International Congress on Environmental Modelling and Software

2nd International Congress on Environmental Modelling and Software - Osnabrück, Germany - June 2004

Jul 1st, 12:00 AM

MOIRA DSS – Architecture, Model Integration and User Interface Design

D. Hofman

MOIRA DSS – Architecture, Model Integration and User Interface Design

D. Hofman

Studsvik RadWaste AB, 611 82 Nyköping, Sweden

Abstract: The acronym MOIRA stands for “A Model-Based Computerised System For Management Support To Identify Optimal Remedial Strategies For Restoring Radionuclide Contaminated Aquatic Ecosystems And Drainage Areas” (EC projects MOIRA, COMETES, EVANET-HYDRA). MOIRA system is an integrated friendly tool for the users with different level of responsibility in decision-making and different experience in computers, environmental modelling and radioecology. With MOIRA GUI the decision-maker works with the custom things – maps, tables, graphs, reports. User can be guided through all the steps of the decision-making or can start from the direct request of the data of interest (it could be for example “ranking of the alternative strategies” for the top-level decision maker or “concentration of Sr-90 in the water” – for expert in radioecology). MOIRA system is developed using component architecture by integration of the components such as GIS (MapInfo), models (.exe, Powersim®-based, MapInfo-based) and Muli-Attribute Analysis Module (.exe) with the MOIRA Software Framework that provides kernel and GUI functionality of the system. To support quick development of the system by distributed international team same models can be used both in the frame of MOIRA and standalone. Exchanging of data sets between system kernel and the models as the text files supports this flexibility. The model’s developer decides format of the file and kernel adapts itself to it using data set description. Each model can be supplied also with the files (in the same format as input ones) containing default, minimum and maximum values for its input parameters. These files compose the contents of the Reference Data Base (RefDB) used to validate user’s input and to help inexperienced users with the default values for the unknown parameters.

Keywords: Decision support system; Model integration; Object-oriented language; User interface;

1. INTRODUCTION

The acronym MOIRA stands for “A Model-Based Computerised System For Management Support To Identify Optimal Remedial Strategies For Restoring Radionuclide Contaminated Aquatic Ecosystems And Drainage Areas” [Monte et al. 2000, Gallego et al. 2000, Monte et al. 2002b]. The decision support process with MOIRA is based [Appelgren et al. 1996] on evaluation via modelling of social, environmental and economic consequences of implementation of alternative countermeasures and ranking the effectiveness of the countermeasures using Multi-Attribute Analysis techniques. Necessary site-specific parameters for models are received from GIS-based data, database or provided by user.

MOIRA DSS contains ten predictive environmental models, dose model, economic model and Multi-Attribute Analysis (MAA) Module [Monte et al. 2000, Håkanson et al. 2000, Gallego at. al. 2002]

Efficient management of these tools requires knowledge in nature of models, their input and output parameters and file formats, knowledge in GIS and data base as well as efforts to keep results of evaluation of different countermeasure strategies consistent with the changes in input data. To make MOIRA DSS user-friendly, accessible, flexible and practical system, which can be used by users – not necessary experts in environmental modelling [Appelgren et al. 1996, Monte, Brittain 1998], the MOIRA Software Framework had been developed [Hofman 1998a, Hofman et al. 2000]. MOIRA Software Framework features have been constantly developed during preparation and issue of updated versions of MOIRA DSS [Gallego et al.2002b, Hofman moiradss.topcities.com 2003, 2004]

Due to the development of the MOIRA DSS by the distributed international team the following considerations was taking into account during MOIRA architecture design and MOIRA Software Framework design and development:

- It must be possible to simultaneously develop models, “system” part and user interface of the DSS
- It must be easy to integrate existing models into the system with minimal changes in the models code. Developers of the model should not concern about technical (programming) aspects of model integration and GUI.
- User interface of the system need to be easy updateable in response to the changes in models input and output parameters.

2. ARCHITECTURE OF THE MOIRA DSS

The component architecture where models are independent applications communicating with the system kernel by exchange of the data had been selected as most suitable for the MOIRA DSS (Fig. 1).

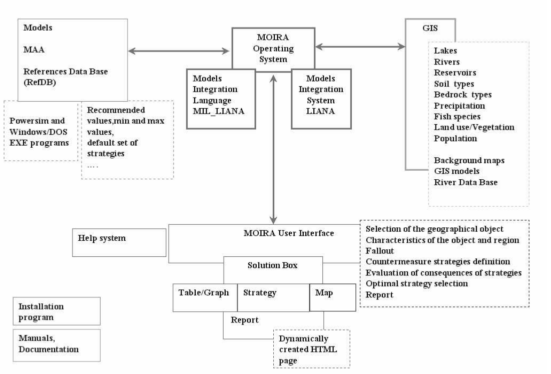

Models
MAA
References Data Base (RefDB)
Powersim and Windows/DOS EXE programs
Recommended values,min and max values, default set of strategies
....
MOIRA Operating System
Models Integration Language MIL_LIANA
Models Integration System LIANA
GIS
Lakes
Rivers
Reservoirs
Soil types
Bedrock types
Precipitation
Fish species
Land use/Vegetation
Population
Background maps
GIS models
River Data Base
Help system
MOIRA User Interface
Solution Box
Table/Graph
Strategy
Map
Report
Dynamically created HTML page
Selection of the geographical object
Characteristics of the object and region
Fallout
Countermeasure strategies definition
Evaluation of consequences of strategies
Optimal strategy selection
Report
Installation program
Manuals, Documentation

Fig. 1. Architecture of the MOIRA DSS

Similar architecture showed it reliability and usefulness in the RODOS system [Erhardt, Shershakov 1996]. There is one important difference between data exchange in RODOS and MOIRA. RODOS models are exchanging with the kernel via shared memory [Bentz et al. 2001]. This simplifies description of the I/O data for the but requires using of special tool (RODOS/2) to convert input and output files to and from shared memory for the model standalone run during the testing. MOIRA models exchange data with kernel by text files. It is possible to use exactly the same model both in the frame of MOIRA DSS and as stand-alone tool. This helps to support development, testing and update of the models with minimal interaction between developers of models and system. In the same time it requires to provide not only description of input and output data types, arrays length etc. (as in RODOS) but also description of corresponding I/O files formats.

Components of the MOIRA system can be subdivided to:

- Modelling components - models, MAA module and MOIRA GIS
- MOIRA Software Framework

Most of the models was realised in the environment of Powersim® 2.5 package [www.powersim.com]. MAA module and economic model was realised as stand-alone Windows and DOS applications respectively.

Specific modelling component is the MOIRA GIS. During the MOIRA project collection of environmental data for 30 Swedish lakes and for river Tevere (Italy) as well as European-wide population, land-use, soil types and precipitation data had been made in the environment of MapInfo GIS [www.mapinfo.com]. Several programs was developed using MapBasic® to help user in querying of available environmental data and in estimation of number of person affected by the contamination of aquatic body. Results of these models are automatically available for the MOIRA users after user’s selection of the geographical object via the MOIRA GUI.

MOIRA Software Framework consists of MOIRA Operating System and MOIRA User Interface and “covers” all the models- and system- specific functionality of the DSS giving user the possibility to work with “native” things – geographical maps, data tables, graphs, report**.** In the same time it supports easy extending of the system with the new or updated models. Detailed explanation of Software Framework architecture and functionality is given in the next chapter.

3. MOIRA SOFTWARE FRAMEWORK

3.1 Software Framework classes and objects.

MOIRA Software Framework is Windows application internally developed (using Visual

C++ language) as based on Document/View architecture (supported by Microsoft Foundation Class library (MFC)) framework of classes. The simplified schema of the key MOIRA Software Framework classes and their associations is shown on figure 2.

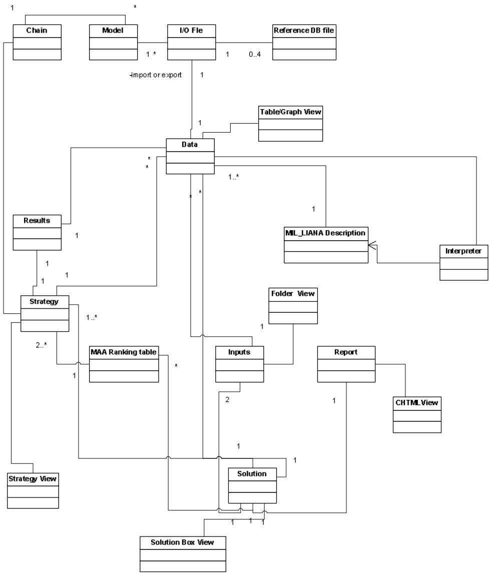


Figure 2. Simplified class diagram of MOIRA Software Framework

The objects of type Data containing values (scalar or time-dependent) for one or more model parameters or results. Content and properties of each data object are kept in persistent form as file (further in this article referred as "data set"). First time properties of particular Data object are set by MOIRA Operating System using correspondent description file written on MIL_LIANA language. Data object properties describe Data object purpose ("input", "result", "countermeasure"), type of information in the object ("scalar", "time-dependent"), templates for object information importing or exporting from/to text file and format of the table presenting object data in GUI.

Each data object is normally related to certain text file which is either directly input or output file of one of the models integrated to MOIRA or file created by the MOIRA Framework. Before run of the model content of the respective Data objects exported to its input files. When model finish execution its output files is imported in other Data objects. As. mentioned above MIL_LIANA description contains template for such exporting or importing operations.

Objects of classes Inputs, Strategy, Results, Solution are collections keeping references to name of each object, its data set and BaseTemplate id. BaseTemplate is the basic reference to the place in array containing Document/View/Frame templates and to array of icons. Object may have several "states" (for example – does not exists, exists but contain undefined values, exists and do not contain undefined values). Depending of "state" corresponding offset will be added to the BaseTemplate. Collection may contain references not only to Data object but also to objects of other classes (for example Solution contains references to all Strategy collections). A content of each type of collection is described by Framework's configuration files. Software Framework always keeps consistency between data referred in Inputs, Strategy and Results collections

User interacts with the MOIRA User Interface "browsing" Solution and "activating" data set referred in collections by clicking of mouse on its icon. After activation system will create corresponding Document/View/Frame objects using information provided by MFC CDocTemplate mechanism. For Data objects if data set does not exist it will be first created using MIL_LIANA description[1] and (in the case of activation of "results" Data object) the chain of models will be invoked (via Chain object) in order to receive object's data.

.

3.2 MOIRA Operating System.

3.2.1 LIANA Model Integration System.

LIANA Model Integration System [Hofman 1998b, Hofman 1999] is the general-purpose framework of classes and functions for the construction of "shell" for model-based DSS. Most of MOIRA Software Framework classes are either directly classes of LIANA system or inherited from classes provided by LIANA or use its low-level API functions.

LIANA Model Integration System consists of the following main parts:

- Classes and low-level functions for management of Data objects and their collections

[1] To save time system use techniques based on idea similar to just-in-time compiling. When certain MIL_LIANA description is used first time system creates on the base of it empty data set, keep it in system directory and then just make a copy of it when object data set need to created.

- Prototype classes for Model and Chain framework classes, realising the basic functionality for running of the individual models as well as for construction and running of the chain of the models
- Interpreters for LIANA and MIL_LIANA languages
- API managing interaction with the Reference Data Base

3.2.2 Integration of models with the MOIRA OS

LIANA system requirement is that model has to be Windows or DOS standalone application (.exe file, .bat file). Specially designed (.exe) application performing link with Powersim® package allow also integrating Powersim® models. MOIRA OS integrates MapBasic® models via connection with MapInfo package based on OLE Automation

Model has to receive inputs and produce outputs as one or more text files. LIANA adapts itself to formats of these files at run-time using description of each input/output file. This description has to be written on the MIL_LIANA language and contain:

- Type of the data set
- Types, names and (optionally) initial values of variables of data set
- Format of the table pesenting data set in MOIRA GUI
- Template for saving data set as the input file of the certain model; or for obtaining the data set information from the output file of the model

MIL_LIANA description can be relatively easy prepared by editing of model's input and output files. All I/O files of MOIRA models were described using MIL_LIANA language. Complete freedom in selection of file format is available with using of LIANA language (see 3.2.4)

During the integration the references to data objects related to model's input and output files must be included in the configuration files describing Framework's collections. In addition each model integrated with the MOIRA Frameworks is described in data set Times by:

- Unique Id number
- Model file (for example neweco.exe or lake1.sim)
- Simulation time-step (if applicable)
- Place in the model chain

During the simulation LIANA system prepares all the input files required for the model, starts model, checking until model will be finished and collect model output information. If model writes data about time passed in special file then this information is transferred to the user during model run-time.

Limited requirements of the MOIRA Software Framework to the model help to integrate model in the form it received from developers. For the model there is no difference either to run in the frame of MOIRA DSS or standalone. This gives the reach possibility for the models testing and development.

3.2.3 Construction and running of the chain of the models

MOIRA models running as a "chain". User has possibility to run complete chain of the models or only part of the chain (if user activates the data set related to the output results of the model situated in the "middle" of the chain).

MOIRA has two "main" chains related to the lake and river scenarios and selects one of them depending on the current scenario. As the next step the models related to the evaluation of concentration of Cs-137 and Sr-90 may be excluded from the chain if user did not provide fallout for one of these radionuclides. Model for estimation of chemical properties of lake water depending on the environmental characteristics of the region can be excluded from the chain if user directly provides the lake water properties. For Framework objects (of Chain class) chain is given as an array containing Id-s of the models on the certain positions. Excluding of the model simply mean placing of 0 instead corresponding model Id.

While present method for excluding model from the chain depending on aquatic object type and provided radionuclide fallout is suitable for the present MOIRA conditions (two types of aquatic objects and two radionuclides) the further development of the system may require automatic construction of the chain using user-defined conditions. This will be done by using LIANA language (see 3.2.4).

During the run of the chain before run of each individual model system verifies if all output data sets of the model are already present for the corresponding strategy (as described earlier due to the data consistency if some of the data in strategy or scenario have been changed then system will delete corresponding results). If all data sets are present then it is no need to run the corresponding model and system simple convert the output data set to text files and place them to the working directory.

3.2.3Reference Data Base

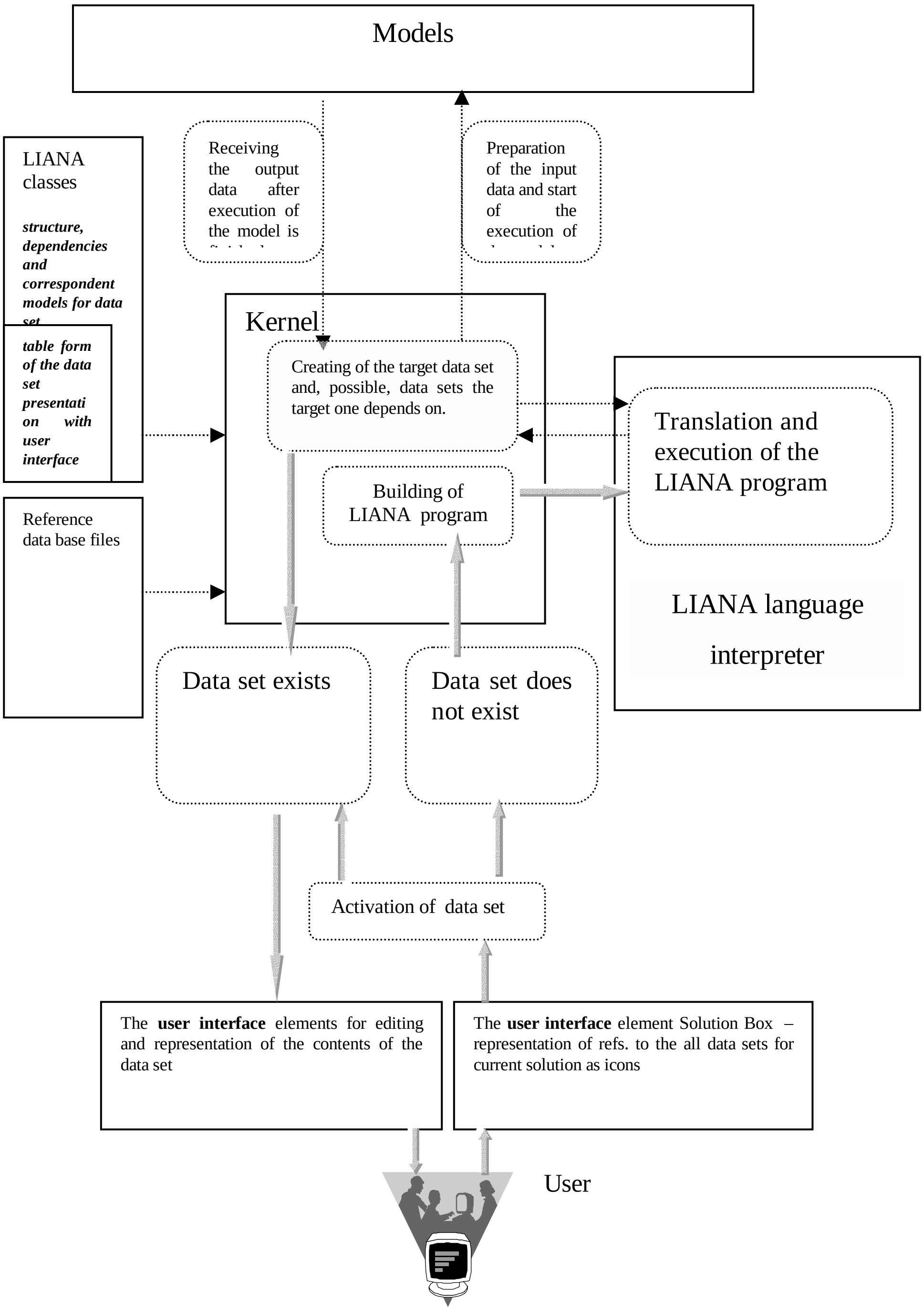
Each model integrated with the MOIRA system may be optionally supplied with

- Files with the default values for certain parameters
- Files with the min values
- Files with the max values
- Files corresponding to the "no countermeasure"

These files have the same format as corresponding input file of the model and compose the Reference Data Base (RefDB). Default data are available for the user during working with the corresponding data tables. Range values are used to check user's input.

3.2. 4 LIANA object-oriented language

LIANA object-oriented language [Hofman, 1999] is the build-in programming language used by LIANA Model Integration System. Schematic functionality of LIANA Model Integration System with using of this language is shown on Figure 3. LIANA language was used in prototype versions of MOIRA [Hofman 1998], while MIL_LIANA



Models
Receiving the output data after execution of the model is
Preparation of the input data and start of the execution of
LIANA classes
structure, dependencies and correspondent models for data set
table form of the data set presentation with user interface
Kernel
Creating of the target data set and, possible, data sets the target one depends on.
Building of LIANA program
Translation and execution of the LIANA program
LIANA language interpreter
Reference data base files
Data set exists
Data set does not exist
Activation of data set
The user interface elements for editing and representation of the contents of the data set
The user interface element Solution Box – representation of refs. to the all data sets for current solution as icons
User

Figure 3. Schematic presentation of functionality of LIANA Model Integration System

have been used in end-user versions [Gallego et. al 2002, Hofman moiradss.topcities.com 2003, 2004]. The version of MOIRA currently in preparation for release (v. 2.4) will allow using both MIL_LIANA and LIANA language.

In LIANA language types of Data objects are described using C++-like class definition, which may contain the following additional sections:

- **Produces** – class members available for access (normally read-only) from other objects. These members only will be saved in the data set.
- **Private** – class members not available for access
- **Needs** – other objects, which must be prepared before execution of the command described in Realization. Conditional creating of the certain data objects can be specified by using "if" statement.
  In order to make description simple, for each model it can be created one class with the Needs section describing model inputs. The classes describing model outputs can refer to object of this class in the Needs section.
- **Realization** – normally the command starting external model, which provides data defined in section Produces (in the form RUN ModelId, for example RUN 3). Alternatively can be a message to the user or SAVE command for the data sets created just by query and pre-processing of information from other data sets. The possibility to specify call to Framework's internal function is in plans.
- **Represented –** LIANA statements for reading of model's output or writing the model's input file. The possibility to specify call to Framework's internal function is in plans.
- **Table -** description of the format of the table presenting given object

The STORAGE, GLOBAL, LOCAL, UNIQUE keywords could be used in LIANA class declaration. The STORAGE identifies that there is a data set corresponding to the object of this class. The GLOBAL, LOCAL, UNIQUE keywords help to identify the location (from the current "solution" directory) and name of the data sets.

LIANA language can help to integrate the models with the complicated or changing structure of the I/O data. System can adapt itself to the I/O files with arbitrary structure selected by developers of models. Language gives also possibility to automate completely model chain construction. It easy to see that creating of object described by LIANA language class with the non-empty Needs section will automatically constructs the necessary chain of the models. In addition LIANA language can help users themselves write (if necessary) complex procedures manipulating with Data objects and in this way automate the system

Differences in the structure of MOIRA Software Framework classes (in comparison with Fig. 2) in the case of using LIANA language are shown on Fig. 4. There are class LianaInterpreter, link between LianaInterpreter and Chain object and possibility to recursively load and parse required LIANA class descriptions (bidirectional association between LianaInterpreter and LIANA class).

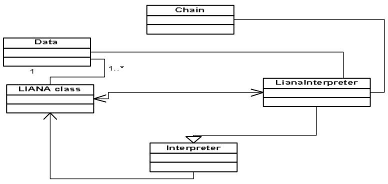
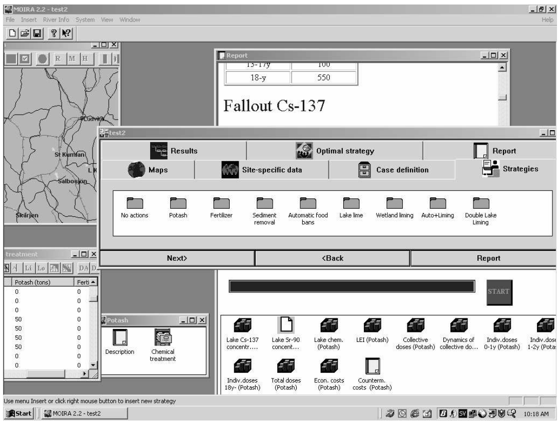


Figure 4. Changes in the MOIRA Software framework class diagram in the case of using LIANA language for the data objects description.

3.3 MOIRA User Interface

MOIRA User Interface (Fig. 5) is the part of MOIRA Software Framework and contains the classes derived from MFC CView and CMDIChildWnd related to browsing, previewing and editing of Software Framework objects as well as following of progress during model chain execution. Browsing of the data sets connected to the geographical location is provided by using of Map view available in MOIRA GUI by making OLE Automation connection with MapInfo (this technique called in MapInfo "Integrated Mapping").

The interesting feature of MOIRA GUI is that model developers indirectly participate in the design of GUI final view. After including in the MOIRA configuration files of the reference to Data object related to I/O file of certain model the corresponding icon will appear in one of the windows of MOIRA User Interface. Now the edition of input or preview of output data is available in table format through Table/Graph Tool after clicking on this icon. This gives model developers the possibility to participate in the DSS user interface construction by data separation in the different files and by description of these files with MIL_LIANA or LIANA language. Such participation helps to construct user interface quickly and utilise broad experience of the model developers in scientific subject and their experience with the communication with users of particular model.



Potash (tons)	Ferti
0	0
0	0
0	0
50	0
50	0
50	0
50	0
0	0
0	0

Use menu Insert or click right mouse button to insert new strategy

Figure 5. MOIRA Graphical User Interface

4. CONCLUSIONS

The architecture, model integration techniques and user interface design implemented in the MOIRA helped quickly create the reliable user-friendly decision support system based on the integration of the predictive environmental, dose and economic models and MAA.

MOIRA software framework could be used as the basis for the creation of the wide range of the user-friendly and easy-to-learn decision support systems.

5. SOFTWARE

The MOIRA software is working on PC under Windows 95/98/NT/2000/XP. The recent version of MOIRA software is the MOIRA 2.3.1 released in October 2003. MOIRA software and documentation web site is http://moiradss.topcities.com

6. ACKNOWLEDGEMENTS

The MOIRA system is the result of collective efforts undertaken along the MOIRA [Monte et al. 2000] and COMETES [Monte et al, .2002a] projects, as well as during ongoing EVANET-HYDRA thematic network [http://info.casaccia.enea.it/evanet-hydra].

Design and development of the MOIRA Software Framework would be impossible without its constant assessment by A. Appelgren (Studsvik, Sweden) J. Brittain (University Oslo, Norway), E. Gallego (UPM, Spain), R.Helling (NRG, the Netherlands), L.Håkansson (Uppsala University, Sweden), S. Nordlinder (Studsvik, Sweden) and L. Monte (ENEA, Italy).

L.Papush (Linköping University, Sweden) had participated on voluntary basis in the design of elements of MOIRA GUI.

7. REFERENCES

Appelgren, A., Bergström, U., Brittain, J., Gallego, E., Håkansson, L., Heing, R., Monte, L., 1996. An outline of a model-based expert system to identify optimal remedial strategies for restoring contaminated aquatic ecosystems: the project "MOIRA". ENEA Technical Report, RT/AMB/96, ISSN/1120-5556, Rome, Italy,1996

Erhardt J., Shershakov V.M. .Real-time on-line decision support systems (RODOS) for off-site emergency management following a nuclear accident. Final Report, EUR 16533, 1996

Monte, L., Brittain, J.(Eds.) Principles for the development and implementation of the MOIRA computerised system. ENEA Technical Report RT/ABM/98/4, ISSN/1120-5555, Rome,Italy, 1998

Hofman, D.,. Design and development of the MOIRA software framework. In: Principles for the development of the MOIRA computerised system (ref. Monte, Brittain 1998), pp.37-57.

Hofman, D., Usage of software system LIANA for the integration of application tasks, GIS and data bases into the decision support systems based on models. Mathematical machines and systems N1, Kiev, Ukraine, p.75-88, (on russian)., 1998

Hofman D..Software framework for integration of the models into the decision support system on ecological safety, , Ph D dissertation (in Russian) , Kiev, 1999

Gallego E., Brittain J., Håkanson L., Heling R., Hofman D. and Monte L.. MOIRA: A Computerised Decision Support System for the Restoration of Radionuclide Contaminated Freshwater Ecosystems, Proc. of IRPA 10 congress, Hiroshima, 2000 (http://www.irpa.net/irpa10/cdrom/00393.pdf)

Hofman D., Appelgren A., Bergström U., Karlsson S., Mathiasson L., Nordlinder S. The MOIRA software framework, geographical information system and integration of models. In MOIRA Final Report (ref. Monte 2000)

Håkanson L., Gallego E., Ríos-Insúa S. "The application of the lake ecosystem index in multi-attribute decision analysis in radioecology", pp. 319-344, Journal of Environmental Radioactivity, 49-3, 2000

Monte L., Van der Steen, J., Bergström, U., Gallego, E., Håkanson, L., Brittain, J. The project MOIRA: a model-based computerised system for management support to identify optimal remedial strategies for restoring radionuclide contaminated aquatic ecosystems and drainage areas. Final report. ENEA Report. RT/AMB/2000/13, 2000

Bentz G., Lorentz A., Rafat M., O. Schüle. Integration of external programs in RODOS, RODOS-TN(1)9601, 2001

Monte L. , Brittain J., Håkansson L., Gallego E., Zheleznyak M., Voitsehovich O., Kryshev I., Marinov Petrov K. . "Implementing Computerised Methodologies to Evaluate the Effectiveness of Countermeasures for Restoring Radionuclide Contaminated Fresh Water Ecosystems. COMETES Project. Final Report". Ente per le Nuove Tecnologie, l'Energia e l'Ambiente (ENEA), RT/AMB/2001/28, Roma (Italia), 2002

Monte, L., Van der Steen, J., Bergström, U., Gallego, E., Håkanson, L., Brittain, J.. A model-based computerised system for management support to identify optimal remedial strategies for restoring radionuclide contaminated aquatic ecosystems and drainage areas (MOIRA). In: Schulte, E-H, Kelly, G.N. & Jackson C.A. (Eds), Decision support for emergency management and environmental restoration, EURATOM, EUR 19793, Belgium, pp. 26-27, 2002

Gallego E., Jiménez A., Mateos A., Sazykina T., Rios-Insua S., Windengård M., Application of Multiattribute analysis (MAA) methodologies to the evaluation of the effectiveness of remedial strategies with the Moira system, pp. 11-120 in COMETES Final Report (Ref. Monte et al. 2002a)

Gallego E., Jiménez A., Hofman D. , The version 2 of MOIRA system. A direct result of COMETES project. pp. 179 – 184 in COMETES Final Report (Ref. Monte et al. 2002a)