Radioprotection 2010 DOI: 10.1051/radiopro/2010024
Vol. 45, n° 5, pages S181 à S189

Article

RODOS re-engineering: aims and implementation details

I. IEVDIN[1], D. TRYBUSHNYI[1,2], M. ZHELEZNYAK[1], W. RASKOB[2]

ABSTRACT **The paper provides details on the re-engineering of the European decision support system for nuclear emergencies RODOS. Being a Java based product, the new RODOS version named JRodos introduces a cross-platform solution capable to run on most operation systems, including Windows and the main UNIX derivates. The re-engineered system preserves computational models from RODOS v6.0, adding a powerful GIS support and applying modern database technologies with flexible configuration possibilities. JRodos import functionality includes the support of EURODEP measurement files and GRIB 1 meteo data. The system operates successfully on Windows (NT based versions from Windows 2000 to Windows 7) and UNIX machines (OpenSuSe, Ubuntu, Solaris), both ×86 and ×64. Any RODOS further development will be realized in the JRodos version aiming to have this as the main operational version for the next 10 years.**

1. Introduction

One of the major objectives of the EURANOS project was to improve the RODOS (Real-time on-line decision support, Raskob and Ehrhardt, 2000) system in its operational applicability. In the first phase of the project two possibilities for the main system development has been discussed: either carry out refinements in the current version of RODOS or to consider a complete software re-engineering. The RODOS Users Group (RUG) finally concluded that the latter option is one to be realised. RUG has been playing an important role by actively participating in the development process with, among others, regular testing of intermediate software builds and suggesting particular use cases. The main work started in the middle phase of the EURANOS project and was successfully completed at the end of the project by delivering a final prototype for testing by the operational emergency centres.

2. Re-engineering: objectives and demands

The RUG reported the following problems and with the operational application of RODOS v6.0 and also defined a so called with list of potential future

[1] Ukrainian Center for Environmental and Water Projects (UCEWP), Glushkova 42, 03187 Kiev, Ukraine.
[2] Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany.

improvements (Gering *et al.*, 2006):

- The system is not easy to use (especially when deployed in emergency situations; *e.g*. too many parallel windows, no intuitive behaviour, it does not have a Windows-like look and feel, it does not completely shield the user from the underlying Unix operation system, and there are only few consistency checks on input). Considerable training is therefore needed to operate the system;
- The GIS tool lacks important functionalities (*e.g*., direct import of raster or vector map data in common formats; aggregation of results; easy configuration of layers);
- Maintenance effort is very high;
- Adaptation to national conditions is very laborious (*e.g*. use of national weather forecast data; adaptation of model parameters);
- Integration of new models is quite difficult;
- Instabilities of the whole system during runs.

As a result the RODOS system required a large effort to be operated which is an ongoing problem in the view or reduced resources in national emergency centres.

The RUG identified the following demands for the RODOS re-engineering (Gering *et al*., 2006):

- The system should be cross-platform with running on Windows, Linux, possibly MacOS;
- Preserve main computational models of RODOS v6.0;
- Use RODOS-Lite as user interface where applicable;
- The solution should use modern database engine;
- Support multi-user approach;
- It should utilize modern GIS subsystem supporting common GIS standards;
- The system should have a distributed architecture to be applicable within different environments;
- User interface should suggest convenient data representation.

3. JRODOS: implementation details

JRodos – the re-engineered RODOS system – is a Java-based product[3]. This decision has been taken as Java can be applied mostly independent from the operating system installed and contains powerful graphic user interface facilities and information management solutions. A large world-wide acting Java community provides a nearly unlimited number of useful open source libraries

[3] http://java.sun.com

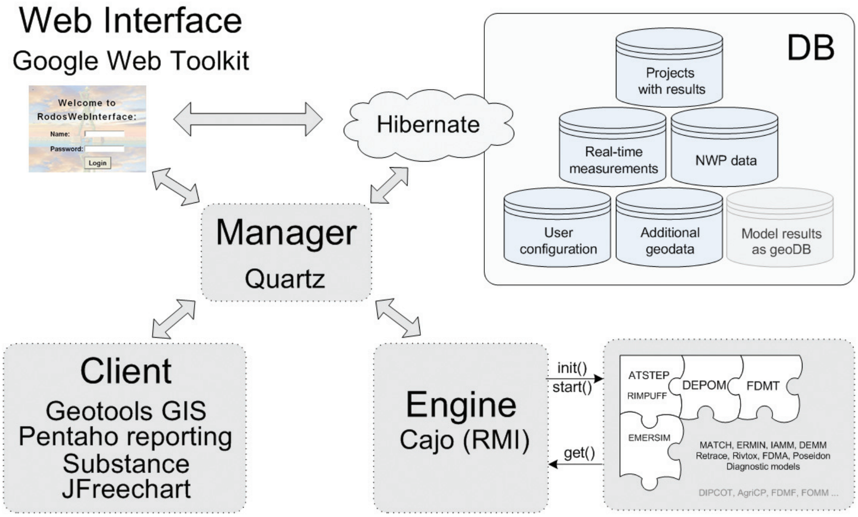


Figure 1 – ***JRodos software structure.***

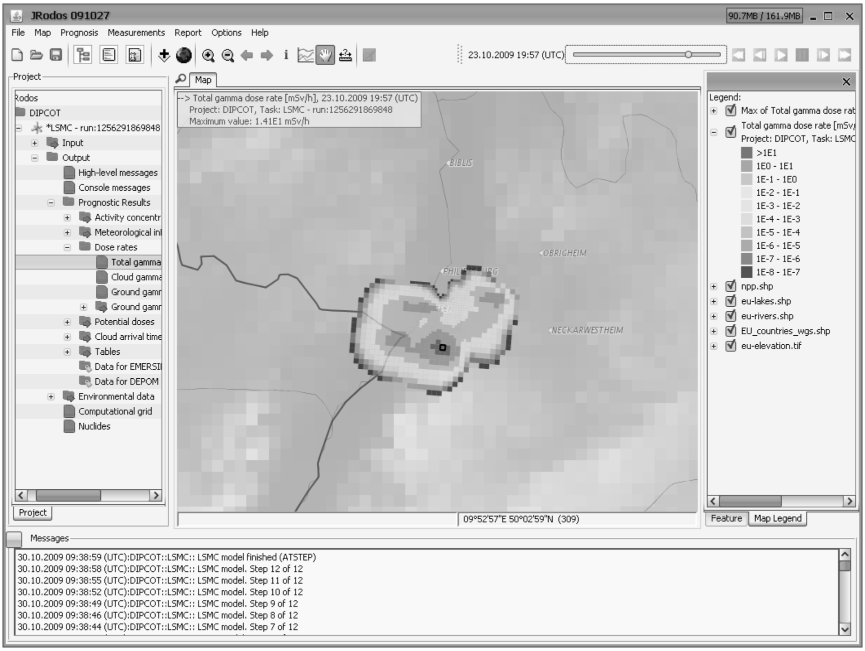
with the proper licensing (LGPL, BSD, MIT[4]) which were used intensively to speed up the development process. Additionally, the newly developed and highly appreciated RODOS-Lite user interface, a first reaction to the complaints of the users, was a Java product, and as this should be part of the re-engineered system, the Java based solution would make this integration easier.

JRodos is developed as a distributed system with the *Computational Engine* for performing calculations, the *Management Server* for database access, providing data exchange between system components, collecting information from other parts of the system and the *Client* for capturing input information, visualizing results and messaging implementation.

3.1. User interface

The JRodos user interface has a modern look and feel and is a convenient tool for the communication with the operator. It is implemented as static window frame with several components such as *Project Tree*, *Message Panel*, *Central Pane* and *Properties Panel.* The central frame is a tabbed pane which serves for data representation/visualization of various types (scalar values, maps, time plots, tables, text, histograms) and collection of user input for the simulation models. For

[4] http://www.opensource.org/licenses



```
30.10.2009 09:38:59 (UTC):DIPCOT::LSMC:: LSMC model finished (ATSTEP)
30.10.2009 09:38:58 (UTC):DIPCOT::LSMC:: LSMC model. Step 12 of 12
30.10.2009 09:38:55 (UTC):DIPCOT::LSMC:: LSMC model. Step 11 of 12
30.10.2009 09:38:52 (UTC):DIPCOT::LSMC:: LSMC model. Step 10 of 12
30.10.2009 09:38:49 (UTC):DIPCOT::LSMC:: LSMC model. Step 9 of 12
30.10.2009 09:38:46 (UTC):DIPCOT::LSMC:: LSMC model. Step 8 of 12
30.10.2009 09:38:44 (UTC):DIPCOT::LSMC:: LSMC model. Step 7 of 12
```

Figure 2 – ***JRodos user interface, presenting time dependent gamma dose rate field.***

time dependent results (numerical weather prediction data, measurements, model results) a time slider – implemented on top of the main window – is used and it is possible to create a time plot of values for a specific location with just two clicks. Maps can be saved both as picture and as geo-shape format .shp. Almost all system configurations are done via user interface, which means files should not be modified manually.

3.2. Model integration

Computational models are integrated into JRodos as plug-ins. This allows easy system expansion and facilitates independent development of new models outside the system. All model specific functionalities *e.g.* user interface, database access, model wrapper, data structure, localization resources are placed inside the corresponding plug-in. The *Management Server* searches and loads plug-ins when needed during runtime sends necessary parts of the plug-in to the *Client* (user interface, localization resources), the *Engine* (java model wrapper, computational library) or executes the code itself (database access).

Simulation models developed in Fortran/C language are realised as shared libraries (.so, .dll) leaving the original code almost untouched. Only minor modifications were performed during the implementation process only in the input and output functions and subroutines. Implementing simulation models as shared libraries and not as standalone executables, allowed using the advantages of robust data exchange concepts, introduced by the developers, and resulted in a substantially decrease of the amount of information processing compared to a file based system. To increase the overall system stability, the computation of the simulation model is carried out completely independent from the main system Java Virtual Machine (JVM), operating the core system with the user interaction. This approach protects JRodos against bugs in the models which can be memory leaks, uncaught exceptions, critical errors and unpredictable termination of the process.

Several models use output from a predecessor model as their input and in this defining so called model chains. Model chains are the base for so called JRodos projects, which contain a chain of models, selected by the user. A user interface as easy-to-use tool allows an intuitive configuration of the model chain.

At the moment the core models of RODOS v6.0 are integrated: the atmospheric dispersion models of the local scale model chain (ATSTEP, RIMPUFF, the particle model DIPCOT for complex terrain), the early phase emergency model EMERSIM, the food chain and dose model FDMT, the deposition model DepoM, the diagnostic model chain for operating the Automatic mode, the inhabited area model ERMIN, the data assimilation model IAMM as pre-processor for ERMIN, the long range atmospheric dispersion model MATCH, the hydrological model chain (ReTrace, RivTox, FDMA, Poseidon), the two data assimilation models DeMM, FoMM for deposition and food chain, respectively, and the forest model FDMF. In the near future additional model such as the countermeasure model for the terrestrial food chain AgriCP and the countermeasure model for the forest system LCMF will be implemented.

3.3. Database solutions

JRodos uses the open source PostgreSQL database engine for storing data[5]. The engine can be implemented and used on a machine physically separated from the core JRodos one, allowing several Clients to be connected to it. Thus, all the system data can be shared between different JRodos Clients.

Two databases were established to store incoming numerical weather prediction data and real-time measurement. The User can configure options related

[5] http://www.postgresql.org

to the data provider via special user interfaces. In addition, the incoming data can be visualized either as map or time plot for one location. Importing of historical data can be performed either manually or automatically once it is located in the corresponding folder of the system. As the format in the data base is specific, the data is converted to model specific formats as soon a calculation is requested.

Runs are organized as projects that include all the input data (user and data bases), messages, calculated results and are stored in the project database. The management of these projects is also implemented via user interfaces that include creating, loading, saving, copying, deleting projects and sharing with other users.

User roles and preferences are stored in the Users database. Preferences are loaded at login stage, so users have the same configuration on any clients that is connected to the same database engine.

Additionally, several models have their own database where they can save fixed model parameters, default values, options with their parameters definition, user created data set, maps and any other specific data needed for communication with the user interface or used for the calculation.

3.4. Geographical data support

To process and display geographical data, JRodos uses the Java open source *Geotools* library[6], a powerful GIS system, which supports spatial data representation and analysis as well as common data standards (.shp, .tiff) and technologies (*WMS*[7], connecting to *PostGIS*[8]). In addition it is possible to use Google Maps as a map background in JRodos.

During model calculations, the geographical data (elevation, land use, population, soil, production etc.) is interpolated to the specific computational grid of the model. A polygon interpolation implemented in JRodos provides a smoother picture than the point data evaluation utilized in RODOS v6.0. User defined data on a finer grid can be combined with default data on a coarse grid.

The PostGIS extension is used for storing large geo-spatial data sets. The map of administrative regions, urban areas, roads and railways are stored in the GIS database and can be filtered and displayed around a particular site if needed. In future, it is also planed to save the project results in GIS databases that can be opened by any other third-party GIS software.

[6] http://www.geotools.org

[7] http://en.wikipedia.org/wiki/Web_Map_Service

[8] http://postgis.refractions.net

3.5. Import / export functionality

The system supports the EURODEP format for measurements[9] as well as GRIB 1 files[10] for input of numerical weather data. Data importing can be done both manually and automatically. The export functionality allows storing calculated 2D fields in the ESRI shape .shp file format for exchange with other GIS systems.

3.6. Reporting

Results of the model calculations can be combined in so called Reports to be forwarded to the decision making team. For the generation of these reports, JRodos uses the open-source Pentaho reporting engine (formally known as JFreeReport)[11]. Generated reports can be printed, saved and stored in pdf, xls, html, rtf, text formats. The system allows manual and automatic report generation. In the manual mode, users can create reports that contain different type of data (maps, tables, texts, time plots) adjust the size, and colour of the components as well as the selected page orientation and margins. The report layout can be customized with the open source Pentaho Report Designer, which is extended with appropriate functionalities and is delivered as part of the JRodos installation. Main layout components are information about the map area, the legend, site, source term, user name and run id. If the automatic generation option is selected and configured, reports in one or all possible formats (pdf, html or rtf) will be generated after each run of a specific model chain. Configuration means here that specific results are predefined as content of the final report. Once a report has been generated it can be also accessed via the Web interface.

3.7. Web interface

Similar to the so called “B-user” of RODOS v6.0, a web interface has been implemented. It allows limited access to JRodos functionalities and the project database via a web browser. With the web interface, the user can perform the following tasks: retrieve information about existing projects and tasks, be informed on their status, share own projects with other users and retrieve results from automatically generated reports. It is also possible to request calculation services within the web browser by using the RODOS-Lite user interface. This will fill in data for two standard model chains (full interactive chain included LSMC, EmerSim, DepoM, FDMT and Automatic model chain) with the same user interface as in the full JRodos installation. After completing the input forms, an

[9] http://rem.jrc.ec.europa.eu/200.html
[10] http://www.wmo.ch/pages/prog/www/WDM/Guides/Guide-binary-2.html
[11] http://reporting.pentaho.org

xml-file is generated and submitted to a web server running as JRodos application and that will trigger calculation automatically. Information about the progress of the calculations requested is displayed in the Web interface.

3.8. Stability solutions

As part of the development process, several important steps were carried out to increase the overall system stability:

- Model calculations are performed in separate JVM processes. This protects the main system from model crashes.
- Communication between models and the system is directly performed via the memory (RMI); data exchange through the file system is highly minimized.
- Additional checks are performed before starting a calculation to assure that the input data is valid.

3.9. Future plans

The highest priority in future developing the system is to keep the operability and stability that is requested by the end user. Further extensions of the system will address the way the results will be treated So called WMS server or/and .shp files might be used to allow visualisation by third party GIS system running in an emergency centre. Additionally, the JRodos development group investigates the possibility to gain advantages of multi-core computers by introducing OpenMP parallelization/GPU acceleration[12,13] into model codes.

4. Conclusions

The re-engineered RODOS system was a request by the end users organized in the RUG. The EURANOS project took these requests as part of their working program and demonstrated with the delivery of the JRodos version for demonstration at the end of the project that this request has been answered successfully. Based on modern IT technologies such as Java and open source projects such as Geotools, the re-engineered system is now an open and easy to use toolbox that allows implementation of external simulation models and provides a similar look and feel independent if operated locally or remotely. Based on a vivid user group, future extensions will be mainly discussed in the RUG, focusing on the operational aspects of the tool.

[12] http://openmp.org/wp/

[13] http://www.pgroup.com/resources/accel.htm

Acknowledgment. *This work has received partial financial support from the European Commission Sixth Framework Programme (Nuclear Fission/Radiation Protection) under the EURANOS integrated project: European approach to nuclear and radiological emergency management and rehabilitation strategies (Contract No: FI6R-CT-2004-508843).*

REFERENCES

Gering F., Gerich B., Duranova T., Bohun L., Ammann M., Fritsch G., Hofer P., Twenhöfe C., de Ridder M. User requirements for the re-design of RODOS in phase 2 of the EURANOS-project, EURANOS(CAT2)-TN(06)-09.

Raskob W., Ehrhardt J. (2000) The RODOS System: Decision Support for Nuclear Off-site Emergency Management in Europe. 10th International Congress of the International Radiation Protection Association, CD published May 14–19, 2000. Hiroshima Japan.