[1] A. Annoni, L. Bernard, J. Douglas, J. Greenwood, I. Laiz, M. Lloyd, Z. Sabeur, A.M. Sassen, J.Jacques Serrano, T. Usländer, Orchestra: developing a unified open architecture for risk management applications, in: Geo-Information for Disaster Management, Springer, Berlin, 2005, pp. 1–17.

[2] Martin Maliska, Branislav Simo, Marek Ciglan, Peter Slizik, Ladislav Hluchy, Service oriented architecture for risk assessment of natural disasters, in: Lecture Notes in Computer Science, vol. 3911, Springer, Berlin, Heidelberg, 2006, pp. 357–363. ISBN: 0302-9743.

[3] Carlos Granell, Laura Dıaz, Michael Gould, Service-oriented applications for environmental models: reusable geospatial services, Environmental Modelling & Software 25 (2010) 182–198. Elsevier Press.

[4] R. Denzer, Generic integration of environment decision support systems—state-of-the-art, Environmental Modelling & Software 20 (10) (2005) 1217–1223. Elsevier Press.

[5] Qing Gu, H. van Vliet, SOA decision making—what do we need to know, in: IEEE Workshop on Sharing and Reusing Architectural Knowledge, 2009, pp. 25–32.

[6] M.J. Mineter, C.H. Jarvis, S. Dowers, From stand-alone programs towards grid aware services and components: a case study in agricultural modelling with interpolated climate data, Environmental Modelling & Software 18 (4) (2003) 379–391. Elsevier Press.

[7] Zongsheng Zheng, Dongmei Huang, Jianxin Zhang, Shengqi He, Zhiguo Liu, A SOA-based decision support geographic information system for storm disaster assessment, in: IEEE 18th International Conference on Geo-Informatics, 2010, pp. 1–6.

[8] C. Granell, L. Diaz, M. Gould, Managing earth observation data with distributed geoprocessing services, in: IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2007, 2007, pp. 4777–4780.

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

[10] Open Geospatial Consortium, OpenGIS geography markup language, (GML) implementation specification. Available at: http://www.opengeospatial.org/standards/gml (last accessed: May, 2010).

[11] Open Geospatial Consortium, OpenGIS web feature service implementation specification. Available at: http://www.opengeospatial.org/standards/wfs (last accessed: May, 2010).

[12] Open Geospatial Consortium, OpenGIS web map service (WMS) implementation specification. Available at: http://www.opengeospatial.org/standards/wms (last accessed: May, 2010).

[13] N. Alameh, Chaining geographic information web services, IEEE Internet Computing 7 (5) (2003) 22–29.

[14] Petraq Papajorgji, Howard W. Beckb, Jose Luis Braga, An architecture for developing service-oriented and component-based environmental models, Ecological Modelling 179 (2004) 61–76.

[15] Michi Henning, Steve. Vinoski, Advanced CORBA Programming with C++, Addison-Wesley, ISBN: 0-201-37927-9, 1999.

[16] X. Zhu, Developing web-based mapping applications through distributed object technology, Cartography and Geographic Information Science Journal 28 (2001) 249–258.

[17] Y.-S. Chang, H.-D. Park, XML web service-based development model for Internet GIS applications, International Journal of Geographical Information Science 20 (4) (2006) 371–399.

[18] L. Dıaz, C. Granell, M. Gould, Case study: geospatial processing services for web-based hydrological applications, in: Geospatial Services and Applications for the Internet, Springer, New York, 2008, pp. 31–47.

[19] Natallia Kokash, Risk management for service-oriented systems, in: Lecture Notes in Computer Science, vol. 4607, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 563–568.

[20] C. Yang, W. Li, J. Xie, B. Zhou, Distributed geospatial information processing: sharing distributed geospatial resources to support digital earth, International Journal of Digital Earth 1 (3) (2008) 259–278.

[21] A. Friis-Christensen, N. Ostlander, M. Lutz, L. Bernard, Designing service architectures for distributed geoprocessing: challenges and future directions, Transactions in GIS 11 (6) (2007) 799–818.

[22] C.D. Michaelis, D.P. Ames, Evaluation and implementation of the OGC web processing service for use in client-side GIS, GeoInformatica 13 (1) (2009) 109–120.

[23] Galip Aydin, Service oriented architecture for geographic information systems supporting real time data grids, Ph.D. Thesis, Dept. of Computer Science, Indiana University, 2007.

[24] S. Oh, J.-H. Kim, G. Fox, Real-time performance analysis for publish/subscribe systems, Future Generation Computer Systems 26 (3) (2010) 318–323.

[25] John Douglas, Thomas Usländer, Gerald Schimak, J. Fernando Esteban, Ralf Denzer, An open distributed architecture for sensor networks for risk management, Sensors 8 (2008) 1755–1773.

[26] IRMOS: Interactive real time multimedia applications on service oriented infrastructures, Integrated Project, European Union. www.irmosproject.eu.

[27] D. Kyriazis, A. Menychtas, G. Kousiouris, K. Oberle, T. Voith, M. Boniface, E. Oliveros, T. Cucinotta, S. Berger, A real-time service oriented infrastructure, in: International Conference on Real-Time and Embedded Systems, RTES, Singapore, 2010.

[28] M. Boniface, B. Nasser, J. Papay, S. Phillips, A. Servin, K. Yang, Z. Zlatev, S. Gogouvitis, G. Katsaros, K. Konstanteli, G. Kousiouris, A. Menychtas, D. Kyriazis, Platform-as-a-service architecture for real-time quality of service management in clouds, in: 5th International Conference on Internet and Web Applications and Services, ICIW, Spain, 2010.

[29] Tommaso Cucinotta, Antonio Mancina, Gaetano Anastasi, Giuseppe Lipari, Leonardo Mangeruca, Roberto Checcozzo, Fulvio Rusinà, A real-time service-

oriented architecture for industrial automation, IEEE Transactions on Industrial Informatics 5 (3) (2009).
[30] M. Humphrey, G. Wasson, K. Jackson, J. Boverhof, M. Rodriguez, J. Gawor, J. Bester, S. Lang, I. Foster, S. Meder, S. Pickles, M. Mc Keown, State and events for web services: a comparison of five WS-resource framework and WS-notification implementations, in: IEEE International Symposium on High Performance Distributed Computing, 2005, pp. 3–13.
[31] G. Moltó, V. Hernández, J.M. Alonso, Automatic replication of WSRF-based grid services via operation providers, Future Generation Computer Systems 25 (2009) 876–883.
[32] K. Czajkowski, et al. WS-resource framework, 2004. http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf.
[33] P. Louridas, Orchestrating web services with BPEL, IEEE Software (2008) 85–87.
[34] W3C, Scalable vector graphics (SVG). http://www.w3.org/Graphics/SVG/.
[35] B. Mathews, D. Lee, B. Dister, J. Bowler, H. Cooperstein, A. Jindal, T. Nguyen, P. Wu, T. Sandal, Vector markup language, Technical Report, World Wide Web Consortium, May 1998. http://www.w3.org/TR/1998/NOTE-VML-19980513.
[36] N. Al-Shamma, R. Ayers, R. Cohn, J. Ferraiolo, M. Newell, R.K. de Bry, K. McCluskey, J. Evans, Precision graphics markup language, Technical Report, World Wide Web Consortium, April 1998. http://www.w3.org/TR/1998/NOTE-PGML-19980410.html.
[37] W3C consortium. http://www.web3d.org/x3d/.

**Vassilios Vescoukis**: Dr. Vassilios Vescoukis was born in Athens, Greece in 1967. He received his diploma and Ph.D. Degrees from the School of Electrical and Computer Engineering of the National Technical University of Athens, in 1991 and 1996 respectively. He currently holds an Assistant Professorship position at the School of Surveying Engineering of NTUA. His research interests include reasoning support in software development, distributed heterogeneous software applications, value-added optimization services and knowledge-based systems.

**Nikolaos Doulamis**: Nikolaos D. Doulamis received his Diploma degree in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) in 1995 with the highest honor and his Ph.D. degree in electrical and computer engineering from NTUA in 2000. He joined the Image, Video and Multimedia Lab of NTUA in 1996 as research assistant. His Ph.D. thesis was supported by the Bodosakis Foundation Scholarship. He has elected as Assistant Professor in the National Technical University of Athens.

Dr. Doulamis was awarded as the Best Greek Student in the field of engineering in national level by the Technical chamber of Greece in 1995. In 1996, he was received the Best Graduate Thesis Award in the area of electrical engineering with A. Doulamis. During his studies he has also received several prizes and awards from the National Technical University of Athens, the National Scholarship Foundation and the Technical Chamber of Greece. In 1997, he was given the NTUA Medal as Best Young Engineer. In 2000, he was served as Chairman of the technical program committee of the VLBV'01 workshop, while he has also served on the program committees in several international conferences and workshops. In 2000, he was given the Thomaidion Foundation best journal paper award in conjunction with A. Doulamis. He is editor in the Who is Who bibliography. He is a reviewer of IEEE journals and conferences as well as and other leading international journals.

His research interests include video transmission, service engineering, content-based image retrieval, summarization of video sequences and intelligent techniques for video processing.

**Sofia Karagiorgou**: Karagiorgou Sophie holds an M.S. degree from the Department of Computer and Communication Engineering at University of Thessaly in Greece. Currently, she is a Ph.D. candidate in the School of Rural and Surveying Engineering at the National Technical University of Athens (e-mail: sokaragi@mail.ntua.gr).