iEMSs 2008: International Congress on Environmental Modelling and Software
Integrating Sciences and Information Technology for Environmental Assessment and Decision Making
4th Biennial Meeting of iEMSs, http://www.iemss.org/iemss2008/index.php?n=Main.Proceedings
M. Sànchez-Marrè, J. Béjar, J. Comas, A. Rizzoli and G. Guariso (Eds.)
International Environmental Modelling and Software Society (iEMSs), 2008

Representing river system behaviour in UML to enhance development of DSS

J. A. E. B. Janssen[a] **and Ioannis M. Dokas**[b]

[a] *Water Engineering and Management Group, University of Twente Enschede, The Netherlands, j.a.e.b.janssen@utwente.nl*

[b] *Cork Constraint Computation Centre, Department of Computer Science, University College Cork, Ireland, i.dokas@4c.ucc.ie, jdokas@yahoo.gr*

Abstract: Water management decisions often have far reaching consequences for the every day life of the inhabitants of a river basin. To take these consequences into account during decision making processes, we aim to develop a software tool that can indicate the expected effects of different management strategies on a number of river functions. This requires knowledge domain representation that fits the perception of stakeholders, and at the same time provides a starting point for modelling. We found that UML provides a language that on the one hand facilitates the communication and is intuitively understood, and on the other hand provides a high level of consistency as well as decomposition into elements that can easily be modelled. Moreover, UML provides the space required to deal with both quantitative and qualitative information, which is desirable to incorporate knowledge of different scientific fields / disciplines.

Keywords: River basin management, Maas river, UML, DSS.

1. INTRODUCTION

The development of software models for environmental management is to a large extent an interdisciplinary activity. During the development of the model, many persons with different backgrounds are involved. These include for instance scientific experts in different fields, policy makers at different administrative levels, inhabitants, software developers, consultants, and so on. Therefore the successful development of these models depends to a large extent on communication [De Kok et al., 2006]. The general approach is the use of some kind of common modelling language, such as a verbal or graphical description of model components. This language is then used to construct a model of reality, which is understood or not by the receiver, depending on whether he or she is familiar with the language, and depending on the clarity and specificity of the syntax (rules, 'grammar') and semantics (meanings) used in the language. Different languages may be required during the different stages in the design process of tools for environmental management.
Here we focus on the development of a conceptual model for the exploration of strategic river management in the Dutch Maas River. The research is in a phase where the relevant actors are known, the indicators have been selected, and management measures have been proposed. Now the conceptual model ought to provide a framework for the software implementation, i.e. the translation to programming language. In this stage we focus on the communication between the software designer and domain experts. While having in mind that the systems need to be implemented in a programming language, it is desirable to

provide an early link with the world of software developers and programmers. This means that a number of technical and social issues need to be addressed, among which adoption of standard specification, notation and documentation on models [Argent, 2001]. For this specification, one can distinguish between a) informal, b) semi-formal or c) formal specification. These specifications vary in the level of precision in notations and semantics. In general, the more formal the method is, the more difficult it will be to apply it. In this paper we propose to use the Unified Modelling Language (UML). UML is a widely used standard language for the specification, visualization, construction, and documentation of the components of software systems. We aim to explore the potential use of UML for the development of a DSS for river management, involving particular issues such as the involvement of different decision-criteria and different stakeholders.
We hypothesize that UML can help overcoming a number of problems that are generally encountered during the domain analysis, like issues in communication and ad hoc decisions during the design. The problems have been formulated as requirements, which the UML needs to satisfy to be of added value. We tested the extent to which UML satisfies the requirements by an empirical investigation concerning the Dutch Maas river.

2. STATE OF THE ART

The development of software tools to support river management begins with analysis of the system at hand. Often, graphical representations are used, such as system diagrams [Matthies et al., 2007]. The most common approach is to strongly link the graphical user interface to the code representing the environmental model [Papajorgji et al., 2004b]. Recently several model developers have started using UML. Barthel et al. (2005), for instance, have successfully used model conceptualizations in UML for the development of DSS. UML has also been applied to water to soil water-balances and irrigation- scheduling modeling [Papajorgji, 2004a]. The object-oriented approach is also used to enable the assembly of simulation models from previously and independently developed component models [Hillyer et al., 2003]. Pahl-Wostl et al. (2008) applied UML in environmental modeling and found the following advantages:

- The enforcement to be precise and concrete forces domain experts to clarify their own concepts and models and to make choices in otherwise fuzzy approaches.
- The incorporation of concepts as part of and interactive implementation process requires mutual explanations of concepts as part of a new project communication.
- Complex system structures can be communicated unambiguously.
- It can facilitate the comparison of competing concepts.

The most undisputable argument for using UML however is the lack of an alternative method that can cover such a broad and easy to understand variety of views [Pahl-Wostl et al., 2008]. In addition, other languages, like the Modular Modelling Language (MML) [Maxwell et al., 1997], which have been used in developing DSS do not have the level of international standardization support such as UML. The use of UML for the development of DSS seems promising. However, a structured approach of the evaluation of domain-specific requirements has not yet taken place. This work aims to provide such an evaluation, and add an assessment of the suitability of UML for this type of domain to the existing knowledge.

3. MATERIALS AND METHODS

In this work we have used knowledge from the water management domain in order to formulate the requirements of a DSS. UML was used as modelling method and it was evaluated empirically in a case study.

3.1 UML

On the surface, UML is a visual language for capturing software designs and patterns. It can be applied to a large number of different areas and can capture and communicate everything from company organization to business processes to distributed enterprise software. It is intended to be a common way of capturing and expressing relationships, behaviours, and high-level ideas in a notation that's easy to learn and efficient to write. UML is visual; just about everything in it has a graphical representation [Pilone & Pitman, 2005].
First and foremost, it is important to understand that UML is a language. This means it has both syntax and semantics [Pilone & Pitman, 2005]. UML provides a way to capture and discuss requirements at the requirements level, sometimes a novel concept for developers. There are diagrams to capture what parts of the software realize certain requirements. Also, there are diagrams to capture exactly how those parts of the system realize their requirements. Finally there are diagrams to show how everything fits together and executes [Miles et al., 2006].

Obviously the focus of UML is modelling. It is a means to capture ideas, relationships, decisions, and requirements in a well-defined notation that can be applied to many different domains. Modelling not only means different things to different people, but also it can use different pieces of UML depending on what you are trying to convey. The use case view captures the functionality required by the end users. The concept of end users is deliberately broad in the use case view and includes the primary stakeholders, the system administrator, the testers, and potentially the developers themselves. The use case view is often broken down into collaborations that link a use case with one or more of the four basic views. The use case view includes use case diagrams and typically uses several interaction diagrams to show use case details.

3. 2 Description of the reference system

Various management measures (river works) can be applied to optimize the functions of rivers. In the Netherlands, of these functions, the conveyance of water while maintaining a safe catchment is considered the most important one. The safety is expressed by an exceedance probability, indicating the expected occurrence interval of extreme discharges. In general for the Netherlands' rivers the interval is 1250 years. The water levels corresponding to peak discharges with a frequency of 1:1250 years are not supposed to exceed an indicated maximum level corresponding to the dike height minus a safety board. In order to maintain these water levels, even when the extreme discharges increase due to climate change, river engineering measures are taken. These engineering measures however, may have side effects on other functions during normal discharges. It is desirable that the negative side-effects on other functions, such as nature, agriculture, shipping and habitation, remain limited. For simplification purposes, we only include three user functions here; landscape impact, agriculture suitability and safety. Also, only three engineering measures are taken into account (Figure 1).
The impact on safety can be assessed by calculating the probability that levees are overtopped. This only occurs during extremely high discharges for which a maximum probability is legislatively determined. Landscape impact is relevant to the inhabitants of the area. It is determined by a combination of measure and area features, such as the changes in width, location in the river

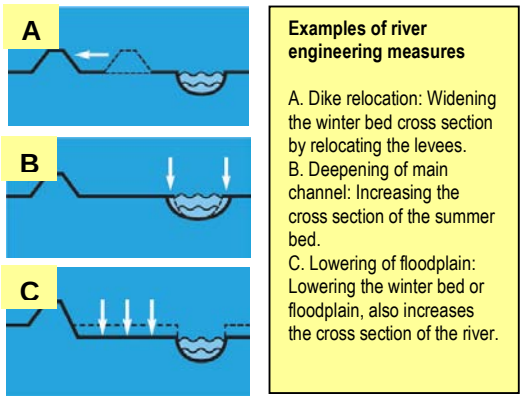

Figure 1. River engineering measures [www.ruimtevoorderivier.nl, 25-04-07]

bed, and the original river cross-section dimensions. The impact on agriculture is determined by calculating the annual effects of drought and effects of floodplain inundation (which is where the agriculture is assumed). Table 1 shows an overview of the relevant indicators in this system.

Table 1. Overview of indicators

	Time scale	**Spatial scale**	**Nature**
Safety	Probability [-]	River [~km's]	Crisp
Landscape impact	Impact [-]	Measure [~m's]	Fuzzy inputs and relations
Agriculture	Process[yr]	Location [km n^{o}.]	Fuzzy inputs, crisp relations

3.3 Evaluation process

The domain of this research is strategic river management. This refers to the early phases of river system planning, in which a trade-off has to be made between different objectives of the river planning process. From this, a number of requirements to the conceptual model can be derived. In fact the evaluation process has started with the identification and selection of evaluation criteria. The aim was to evaluate UML based to these criteria. The second step of the evaluation process was to design the river model. In this step the Maas river was used as case study. The third step was to discuss and evaluate the use of UML with domain and software experts to evaluate UML based on the criteria defined in the first step.

3.3.1 Evaluation Criteria

The identified evaluation criteria are listed below.

- *Different temporal and spatial scales.* A variety of functions is taken into account, and depending on their accompanying evaluation criteria the system description contains many different temporal and spatial scales.
- Because in our case, stakeholders or end-users determine what needs to be included the descriptions of the relations vary from *qualitative to quantitative.*
- *Different actors require different types of information.*
- Implementation of the design in a software tool requires a *consistent* design.
- *Flexibility with respect to model extension.* The stakeholders' interests represented in the model may change over time. We want to be able to anticipate such changes.
- For iterative development of the model, *the communication with people from different backgrounds needs to be supported.*

3.3.2 River Model Development

The model was developed based on a readily existing system diagram. The problem of this diagram however was, that it is not sufficiently specific to provide a sound basis for programming. The system was redesigned by the authors, from a river engineering and software engineering background, respectively. Part of the design is based on literature research, and on study of the case documentation. To further specify the system, additional knowledge was collected during meetings with river experts, both with a practical and research background. By bringing together the knowledge, a number of diagrams applicable to the river domain were developed.

3.3.3 Expert Evaluation

After designing the model in UML, the eventual product and the process underlying it were again discussed. During a workshop with model designers in environmental modelling a number of pros and cons of using UML came to light. The actual designs were again discussed during face to face meetings with people with a background in river research, both from their practical experience as well as from research experience. The focus of these discussions was both on the use of the diagrams during the design of the system, as well as on the later 'usability' of the diagrams. The expert opinions were categorised corresponding to the requirements that were outlined earlier.

4. EMPERICAL STUDY RESULTS

Our research aims to develop a conceptual model that outlines how a software tool can support various information needs in the trade-off between different river management measures. A number of UML diagrams will be used. Some indicative examples are shown below.

4.1 Use cases

In this example, there are three main groups of actors who will use the information provided by the tool (Figure 2). In the first place, there is a river manager. He has a set of measures at his disposal. He may want to implement different measures at different locations, and with varying parameters. His initial interest in proposing measures, in this case, is an improvement of the flood safety. He wants to know how safety changes, at different locations along the river, as a result of the measures. Second, inhabitants are interested to know what the impact of the river works (measures) on the landscape will be. They prefer measures that fit in the landscape to measures that severely disrupt it. Third, farmers want to know how the measures affect the ground water levels in the area, and the frequency with which floodplains are inundated. Also the duration and depth of this inundation, as well as the season during which it takes place, is important in the evaluation of preferences.
This diagram clearly illustrates which information is required by which actor, and it shows at a very abstract level how the information coheres with the river system. Consistency is approached from two angles here; first, all the relevant actors need to get a place in the diagram, and second all information needs require an actor perceiving it. In the initial stage of the DSS design this is a very helpful diagram to draw, because it shows which information needs be derived from the eventual model. By keeping the diagram very abstract in this stage, one prevents looking into the relations too much. Too much detail in the relations may shift the focus of the design to the availability of knowledge, models and data, something not yet important in this stage.

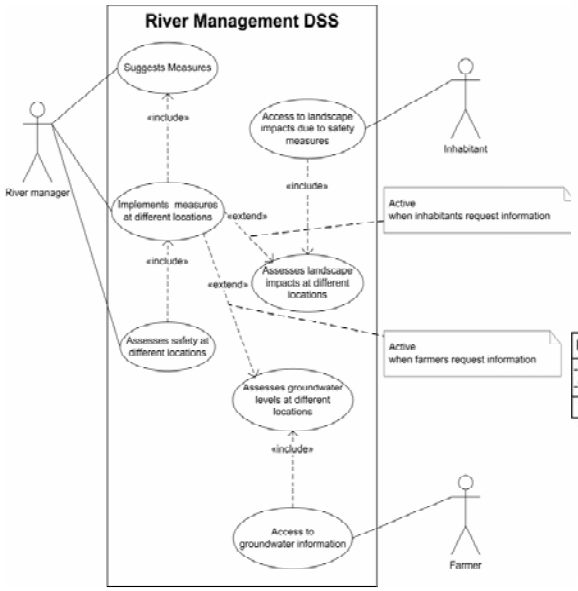

Figure 2. Use case diagram

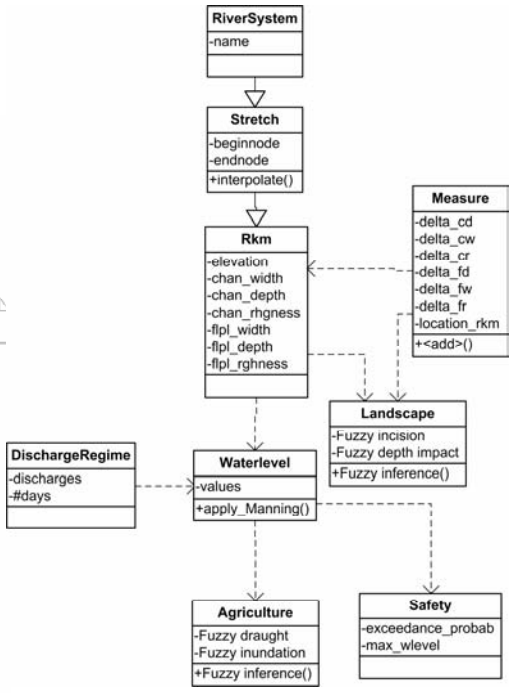

Figure 3: Static structure of the river system

4. 2 Static structure

The static structure, or class diagram, depicts the time-independent relations between the objects in the model. Classes are collections of objects. In our case, the model applies to a certain river system. Although it is common to omit attributes and relations in a conceptual model, we have added some of them here to clarify the model structure (Figure 3; previous page). Including attributes allows to more easily check the consistency of the conceptual model; if attributes are not included it is easier to define overlapping or inaccurately defined classes.
The river system consists of different, geologically distinct, stretches. Stretches have a begin- and end-node. Within each stretch a number of river kilometres (rkm) is located. For each rkm a cross-section is described along with roughness data. The measure class contains the different measures. Attributes hereof are the changes (delta) in cross section (channel depth and width, floodplain depth and width) and roughness data (channel roughness, floodplain roughness). The measures perform operations on the cross sections, for the rkm's at which the measure is located. Measure and location dimensions are the input for a fuzzy inference [Zadeh, 1973] on landscape impact. The new cross section, combined with the discharge regime of choice, determine the new water level. The water level, next, is the input for safety and agriculture indicators, one of which is evaluated using probability theory, and the other using another fuzzy inference.
The static structure diagram shows how information in the system at hand is related. It clarifies which properties are inherited, and it indicates how the different objects relate to each other. Thinking about accurate classes and subclasses provides a better insight into the system and provides a sound basis for model implementation. Moreover, it leads to a modular structure, which makes it easier to attach new functions or indicators.

5. SYSTEM LEVEL MODELING

An illustrative example showing how the UML models were transformed into code (i.e. a system level representation), using Matlab, is shown below.

```
(1)   river = load ('stretches.txt');
      measure = xlsread ('measure.xls');
      Q = load('QBorgharen93.txt')';
(2)   ini_prof = initiate (river); % interpolate profiles between nodes
(3)   wlevels1 = manning(Q,ini_prof); % Q in rows, nodes in cols
```

In step (1), the data-files area read. The stretches.txt file contains a number of river stretches nodes, defined between the "beginnode" and the "endnode", each characterized by the attributes summarized in the UML static structure. Example of the data set of the text file is shown in Table 2. Other data-sets (not shown in the example code above due to space limitations) are the "measures" and "discharges" which are required for the calculations, as it is depicted by the static UML model. A series of average discharges per day is given, in this case for the year 1993. In step (2) the initial profiles per stretch node (shown in the model with the class "Rkm"; river kilometer) are interpolated from the stretches, by the subroutine 'initiate'. Step (3) then applies the subroutine 'manning' to calculate "Waterlevels" based on the given discharges and on the profiles per "Rkm". In the following steps, not depicted here, the measures are applied to their respective locations (Rkm's).

Table 2: Example of dataset 'stretch'

Elevation [m+NAP]	Channel width [m]	Channel depth [m]	Channel roughness [-]	Floodplain width [m]	Floodplain depth [m]	Floodplain roughghness [-]
45.62	130	7.03	0.03	10	14.38	0.029
18.22	90	10.04	0.03	2590	2.78	0.026
…	…	…	…	…	…	…

Figure 4 shows an example plot produced by the DSS of the original river stretch, and the river stretch after measure implementation. On the latter the 'manning' subroutine is applied again to calculate changes in water levels. The water levels, and old and new river descriptions, next provide input to the fuzzy agriculture and landscape modules.

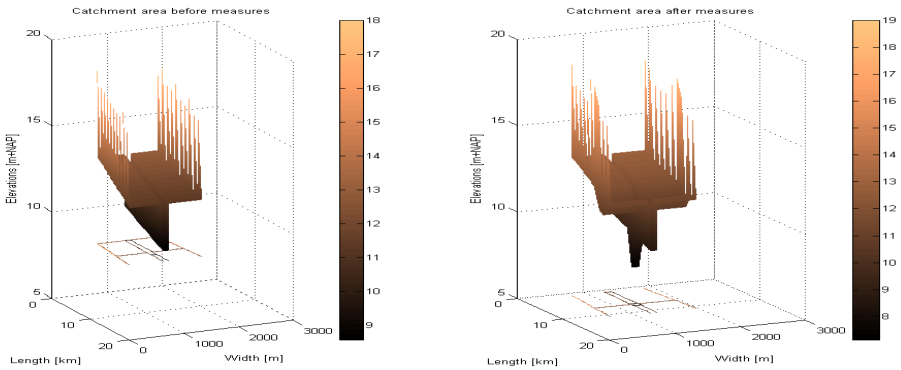

Figure 4: 3D plots of a river stretch before (left) and after (right) measure implementation

6. CONCLUSIONS

To conclude this paper, we present the results of our empirical evaluation that describes the "performance" of UML with respect to the criteria defined in section 3.3.1.

Ability to handle different spatial and temporal scales.
The static structure shows the coherence between objects on different spatial scales, and the temporal scales can be addressed in attributes of the objects (which for that purpose need be included in the conceptual model). Inclusion of differently scaled variables potentially increases the solution space and supports the comprehensiveness of the conceptual model.

Ability to deal with quantitative and qualitative relations.
UML semantics provides sufficient accuracy to come to a unified understanding of the meaning of the relations in the system without comprising the need to be overly formal. A disadvantage is the fact that in the natural system most relations are of a causal nature, requiring the incorporation of diagrams representing dynamics to clarify the way the system works.

Ability to deal with different actors and their respective information requirements.
The application of the use case diagram is the initial step that helps clarifying which actors need information from the model, and how the different kinds of information cohere with the natural system. It requires an accurate specification of the model users and possible other actors involved.

Design consistency.
Unlike many other graphical representation methods, the UML syntax and semantics strongly force one to think about the consistency of the design. Inconsistencies lead to overlaps, incompleteness or in the worst case to a failure to include all components in the same diagram set. The occurrence of either of the three therefore forces one to rethink the design, and in this manner avoid inconsistency issues during further implementation.

Flexibility with respect to model extension.
UML provides a sound basis for software documentation. This makes it easier to extend the model, than in comparable cases in which the documentation is less specific. Using the object-oriented approach allows treating objects as 'libraries', which can easily be extended.

Once matching w.r.t. attributes and operations, new classes, or for instance 'user functions' can easily be added. A disadvantage is that changes in relations between objects or users still require extensive re-engineering.

Support the communication with people from different backgrounds.
A disadvantage of using UML is that although the diagrams are relatively easy to intuitively understand, building them is not so straightforward. Particularly when more different diagram types are involved, quite some study may be necessary before the benefits of UML can be gained to the fullest.

ACKNOWLEDGEMENTS

We would like to thank Dr. Jean-Luc de Kok of University of Twente for his comments on the manuscript of this paper. We would also like to thank Georg Holtz and Dr. Marcela Brugnach of Osnabrück University for organizing a discussion session on UML, and we are grateful to all participants for sharing their experiences and commenting on our work. During the last phases of this work, Dr. Dokas, was supported by the research project SCEWA (Grand No 2007-DRP-2-S5), funded by the Irish Environmental Protection Agency under the DERP grand scheme. Miss Janssen was supported by the Cornelis Lely Stichting in the framework of her PhD research.

REFERENCES

Argent, R. ad B. Houghton. Land and water resources model integration: software engineering and beyond, *Advances in environmental research*, 5, 351-359, 2001

Barthel, R., D. Nickel, A. Meleg, A. Trifkovic, and J. Braun. Linking the physical and the socio-economic compartments of an integrated water and land use management model on a river basin scale using an object-oriented water supply model, *Physics and Chemistry of the Earth*, 30, 389-397, 2001

De Kok, J.L and S. Kofalk. Towards a user-oriented design of a DSS for Integrated River-Basin Management: the Elbe Prototype DSS. In: A. Voinov, A. Jakeman, and A. Rizolli (Eds.), Proc. of the IEMSS Third Biennial Meeting: "Summit on Environmental Modelling and Software", July 9-12, 2006, Burlington VT, USA, 2006

Hillyer, C., J. Bolte, F. van Evert and A. Lamaker. The ModCom modular simulation system, *European Journal of Agronomy*, 18, 333-343, 2003

Maxwell, T. and R. Costanza. A language for modular spatio-temporal simulation, in *Ecological modelling*, 103, 105–114, 1997

Matthies, M., J. Berlekamp, S. Lautenbach, N. Graf and S. Reimer. Systems analysis of water quality management fort he Elbe river basin, *Environmental modeling and software*, 21, 1309-1318, 2006

Miles, R. and K. Hamilton '*Learning UML 2.0*', O'Reilly, 2006

Pahl-Wostl, C., J. Möltgen, E. Ebenhoeh, and G. Holtz. The NeWater management and transition framework – state and development process, in: C. Pahl-Wostl, P. Kabat and J. Möltgen (eds.) 'Adaptive and integrated water management – coping with complexity and uncertainty', CAIWA-conference, Nov. 12-15 2007, Basel, Switzerland, 2008

Papajorgji, P. and T. M. Shatar. Using the Unified Modeling Language to develop soil water-balance and irrigation-scheduling models, *Environmental modeling and software*, 19, 451-459, 2004a

Papajorgji, P., H. W. Beck and J. L. Braga. An architecture for developing service-oriented and component-based environmental models, *Ecological modeling*, 179, 61-76, 2004b

Pilone, D. and N. Pitman. UML 2.0 in a nutshell, 234pp., O'Reilly , 2005

Zadeh, L. A. Outline of a new approach to the analysis of complex systems and decision making, *IEEE Transactions on systems, man and cybernetics*, SMC-3, 28-44, 1973