A Flexible and Extensible Open Source Tool for Urban Drainage Modelling: www.Water**Aspects**™.org

Morten Grum, Eric Longin and Jens Jørgen Linde

PH-Consult, Ordruphøjvej 4, 2920 Charlottenlund, Denmark
(E-mail: mg@phc.dk)

Abstract
Over the past twenty-five years several modelling tools have emerged each with their specialisations with respect to methods and markets. However, within both research and practise there is a need for a more flexible water-modelling environment. From an engineering point of view, today's tools are often too rigid to allow for the analysis and evaluation of the very frequent case-specific issues of today's environmental and urban flooding projects. From a scientific point of view, the exploration of new methods and postulates requires more flexibility than is usually available in today's urban drainage tools. From a technological point of view, the poor flexibility means a long time lag between the availability of scientific research results until their incorporation into engineering practise.

Water**Aspects**™ is a water-modelling framework that has been developed to meet this demand for higher flexibility and extendibility. With its focus on planning and model assisted real time monitoring and control, Water**Aspects**™ is a platform directed towards simple urban drainage models for tasks stretching from long term simulations with historic rainfall series to real time adaptive modelling using a stochastic state-space modelling approach. Water**Aspects**™ includes flexible event definition, event handling and presentation tools. The software has been coded in Java and uses standard XML-formats of data persistence and data interchange. Water**Aspects**™ will during 2004 be launched as an open source project managed by PH-Consult.

Keywords
Water, Modelling, Flexibility, Extensibility, Framework, Open Source, Java, XML

INTRODUCTION

Modelling and data analysis are key elements of urban drainage management. Over the past twenty-five years several modelling tools have emerged each with their specialisations with respect to methods and markets. Most modelling tools were, and many still are, associated with a method or even with a single methodology. The modelling tool often dictates the mathematical representation of the system and its behaviour allowing only for minor case specific choices and hardly any variation in level of detail. Within both research and practise there is a need for a more flexible water-modelling environment.

From an engineering point of view, water modelling tools need to be able to meet the need specific to the project at hand. This could involve a particular non-typical local infiltration into the system or this could be a special type of detention basin behaviour that has been observed. One could also be in the situation local pollutant measurements have given a basis for a more accurate description of the pollutant variations than is available in existing tools. A particular case may involve hazardous wastes which require additional calculations that are not present in a typical urban drainage tool. In all these cases the engineering team needs a highly flexible tool that will allow these analyses either through its built-in user flexibility or via the creation of one or two add-ons. These examples illustrate some situations where Water**Aspects**™ aims to provide flexibility.

From a scientific point of view there is a need for a framework which allows for the exploration of new ideas to a much larger extent than would normally be the case in practice. Today this sort of exploratory work is done using anything from spreadsheets through tools like MatLab to the researcher's own programs tailored to the particular issue being studied. These are all highly flexible tools indeed and powerful in each their way, but the common aspects of water and substances modelling such as rainfall-runoff processes, substance mixing and substance processes are not standard elements of these flexible tools. They all therefore require a large overhead in considering these issues to the extent that is needed for the research. Alternatively, a researcher may choose to use an existing modelling tool and then limit his or her exploration to the existing features within the tool. The long term effects of this would of course be detrimental to the field (or at least to the research team). Innovation that in this way limits itself to state-of-the-art is not innovation. Water**Aspects**™ aims to provide the water research community with a modelling and data analysis framework dedicated to water issues whilst at the same time providing the sort of flexibility that can be obtained using tools like spreadsheets, MatLab and own programming code.

Bridging the gap between research and practice will always be a topic in need of attention. Whilst practitioners have a tendency to stick to the overly simplified and known, researchers often do not appreciate the engineers' day to day issues and the broader context of the research being conducted. Additionally, the expenses of creating the tools that would bring any clearly beneficial results from research to practice are often high. As a tool dedicated to both research and practice, Water**Aspects**™ aims to assist in bridging this gap between research and practice.

Another of Water**Aspects**™ objectives is to reduce the time from scientific results to practice. Limitations in engineering tools often determine "general practice" and they do so, not because this is the best way of evaluating the issue, but simply because there exists a modelling tool to look at issues in this or that way. The adaptation or exploration of new practices has over the past many years been slowed down by non-dynamic water modelling tools bound to one method and/or approach. Any attempt to address these issues would require not only a truly object oriented software design but direct access to the code. Open source allows organisations, wishing to make extensions or add-ons in order to adapt or explore new approaches, to manage and prioritise their resources more closely. This sort of flexibility can also be achieved to some extent in a close source application by providing a highly modular architecture allowing users to create their own add-on modules.

Scope

Water**Aspects**™ is a framework for discrete-time deterministic and stochastic water modelling and data analysis using simple conceptual models. The modelling tool was initially developed to meet requirements of the MANTISSA research project (Grum et al., 2002). The framework assumes unidirectional flow where upstream dependencies and loop flows must be lagged at least one time step. In all essence at each time step Water**Aspects**™ allows for the "*bookkeeping*" of water and substances moving downstream at each time step. The computation of how much water comes out of a water component at each time step is defined by its implementation and independent of the framework.

Although Water**Aspects**™ aims to be an extensible water modelling framework geared to expansion-on-demand, its creators are conscious of scope. Hydrodynamic modelling based on solutions to the Saint-Venant equations, as known from the highly physically based urban drainage models like MOUSE, InfoWorks and SWIMM, lies clearly out of scope. The frameworks' object oriented design is not well suited for these types of simulations. This also means that Water**Aspects**™ is not well suited for detailed urban flood evaluations requiring a good description of backwater effects.

WATER COMPONENTS

Water modelling in Water**Aspects**™ is component based. That is a water system is built up of interconnected water components. Generally speaking, a water component is a model component into and/or out of which water flows. Examples of water components are a simple surface, a detention basin, a linear reservoir, a joiner that joins flows, a splitter that divides a flow and even a rainfall component. Water**Aspects**™ is open source and it is therefore relatively easy for a user or an organisation to create or have created his or her own water component. The Java programming language is relatively easy for engineers to learn and developer's documentation will include a guide on "How to create your own base water component".

Connecting Water Components

Water components have inlets and outlets and can be connected to one and other by connecting the outlet of one component to the inlet of another. In each component is defined a function that specifies the amount of water and substance masses that leave its outlets within each time step. The amount of water and substance masses entering a water component through its inlets is thus defined by its immediate upstream water components.

There are two distinct types of connection points (inlets and outlets) whose dimensions are incompatible. These are "volume" and "depth" connections and are by default handled in the units [m^3] and [mm] respectively. Inlets and outlets of a detention basin are examples of "volume" connection points. In contrast the outlet of the rainfall component is a "depth" connection point. Surface areas have a "depth" inlet and a "volume" outlet because they convert rainfall amounts from [mm] to a corresponding volume of water [m3].

A special type of component is a water network. This is a composite component consisting typically of a number of interconnected water components. Some of the contained components may themselves be water networks. Inlets and outlets can be defined on the water networks and these can be attached to the inlets and outlets of the water components contained within the network. This gives great flexibility in combining water components to network units which are meaningful and useful to the modeller.

Volume Balance and Mass Balance

In water modelling it is useful or even essential at different levels to know how much is going in (and from where it is coming), how much is getting produced or added internally, how much is coming out (and where it is going), how much is getting removed or lost, how much is accumulated and finally that they all more or less balance to zero. This sort of summary information (either by month, year or for a whole series) is valuable in understanding a system and appreciating the relevance of various elements. Due to the object oriented design used in Water**Aspects**™ this information can be readily extracted from any water network.

PARAMETERS AND VARIABLES

In water modelling the distinction between parameters and variables is highly context specific. A value considered to be a parameter at one time- or spatial-scale may well be considered a variable at another. This is purely a question of the model's underlying assumptions. Whether these assumptions are reasonable or not will typically depend on the scales and extent being considered. A flexible modelling framework should not a priori dictate whether a given value should be considered a parameter or a variable.

Variable Formulations

In Water**Aspects**™ no distinction is made between parameters and variables. All are called variables. A variable can have different formulations. It can be:

- a constant
- an arbitrary function of any other variables in the model. Where cyclic dependencies occur the dependency must be lagged at least one time step. If, for example, the outlet capacity of a detention basin is a function of the water volume in a downstream detention basin then the downstream volume in the previous time step must be used.
- a predefined function of other variables (e.g. sum, average, maximum, minimum or bound to one and other)
- a tabular dependency of another variable (lagged when downstream)
- a time series value read from a file

Users can also create their own variables, define their type and dependencies.
Changing the formulation of a variable is not always logical and may for some component variables even result in obscure behaviour. A suitable strategy for access rights with respect to adding and changing variable formulations is on the drawing board as part of profile based user interface (see below).

Predefined Variables

In some cases it is useful to have variables with predefined types. Examples of these are a difference variable which subtracts one variable from another and a sum of squares variable which gives the accumulated series of the squared value of a series. These are used when, for example, fitting a model to measured values using a least squares estimation criterion or simply when comparing any two series. Another simple but often very useful example, is an accumulation variable which gives the accumulated series of a variable under consideration of the time step and the original variables units.

Missing Values

Any variable can go missing for some reason or other. Both in the real world and in research missing values in measured data series are more the rule than the exception. Models may also have a validity-range and a modeller may choose to present situations outside the models validity range as missing values. In stochastic modelling differences in sampling frequency of various observed series are often handled by assuming the less frequently sampled variable to be missing at instances where it has not been observed.

Water**Aspects**™ has several features geared towards handling missing values. We'll look at three. Firstly, as discussed later (see "Time Step Flexibility") there are several options when reading time stamped data from files. Secondly, the user can define a given constant value that a variable should take when it is missing. Thirdly, Water**Aspects**™ allows the user to define replacement variables whose value should be used when any given variable is missing. In periods where a rainfall measurement is missing it is often useful to be able simply to use data from a nearby gauge.

Criteria and Inequalities for Flexible Event Definition

A special group of time series variables in Water**Aspects**™ are the criteria variables. These are variables that at any moment in time are either zero or one corresponding to false and true respectively. Criteria variables can be combined in Boolean logic using "and" or "or" criteria variables.

A special criterion is the inequality time series in which the value of two arbitrary expressions of any other variables in the model can be evaluated at every time step and compared using the standard operators <, <=, > and >=.

Event frequency and the return period of a given event are common indicators of urban drainage performance. This could for example be the frequency of a combined sewer overflow or the frequency of critically low oxygen concentrations in the receiving waters. Attached to any event statistics is an event definition which often does not relate to the effect being studied but to some other state of the system or even simply to the inputs. Water**Aspects**™ gives the user the flexibility of making his or her own event definitions and having these used for the event summary and extreme statistics.

FLEXIBILITY IN TIME RESOLUTION

Different phenomena need modelling at different resolution in time. Different rainfall-runoff systems require different time steps. For example, an urban drainage problem may need five-minute time steps whereas a rural river catchment may need hours or days. Depending on observation frequency and pixel grid size a rain plane model as described in Grum et al. (2003) and Krämer et al. (2004) may use a time step between 1 and 10 minutes.

A modelling framework like Water**Aspects**™ should allow for a user defined time step which is not restricted by the frequency with which the input measurement data has been provided. It should for example be possible to work with a time step of one hour even though rainfall data comes as one minute intensities or even as arbitrarily placed tip times. The Water**Aspects**™ framework works with time series in two forms:

- Arbitrarily time stamped data. Typically the original measured data series or series generated by other tools.
- Equidistant time series. The time series which are actually used as inputs or observed outputs in the modelling work.

Before arbitrarily time stamped data can be used in relation to modelling it needs to be converted into equidistant time series. This can be done in several different ways. In Water**Aspects**™ a number of different transformation methods have already been implemented including:

- Rainfall from tip time stamped data
- Rainfall from time stamped intensities
- Rainfall from time stamped cumulative rainfall
- Interpolation
- Aggregation
- Nearest value
- Last value

Note that the applied conversion strategy is independent of the original file format. All data formats (files, databases or others) are read through a common interface for time stamped data to which the conversion strategy is applied. Users that comes with a new file format needs only develop a reader implementing the common time stamped interface after which any conversion strategy to equidistant time series can be applied.

However, the desired transformation strategy can sometimes be deduced on the basis of data source of origin. In future versions it is likely that Water**Aspects**^{TM} will make some automated best guess of the required transformation and then allow the user to modify this if needed.

The Water**Aspects**^{TM} team has juggled with the idea of allowing different water bodies within a single model to operate with different time steps. This would allow modelling, for example, of an integrated model involving an urban drainage system, a stream and a lake where the urban drainage system could work with five-minute time steps, the stream with 1 hour time steps and the lake with time steps of a day. However, unless prioritised by a specific need, the use of different time steps for different water bodies is not likely to be possible in Water**Aspects**^{TM} within the near future.

USABILITY AND FLEXIBILITY

As much as flexibility is a glory for those who need it, so is it a menace to those who do not. In any software development flexibility will to some extent always come at the cost of usability. Water**Aspects**^{TM}'s user interface today consists of select-and-view windows as illustrated in figure 1. Components selected in the tree on the left are viewed on the right. In Water**Aspects**^{TM} the development team is working on the implementation of a system of user profiles, controlling issues ranging from the appearance of the user interface including menus, access rights to modify values, to change system structure and the visibility of components within the tool and model. Users will then be able to create their own profiles dedicated to their needs or even to specific tasks based on one of default profiles that will be available. This is on the drawing board today.

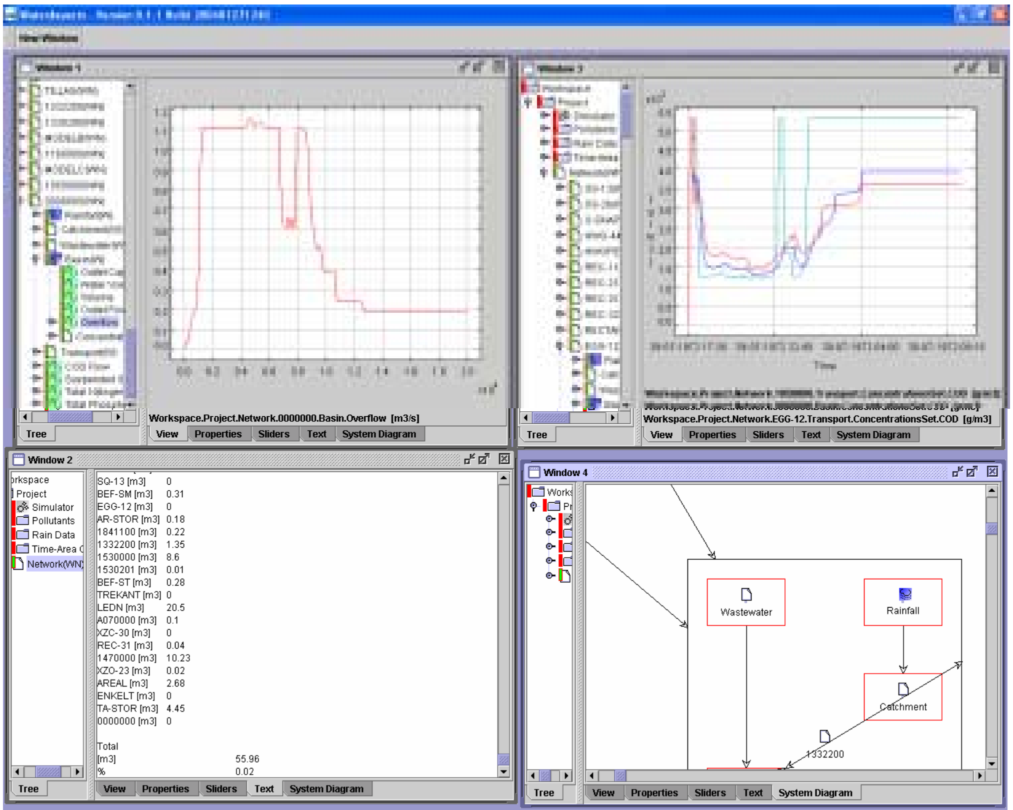

Figure 1. Today Water**Aspects**^{TM}'s user interface consists of select-and-view windows providing rapid browsing of model components and results.

CONNECTIVITY

Any modelling tool and software developing organisation for that matter should be conscious of scope. Illusions of "one tool that does it all" have now truly been overtaken by a conscious strive towards open standards and better connectivity. Water**Aspects**^{TM} adheres to a standard Java XML object (JavaBeans) format definition. This provides immense openness toward anyone's wish to create or edit these files, extract data from them or simply convert them into other formats using XSLT styling sheets. Most common column arranged time stamped time series text files can be loaded into Water**Aspects**^{TM} using the multipurpose text file reader which is similar to the text-to-column converters known from spreadsheets. Some users may wish to connect to other applications via open interface definitions such as OpenMI (Blind and Gregersen, 2004). Placing a Water**Aspects**^{TM} sub-system within an OpenMI framework could readily be done by coding a wrapper/adapter object for the Water**Aspects**^{TM} sub-system. Others may wish to use an existing OpenMI compatible program within the Water**Aspects**^{TM} framework which would require the coding of a new water component within which to wrap the existing OpenMI compatible program. As an open source project the connectivity can be created when needed and find it a wreath while investment. Code submitted and accepted into the Water**Aspects**^{TM} code pool will be maintained with respect to refactorings of the Water**Aspects**^{TM} code.

DETERMINISTIC/STOCHASTIC MODELLING

Measured data is playing an increasingly important role in environmental and water engineering. Measurements are becoming cheaper and many have now for some decades played a major role in real time management of waste water systems. Arising today are also database systems geared to storing and managing these large amounts of data and utilising them for planning purposes. This trend gives rise to many interesting new approaches to model building, calibration and data assimilation in general.

The MANTISSA project involved the assimilation of a multitude of rainfall and runoff data (Grum *et al.* 2002). This required a tool able to perform both deterministic and stochastic state space modelling. Water**Aspects**^{TM} has therefore been developed with stochastic state space modelling as a part of its primary focus. Though beyond the scope of this paper there do lie many interesting possibilities both in terms of planning and real time modelling in exploiting these data assimilation features of Water**Aspects**^{TM}.

ARCHITECTURE, DESIGN, EVOLUTION AND THE WAY WE WORK

Water**Aspects**^{TM} has an object oriented design and has been coded in Java. Its architecture consists of a classic separation of model (in a software design sense of the word) and graphical user interface (GUI) design. Thus system logic is separated from and independent of its representation in the GUI. The two are joined by a model-view-controller design which is implemented using the standard Java listener design. System persistence is obtained using Java's standard API for XML-object serialisation.

Agile Software Development

The core Water**Aspects**^{TM} development team has found much inspiration in the world of Extreme Programming (Beck, 2000 and Jefferies *et. al.* 2000) and agile software development methods in general (Cockburn, 2002). In contrast to the classical waterfall software development methodology (requirement/idea-analysis-design-coding-tests-release) these agile approaches are built up around small iterations. Each iteration typically involves:

- isolating a need into a requirement (documented by keywords and hand-drawn sketches)
- quick pattern based object oriented design (Gamma *et al.*, 1994)
- test-first coding:

- coding of automated tests (making requirements specific)
- refactoring (when needed)
- production coding

These small iterations allow for continuous adaptation to changing requirement and therefore play an important role in achieving flexibility. Tomorrow's needs are unknown today.

Refactoring. Another important and indispensable element of the agile approaches is the continuous refactoring. Refactoring is improving the design of existing code without changing its functionality (Fowler, 2000). As future requirements are unknown so are the software designs that best meet these requirements. Many software development teams have experienced the price of designing and maintaining code designs based on anticipated requirement rather than an approach of continuous refactoring to meet the requirements at hand. Anticipated requirements rarely turn out as they were anticipated and sometimes never materialise into actual requirements.

Test-First. The creation of test code before production code brings focus to what an object does rather than how it does it. Focus is foremost on the requirements and on how an object will be used and not on its internal implementation. The large tests (acceptance tests) serve as a concretisation of requirements. The smaller object specific tests (unit tests) serve as quality control at the finest level. The whole test pool can be run at any time and is run at least once a day as a part of the nightly build. These tests also play an important role in providing flexibility and making the code refactorable because unanticipated consequences are disclosed immediately.

CONCLUSIONS

The demand for a flexible and extensible water modelling tool has been identified and characterised. The vision of a water modelling tool that meets a part of this demand has been presented and implemented. The water modelling framework Water**Aspects**™ was developed with an initial focus on the methods and requirements of the MANTISSA project and has now been used successfully on several other projects. The water modelling framework will be launched and managed as an open source project at www.Water**Aspects**.org. The framework's reception within the world's water modelling community and its ability to meet the identified need for flexibility and extensibility are yet to be seen.

ACKNOWLEDGEMENTS

Much of the work behind Water**Aspects**™ and this paper have been carried out as a part of the EU sponsored MANTISSA project. We wish to thank all project partners, particularly Gavin Robbins, Stefan Krämer and Antonello Gaviano who were the very first true users of Water**Aspects**™.

REFERENCES

Beck, K. (2000). Extreme Programming Explained: Embrace Change. Addison-Wesley. ISBN 0-201-61641-6.

Blind, M. and J. Gregersen (2004). Towards an Open Modelling Interface (OpenMI) – The HarmonIT project. In Pahl-Wostl, C., S. Schmidt and Jakeman, A. J., (eds.), Proceedings of the Second Biennial Meeting of the International Environmental Modelling and Software Society, iEMSs, 2004.

Cockburn, A. (2002). Agile Software Development. Addison-Wesley. ISBN 0-201-49834-0.

Fowler, Martin (2000). Refactoring : improving the design of existing code. Addison-Wesley. ISBN 0201485672.

Gamma, E., R. Helm, R. Johnson and J. Vlissides (1994). Design Patterns: elements of reusable object-oriented software. Addison-Wesley. ISBN 0-201-63361-2

Grum, M., P. Harremoës and J. J. Linde (2002). Assimilating a multitude of rainfall and runoff data using a stochastic state space modelling approach. 9th International Conference on Urban Drainage (9ICUD), Portland, Oregon.

Jeffries, R., A. Anderson and C. Hendrickson (2000). Extreme Programming Installed. Addison-Wesley. ISBN 0-201-70842-6

Krämer, S., M. Grum, H.-R Verworn and A. Redder (2004). Runoff Modelling using Radar Data and Flow Measurements in a Stochastic State Space Approach. Presented at the Urban Drainage Modelling Conference (UDM04), Dresden.