International Congress on Environmental Modelling and Software

1st International Congress on Environmental Modelling and Software - Lugano, Switzerland - June 2002

Jul 1st, 12:00 AM

Generic Integration in Environmental Information and Decision Support Systems

Ralf Denzer

Generic Integration in Environmental Information and Decision Support Systems

Ralf Denzer

Environmental Informatics Group (EIG)
Goebenstr. 40, 66117 Saarbrücken, Germany, www.enviromatics.net
Ralf.Denzer@enviromatics.org

Abstract: Environmental Information Systems (EIS) and Environmental Decision Support Systems (EDSS) are major building blocks in environmental management and science today. They are used at all levels of public bodies (community, state, national and international level), in science, in management and as information platforms towards the public. EIS and EDSS are usually said to have certain characteristics, which distinguish them from standard information systems, e.g. information complexity in time and space or uncompleteness or fuzzyness of data items. By the very nature of the complex tasks involved, different methodologies can be an option while developing a new system, for instance modelling, decision theoretic approaches, artificial intelligence, geographical analysis, statistics and many more. As software developers, we face the situation that we have to recompose these different methodologies in different application scenarios over and over again. This is rather cumbersome, because the tools implementing certain methodologies are usually not very helpful in the integration process. This paper discusses the question, how different EIS and EDSS tools can be integrated in a generic way. For this purpose, we discuss a number of integration strategies and give 2 examples of current EU-funded projects.

***Keywords**:* *Generic integration, Environmental Information Systems (EIS), Environmental Decision Support Systems (EDSS), EU FP5, GIMMI, IMARQ*

1. EIS and EDSS

There have been many approaches to identify what EIS and EDSS are and all of them are probably as right or as wrong as the perspective of the respective reader allows them to be. A definition of the systems depends on the viewpoint of the person defining them and may be very different, e.g., between a modeller and a software engineer. This introduction does not give yet another overview on EIS and EDSS, it rather focusses on the key issues necessary to understand the integration issue. For general overviews on EIS and EDSS, seen from the software point of view, see Swayne [2000] or Denzer [1999].

Key elements of EIDSS (as shortcut for EIS and EDSS) are usually said to be

- Complex, time and space related data which is often incomplete, fuzzy or of the wrong scale needed for a given task
- Complex algorithms resulting in complex software tools which may come from any domain of information technology, e.g. databases, meta information systems, real time monitoring systems, geographical information systems, networking, artificial intelligence, etc.
- Complex data management issues due to the variety of autonomous data providers and consumers
- The absence of *real* data and metadata standards for many domains (meaning those which are used by a broad community, not the many so-called standards which are defined by single individuals or organisations)
- The fact that for many problem solving issues, you need to bring different tools into one holistic solution for end users, where the tools may use different algorithmic and/or data management strategies

The latter point is the main focus of this paper: those cases, where you can *not* solve the problem with one single tool, meaning that you need to combine at least two or even more tools into a software solution. The point of complex data management issue needs to be addressed in this comtent as well, because tools can not be separated from their data management.

In the following sections, we discuss: a) the tools involved in EIDSS; b) typical today's concepts of integration; c) the goals and steps towards generic integration; and d) two examples of current projects of the authors and others.

2. EIDSS Tools

This paper considers four main *building blocks* of typical EIDSS (figure 1)

- Models
- Geographical information systems (GIS)
- Decision support systems (DSS)
- Data management systems

Many EIDSS have a combination of at least two of these building blocks. Many others have only one building block (models, GIS or DSS), but should at least have a second one, namely a proper data management system. Those which only have one data management system (e.g. a typical environmental facts database like an emission inventory) are not of interest in this context.

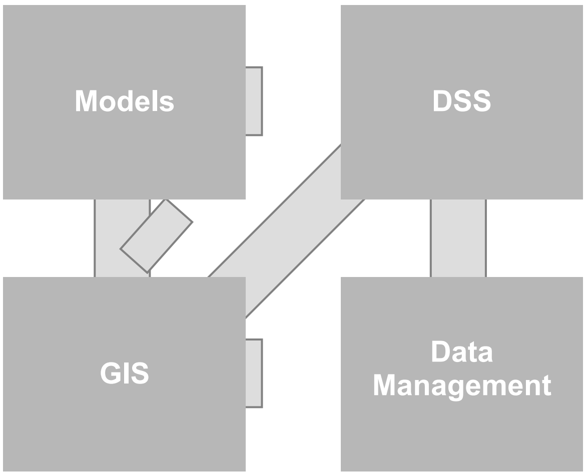

Figure 1. Building blocks of EIDSS

This selection of typical building blocks might already be questioned, depending on where a certain reader might stand. E.g. some people migth say that if you use a bayesian belief network or a rule base in a DSS, then you are doing modelling. Others might say that if you model geographic relationships that you work in a GIS per definition. These types of discussions have been around a long time and from a software engineering point of view are not very fruitful.

The selection of these four building blocks in this article is solely based on the need to develop integrated software systems for reasonable cost, which means that you can not start from scratch every time again and that you want to use existing software as much as possible. This is where the question of existing *software tools* comes in. Seen from this viewpoint, the reader will understand why these four blocks are chosen. In this context

- The term "models" denotes stand-alone models or modelling suites.
- The term "GIS" denote the well know geographical software tools.
- The term "DSS" denotes tools based on AI techniques.
- The term "Data management systems" denotes database systems, including meta databases and networked information infrastructures.

Again, this definition may be questioned, e.g. by noting that GIS's are or can be DSS's, e.g. if your decision strategy is of such nature that you can build it using a GIS (e.g. the one described in Veitch[2000]). And again, this discussion is not fruitful.

The approach discussed in this paper is that there are four main methodologies or technologies, which you may find in an EIDSS, namely

- an approach based on numerics (models),
- an approach based on geography (GIS),
- an approach based on AI (DSS),
- an approach based on data management and networking (data management systems),
- taking into account possible overlaps,

and that we often need to combine more than one of the respective tools in a *software solution* for a given application. Therefore, our main discussion is related to the *interoperability* of systems.

If you mention this particular term, software vendors (in particular those of one class of tools), say that these issues are all solved, because they sell – what they call – "open systems". Everyone who has ever tried to do this type of integration knows that these issues are not solved. How can they be if most tools are not or are just starting to get interoperable amongst the *same* class of tools.

3. Integration Today

A very common integration strategy leaves the end user with the problem of integrating different tools by providing data import and export facilities only (typically files). This results in the end user wasting incredible amounts of time importing, exporting and converting data between different systems instead of being able to spend more time on the task itself. It also produces large amounts of data files which noone can really manage over longer periods of time. This strategy, which we might call a *null integration strategy* may be tolerated for scientists but is inacceptable for other user groups like managers or the public. Therefore, this strategy is not the concern of this article.

The next level, which is a typical state-of-the-art integration concept, is shown in figure 2. The example is from a publication of Leon [2000]. The original diagram has been slightly modified in style, not in content, for this paper.

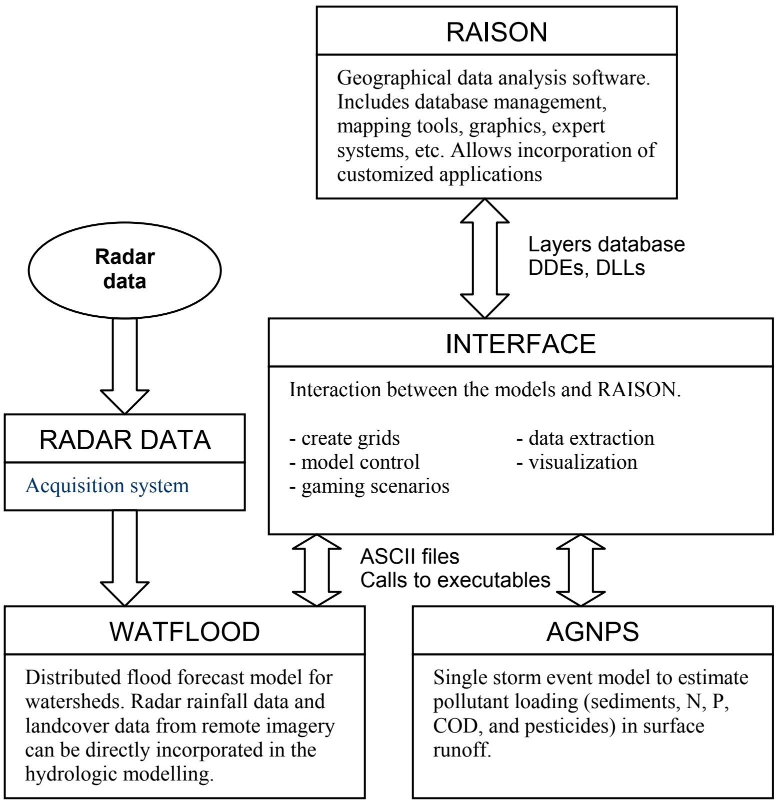

Figure 2.
EDSS for Sedimentation and Nutrient Transport after Storm Events (courtesy Luis Leon)

The system implements an end user decision support solution for nonpoint source pollution in surface waters. It uses 2 models and the decision support system RAISON.

In Lam and Swayne [2001], the same group of developers discusses design issues raised with the development of integrated EDSS, in particular the question whether you should design specific systems, tailored to the needs of only one application, or whether you should try to generalize from the start, which clearly means that there is a heavier investment. The authors' experience is that the second approach is more effective for complex EIS systems and pays off in the long run. Technically, they regard the integrational EIS as the sum of many parts (i.e. software components), which stands in the tradition of the software engineering goal to reuse as much code as possible. In short, this means that we would like to use off-the-shelf tools and just "plug" them together. Regarding this wish, they come to the conclusion: "In short, it was difficult to construct an integrative EIS system from simple connection of many existing software packages." They also remark that one key feature to achieve this is interconnectivity. We all know that unfortunately many available tools today are still very bad with regards to this requirement.

The key issue involved with this type of integration is that linking the different software pieces together is a complex task, in particular if the intention is to deliver an easy to use environment for end users. The process usually involves heavy programming on data exchange and the user interface. Depending on available data exchange facilities of the individual components, a "zoo" of data exchange mechanisms is used (see figure 2). Key specialists are also needed to complete the task and the development can come with considerable cost.

Another issue, which the diagram does not show, is that each tool may have it's own data management facility and that they may use completely different technologies. In many cases, the development process is expensive enough that an integrated data management concept is of low priority. This may result in poor data management which leaves the user with this problem.

We may call this type of integration *in-project integration strategy*. The way the integration is performed is done in a way which only suits the particular project. A different project starts with the same problems over again and involves the same costly development process. Unfortunately, this is the best which can be done today with the existing base tools and with the existing state-of-the-art integration concepts. However, there are recent achievements which promise more generalized solutions in the near future. These will be discussed for the remainder of this article.

4. Generic Integration

The term "generic" is a word which is very trendy but usually not defined by many of those using it. The term is also often not very well understood. In this respect it is similar to the terms "open" or "interoperable". IFIP Working Group 2.1 (www.ifip.org) is conducting a working conference on generic programming in 2002 (WCGP'02) and on their home page they say: "Generic programming is about making programs more adaptable by making them more general." In computer science, the focus of generic concepts seems to be on formulating generic versions of algorithms and program structures at the moment.

In generic integration, we talk about *whole systems*. Systems are composed of system components or *services*, if we are in a distributed environment. Therefore generic integration means *generic systems* composed of *generic services* through *generic communication infrastructures*. Our particular research goal is and was to make components and interfaces general enough that reprogramming is avoided if you move from one application to another.

To make this a more practicle definition, we may propose the following tests:

> <u>Test A</u>
>
> A service is generic, if it is independent of the application domain.
>
> This means that you must be able to use the same software for different end user scenarios (or data types), e.g. for pesticide management and marine ecosystems (examples of section 6), without re-programming (ideally).
>
> <u>Test B</u>
>
> A communication infrastructure is generic, if it is independent of the application domain.
>
> This means that you do not have to reprogram the communication for different end user scenarios (or data types).
>
> <u>Test C</u>
>
> A system is generic, if it is independent of the organisational structure of a given user site (or a group of user sites).
>
> This means that if you transport the system from one organisational structure to another, that there is no reprogramming either (different pilot regions of section 6).

It is clear, that not all functionality of an application can be abstracted in a way that they become generic. There will always be components which can be made to look similar to different application domains (e.g. a data catalog of a pesticide management system and the one of a marine system) and there are others which are very specific to the given domain.

We also need to consider all sorts of legacy data and models, meaning data and models which have not been designed with a generic approach in mind. Therefore we may propose a further test:

> <u>Test D</u>
>
> A system is generic, if it allows the integration of non-generic components into generic services, in a way that they look like generic services to the rest of the generic system.
>
> This means that the system defines wrappers in a way that legacy data and software can be made to look like generic services to the outside world.

Our main approach to generic integration consists of a distributed infrastructure which is independent of the data types, which implements the most important general services and which allows a very scalable design of a particular network. The work is basically lifted one level up – from programming to administration – and therefore pays off the high initial investment once you build 3 or 4 concrete applications.

5. Generic Integration of EIDSS Building Blocks

This section presents a brief discussion of the state of the art of integration of the major EIDSS building blocks introduced in section 2, with respect to the definitions given in sections 3 and 4.

Without disrespecting existing systems and projects, a very rough and global assessment is given: where do we stand at the moment in the author's opinion. This assessment is mostly based on practice and little on literature.

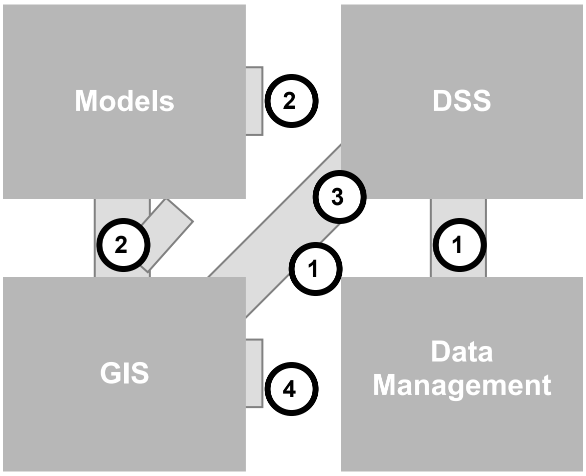

Figure 3. Building blocks of EIDSS revisited

1 Interfacing of models and DSS with data management systems is generally recognized as not being optimal. Generic approaches yet to be developed.

2 Interfacing of models with GIS and of models with DSS is usually done as "in-project integration" as described in section 2. Generic approaches yet to be developed.

3 Interfacing of GIS and DSS has been demonstrated in two ways mainly: either the decision support strategy can be implemented in the GIS (like in Veitch [2000]), or the DSS implements it's own GIS user interface (like in Swayne [1992], due to the fact that, at least in the past, most GIS's were very closed systems). Generic approaches yet to be developed.

4 Interfacing of GIS with data management systems has successfully been demonstrated and is in practical use, with respect to Test A, Test B and Test C (Güttler [2000]), with respect to Test D there is ongoing research (Denzer [2001]).

One example is the WuNDa system (Güttler [2000]), an information platform for the city of Wuppertal, Germany (see figure 3). We succesfully demonstrated, that the system requires no reprogramming if you add new data sources from new application domains to the network. The communcation system, the meta information database, all distributed services, the links to legacy systems and the user interfaces are implemented in such a general way, that they are independent of the data types of the data sources attached to the network.

6. Recent Research

This section introduces two research projects which have started in spring 2002 within the 5th European Framework Program. One of the calls in the program (Cross Program Action 3, or CPA-3) is dedicated towards projects promoting the use of geographic information. Amongst other activities, a strong focus is put on interoperability issues, in particular, citing the call: "Develop engineering techniques for designing and providing generic spatial data services and software components based on standardised interfaces that can be reused and integrated in future applications and other contexts."; "Improve accessibility, usability and exploitability of GI and related reference data with focus on metadata interoperability, standards, and semantic and ontological compatibility issues."; "Support the setting up of the European Spatial Data Infrastructure and a Common Reference System, and contribute to global initiatives like GMES, GSDI, OGC, ISO, JTC, in conjunction

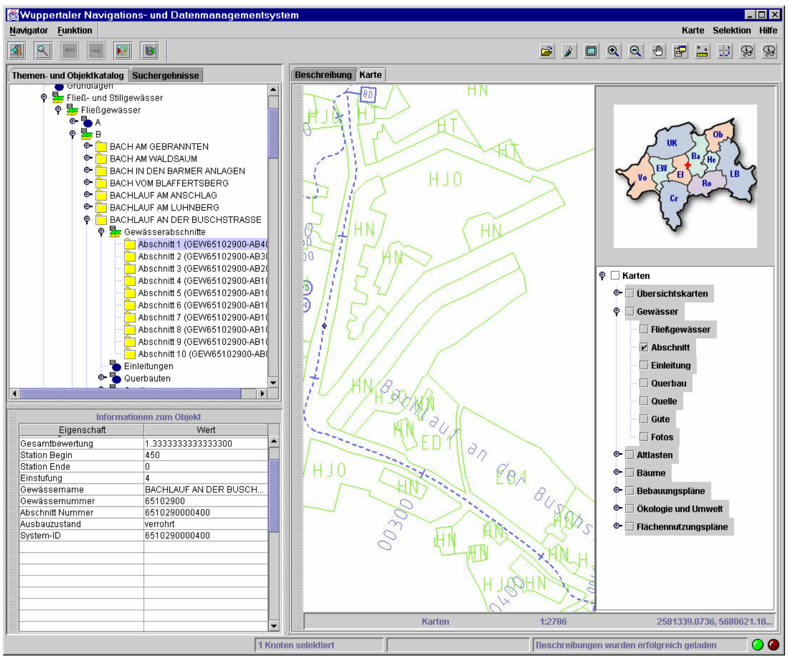

Figure 4. WuNDa navigator

with existing national and international work, through national mapping agencies and others.”

EIG, the author’s institute, is a major partner in 2 projects funded under this call. The projects deal with very different application domains but have remarkable common intentions, although directed under two completely different consortia and coordinated and planned by individuals not knowing each other. EIG is the only common link.

6.1 GIMMI

GIMMI (Geographic Information and Mathematical Models Interoperability) aims at bridging the gap in Pesticide Impact Assessment domain between data and service providers, scientists and end users: In particular:

- allowing the inter-operability via web of geographic information (GI) based environmental protection services physically distributed and locally managed and maintained by their own inventors and generators
- providing the proper IT structures to represent and manage temporal knowledge inside a GI system.
- integrating in the IT infrastructure state-of-the-art legacy systems for document management and report generation

GIMMI intends to implement 3 kinds of services:

- on-line data access, when the user seeks to "drill down" into the huge amount of GI distributed in different formats and in different sites
- on-line simulation, when the amount of data involved and the time required to answer allow it
- off-line study, when the requested services require huge amounts of data, long time or human experts

TXT e-Solutions	Italy	Software company
Fraunhofer AIS	Germany	Research Center
EIG	Germany	Academia (Saarbrücken)
LABSITA	Italy	Academia (Rome)
ERSAL	Italy	Region of Lombardy
SARA	Spain	Region of Catalonia
INAMHI	Ecuador	Meteo/hydro of Ecuador

Table 1. GIMMI Consortium

GIMMI will be based on a distributed service concept integrating GIS, legacy models and data/ metadata management and will also interface with other, existing services like E-commerce engines, data mining and workflow systems.

Table 1 shows the consortium partners. GIMMI. Pilot end users are the region of Lombardy and the Region of Catalonia.

6.2 I-MARQ

I-MARQ (Information System for Marine Aquatic Resource Quality) will deliver real-time information on coastal water quality into a variety of end-user markets, via a dynamic GIS-based system. The project will develop and validate advanced data fusion, modelling and management algorithms to generate high-quality data content. The system will support decision-making by various end-user groups including:

- Citizens concerned about environmental quality in recreational waters;
- Local authorities seeking a quality tourism cachet and wishing to avoid hazard to public health
- Companies seeking to validate environmental performance and avoid liabilities from pollution incidents.

The overall goal of the project is to develop a GIS which can exploit diverse data resources in order to deliver ‘best estimate’ information on environmental quality of coastal waters. Thus the overall sequence of objectives is:

- To specify a system for monitoring & displaying coastal & estuarine water quality. This will be based on the needs of significant user categories, defined through a combination of survey and analysis
- To develop a system which meets the above specification, using novel techniques in data processing, management & GIS. This will offer significant improvement in timeliness of information, compared with existing on-line systems which present information based on historic, regulatory measurements
- To pilot the system and evaluate its performance against user requirements. This will generate pilot operating experience in three different EU coastal and estuarine regions: one coastal region in the Mediterranean; one coastal region in the NE Atlantic; and estuary system also in the NE Atlantic
- To plan for enhancement & commercial application of the validated system. This

will aim to define a feasible action plan for future development of a commercial-scale system.

By integrating data from many sources and models, the system will be designed to meet the varying information priorities within different markets. Development of the system will require novel advances within a GIS context of data management, fusion and modelling techniques, which have not been achieved in previous work.

NERC	UK	Research Center
GKSS	Germany	Research Center
EIG	Germany	Academia (Saarbrücken)
Telespazio	Italy	Satellite company
BMT Systems	UK	Intern. RTD organisation
SOC	UK	Academia (Southampton)
Marinetech South	UK	Consultancy
IOPR	France	French ocean. Institute
ABP Marine Res.	UK	Consultancy
AMRIE	Belgium	European Interest Group
Ville d'Antibes	France	City

Table 2. I-MARQ Consortium

Table 2 shows the consortium partners. The system will be piloted in three regions: Ligurian Sea (France & Italy), Helgoland Coast (Germany) and Solent (UK).

6.3 Common Issues

The two projects are considerably different with respect to what we will deliver to end users. The application domains are very distinct, some of the tools which will be connected to the information infrastructure have completely different foci (e.g. E-commerce engine in GIMMI, real-time in I-MARQ), and there are different technological partners who wish to incorporate their software systems into the network (e.g. different GIS systems). However, both projects show a remarkable similarity in technological objectives with respect to generic approaches. With EIG being partner in both consortia, there is a unique opportunity to assess how far generic approaches can be applied today.

7. Conclusions

The purpose of this paper is to motivate why integration in EIDSS is a very difficult undertaking. In particular, there is no need to put a blame on systems which today use less elegant integration strategies than others. We should not forget that the basic computer science concepts allowing generic integration are to be developed yet and that basic IT methodologies easing the way are just coming into the market, e.g. the concepts of componentware or design patterns. Some of these terms were not even known only 3 or 4 years ago. Some of the better programming environments allowing better systems design (most notably Java) have only reached full maturity about a year ago. Pressure from users and software developers has also forced tool providers to offer component based packages. This will change the situation considerably.

Integrated data management strategies are still to be given extended attention. Many solutions are still very poor with respect to this issue.

The question how to deal with "integrated tool / integrated data" environments is open to computer science research.

The situation has changed considerably during the past three years and encourages the development of new EIDSS, which provide better and easier-to-use functionality to end users.

8. Acknowledgments

I wish to thank Luis Leon of University of Waterloo for the permission to use one of his projects as an example for the complexity of the integration task.

A big thank you goes to my colleague and friend Reiner Güttler, with whom I am co-directing a growing research institute in Saarbrücken, and with whom I have been working on these issues for over 10 years.

Projects GIMMI and I-MARQ are funded under the 5th framework program of the European Union under contract numbers IST-2001-34245 and IST-2001-34039.

9. References

Denzer R., Environmental Software Systems, *Encyclopedia of Computer Science and Technology, Vol. 40(25)*, pp. 51-61, Marcel Dekker Publishers, 1999

Denzer R., Güttler R., Swayne D.A., Integrated Spatial Decision Support, *Environmental Software Systems Vol. 4 – Environmental Information and Indicators*, pp. 227-235, IFIP Conference Series, 2001

Güttler R., Denzer R., Houy P., An EIS Called WuNDa, *Environmental Software Systems Vol. 3 – Environmental Information and Decision*

Support, pp. 114-121, Kluwer Academic Publishers, 2000

Lam D., Swayne D. A., 2001. Issues of EIS software design: some lessons learnt in the past decade. *Environmental Modelling and Software, Volume 16 No. 5*, pp. 419-425, Elsevier Science, 2001

Leon L. et al., Water Quality Model Integration in a Decision Support System, *Environmental Software Systems Vol. 3 – Environmental Information and Decision Support*, pp. 187-194, Kluwer Academic Publishers, 2000.

Swayne D.A. ,Lam D.C.L. ,Fraser A.S. and Ongley E.D., RAISON - Information Systems for Managing Water Resources Models and Data, *International Journal of Water Resource Development, Vol. 8(3)*, pp. 166-172, 1992.

Swayne, D.A., Denzer R., et al., Environmental Decision Support Systems – Exactly what are they? *Environmental Software Systems Vol. 3 – Environmental Information and Decision Support*, pp. 257-268, Kluwer Academic Publishers, 2000

Veitch S., Lessons from an Environmental Information System Developed to Select a Radioactive Waste Disposal Site, *Environmental Software Systems Vol. 3 – Environmental Information and Decision Support*, pp. 177-186, Kluwer Academic Publishers, 2000.