KETALE Web application to improve collaborative emergency management

Michael Ammann, Tuomas Peltonen, Juhani Lahtinen, Kaj Vesterbacka
Radiation and Nuclear Safety Authority, Finland
{firstname.lastname}@stuk.fi

Tuula Summanen, Markku Seppänen, Pilvi Siljamo, Annakaisa Sarkanen, Minna Rantamäki
Finnish Meteorological Institute, Finland
{firstname.lastname}@fmi.fi

ABSTRACT

KETALE is a database and web application intended to improve the collaborative decision support of the Finnish Radiation and Nuclear Safety Authority (STUK) and of the Finnish Meteorological Institute (FMI). It integrates distributed modeling (weather forecasts and dispersion predictions by FMI, source term and dose assessments by STUK) and facilitates collaboration and sharing of information. It does so by providing functionalities for data acquisition, data management, data visualization, and data analysis. The report outlines the software development from requirement analysis to system design and implementation. Operational aspects and user experiences are presented in a separate report.

Keywords

Emergency management, radiological and nuclear disaster, collaboration, web application

INTRODUCTION

Emergency management of nuclear or radiological disasters is all about collaboration: one team assesses the likelihood and magnitude of a release; another team is in charge of making dispersion calculations with the given release; their results are given to yet another team so that they can plan suitable measurement campaigns; still another team assesses possible health effects and the need for interventions by taking into account the most likely dispersion situation and already available measurement results; and there is a need for coordinating all this effort and communication to the public, which in turn can only be done successfully if first hand and timely information about the current state of affairs and about likely future developments are readily at hand. Of course collaboration does not stop here (there are many more intra- and inter-institutional needs for information exchange at various political levels) but this does roughly picture the collaboration that is needed between and within STUK and FMI to cope with emergencies.

The working of all these groups relies on information exchange and on adequate computer support. Although there are many tools available today to support the different groups in their various tasks, there is no one tool (or decision support system for that matter) that can do all for all users. Admittedly, some decision support systems (e.g. RODOS or ARGOS) go a long way in supporting many of the above mentioned tasks, but there are practical obstacles (institutional traditions, user profiles, IT infrastructure, etc.) to a tight one-system-for-all approach. In addition, none of the systems or tools that we are aware of is particularly good at supporting collaboration during the management of nuclear or radiological emergencies.

On the other hand the Internet is full nowadays of software that enables collaborative work and allows users to interact and share data (social software, groupware, instant messaging, blogs, etc.). Sometimes all such software goes by the term *Web 2.0*, a term which is "associated with web applications that facilitate interactive information sharing, interoperability, user-centered design, and collaboration on the World Wide Web" (http://en.wikipedia.org/wiki/Web_2.0). This definition mentions many useful concepts and associated with these concepts is a vast array of technologies that help software developers to bring their software closer to the actual use cases.

Our most important use case regarding nuclear or radiological emergency management is the production of timely and high quality reports of all sorts, like weather analyses, measurement reports, what-if analysis,

Reviewing Statement: This paper represents work in progress, an issue for discussion, a case study, best practice or other matters of interest and has been reviewed for clarity, relevance and significance.

recommendations, summary reports, etc. A report typically aggregates information from different sources and for us the most natural way of collecting information is to allow different experts to contribute their share. To support this and other use cases we have developed a collaborative web application. The purpose of the application is to make it easier to exchange information and to keep all users up to date with the development of the situation. The application bridges the spatial remoteness of expertise, hardware, software, and provides a consistent user interface to tools that could be used before only with special training. In addition, it provides the different users with their views of the situation and supports the process from source term input, over dispersion and dose assessments, up to the approval of various reports.

EMERGENCY MANAGEMENT IS ALL ABOUT COLLABORATION AND WHAT DOES THIS MEAN FOR SUPPORT SYSTEMS

Use cases

As already mentioned, the main purpose of the system is to support the management of nuclear or radiological emergencies. The main use cases are then the selection of the most suitable numerical weather prediction dataset, the request of dispersion assessments, the planning of protective actions, the production of timely reports, and the exchange of information. Luckily, emergencies are extremely rare in our field, and we have to maintain proficiency and remain alert by arranging exercises with simulated input data. Still other purposes are the support of case studies, training events, demonstrations, or testing.

For illustration purposes, Table 1 gives a narration of the use cases involved in the production of a situation report. In this simplified use scenario we would have several actors (management group, accident assessment group, consequence assessment group, trajectory service, dispersion service) that need to interact in order to achieve the desired goal.

1. The management group requests a situation report and specifies name, deadline, report status and type.
2. The consequence assessment group requests trajectories from FMI, creates suitable portrayals of the returned trajectories and adds a fully annotated image (incl. metadata) to the report.
3. The accident assessment group publishes their release assessment.
4. The consequence assessment group requests from FMI a dispersion forecast for the given release, performs dose calculations, creates suitable portrayals of the returned results and adds sections to the situation report. The sections contain fully annotated images of the *dispersion area*, the *plume arrival and leaving times*, and the *durations that intervention levels are exceeded*. All images are fully annotated and furnished with relevant metadata.

Table 1. Schematic use cases to produce a situation report

Aspects of collaboration

The picture given above is a rather crude description of the actual human interaction that is needed. In reality up to 100 persons can be involved in STUK alone during an emergency (or exercise) and have to coordinate their work. This is the human aspect of collaboration.

But there is a software and hardware aspect as well. Experts use all sort of tools (source term assessment codes, dispersion models, environmental transfer models, dose models, etc.) when making their assessments. Some of these tools work only on particular operation systems and on dedicated hardware. For example the long-range dispersion code runs on a supercomputer at FMI, and the dose model is a traditional single-user Windows program. Users need various communication software (e.g. FTP, email) to transfer results between tools and other users; and they need office tools, GIS, image processing software, and other productivity software for various purposes. An additional complication is the need for authorization and authentication in order to gain access to the various resources. In practice, it can be therefore quite challenging to timely produce the high-quality reports that are required.

In summary the following observations can be made:

- Emergency management is a collaborative effort. An easily accessible and constantly updated audit trail of all activities such as model requests or communications is needed.

- Modeling applications and expertise are distributed (weather forecasts and dispersion predictions by FMI, source term and dose assessments by STUK) and have to be integrated.
- Modeling applications run on different hardware and software and often have scientifically motivated user interfaces. The users, however, should not have to care about these peculiarities.
- Modeling applications often produce static images that are hardly suitable for all users in all cases.

User requirements

Use cases and observations lead us to the following user requirements (Lahtinen et al., 2008):

- Collaboratively multi-user: users work together on a single case and need remote access to resources;
- Ease-of-use: there should be only very few client side requirements; the system should not require any special or advanced computer skills; different users need different views of the system;
- Multi-purpose: emergencies, exercises, training, comparison exercises, source detection should all be supported; the system should have at least European wide coverage;
- Multi-lingual: Finnish, Swedish, and English should be supported at least as far as products and reports are concerned;
- Reliable: the system should be fit for operational use in emergency centers; this must be assured by constant quality control;
- Open: data import and export should be easy; the system should integrate well with other systems;

SYSTEM DESIGN

The KETALE project was initially launched to streamline the data and information exchange between STUK and FMI. The goal was to create an application that integrates the distributed modeling applications and facilitates collaboration and sharing of information. This has been achieved by providing functionalities for data acquisition, data management, and data visualization.

Data acquisition

The data acquisition component is the glue layer between the distributed modeling applications. It implements a simple request-response pattern where a request is issued to an application server, which returns either the calculation results or some fault condition. As most services need considerable time to respond to requests, asynchronous web service invocations are the preferred choice. Data can also be imported into KETALE by means of simple file upload.

KETALE provides the users with a consistent Web browser interface to all registered models. This bears the advantage that users can access these models without having to care about the particular hardware and software the respective application is installed. The users do not need to be trained on the use of the sometimes rather scientifically motivated user interfaces of these applications.

Whenever possible, model results are obtained in form of raw data instead of readily rendered images. Raw data can be portrayed by the KETALE users in ways that best serve their purposes. In addition, raw data can be manipulated in various different ways: it can be scaled, for example, to account for a somewhat higher release than initially thought of; doses from different pathways can be added; the plume arrival time can be calculated from the activity concentration in the air; or results can be compared to intervention levels. Finally, the output data of one model can be fed into another model to obtain additional results.

Data management

Data management is concerned with keeping account of who was doing what and when. All users are authenticated and assigned to groups. These groups are authorized then to access predefined resources. The notification page is accessible to all users, but *new, edit, delete* functions depend on group permissions.

A relational database is used to persistently store all data objects, be it metadata related to events, sites, requests, responses, and reports; or release information or any other information in the form of free text. In addition, all imported data is registered in the KETALE database. The necessary metadata is either extracted from the response, or it can be supplied manually during import. All changes to the database are seen immediately by other users.

Users are presented with a notification table (Figure 1) that shows all recent database transactions (emergency declaration, bulletins, release assessments, recommendations, issue of a request, receipt of a response, and addition of a report). By clicking on an entry they can view the notification in detail and, in the case of a model result, visualize it. Filters are provided so that users can list only entries of interest.

Data visualization

The data visualization page consists of a Web GIS component and of control widgets (Figure 1). An action widget allows the user to change style, scale, color map, and map projection of the data portrayal. Users can zoom and pan to the area of interest, and add or replace underlying map layers. The customized map can be annotated and added to a report that can be later exported as a PDF document. Selection widgets are available if the data has time and height dimensions. The evolution over time can be animated, or the respective time function pertaining at any location can be displayed just by clicking on the map.

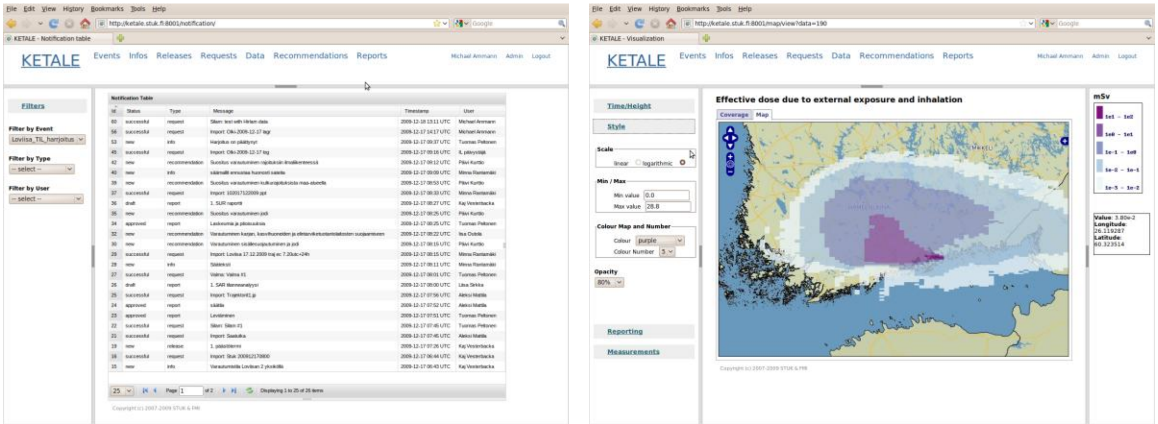

Figure 1. Notification page (left) and Visualization page (right)

SYSTEM IMPLEMENTATION

KETALE was developed using the TurboGears Web application framework. The design follows the Model-View-Controller (MVC) design pattern. This is a popular design pattern in Web applications encouraging modularity as it separates database (model), application logic (controller), and user interface (view). The user interface is expressed in a templating language (Genshi) with the addition of JavaScript (jQuery). It is dynamically populated during runtime with data from the database. Ajax technology is used to make the application more responsive by exchanging data with the Web server behind the scenes, instead of reloading an entire Web page each time a user makes a change. The data objects are linked to a relational database management system (PostgreSQL) with an object relational mapper (SQLAlchemy). This bears the advantage of being independent of a particular database system and of easier coding. Model results are mostly stored using NetCDF, which is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. The application is written in Python, a general purpose scripting language that is well suited for gluing together the different components of a system of this kind.

Maps are displayed using the OpenLayers library. OpenLayers is a JavaScript API for displaying map data in Web browsers and adds functionality for navigation and zooming. It implements industry-standard methods for geographic data access, such as the OpenGIS Consortium's Web Mapping Service (WMS) and Web Feature Service (WFS) protocols. This makes it easy to switch between different map services without demanding changes to the code. The map in the Figure 1, for example, is produced by STUK's own WMS that uses the GeoServer software. The data layer on the map is requested with a WMS-like call with extra parameters.

We want to highlight the following system properties:

- Web application: The system can be accessed with a modern Web browser over the Internet, and no client-side installation is needed.
- Database centric but vendor neutral: All information is stored in a central database; users are provided with context-specific views of the same database. The system does not depend on a particular database management system.
- Modular: The model-controller-view design pattern naturally separates the various parts of the system and enforces modularity.
- Web map visualization: Model results can be shown on top of any geographic map that is served by a Web Map Service; basic navigation tools are provided.
- Open source software: The system is build completely with freely available open source software.
- Platform independent: The system can be deployed both on Linux or Windows platforms; users access the system from a Web browser interface that has no client-side requirements.

CONCLUSION

Modeling applications often unduly impose their requirements on the users: applications have to be installed on particular hardware; users have to get acquainted with the operation system of that specific hardware and with the application logic of the model. Furthermore, they have to content themselves with predefined and often static results. KETALE tries to correct that. A modern Web browser is all that is needed to access and interact with the system. The users can access the platform independent system virtually from anywhere and they are confronted with a consistent user interface that they can use to access a multitude of application programs.

The system can be (and is typically) used simultaneously by many users with various needs. All of them are provided with their own view of the data. In addition, the system is multi-lingual. Finnish, Swedish, and English are the currently supported languages. Users can select their default languages and switch to another language whenever they want.

Finally, emergency management demands security and operational reliability. Security is provided by the secure socket layer (SSL) and, in the future, by a dedicated governmental backbone network, whereas the operational fitness is achieved by constant quality control.

The application supports the process from source term input up to the issue of reports. In doing so it preserves a complete audit trail. Data and information exchange is streamlined, transparent, traceable, and routinely tested. Reports are produced much faster than before, they are standardized and better deliberated, and data portrayals are tailored to the needs of the users. Although still developed, the application has been in use in STUK and FMI since 2008. So far we have received mainly positive feedback.

ACKNOWLEDGMENTS

There were many more people involved in the project than could be accommodated in the list of authors. We thank the project members K. Alm-Lytz, L. Pöllänen, T. Ilander, S. Kuukankorpi, H. Lemmelä, and T. Salonen from STUK and J. Vira, J. Korhonen, M. Visa, M. Aalto from FMI for their valuable contributions. We thank the members of the steering group H. Aaltonen, S. Salomaa, R. Mustonen, and H. Toivonen from STUK and M. Heikinheimo, H. Juntti, J. Korhonen, T. Tourula and K. Soini from FMI.

REFERENCES

1. Lahtinen, J., Ammann, M., Ilander, T., Kuukankorpi, S., Lemmelä, H., Peltonen, T., Rantamäki, M., Salonen, T., Sarkanen, A., Seppänen, M., Siljamo, P., Summanen, T., Vesterbacka, K. (2008) KETALE – A centralised data system for the management of dispersion and dose calculation results, *Proceedings of the International Conference on Radioecology and Environmental Radioactivity*, Bergen, Norway.