

User friendly application of RODOS

T.O. MÜLLER¹, W. RASKOB¹, S. VAZIRI ELAHI¹

ABSTRACT The decision support system RODOS (Ehrhardt and Weis, 2000) contains a large set of simulation models, data bases, user interfaces and system functionalities. However these many options resulted in heavy-weight and complex user interfaces difficult to handle by non-experts. For this reason a user friendly and light-weight access to the RODOS functionalities was requested by the end users. This was fulfilled by the design and implementation of a new user interface named RODOS-Lite. The main purpose of this redesigned user interface is to gently lead operators through a sequential line of input masks, hiding rarely used parameters (defaults are set) from the operator but still providing the possibility to access them if necessary. Additionally, the interface is strongly safeguarded to prevent operators from entering wrong input values. This is achieved by systematically checking input values and even suggesting occasionally best estimate parameters. RODOS-Lite is complemented by various simple but nevertheless extremely useful features such as online access to Google maps² or source term visualisation.

1. Introduction

The decision support system RODOS is a comprehensive software tool containing many different and complex models for the simulation and evaluation of nuclear and radiological emergencies. During its lifetime it was constantly improved and enhanced by adding further requested functionalities. Consequently, the increase in functional complexity resulted in a similar increase of the complexity of the user interface. Therefore, it became difficult for non-expert operators to operate RODOS in a simple and proper manner. Furthermore, the original user interface was developed applying standards and methods that were state of the art 15 years ago, but missing now the highly desired instant recognition effect when working with up-to-date applications.

Therefore, the RODOS User Group requested the redesign of the original user interface and to implement it in a slim and light-weight fashion following the actual state-of-the-art IT-technology. As further result of reviewing the original user interface, several additional requirements have been identified:

- The most important requirement was found to be a simple and easy to use user interface in contrast to the original one. Ideally, the features of the user

¹ Forschungszentrum Karlsruhe GmbH, Institut für Kern- und Energietechnik, Eggenstein-Leopoldshafen, Germany.

² <http://maps.google.com/>

interface should be self-explaining and the intention of the input mask should be understandable even without a manual. In particular special and rarely used features should be initialised with default values and hidden in the everyday usage of RODOS. Nevertheless most of the features should be accessible if an expert actually intends to modify them.

- Entering input values during an emergency should be as fail-safe as possible. The operator should be guided through the input mask and restricted to prevent entering of invalid inputs. However an optimal balance has to be established not to dominate an expert operator more than absolutely necessary.
- The user interface should be realised as a separated application establishing an open interface to the RODOS core. Thus other applications can also use the interface to control the RODOS core.
- The redesigned user interface should be easy to transfer between different operating systems. At the time of the redesign, the implementation for HP-UX and Linux was different. Ideally the same implementation should be used on different operating systems to reduce implementation requirements and eliminate potential errors by having multiple development branches. This issue became even more relevant as the re-engineering of the RODOS system and porting to the Windows operating systems became apparent.
- Originally, the RODOS system provided an environment for several separate models that required specific knowledge to be operated. The system now has spread into the non-expert community. Therefore, the user interface should now abstract from the models, *i.e.* instead of accessing input interfaces for several models an operator should select only the results of interest as outcome without knowing which model will be operated to generate them.
- Last but not least, the redesigned user interface should be based upon well established and state-of-the-art IT-standards to assure long-term support and acceptance in the RODOS community. Also using freely available and open-source software reduces the overall costs of the redesign and its error-proneness.

Because the new user interface focused on being slim and light-weight it was named **RODOS-Lite**.

2. Under the hood: applied techniques and standards

Several state-of-the-art programming languages were considered in the redesign process. Finally, RODOS-Lite was implemented using the **Java**³ programming language. Java fulfils the platform independence requirement as it uses an

³ <http://java.sun.com>

abstraction layer from the actual hardware it is executed on. Methods to provide state-of-the-art graphical user interfaces are naturally included in the Java specification. Java is available for almost every major computer architecture and operating system especially for HP-UX, Linux and the Windows family. Also for the Java environment, a huge amount of open source tools and libraries is freely available thus significantly reducing implementation efforts when used as basis for a new application. Several wide spread libraries with large communities have been incorporated into RODOS-Lite: *e.g.* to visualise the source terms the chart library **JFreeChart**⁴ has been used. To follow the **International System of Units (SI)** (Taylor and Thompson, 2008) the **JScience**⁵ library has been integrated guaranteeing an international accepted convention to manipulate and display units. To support non-expert operators an internal help system has been set up based upon **JHelp**⁶. To quickly evaluate the location of an event access to **Google maps**⁷ respectively **Open Street map**⁸ has been integrated.

The open interface between the standalone RODOS-Lite application and the RODOS core is realised through simple file exchange. RODOS-Lite exports the values defining a RODOS run as XML file into a special incoming directory of the RODOS core. A background service regularly polls this directory and triggers a RODOS run based on the content of the found files. The structure of the XML⁹ file is well documented and can be modified even using a simple text editor. This open interface enables third party tools to access to the RODOS core components.

2.1. Addressing the fail-safe requirements:

RODOS-Lite is designed to have a consistent and correct state at any time. Almost every input value of RODOS-Lite can be checked independently from other parameters. Therefore entering values follows a simple rule: once the operator entered an input field he/she is not allowed to leave unless the entered value is valid within the context. If the operator tries to leave the input field, first of all a syntax check is performed, *i.e.* if an input field should provide a numerical value then every input that is not a number is rejected. In this case a message box is displayed informing the operator about the reason why the input is rejected. Afterwards the focus is moved back into the invalid input field. In many cases after the syntax check a semantic check is performed also, *i.e.* the entered value is rejected if it is not reasonable in some aspect *e.g.* if it is smaller then zero but

⁴ <http://www.jfree.org/jfreechart/>

⁵ <http://jscience.org>

⁶ <http://java.sun.com/javase/technologies/desktop/javahelp/>

⁷ <http://maps.google.com>

⁸ <http://www.openstreetmap.org/>

⁹ <http://java.sun.com/javase/technologies/desktop/javahelp/>

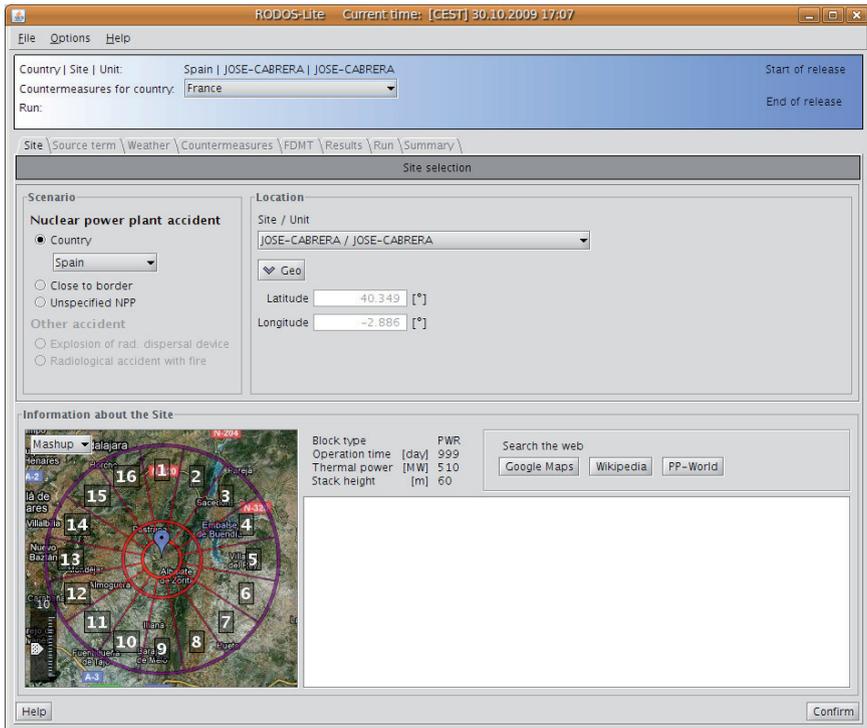


Figure 1 – *RODOS-Lite is organized in a sequential row of input masks. The image shows the very first mask which is used to define the location of an event. At the lower left a map image around the event's location is presented, which was online requested from Google maps during runtime.*

expected to be greater than zero. If all these checks are positive the operator is allowed to leave the current input field. This simple feature is highly efficient to prevent operators from entering invalid parameters and in contrast to late checks no final clean-up is required annoying the operators who believes having completed already all necessary input forms.

3. Design and realisation

Figure 1 shows a screenshot of RODOS-Lite after it has been started. The application window is split into a permanently displayed information panel at the very top (the white-blue component) and a main body that holds the input fields. The information panel is always present and displays the most relevant information of an event at a glance. To keep the user interface clear and slim the

	<input checked="" type="checkbox"/>	Kr- 85m	[Ba]	8.25E13	8.25E13	4.30E13	4.30E13
	<input checked="" type="checkbox"/>	Kr- 87	[Ba]	1.50E14	1.50E14	1.93E13	1.93E13
	<input checked="" type="checkbox"/>	Kr- 88	[Ba]	3.73E14	3.73E14	6.97E13	6.97E13
	<input checked="" type="checkbox"/>	Xe-133	[Ba]	9.05E14	9.05E14	1.73E15	1.73E15
	<input checked="" type="checkbox"/>	Xe-135	[Ba]	1.91E14	1.91E14	1.90E14	1.90E14

Figure 2 – Automatic highlighting the maximum nuclide value of a time interval. Xe-133 is the nuclide with the maximum value for all intervals. Exceeding values of other nuclides will be detected easily.

many input fields in the main body have been organised into several logical groups that are displayed only when selected. Initially these groups are presented in a sequential flow. Each group has to be processed and confirmed to allow advancing to the next group. Once confirmed groups can be accessed directly and edited using the labelled tabs located beneath the information panel.

In **Figure 1** the first group is displayed which allows to define the location of the event, *i.e.* selecting a predefined site or enter the coordinates of a mobile event. Depending on the location of the event, a small image from Google maps, respectively Open Street map, is displayed. This feature is very useful especially for events with previously unknown locations such as an explosion of a Radiological Dispersal Device (“dirty bomb attack”), because it allows to quickly check the correctness of the entered coordinates. For convenience, multiple methods to enter a location are provided like *e.g.* UTM, geo-coordinates, or even Google geo-coding where a regular postal address is transferred into geo-coordinates. **Figure 1** shows in the upper right the input mask for latitude and longitude, both disabled because the event location is a predefined site in the database. The lower right presents additional information on the site if available, *i.e.* if the system administrator has added HTML pages with additional information on the site. The three buttons in the right centre provide an online search for additional information on the site using Google maps, Wikipedia (<http://www.wikipedia.org/>) and Power Plants of the World.

Once the input of a group is finished pressing the “Confirm” button on the lower right moves the operator to the input mask to the next logical group.

Source term visualisation

The original user interface of RODOS presented source terms as matrix of values. Such a visualisation, though functional, is difficult to understand. Type errors can be introduced easily, because there is no feedback other than evaluating the entered values with the human eye. Therefore several features have been introduced to improve understanding and reduce the possibility to enter invalid values.

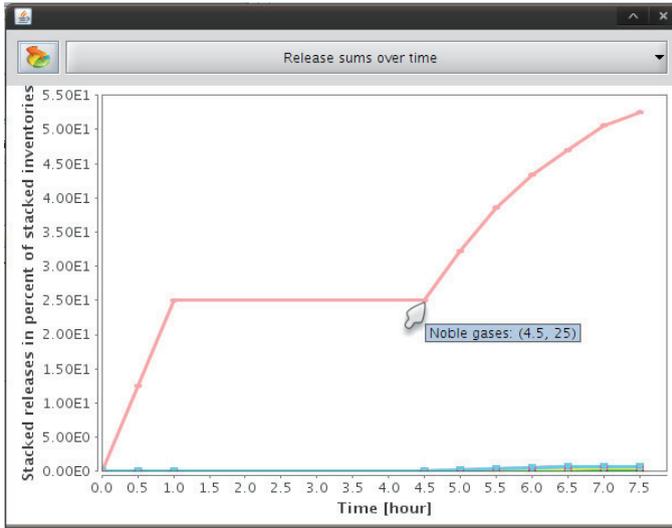


Figure 3 – Source term visualisation as sum of release groups (in percentage) over elapsed time (in hours).

In RODOS-Lite the source term is still entered as matrix of values. However each column (time interval) highlights the maximum value of all nuclides within the column (see Fig. 2). Thus type errors like *e.g.* setting the value 3.73E15 instead of 3.73E14 for Kr-88 in the first column may be detected easily if the operator expects the maximum value for the nuclide Xe-133. This feature, though simple, already helps to detect some errors.

Additionally, the matrix notation of source terms conveys no impression of the relative amounts of values and durations. Therefore, several chart visualisations of a source term have been implemented. Figure 3 shows the visualisation of a F1 source term. The different release groups are displayed in different colours. The x-axis represents the time in hours since the start of the release and the y-axis represents the summed percentage of a release group (the maximum for each release group can reach 100 percent). Obvious through this visualisation is the fact that the noble gases play the major part in this event. After 7.5 hours almost 55 percent of the total amount of noble gases has been released. Also there is a time period from 1.0 to 4.5 hours after the start of the release when nothing is released. These facts are not easily to encounter when using the matrix notation of the source term.

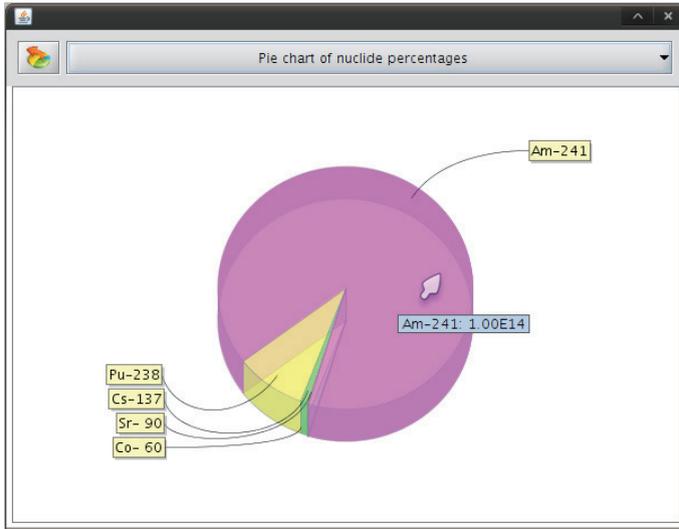


Figure 4 – Visualisation of a source term of a radiological dispersal device as a pie chart.

Figure 4 presents the visualisation of the source term of a radiological dispersal device as pie chart. This representation helps to quickly understand the fractions of the different nuclides.

Manual weather input

Figure 5 shows the input mask to manually enter weather data. This mask is special in two ways: first the value for the diffusion category is automatically selected from the context of the other input fields. This feature is very helpful for an operator of the system who is not a meteorologist, but can be overridden by experts. Secondly, the input fields of an interval form a context depending on each other. Therefore changing a single value may cause an interval to temporarily become invalid. This is displayed at the bottom of an interval by an interval validity label. A red (invalid) label reminds the operator that the interval still needs to be taken care of. Unless all intervals are valid the operator cannot leave the weather input mask. The syntax errors of input fields are immediately rejected.

4. Conclusions

The RODOS Users Group has dominated the process of improving the user input interfaces. Based on their recommendations, the formally deprecated RODOS user

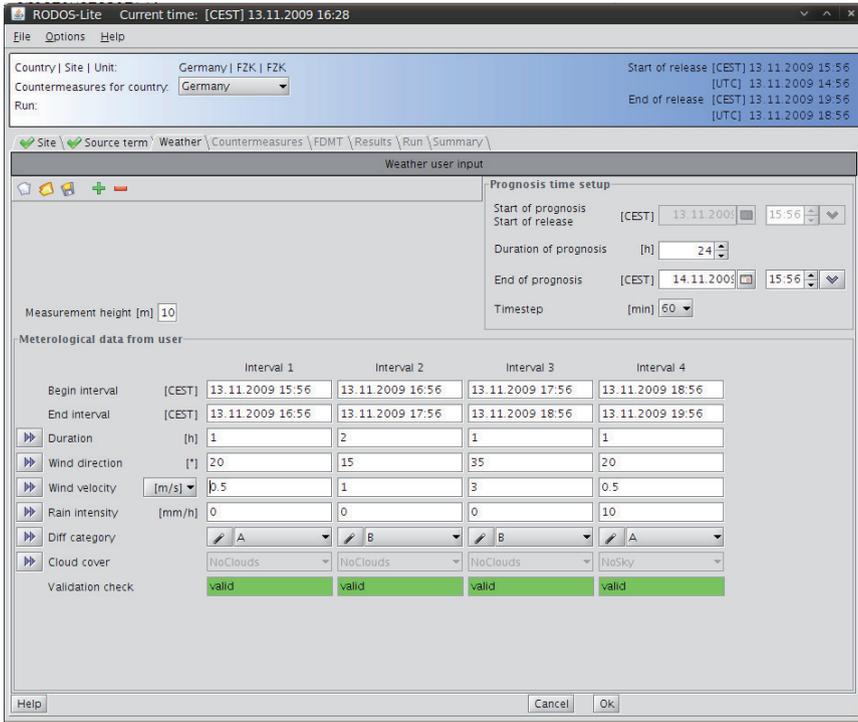


Figure 5 – *The third logical input group allows to manually define weather data. If requested the most likely diffusion category is suggested. In this special mask not all input fields can be checked for validity independently from other input fields. Therefore an interval may become temporarily invalid (visualised by the interval (in)valid label). Unless all intervals are valid it is impossible to leave this input panel.*

interface has been redesigned, re-engineered and implemented using state-of-the-art technologies. The focus was placed on user friendliness with many additional features to support especially non frequent operators. The constant check of input parameters protects the operator from entering invalid values. The sequential lead through the input masks and the simplified entering of values gently guides the operators without patronizing them. Many simple but helpful features like displaying Google maps, easy customisation of *e.g.* local languages and personalisation by using *e.g.* a different Look&Feel were highly appreciated by the end users and facilitated the overall acceptance of RODOS-Lite as the new standard user interface for RODOS.

As the implementation was performed in Java, RODOS-Lite can be transferred to all major operating systems without any changes or recompilation. The Java basis and the modular design approach allowed to integrate RODOS-Lite easily into the new Java version of Rodos (the upcoming further development of the Linux based RODOS). Furthermore, RODOS-Lite can be used as an applet in a web browser thus providing a web interface for the Java based RODOS.

The structured design and usage of standard IT-technologies like XML assure on the one hand sustainability of RODOS-Lite and on the other hand a long product life time.

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

- Ehrhardt J., Weis A. (2000) RODOS: Decision Support System for Off-site Nuclear Emergency Management in Europe. European Commission, Brussels, Report EUR 19144.
- Taylor B.N., Thompson A. (2008) The International System of Units (SI). Gaithersburg, MD: National Institute of Standards and Technology. p. 23.
<http://physics.nist.gov/Pubs/SP330/sp330.pdf>