Governance of a Service-Oriented Architecture for Environmental and Public Security

Apostolos Papageorgiou, Stefan Schulte, Dieter Schuller, Michael Niemann, Nicolas Repp, Ralf Steinmetz

Multimedia Communications Lab (KOM), Department of Electrical Engineering and Information Technology, Technische Universität Darmstadt, Germany

Abstract: Research for the application of technical support in environmental projects has grown rapidly due to the progress of sensor network systems and the large, scalable IT infrastructures that can unify and expand them. There are numerous attempts to make use of new technologies in order to serve environmental purposes. The research presented in this paper focuses on critical aspects of the core infrastructure of a project that aims to serve these purposes as well. The nature of the actions that must be taken against environmental disasters, which are our target use cases, dictate that extended, cross-organizational information and communication systems are engaged. In order to integrate information from different sources and process heterogeneous services, the IT infrastructure is based on the paradigm of Service-Oriented Architectures (SOA). However, the distributed and often very large nature of an SOA leads to a need for governance mechanisms. We present an SOA governance approach targeted to serve the purposes of large-scale distributed IT landscapes for environmental and public security.

1 Introduction

Environmental engineering took a significant leap forward thanks to the advances in sensor networks and it is not by coincidence that environmental monitoring is an important application domain of them. Still, complete solutions for environmental support require much more than that. Services such as communication services, geo-location services, and many others must be part of any ambitious solution. When it comes to large-scale environmental projects with so many services involved, SOA appears to be an adequate approach [11].

Some studies are already dealing with the lower levels of an architecture that enables technical support for environmental purposes. Examples of these studies that have been also developed in the context of the same scenario as introduced in this paper (cf. Section 2.1) vary from energy-concerned studies to sensor network overlays and geo-information services, whereby [9] and [11] are representative of such research work.

Both SOA infrastructure and SOA governance are hot issues for research both in a theoretical and in a practical sense. Although some frameworks have appeared (e.g., [8]) that can be utilized as an SOA infrastructure, SOA governance remains often only superficially dealt with in most cases. Furthermore, SOA governance lacks a common definition and can indeed mean many things according to the respective project. The most common cause of confusion is the attempt to analyze or evaluate SOA governance as a system, a product, or something fully automated. In our previous work [6], we have proposed the following definition of SOA governance which will be the basis of our approach to governance in this paper:

"SOA governance is a management model that provides the ability to guarantee sufficient adaptability and integrity of an SOA system as well as to check services concerning capability, security and strategic business alignment. Its overall goal is SOA Compliance, i.e., compliance of legal, technical and intracompany regulations, respectively. In particular, it ensures the reliable long-term operation of an SOA."

As this definition implies, the first step is to distinguish between the overall governance procedure and the corresponding technical support mechanisms, i.e., the system that is included in the architecture in order to facilitate, enable, and/or automate governance aspects. In this paper, we present an approach on how to implement a concrete SOA governance support system based on the abovementioned definition.

After this introduction, Section 2 gives a short overview of our use case scenario, i.e. the research program SoKNOS. Our SOA governance approach is analyzed in Section 3. Concrete governance use cases that are directly dictated by the environmental needs of our project are described in Section 4. After referring to the related work in Section 5, we conclude our paper with a summary and our plans for future work in Section 6.

2 Application Domain Overview

Even though we develop an SOA governance approach with respect to a specific use case, this suggestion is intended to be universally valid. Nevertheless, in the following, we give an overview of the goals use case scenario. Furthermore, we introduce the set of services involved, so that a better understanding of the SOA governance's needs and challenges can be achieved.

2.1 Scope of the SoKNOS Project

Against the background of the security research program of the German Federal Government, the SoKNOS research project aims to develop concepts that are valuable in the support of governmental agencies, private companies, and other organizations active in the handling of disastrous events in the public security sector. In the event of a large-scale environmental disaster, SoKNOS can provide a cross-organizational view of the situation, supports the intense cooperative action between the response organizations involved, and allows for the efficient exchange of information between these organizations. The use case scenarios of SoKNOS are a flood and a chemical plant explosion, but it is flexible enough to be used for any environmental crisis that would require cross-organizational cooperation of the public security sector such as fires, earthquakes, extreme weather conditions, etc.

SoKNOS is meant to provide an extensible platform based on the SOA paradigm, which will enable the integration of any service that can be useful in the scenarios mentioned. Therefore, some of its main services will be sensor network services (for environmental monitoring), messaging services (for cross-organizational cooperation), geo-information services (for map-based control interfaces), and many more. The platform will be extensible in order to integrate any other, possibly third-party services that will be available in the future.

2.2 Service Providers and the Need for Governance

A coarse classification of services within an SOA system is the distinction between internal and external services. Both external and internal services can be objects of governance, especially in large-scale architectures. The understanding of the types of services that co-exist in such a system leads to useful conclusions about the need for governance. In the following, we try to provide a better understanding of internal and external services and of the types of service providers.

Internal services may be either new services, engineered for the purposes of the project, i.e., all services actually developed during the course of the project, or existing services that are provided by project partners. This means that the existence, the endpoints, and the functionalities of these services are already known from the beginning of the project. Typical examples of internal services are the geo-information and sensor data services (provided by project partners), the messaging services (provided by the architects of the infrastructure), and even our governance services. But many more providers offer services that are to be used in core functionalities (system core) or for the implementation of basic capabilities (system usage).

External services are offered to SoKNOS by providers who are not project partners, but rather third-party organizations. For example, these could include services from weather forecasting institutions that are outside of the project domain, sensor data from third-party providers, and communication and messaging services that are housed and managed outside of the project network.

Taking into account the need for integration of so many heterogeneous services and the need for control mechanisms of such a wide system, the need for SOA Governance becomes clear. This need is even more obvious when considering external services, because of the lack of control on them, as they are hosted on remote hosts and their providers are third parties.

Despite the lack of commonly accepted definitions and standards of SOA governance, there is a prevailing view on its main goals. These are to provide policies, best practices, and other info for the usage of an SOA and, if possible, to automate their enforcement. This means that the governance system supports the decision-making based on governance-related data. The decisions may involve human intervention or be automated.

3 Governance Support System

In previous work [6], an approach for SOA governance has been described. This approach described concepts that lie within the foundations of our work. We will refer to its models and use them in our implementation. The work at hand introduces a governance support system based on the holistic approach of [6]. It shows how the ideas described in [6] can be applied in a real system, namely our environmental application domain, adjusting it to its purposes and focusing on the system, not the model.

Fig. 1. SOA Governance Control Cycle (cf. [6])

First we summarize shortly the concepts and the goals provided in the generic model of [6]. Then we describe the extended model, new features, and technical implementation of our SOA governance system. The work presented in 3.1-3.4 (except the reference to the cube model, cf. Fig. 4) is novel and application-oriented, extending the model of [6], and is part of the SoKNOS system, though some services are still being extended or enriched.

The SOA Governance Control Cycle (cf. Fig. 1) is a simple concept that leads to a lifecycle approach, ensuring that the governance process will be dynamic, enduring and robust. It shows a sequence of the main phases that involve governance and it implies that the appearance of new challenges during the *operation* phase maintains the need for governance. Loyal to this control cycle, the SOA Governance Operational Model (cf. Fig. 2) offers a view of how these phases are translated in the context of an organization, and also gives some first hints of what a supporting system should look like.

The operational model is made up of seven entities: *SOA Goals* refer mainly to the adherence to legal, technical, and internal regulations. *SOA processes* are the entities that actually form the system that has to be governed, and they can be perceived as business processes, services, service compositions, or any other organizational processes involved in the IT solution. *Organizational Governance Entities* constitute one or more governing teams. They can have many different forms depending on the nature of the organization or project, and they usually follow a hierarchical or coordinating pattern. The existence of *metrics* actually guarantees the endurance of efficient governance, and the existence of a complete control cycle. Some kinds of metrics, like Quality of Service (QoS) metrics or ratings will be further analyzed in Section 3.2, but the spectrum and significance of metrics are wider than this and can actually mean a lot more. *Best Practices* are documented results of prior experiences in form of policies and ratings, which lead in turn to new policies and optimal governance operation. *Policies* are core ingredients of any SOA governance model as they define procedures that an SOA must adopt. These policies range from technical specifications to security standards, and more. A schema for governance areas concerning policies is defined in [6] and it will be presented in more detail in Section 3.2. Finally, the purpose of the *SOA Maturity Measurement* element is to provide feedback to the SCE, concerning the maturity level of processes.

Given this background, the next sections describe our approach to implementing and applying a SOA governance system based on the principles of the presented model. In Section 3.1 we explain how our approach constitutes a distributed governance scheme by exploiting the meaning of user roles. The understanding of the underlying data model is crucial for the understanding of the whole governance procedure and is therefore separately presented in Section 3.2. The services offered by our governance system can be coarsely divided into those that facilitate manual governance and those that enable automated governance. The former are the subject of Section 3.3, while the latter are described in Section 3.4.

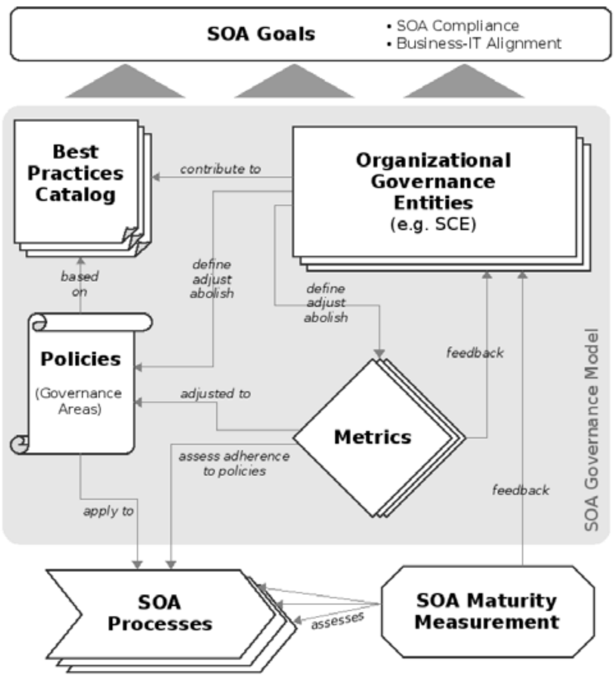

Fig. 2. SOA Governance Operational Model (cf. [6])

3.1 Distributed Governance Using Roles

In order to understand how we define distributed SOA governance, we have to distinguish between the governance approach and the governance supporting system. A distributed governance approach does not mean the existence of a distributed governance system and vice versa. Distributed governance (as described in [5]) addresses the distribution of responsibilities among independent authorities, whereby no central authority has complete access or full rights to every part of the process. This approach requires clearly defined rights and therefore many governance roles. However, roles do not appear only in distributed approaches. They can appear in many solutions, having different meanings (as in [2] and [10]).

There are some generic governance roles that should be introduced in general-purpose governance models. The roles in the use case at hand are governance-oriented but are derived from the roles that already exist in the project. In an SOA, user roles are important not only for governance but also for other aspects of the system. So, with the knowledge of the predefined set of roles of our project and a set of proposed generic governance roles, we can assign governance capabilities to general project roles, keeping in mind that governance roles and project roles are different.

In more detail, we map the set of *General Project Roles* to a smaller set of *Governance Roles*, including *Governance System roles* as well as *Governance Project roles*. As Governance Project Roles, we denote any roles that take part in the governance process, but do not use the supporting system. In particular, this applies to governance actions that are not system-supported. Then, responsibilities/capabilities are assigned to these roles. This schema is denoted in Fig. 3, where adequate project roles have to be mapped to one or more governance roles. Some governance roles have no counterpart in the project role model, so, these roles are defined especially for governance purposes. Such roles need to be assigned to proper experts.

3.2 Governance Data

All governance-related data is stored in an adequate database. An overview of the data model used by the governance system is necessary in order to understand the nature of the provided services. The description that follows implies which entities and metadata form the core of our governance database, and it gives a first hint about how they can be exploited in order to facilitate manual governance and enable automated actions.

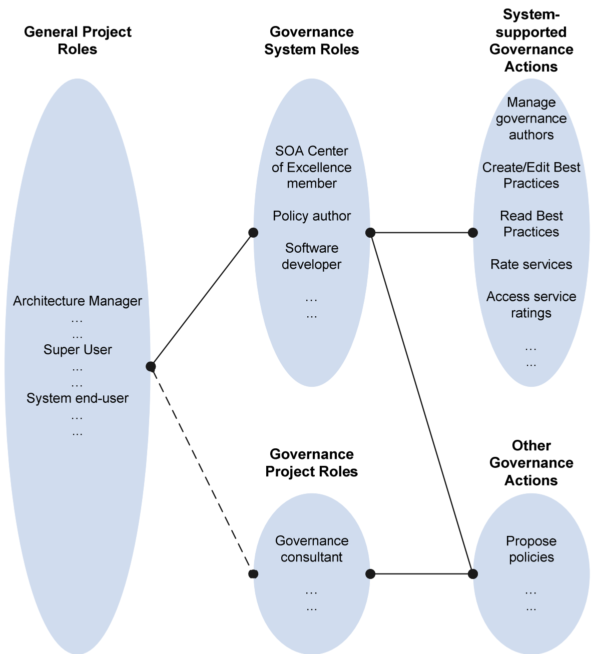

Fig. 3. Governance roles and responsibilities mapping scheme

User Roles

The idea behind user roles has already been resolved in the previous section, and it is easy to imagine a data model that can sustain this kind of information. The existing data should be adequate in order to assign roles to system users, and at a higher level, capabilities and access rights to roles.

Policies and Best Practices

These are the objects that explicitly dictate tactics, policies, and ways to improve the functionality and efficiency of the architecture. They are interrelated, in that the latter contain some kind of rated tactics/policies – negative or positive. A set of such policies and best practices would not be of much use if they were not related to services, implicitly or explicitly. In this respect, our system follows the cube model described in [6] in order to define governance areas and classify policies and best practices inside these areas. Services can use a similar grouping technique, and despite the differences in the domains used, useful mappings can be performed.

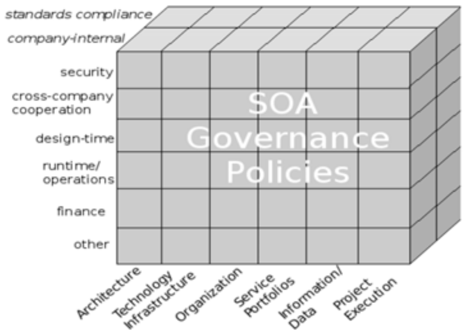

Fig. 4. Fraction of the data model for policies and best practices (cf. [6])

The mentioned cube model (cf. Fig. 4), defines six major governance areas, six cross-sectional areas, and two major features [6]. All policies and best practices can then be mapped to a cell in the cube by the assignment of a triplet (*m*, *s*, *d*) to them. The letters stand for *m*ain aspect and *s*ide aspect, each of which have six possible values, and *d*imension, which has two possible values. There are two reasons for having this analytical description and fine categorization not only in theory but also inside the system. First, there is the need for a foundation for a possible real-time mapping of policies to services, and second these objects are supposed to be managed mainly as standalone documents by non-technical staff, with the system just assisting the procedure. In such a case, all these details must accompany the governance objects.

Service Metadata

The objects that must ultimately be exposed to governance in an SOA system are the services themselves, with all of the remaining system having implicit or explicit relations to them. However, at the level where our governance support-system functions, we can safely minimize the problem, abstract things from the architecture and refer only to technical services (i.e., SOAP-based Web services or other callable services – HTTP, JSONRPC, etc.), and of course, technical business processes (e.g., compositions of services).

The service metadata from the governance data model extend the Web service descriptions.. Before describing the needed extensions, it is crucial to understand that, while the entities that have been described until now focus on the facilitation of *manual* governance (cf. Section 3.3), the main goal of the usage of services' metadata is to enable *automated* governance (cf. Section 3.4). Still, some data are involved in both cases.

Services' metadata are divided into three categories. Functional description data are used to support the notion of service domains (see also [7]). QoS and monitoring data are used to support service evaluation and selection [1]. Finally, rating data are used to allow human intervention in the system's automated procedures. In more detail, we need the following information about Web services in order to enable automated governance:

- Functional description data as a 4-tuple (*o*, *d*, *i*, *f*) that assists the grouping of services in many respects and the orientation of the services inside the architecture. For organization, *o* indicates the provider but as an institution or organization rather than in a technical sense. All necessary information about the provider in a technical respect is normally included in a technical description like a WSDL document. *d* assigns a project-wide functionality domain to the service, while *i* (for internal) categorizes the service further, but inside the service provider's own domain. Finally, *f* describes the service *functionality* semantically, based on a project-wide established ontology. The latter is crucial for real-time service selection, as it can be used to identify "alternative" services.
- QoS data as an average response time, which is a result of monitoring performed by various parties both in advance and during the usage of the system. Still, the monitoring can only be performed by parties with some certain governance-related rights.
- Rating data, which is currently designed to be manual. It could be automated with calculations based on existing data (QoS values, ratings of related Best Practices, etc.).

3.3 Facilitating Manual Governance

As "system-supported facilitation of manual governance" we denote our service-oriented implementation that provides distinct governance functionalities for different users, so that the architecture can be "governed" according to the exact regulations that a model implies.

We list here the main categories of Web services offered by our system in order to assist manual governance:

- Web services for role-based access to policies and best practices, as well as their author objects.
- Web services for management of user roles and their corresponding rights and capabilities.
- Web services for access to domain information of the services, i.e., for the organizing and categorization of Web services.
- Web services for role-based access to the rating of services.

3.4 Enabling Automated Governance Mechanisms

There are many governance actions that a system can automate, ranging from service substitutions and automated workflow re-planning to automated monitoring and usage of monitored data. The foundation needed to achieve these goals is a set of suitable services, whose adequate application and combinations lead to partly automated governance systems.

We list here the main categories of Web services that our system offers in order to support automated governance. They may be better understood with the automation example of the next section.

- Web services for reporting and accessing monitoring data.
- Web services for exploiting the *functional description data* of services, e.g., for finding services that have equal functionality, etc. This can also be enhanced by semantic descriptions of services.
- Web services for accessing QoS and rating data.

4 Environmentally-dictated Governance Use Cases

An example of an automated governance mechanism that we employ was initially presented in [1]. As analyzed there, it is possible to compose business processes and workflows based on QoS parameters. This approach is based on the assumption that for each task in a workflow, a set of alternative Web services with similar functionality is available, and that these services have different QoS parameters. Using heuristics, it is possible to solve the optimization problem of how to select Web services for each task in a process or workflow so that the overall QoS and cost requirements of the composition are satisfied. But apart from optimization, this technique provides service reliability. Our application domain aims to face environmental disasters. So, service failure or unavailability must have been foreseen and dealt with. This is done with runtime service substitution based on the described technique. For example, our project involves Web services from many different providers (fire agency systems, a geo-informatics institute, as well as new, project-driven sensor services) that provide sensor data for the same areas. So, alternative services can be found in cases of failure or low performance. Having an abstract implementation of this service substitution logic and its prototypical integration in the project, its full integration is currently under development.

Another aspect of our environmental scenario that raises the importance of our governance system has to do with the working methods of the organizations that face environmental disasters (e.g. fire agency, police). Because of the huge extent and the vast number of documented tactics and policies of these organizations, a support system that provides best practices (in the context of the used technology) at any time is of great use. Same services can be used against same environmental problems, and services that proved inefficient can be banned. Furthermore, the role-mapping of our governance implementation, pictured compressed in Fig.3, is dictated directly by the needs of an environmental project. Among the project roles that have to be mapped to governance roles are users such as firemen, sensor network appointees etc.

5 Related Work

Among environmental projects, past and present, many approaches can be found that are interesting in our project's general context, but to our best knowledge no such project follows a similar architectural approach. Thus, the comparison and the exchange of ideas could not answer the questions: why to apply an SOA, why SOA governance is needed, and what kind of SOA governance should be adopted.

Some interesting insights lie among the results of an environmental programme named PETIT-OSA. Such an outcome is the Italtel Softswitch System [4], which provides environmental monitoring services. It is built on top of an extensible platform where it is possible to add services (e.g., a fire simulator, a service providing real-time data for damages). Apart from the fact that it is mostly targeted to telecommunication services, it is *not* based on open standards, hiding the service governance aspect inside the architecture of its platform.

In any case, the main focus of our study at this point is the answer to another question, namely what is the best way to practically integrate SOA governance in a large and heterogeneous IT system. The state-of-the-art in this field is what most concerns our work. In the following, we summarize three approaches whose concepts have also appeared in our proposal. However, our contribution is not limited to merely offering a proposal for an extended SOA governance model, but lies also in the application of such a model to our environmental scenario, and finally, the presentation of a system which can be seen as the implementation of such a model.

The SOA governance lifecycle described in [7] offers a suitable foundation and ideas on building a more complete and system-supported SOA governance. We have enhanced this lifecycle model by defining service domains more concretely. The grouping of services in domains of various types is indeed necessary for governance. In [7], different types of structuring criteria are proposed in order to define four domains (process, product, geographical, and functional) so that services are grouped within each domain and are more effectively and easily governed. The notion of service domains as part of our system has been discussed in Section 3.2.

Some kind of distinction between *governance as a holistic approach* and *governance as a supporting system* can be found in [2] and [10]. Furthermore, both publications highlight the importance of user roles with regard to the service life cycle [2] and the software development process [10]. As analyzed in Section 3.1, we built our own user role model which can be applied to our use case scenario. Thus, we incorporate user rights, authorization issues, and a mapping from project roles to governance system roles. Studies [2] and [10] include governance in the whole process of design and development management. While they present interesting approaches to SOA governance, the governance of the actual operation of an SOA-based IT landscape is only one minor part of them. In our work, we focus on the control and steering of an SOA during the actual operation of the IT system. Hence, we have separated governance of system design and development from governance during run-time, focusing on the latter.

6 Conclusions and Future Work

Facing the governance needs of a service-oriented system for environmental and public security, we presented a system-supported SOA governance approach that focuses on the control and steering of an SOA during the actual operation of the IT system.

We described data models that we consider critical for successful SOA governance and for the achievement of our system's two main goals, namely the facilitation of manual governance and the enabling of automated governance. Handling of policies, best practices, and monitoring data, as well as automated service replanning are all aspects of great importance in our approach. Focusing on the above, the issue of SOA governance is developed beyond simple system supervision or management.

Subjects of future work are a possible integration of our governance services with existing security mechanisms, and the further automation of governance actions. To this end, we are investigating the exploitation of semantic service descriptions in order to automate and enhance the service substitution system. Furthermore, we are investigating distributed mechanisms, such as distributed exchange and discovery of services metadata, which could possibly better support our approach.

Acknowledgements

This work is supported in part by the E-Finance Lab e. V. (www.efinancelab.de) and the BMBF-sponsored project SoKNOS (www.soknos.de).

References

[1] Berbner R, Grollius T, Repp N, Eckert J, Heckmann O, Ortner E, Steinmetz R (2005) An approach for the Management of Service-oriented Architecture (SoA)-based Application Systems. In: Proceedings of the Workshop Enterprise Modelling and Information Systems Architectures, EMISA 2005, pp 208-221

[2] Derler P, Weinreich R (2006) Models and Tools for SOA Governance. International Conference on Trends in Enterprise Application Architecture, Berlin, Germany. Springer, Berlin Heidelberg

[3] Erl T (2005) Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, Upper Saddle River

[4] Italtel (2007) press release: The Italtel Softswitch, an intelligent sentinel for environmental monitoring in the Vesuvius area.

[5] Nadhan EG (2004) Service-Oriented Architecture: Implementation Challenges. Microsoft Architect Journal 2, April 2004, pp 24-32

[6] Niemann M, Eckert J, Repp N, Steinmetz R (2008) Towards a Generic Governance Model for Service-oriented Architectures. In: Association for Information Systems, Proceedings of the Fourteenth Americas Conference on Information Systems, AMCIS 2008, Toronto, ON, Canada, August 2008

[7] Schepers TG, Iacob ME, Van Eck PA (2008) A lifecycle approach to SOA governance. In: Proceedings of the 2008 ACM Symposium on Applied Computing, Fortaleza, Brazil, March 2008, pp 1055-1061

[8] SOPERA Open Source SOA Framework, SOPERA AG, http://www.sopera.de

[9] Stasch C, Walkowski AC, Jirka S (2008) A Geosensor Network Architecture for Disaster Management based on Open Standards. In: Digital Earth Summit on Geoinformatics 2008: Tools for Climate Change Research. Heidelberg, pp 54-59

[10] Strnadl C (2007) Bridging Architectural Boundaries Design and Implementation of a Semantic BPM and SOA Governance Tool. ICSOC 2007, pp 518-529. Springer, Berlin Heidelberg

[11] Walkowski A (2008) Model-based Optimization of Mobile Geosensor Networks. In: The European Information Society - Taking Geoinformation Science one step further. Berlin, pp 51-66