

# Towards automation of model execution from a decision support environment

**R. Denzer**<sup>1</sup>, **S. Schlobinski**<sup>2</sup>, **T. Hell**<sup>2</sup>, **R. Güttler**<sup>1</sup> and **L. Gidhagen**<sup>3</sup>

*Environmental Informatics Group, Goebenstr. 40, 66117 Saarbrücken, Germany*

*cismet GmbH, Altenkesseler Str. 17 D2, 66115 Saarbrücken, Germany,*

*Swedish Meteorological and Hydrological Institute, Folkborgsvägen 1, 60176 Norrköping, Sweden*

*(Email: [ralf.denzer@enviromatics.org](mailto:ralf.denzer@enviromatics.org))*

**Abstract:** Environmental decision support systems (EDSS) are often constituted by a combination of several heterogeneous tools for data and information management, visualization, modelling and simulation, data analysis, reporting and so forth. EDSS need to be tailored towards specific decisions by human decisions makers and often require highly interactive components based on good human factors engineering. They are often expensive to build.

This paper presents early results of project SUDPLAN, an EU funded R&D project in the 7<sup>th</sup> EU Framework Program. SUDPLAN pursues several objectives. From an environmental management perspective, the project aims at combining (larger scale) regional climate models (RCMs) with (smaller scale) local models in order to support urban management decisions related to climate change. From an end user perspective, SUDPLAN aims at supporting urban management decisions with an easy to use Scenario Management System (SMS), which supports end users in the assessment of decision alternatives. This end user support shall in particular include the automation of recurring tasks. From an informatics perspective, the R&D in SUDPLAN aims at building a generalized framework to build EDSS which make heavy use of models and interactive visualization (part of this, not discussed in this paper, is the extension of geo-spatial towards 3D interaction). This paper focuses on the latter perspective.

Over recent year, generic approaches in environmental software systems have concentrated on generic ways of integrating data, and have lead to *data infrastructures*. Often the term *service infrastructure* is used in the literature but the majority of *generalized* or *standardized* service types are at face value data access services. However, at least the data integration part of EDSS has become much more efficient than it used to be. For highly interactive end user applications including use of models, the actual design and implementation of the graphical user interface (GUI) is still a heavy engineering and development task. This is due to the fact that a) end user applications (including the GUI) are still often developed from scratch, without systematic approach and b) the integration of models into workflows is done in a similarly unsystematic fashion. The SUDPLAN SMS aims at overcoming these software engineering deficiencies.

The combination of two aspects constitutes the core of the approach. First, a generalized data model to cope with managing models in the application is used to support the integration of models with the EDSS. This approach is not new and has been proposed by other researchers before. Secondly, an interactive geo-spatial development environment (named *cids*) is used, which allows to incorporate specialized application themes into a general framework which supports a wide range of applications. This development environment is “receptive” in the sense that it does not only generalize data integration but also the integration of GUI and visualization elements which interact with the underlying information management platform. This is done through a number of software patterns. In the SUDPLAN SMS both approaches are combined and extended towards models. This means that to integrate a model into an EDSS workflow means to instantiate a number of model managers, which support the interaction with the models, and to implement a number of renderers and editors which constitute the end user’s GUI.

This paper suggests that a *combined systematic approach* to both model integration and interactivity engineering leads to better engineered software and reduction of development efforts.

**Keywords:** *Environmental decision support systems, model integration, climate change, urban management*

## 1. INTRODUCTION

### 1.1. Model Integration in Environmental Decision Support Systems

Numerical models are essential components of Environmental Decision Support Systems (EDSS). Models appear in many EDSS as part of the decision strategy and offer users assessments of consequences related to decisions taken. Therefore it is not a miracle that the literature offers a wealth of papers concerned with integration of models into EDSS. Many papers describe how models have been integrated into EDSS in a concrete case. An early example can be found in (Leon, 2000). Over the years researchers have come up with more general concepts for and classifications of model integration (Argent, 2004) and the idea of a generalized ways to handle models also penetrates from more general publications (Deokar, 2011) into the environmental domain. Innovative technologies like semantics are also considered to support integration of models (El Gayar, 2009).

Another flavour of the EDSS problem domain is concerned with general integration issues related to environmental data, information, knowledge and services. There has been a flood of projects and related publications which deal with environmental / spatial data and service infrastructures. This work shares the common goal to provide an underlying ICT capacity which can support environmental information systems (EIS) and EDSS, and cross-cutting initiatives intend to provide basic services on a common and regular basis (Hrebicek, 2011). Many projects have produced valuable results in terms of concepts, specifications and implementations of architectures for environmental information and service infrastructures, see for instance (Usländer, 2010). Their main goal is to make basic infrastructures *generic*, i.e. fit for very general, application-independent and application-cross-cutting use (Denzer, 2002).

### 1.2. The SUDPLAN Scenario Management System

SUDPLAN (Sustainable Urban Development Planner) is a project within the 7<sup>th</sup> European Framework program. SUDPLAN aims at supporting city managers and urban planners in their decisions related to the anticipated changes of the earth's climate (Gidhagen 2010, Denzer 2011). This involves heavy use of models from the continental down to the local scale. The SUDPLAN Scenario Management System (SMS) is intended to be a highly interactive application which allows users to play "games" with many alternative decisions, which allows them to assess the different consequences of such decisions. The approach is validated in several application areas such as urban air quality, drainage and waste water (Engart 2011, Sander 2011).

As part of the work carried out in SUDPLAN, it is essential that models are integrated into the SMS, their executions are carried out and results of alternatives are shown to end users. This, as such, is nothing uncommon.

### 1.3. Proposition of this paper

One of many definitions of EDSS states: 'An Environmental Decision Support System is an information system containing at least one component whose purpose is to support human decision making about an environmental issue' (Swayne, 2000). This definition enforces the idea that the human and the decision are in the centre, and that the software system is the supporting tool. Therefore interactive graphical user interfaces (GUI's) and visualization play an important role. When considering human factor of EDSS any generic approach towards EDSS appears to be in contradiction with generic approaches, as expressed in (Frysiner, 2003): "... EDSS will only help the decision maker if it is useable by them. It must provide everything they need – but *only* what they need. It must represent the problem and its elements in terms with which they are familiar, and offer them choices which they are equipped to make. This suggests the futility of a generic EDSS". From an *end users point of view*, we agree with this statement, as it is hardly to imagine that one EDSS tool can cope with the tremendous variety of environmental decision making. However, from a *software engineering point of view* generalized approaches to EDSS are possible.

Traditionally such generalized approaches are data centered and have mostly been concerned with generic data and service infrastructures, like those mentioned in section 1.1. This paper – however – demonstrates that a generic approach can be pushed "further up" towards the end user and into the GUI of an EDSS, and *that this can also be done with numerical models which need to be integrated into an interactive work flow*. On top of this, *automation of repeated model parameterization and execution* can be integrated into a DSS work bench in a fashion which is *independent of the model(s) involved*. This is done by a) separating model-independent from model-dependent functionalities, b) use of customization patterns for model specifics, and c) using an appropriate platform including an interactive tool environment which allows to plug-in such functionality in an easy way.

## 2. MODEL INTEGRATION CONCEPT

### 2.1. Levels of model integration

Models are made part of an EDSS in several ways which may vary depending on the level of interconnection, or in other words the degree of separation of model logic and EDSS application logic. Often models are the integral core of an EDSS. In this case the EDSS is built around a particular model or set of models. If an EDSS is implemented using this approach then it cannot be easily adapted (e.g. to a different location) without the need to change the entire software. This impedes the easy re-use of model based EDSS for very similar but slightly different problems.

The situation gets even worse if this tight relationship between model logic and EDSS application logic is implemented relying on specific properties of specific software tools, in particular if these tools are proprietary.

Obviously there are other ways to integrate models in EDSS which enhance the re-usability and adaptability and thus sustainability of EDSS solutions considerably. When we look at existing EDSS we can find different levels of model integration. Argent, for instance (Argent, 2004) classifies models in a way which one could call a 'life cycle', in which they move away from the original developer towards use in policy analysis.

The levels of model integration can also be defined according to the level of separation of concerns which is one of the key concepts of scientific thought (Dijkstra, 1982). The criteria for separation are in our case the affiliation of the EDSS application logic with the model logic. The assumption is that the more model implementation and EDSS implementation are separated, the better the resulting solution will be regarding aspects of re-usability and adaptability.

The following four cases help to understand this proposition:

1. Monolithic integration

Models are integral part of the EDSS application logic (model code is included in the application code, 1 runtime binary).

2. Separated integration

Model logic is isolated from EDSS application logic (2 binaries), both parts communicate via proprietary interface, as defined by the model logic implementation.

3. Well defined integration

Like in 2, but the interface is well defined in the sense that the interface definition is independent of the model instance; this is key to interchangeability of the model implementation and means that a change in the model will cause little or no effect in the EDSS application logic.

4. Standardized integration

Like in 3, but the interface is standardized and its specification is published, which fosters the re-use of models on a larger scale.

Following either approach 3 or 4, well defined models can be handled (parameterized, executed, results analyzed, results visualized, exported and so forth) from a more abstract viewpoint. This allows for the separation of *model-generic functionality of an EDSS workflow* and *model-specific functionality of an EDSS workflow*. The *generic functionality can then be implemented for a large class of models*, and the *specific functionality can be embedded into an application using well defined (and re-usable) software integration patterns*. If this separation is successfully carried out, the concept will be particularly useful for the automation of recurring tasks, like the execution and analysis of models in ensemble mode.

### 2.2. Generic parts of EDSS application logic

The borderline between generic and model specific parts of the EDSS application logic can be defined by separating those parts of the application which need to rely on the model logic, and those parts which are independent of the model logic. We call the set of common functionality "Generic EDSS Application Logic". An EDSS environment which can be used as the software platform for a large variety of concrete EDSS would provide as much as possible of generic logic, and would at the same time provide concepts

and tools to implement individual, model-specific logic. The SUDPLAN SMS platform development is an example of how this can be achieved.

As part of the scoping of the project, an analysis of requirements for different user groups and applications in the project was carried out, and documented in a requirements specification (SUDPLAN, 2011), a summary is published in (Schlobinski, 2011). Similarly to (Argent, 2004), different user groups were considered, as they have different needs of an EDSS.

Model-Independent Application Logic	Model-Specific Application Logic
Model integration capabilities	Can become arbitrarily complex, will usually include:  Parameterization  Model-specific end user GUI components  Model-specific geo-spatial functionality and visualization  <u>and</u>  Will use model-independent functionality
Model input-/output management	
Data management including management of simulation results	
Visualization	
Geo-spatial functionality	
Search / discovery / navigation	
Management of information products	
System manager functionality	
Automation functionality	

**Table 1.** Generic and specific functionalities.

It is important to note that automation tasks, for instance running the same model over different parameters, different models over the same parameters, running whole ensembles and so forth *becomes a generic property, once we have established a general software pattern how to embed model-specific functionality*. The main value proposition of the SUDPLAN SMS is that: a) end users of all groups can be heavily supported by providing them with an automation environment, and that b) the actual software engineering costs for an individual EDSS can be reduced considerably once the generic parts are established and standard software patterns are applied to the specific parts.

### 3. INFORMATION MODEL FOR HANDLING MODELS

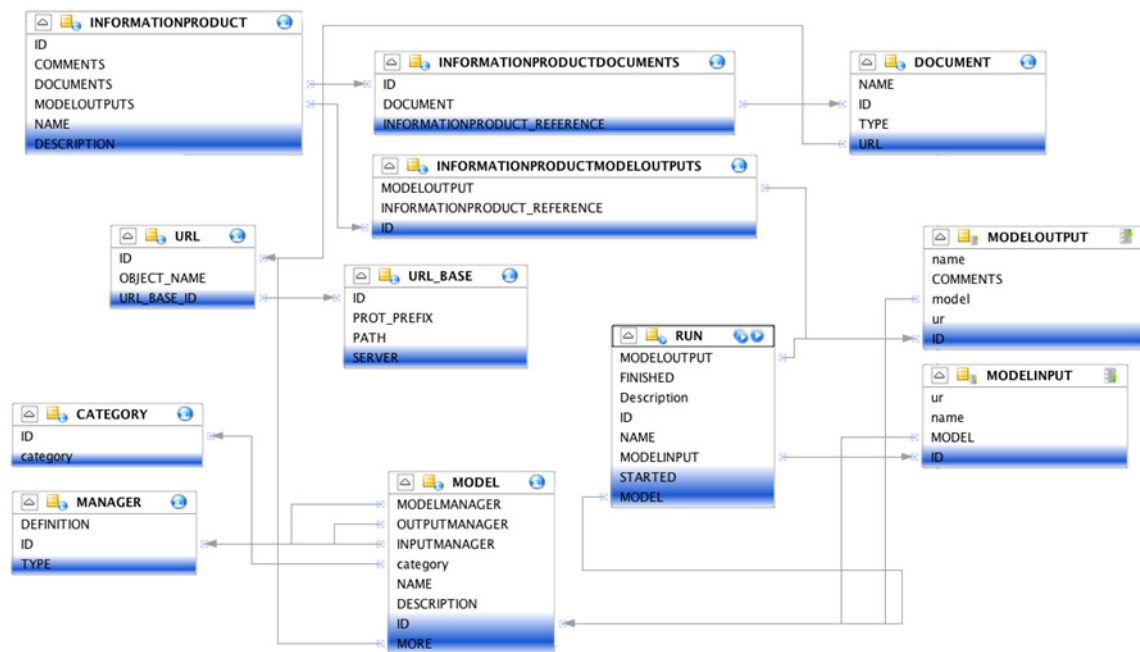
Figure 1 shows the core part of the information model used in the SUDPLAN SMS. It is flexible and adaptable such that it supports arbitrary computational processes can be embedded into the interactive environment.

The Model Entity defines the general properties of a model, associates the model to several Manager Entities which identify a Manager Component (model-specific implementations) which deals with the specifics of a particular model instance. Figure 1 shows that there is a *ModelManager*, a *ModelInputManager* and a *ModelOutputManager*.

The *Run* entity relates information for execution of models, associated through *ModelInput* and *ModelOutput* entities with the I/O data of the model. The *ModelInput* entity identifies the model input data resource, the *ModelOutput* entity identifies the model output data resource.

The concept used is called *model manager*. A manager component is designed as a (software) component which hides the complexity of certain characteristics of a complex process and provides a well defined interface to the EDSS application logic. The manager in our approach has three specializations. The *ModelInputManager* is a piece of software equipped with knowledge about the inner workings of a particular model and its input requirements regarding data, parameterization, formats etc. *Moreover, it provides GUI elements that support the user interaction with the respective model*. It collects the information that is necessary to start the model. It comes with the functionality to persist the complete set of information necessary to reproduce a particular model run. The *ModelOutputManager* is a piece of software equipped with knowledge about the knowledge about the inner workings of a particular model and its output. It processes the data produced by the model and presents the results tailored to the DSS end user requirements this process may be asynchronous (ad hoc or in advance). The *OutputManager* comes with the

functionality to persist the complete set of information of a specific model result. The *ModelExecutionManager* is a piece of software that handles an arbitrary complex model execution workflow. It interacts with the *InputManager* and *OutputManager* which handle model input and output specific processing.



**Figure 1.** Information model for model integration.

The information contained in this information model is enough to perform model information management e.g. the integration of model results or the execution of models in a specific EDSS workflow not considering aspects of specialized visualization, as the manager component contains the entire knowledge necessary to deal with a particular aspect of the model logic.

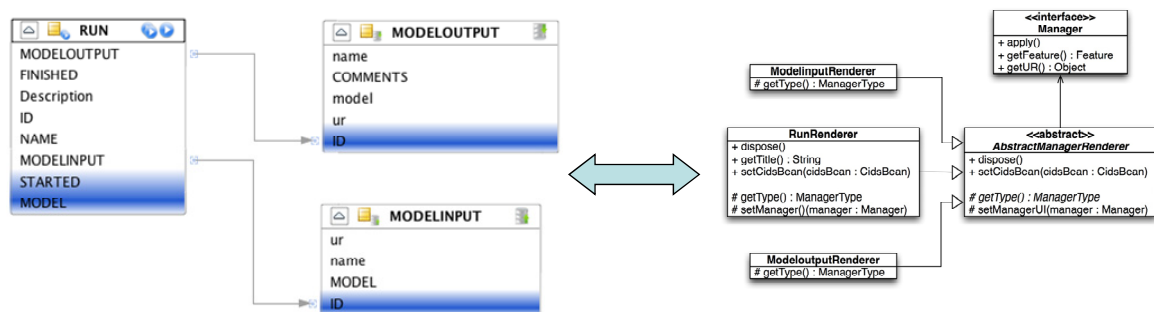
#### 4. EXTENDING THE CONCEPT TOWARDS THE END USER GUI

The approach described in the previous section is nothing new, and has been proposed before, for instance in (El Gayar, 2009). In the SUDPLAN SMS, we extend this approach towards the end user GUI. This is only feasibly by using a platform which provides a framework to implement the concepts easily.

The SUDPLAN SMS is based on the *cids* geo-spatial application and service platform. This platform has been used to build EIS and EDSS since 1999 and has supported environmental applications in various aspects of city management, water resource management and others. Most recently it has been prototypically extended towards model integration in an application for the oil industry (Denzer, 2011a). The most important characteristic of *cids* is that while at the surface it often *looks* similar for different applications, it extensively supports adaptation of the end user GUI towards specific application needs.

This is achieved through several mechanisms, the most important of them being: a) a completely generic information model which supports the definition of so-called *application themes* (they are in essence the baseline to plug-in whole applications, with their information management and work flow, into the platform); b) a catalogue which is associated with a so-called *navigator* as a standard interaction paradigm; c) an interactive geographic component called *cismap*, and d) theme-specific (i.e. application-specific) *renderers* and *editors* which support the interaction of end users with a theme (an application).

In SUDPLAN this concept has been extended to support the integration of models. The work presented in this paper is a first version of this generalization.



**Figure 2.** Model management and associated user interface interaction patterns.

What happens in practice is that for each of the given entities in figure 1, editing and viewing (*rendering*) patterns are defined. This is a concept which has been successfully used for information (data) management in *cids* applications for many years, and which considerably reduces development costs for an individual system. The model management patterns are just an extension of this general idea.

For instance, an individual model run and its associated inputs and outputs are reflected in an interaction pattern which is used to “plug-in” the required GUI elements into the interactive user interface. Technically this is done using customized GUI components that are provided by the manager implementations which are plugged into the general GUI.

## 5. STATUS OF WORK AND CONCLUSION

The approach presented in this paper is currently validated in several real world applications (Gidhagen, 2010). This validation is part of the further work in the project. So far the work concentrates on the individual model(s) execution from an EDSS application and how to embed the model into the SMS work flow. Results so far suggest that the software engineering of an EDSS can be generalized *beyond* generic concepts of data infrastructures without losing the necessary individuality of specific EDSS’s end user work flows. The work also shows that that it is more efficient to develop an EDSS on the basis of an EDSS environment, as this environment provides sophisticated features which would be very expensive to implement from scratch in every case.

Several more steps are logical from this point on. First, it would be advantageous if interfaces to call models were well standardized. We see considerable gaps in this area. Secondly, it would be straight forward to define what SUDPLAN calls a *scenario* (i.e. a collection of model(s), parameters and so forth to create one information product) in an abstract way and to standardize this definition. This would allow us to exchange the whole workflow behind information products based on models, and such products could be executed in a service environment as well. Finally, modelling environments and work benches could support tools and interfaces which generate model adapters, as already pointed out in (Argent, 2007), which could be plugged into an SMS, potentially even at runtime. These and similar advances in EDSS software engineering can contribute to considerably cut cost for the development of concrete systems.

## ACKNOWLEDGEMENTS

SUDPLAN is a Collaborative Project (contract number 247708) co-funded by the Information Society and Media DG of the European Commission within the RTD activities of the Thematic Priority Information Society Technologies.

## REFERENCES

- Argent, R. M. (2004). An overview of model integration for environmental applications – components, frameworks and semantics, *Environmental Modelling & Software*, 19, 3, pp. 219–234.
- Argent R.M. et al. (2007). Modelling Environments and Use of service Infrastructures, In: *Proceedings of MODSIM 2007, International Congress on Modelling and Simulation*, Modelling and Simulation Society of Australia and New Zealand, December 2007, pp. 833—838.
- Deokar A.V., El-Gayar O.F (2010). Decision-enabled dynamic process management for networked enterprises, *Springer Science+Business Media, LLC 2010, Inf Syst Front*, DOI 10.1007/s10796-010-9243-3

- Denzer R. (2002). Generic integration of environmental decision support systems – state-of-the-art, *Environmental Modelling & Software, Volume 20, Issue 10*, pp. 1217–1223.
- Denzer, R., Schlobinski S., Gidhagen L. (2011). A Decision Support System for Urban Climate Change Adaptation, In: *Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS-44)*, CDROM, IEEE Computer Society.
- Denzer R. et al. (2011a). An Environmental Decision Support System for Water Issues in the Oil Industry, In: *Environmental Software Systems Vol. 9, Frameworks of eEnvironment, IFIP AICT 359*, Springer, pp. 208—216
- Dijkstra, E. W. (1982). On the role of scientific thought. In: *Dijkstra, E. W. Selected writings on Computing: A Personal Perspective*. Springer New York, ISBN 0-387-90652-5, 60–66.
- El-Gayar, O. F., Deokar, A. V. (2009). An ontology-based model management architecture for service innovation. Workshop on Network based Decision Support and Modelling for Service Science Management and Engineering SSME. IFIP Working Group 7.6, Lecture Notes in Business Information Processing, Springer.
- Engardt M. et al. (2011). Web Services for Incorporation of Air Quality and Climate Change in Long-Term Urban Planning for Europe, In: *Environmental Software Systems Vol. 9, Frameworks of eEnvironment, IFIP AICT 359*, Springer, pp. 558—565
- Fryssinger, S. (2003). Human Factors of Environmental Decision Support Systems. In: *Environmental Software Systems Vol. 5, Environmental Knowledge and Information Systems*. IFIP, pp. 318-327.
- Gidhagen L. et al. (2010). Sustainable Urban Development Planner for Climate Change Adaptation (SUDPLAN), In: *Proceedings of ENVIP'2010 workshop at EnviroInfo2010, "Environmental Information Systems and Services - Infrastructures and Platforms"*, CEUR-WS, Vol-679, ISSN 1613-0073, urn:nbn:de:0074-679-9.
- Hrebicek J., Pillmann W. (2011). eEnvironment: Reality and Challenges for eEnvironment Implementation in Europe In: *Environmental Software Systems Vol. 9, Frameworks of eEnvironment, IFIP AICT 359*, Springer, pp. 1-14.
- Leon, L. et al. (2000). Water quality model integration in a decision support system. In: *Environmental Software Systems Vol. 3, Environmental Information and Decision Support*. Kluwer Academic Publishers, pp. 187–194.
- Schlobinski, S. et al. (2011). Vision and Requirements of Scenario-Driven Environmental Decision Support Systems Supporting Automation for End Users. In: *Environmental Software Systems Vol. 9, Frameworks of eEnvironment, IFIP AICT 359*, Springer, 51—63.
- SUDPLAN (2011). Sustainable Urban Development Planner for Climate Change Adaptation - Requirements Specification, *project deliverable, a public version will be available in 2011 at <http://www.smhi.se/sudplan/Results>*.
- Sander S. et al. (2011). Integrating Climate Change in the Urban Planning Process – A Case Study, In: *Environmental Software Systems Vol. 9, Frameworks of eEnvironment, IFIP AICT 359*, Springer, pp. 631—640.
- Swayne D. et. al. (2000). Environmental Decision Support Systems – Exactly what are they?, In: *Environmental Software Systems Vol. 3 – Environmental Information and Decision Support*, Kluwer Academic Publishers, pp. 257—268.
- Usländer T., Denzer R. (2010). Requirements and Open Architecture of Environmental Risk Management Information Systems, in: *B. van der Walle (ed.), Information Systems for Emergency Management, M. E. Sharpe Publishers*, ISBN 978-0-7656-2134-4, pp. 344—368.