Decision-Support System Workbench for Sustainable Water Management Problems

M.S.Morley[a], **C.K.Makropoulos**[b], **D.A.Savic**[a] **& D.Butler**[b]

[a] *Centre for Water Systems, University of Exeter, Devon, United Kingdom*
email: M.S.Morley@exeter.ac.uk

[b] *Urban Water Research Group,*
Department of Civil and Environmental Engineering,
Imperial College London, United Kingdom

Abstract: Decision Support Systems (DSS) comprise a wide-range of computer-enabled applications that are based on some form of analytical model, commonly linked to a database. Coupled with the visualization and spatial analysis facilities provided by a Geographic Information System (GIS), a unifying framework can be developed to promote the uptake of advanced decision support technology across a wide range of stakeholders. This paper describes an architecture for a Decision Support Workbench to facilitate the rapid development of DSS applications. Custom DSS implementations supporting decision-making for a wide range of problems may be generated through an extensible, interactive environment, featuring "drag and drop" object-oriented components and dynamic connections between them. Also presented are a number of components for the workbench derived from existing tools for integrated modelling, spatial visualization and advanced decision support. The DSS Workbench prototype is demonstrated through an example development of a DSS for water distribution systems modelling and optimization. The developed DSS includes: (1) simulation modelling tools (e.g. EPANET), (2) optimization algorithms based on the Evolutionary Computing principles, (3) database tools for data storage and manipulation, and (4) spatial analysis tools based on GIS. The new DSS is tested on a case study of water distribution network design. Evolutionary Computing, which uses a computer model of the principles of Darwinian evolution to "evolve" good designs, is used here to design a pipe network. This is a highly complex problem for which classical solution techniques such as linear programming or gradient-based methods are often inappropriate or sometimes hopelessly inadequate. The solutions obtained demonstrate the feasibility of developing a DSS Workbench and its useful implementation for water distribution network modelling and optimisation.

Keywords: Evolutionary Computing, Decision Support Systems, Network Optimisation, Workbench

1. INTRODUCTION

Decision Support Systems (DSS) in general and Spatial Decision Support Systems in particular, represent an attempt to assist the decision-making process by a set of intelligent, knowledge-based techniques. These techniques are of particular importance when the problems are complex and semi-structured [Densham, 1991], as is the case in the urban water management environment [Makropoulos et al., 2003]. A major UK EPSRC Research initiative, the Sustainable Urban Environment Programme (SUE) is currently investigating, through its various projects, issues of spatial decision support for sustainable decision-making (CoDES Project), including the sustainable management of the urban water cycle (WaND Project). Due to the complexity and extensive scope of this aim, the development of an extensible environment facilitating the generation of diverse DSS Tools for multiple end-users was considered *sine qua non*. The development and a first application of this environment (Workbench) will be discussed.

2. DECISION SUPPORT FRAMEWORK

The Decision Support System Framework (Framework) is a software engine that underpins a Workbench application and coordinates interaction between a number of modules that can be combined to constitute a specific Decision Support Tool. The Framework may be used either in the context of the Workbench, facilitating the interactive generation of Decision Support Tools or as a library suite that can be used with conventional programming languages for tightly integrated solutions.

2.1.1 Modules

The basic element of the Framework is the Module. The basis of the Module system is the Model – Control – View paradigm popularised by SmallTalk [Goldberg and Robson, 1989] in which software is decomposed into three distinct "layers" for presentation, operation and data analysis/simulation. These layers are also well suited for dividing the application across client/server boundaries. Modules could include analytical engines, input/output interfaces, display libraries, generic optimization routines (e.g. a GA library) etc. Within each "layer" of the Framework, there are prototypes for the different types of modules, which define certain core behaviours for that type – for example, the Data Provider type has to implement a mechanism for discovering the type and structure of the data therein. Figure 1 shows the available module types and their use in constructing a Decision Support Tool.

2.2 Model

The model layer comprises the information upon which the Decision Support System Tool acts. These data may be in many forms: external database connections, spatial data, temporally variable data and software models.

2.2.1 Data Provider

The Data Provider Module type incorporates raw data sources such as database tables, GIS spatial data etc. along with a number of core data components that provide generic access to relational and hierarchical database forms and manipulation of spatial data. At present, access to relational databases in the Framework is achieved through an abstraction layer, which "wraps" a Microsoft ODBC connection to a database. The abstraction layer provides a standard interface to the Framework that allows it to query the relational database as to its data structures and to forward SQL queries to the database. Use of a standard interface allows the abstraction layer, in future, to present an unchanged interface to other database technologies, not limited to ODBC, which would be more appropriate for other platforms. Hierarchical databases – which are, naturally, more appropriate for representing some types of data - are supported in the form of XML-based databases. XML documents are commonly parsed into tree-like structures for manipulation using the Document Object Model (DOM). DOM was initially a proprietary interface specification developed by Microsoft but, unusually, the interface has been adopted by the wider community as a standard. The Framework exploits the hierarchical nature of DOM documents to manipulate databases with heterogeneous structures.

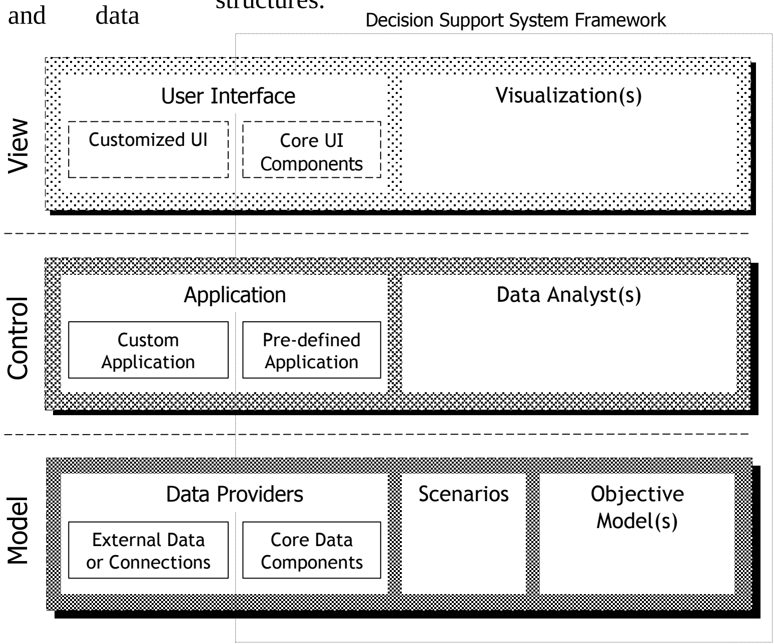

Figure 1: Anatomy of a Decision Support Tool

Access to GIS data from multiple sources is facilitated through the exposure of the ODBC connection if an ODBC aware spatial middleware is being employed. For local GIS tables, the Framework provides core data components for the inclusion of tables from MapInfo and Arc/Info. There is a presumption, however, that the GIS routines employed for visualization are able to accept the data from the data provider – as no translation facility is provided.

2.2.2 Objective Model

The Objective Model Modules represent the real-world environment on which the Decision Support System acts – a necessarily "fuzzy" constituent. For clean and wastewater systems, for example, the key Objective Model component is that of the *pipeline* network model.

2.2.3 Scenario

In order to simplify the structure of the Objective Model, the Scenario concept has been introduced. Scenarios represent instance-specific or temporally variable data for use in "what-if" analysis to record changes in state of the Objective Models or to act as a repository for results from Data Analyst components. Scenarios operate on the metadata exposed by the Objective Model and can be generated automatically as a snapshot of an objective model at a particular time or semi-automatically on a subset of the Model at a given stage. Once created, a Scenario can be used to revert the objective model to the state that it describes. Returning to the pipeline network example, Scenarios could be used to describe the control settings of valves, reservoirs and tanks across the network as well as for storing the results of a 24-hour hydraulic simulation.

2.3 Control

The control layer is responsible for the functional operation of the Decision Support Tool. Despite appearances in the View layer, the Control layer implements all but the most basic functionality of the application. This distinction is vital as it allows for a high-degree of independence between the View and Control layers allowing for the efficient replacement and variation of user interfaces. Whilst the View layer might be responsible, for example, for the processing of mouse clicks, the Control layer will ultimately determine what action that click performs.

2.3.1 Application

The Application module type is responsible for creating and managing all the components in a Decision Support Tool. At present, the Framework provides a standard Application type for undertaking optimization of an Objective Model. Other standard types will be added in future. Custom Application types can be created and added as with any other Module.

2.3.2 Data Analyst

Data Analysts are the actors in the Framework that perform some analysis or act on the data in Objective Model in some fashion. Examples of these components include Genetic Algorithm optimization, GIS spatial analysis and a hydraulic solver etc. Data analysts commonly store their results in Scenarios and present their results to Visualization components.

2.4 View

The topmost layer of the application is responsible for the interaction with the end user and the presentation of results. At the time of writing, the view layer is sparsely populated with key Visualization components – most notably a GIS routine for viewing a network. All user interface processing takes place within the View layer. The key characteristic of View layer Modules is that they are inherently passive – in that they do not respond to actions, as such, rather they utilize data exposed by the Control layer.

2.4.1 User Interface

The User Interface Modules can be neatly broken down into those constituents provided as core components by the Framework and those added to extend the core components in order to add support required by custom Application types. The core components envisaged for the system include common visual GUI elements such as forms, buttons and the common graphical controls associated with dialog boxes. Because of the variation in implementation of GUI elements across operating systems it is anticipated that only a sub-set of controls will ultimately be included as part of the Framework provided UI. Additional elements can be added to the system programmatically provided they conform to the Framework's interface requirements.

2.4.2 Visualization

Visualizations encompass facilities such as GIS maps, charts, 3D rendering etc. and access data exposed, particularly by Data Analysts, for display. Currently, the visualizations implemented include a GIS window, which is used to provide a graphical display of a network model with thematic mapping capabilities as well a basic charting Module, which can be use for displaying result comparisons.

2.5 Intra-Module communication

Behind the scenes of the Framework is a sophisticated software component, which has been termed an Object Interoperation Manager (OIM). This component is responsible for marshalling the interactions between the Modules in a Decision Support Tool. The OIM implements a standard interface for components to interact. In doing so it obviates the need for the components to be implemented in the same technology. The DSSF provides template wrappers for Windows DLLs, Windows COM objects, Windows .NET assemblies and CORBA objects. This interface can be developed to include support for web services accessed over SOAP (Simple Object Access Protocol) – this being functionally similar to CORBA's IIOP. Modules need to be registered with the Framework before they can be used in a Decision Support Tool. Registration takes place by passing an XML document to the Framework containing basic information on the purpose and functionality of the Module and, crucially, where the Framework can find the Module in question to instantiate it. Once registered with the Framework any Tool can have access to the Module.

2.5.1 Metadata

In order to meet the extensibility design criterion, it became apparent that the component system needed the ability to "publish" information about itself in such a fashion that Modules could query the capabilities of their peers and the data contained therein. At a more fundamental level, the attributes of individual elements within the Modules – particularly those associated with the model implementation – should also be able to be queried by user-interface Modules.

The absence of a built-in metadata facility in standard C++ and concerns about extending such functionality into other languages prompted the adoption of a simple metadata engine that is used for the representation of public data within Modules as well as facilitating the transfer of data between them. The desire to support diverse object implementations such as COM and CORBA also militated against adopting anything other than a proprietary metadata engine.

2.5.2 Thread safety

Multithreading enables an application to run many processes simultaneously and to take advantage of the multi-processor support provided by the Microsoft Windows NT and Linux operating systems. The use of this symmetric multiprocessing (SMP) allows operations to run concurrently on different processors in the same machine, or to parallelise the execution of object functions. The extensible nature of the Framework militates against adopting a standard threading model, as it is difficult to predict the needs of the potential modules that might be incorporated into a Decision Support Tool. Consequently, the OIM implements a centralized messaging system that facilitates communication between the Modules in a thread-independent fashion. Each Module is required to poll periodically this queue to discover if there are any messages waiting for it. The mechanics of this system are hidden from Module developers in the stubs of the Module code.

2.6 Workbench

The workbench application provides a “front-end” to the Framework – allowing the “drag-and-dropping” of the Modules registered with the Framework to create bespoke Decision Support Tools. In future, it will further allow rudimentary visual development of interfaces for these tools. Object-oriented “inspectors” allow the end-user to define the relationships between the Modules in terms of their attributes and events that can be .

One of the virtues of developing such Modules for use in the Workbench is that will implicitly have a full, coherent API. In turn, this means that it will be relatively straightforward to reuse the components within a more conventional programming context. This allows for the development of more complex applications than would otherwise be possible using the interactive Workbench technique.

3. DECISION SUPPORT TOOL: PIPE NETWORK DESIGN

Solving optimisation problems related to water distribution networks is recognized as an NP-hard analysis that has conventionally been approached using a number of techniques including hill climbing, linear and dynamic programming. Evolution algorithms represent an alternative, proven strategy for approaching these problems. The prototype application (Figure 2) seeks to demonstrate that the optimisation approach used in later versions of GAnet (Morley et al. [2001]) can be replicated in a user-extensible form within the Framework using a number of Modules. This application allows for the optimal design, on a least-cost basis, of a hydraulic network based on user-defined performance criteria.

3.1 Genetic Algorithm Library

Data Analyst. Genetic Algorithms (GAs) are stochastic algorithms whose search methods model mimic genetic inheritance and Darwinian mechanisms of survival [Michalewicz, 1996]. Individuals in nature evolve by developing characteristics allowing them to adapt in a particular environment. These characteristics are incorporated in their genes and the best ones survive through the process of natural selection, from one generation to the next. The individuals in the case of GAs are solutions to a given problem. The more robust ones survive and evolve. The metaphor is conceptually a powerful one: Darwinian natural selection and survival can be considered as one of the most complex and demanding “optimisation” problems ever encountered. Evolution is a natural process and is only “natural” that a mathematical equivalent would be used for some of our own complex optimisation problems [Walters & Savic, 1994, Makropoulos, 2003].

3.2 Network Model

Objective Model. Previous commercial work on optimisation software for water networks, (including Atkinson et al. [1998]), has highlighted the absence of an agreed standard for representing water network infrastructure and operating conditions. It has proved necessary to translate the representation of network infrastructure from the clients’ network-modelling software into a form that can be understood both by GIS applications and for by a hydraulic network solver that can be automated for the purposes of performing the optimization. This earlier version used a commercial hydraulic network solver - directly manipulating the network structure within the solver. Extending this software to accommodate the differing modelling packages used by water companies was hampered by differing conventions

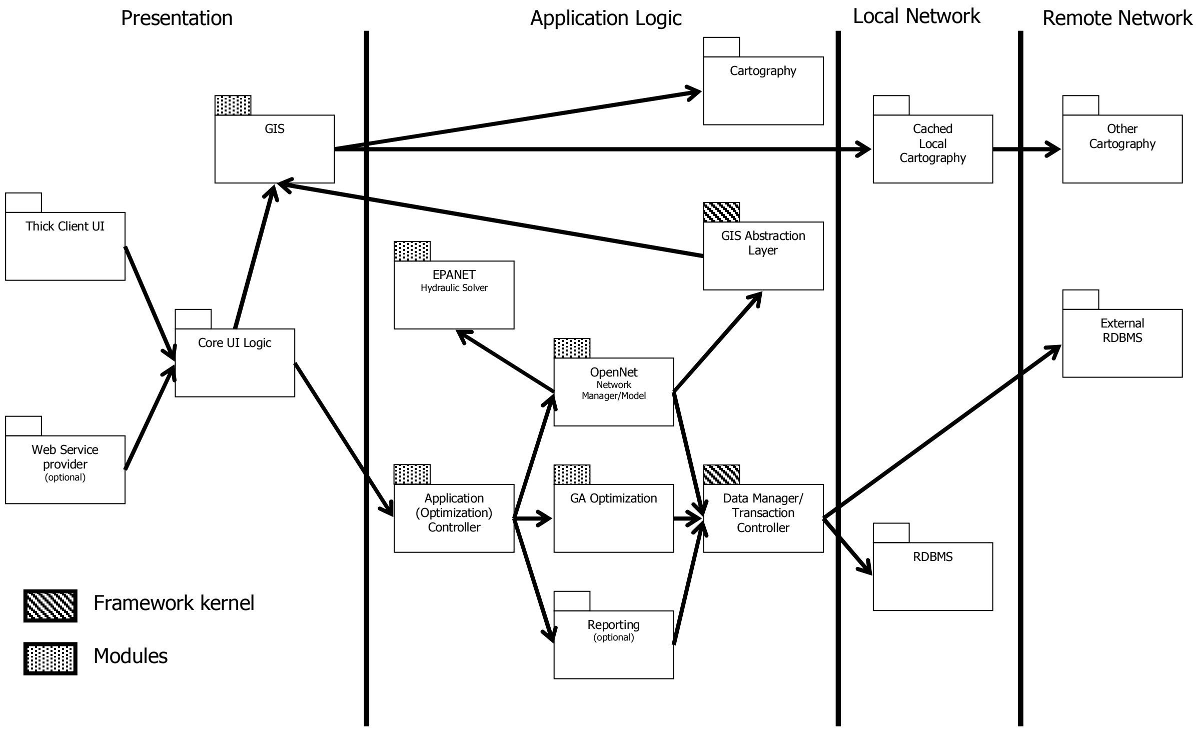

Figure 2: UML Deployment diagram for DSS Framework prototype application

and, in particular, different representations of similar hydraulic elements such as valves. To this end, an object-oriented class-library, known as OpenNet [Morley et al., 2000], has been developed as an abstraction to hide the inner workings of the network solver from the optimisation software. This class library is allied to an XML metafile representation of the network, which is designed to facilitate easier dissemination of network infrastructure data as well as storing the data in a human readable form. For the purposes of integration with the Decision Support System Framework, OpenNet has been completely revised and rewritten to accommodate the meta-data engine used by the Framework that ensures that all properties of the network may be enumerated and accessed by other Modules.

3.3 Hydraulic Network Solver

Data Analyst. A customized version of the EPANET 2 hydraulic solver [Rossman, 1993] has been abstracted and encapsulated in a Windows Dynamic Link Library (DLL). In order that the solver may continue to be used with network model Modules other than OpenNet, a small helper class – not shown in Figure 2 – facilitates the seamless hydraulic solution of an OpenNet pressurised network by automatically reflecting within EPANET, changes made in the OpenNet network.

3.4 Relational Database

Data Provider. In this prototype, the provision of a relational database connection is undertaken to demonstrate the feasibility of such a connection. The data table used by the Decision Support Tool is a simple look up table of available pipe diameters.

3.5 Thematic map

Visualization. A GIS interface component is provided, based on a third party COM object, which allows the visualization of the hydraulic network configuration. As with the hydraulic network solver, a helper class facilitates the seamless data transfer between network and GIS representation. In future the interface between such modules will be further abstracted removing the need for the helper classes.

4. CASE STUDY: NEW YORK TUNNELS (BENCHMARK) PROBLEM

The objective of the New York Tunnel (NYT) problem is to determine the most economically effective design for addition to the existing system of tunnels that constituted the primary water distribution system of the city of New York [Schaake and Lai, 1969]. The twenty-one existing pipes in the network are each considered for duplication to reinforce supply capacity. For each pipe 15 discrete pipe diameter choices available: 36, 48, 60, 72, 84, 96, 108, 120, 132, 144, 156, 168, 180, 192 & 204 inches – along with a “do nothing” option that leaves the existing pipe unduplicated. This gives a solution space of $16^{21} = 1.93\times10^{25}$ possible network design combinations. The minimum available head requirement at all nodes is fixed at 255ft except for nodes 1, 16, 17 that are 300, 260 and 272.8ft respectively. The objective function for the optimization is non-linear: $C_{ij} = 1.1 D_{ij}^{1.24} L_{ij}$ where: C is cost in dollars, D is diameter in inches and L is length in feet. In operation, the Decision Support Tool constructed within the framework, determined the minimum cost option to be $38.80m in line with the result reported by Murphy et al. [1993] and Savic and Walters [1997]. Work to extend the validation by implementing a multiobjective GA analysis of the same problem [Savic, 2002] is underway.

5. CONCLUSIONS

The development of a software Decision Support Framework provides a means of creating Decision Support Tools from a suite of modular building-blocks as part of either an interactive "Generator" application or for use as a more conventional software library. Within the framework additional modules have been developed for innovative optimization and modelling strategies as well as providing the platform for the development of the sustainable urban water tool deliverable from Work Package 6 of the WaND project. This Work Package envisages the development of a GIS-based toolbox to support the development-specific planning and preliminary design of sustainable urban water management practices under different new development scenarios and will serve as one of the main innovative deliverables of this project.

6. ACKNOWLEDGEMENTS

This work is jointly developed by the 'Water Cycle Management for New Developments' (WaND) project [*www.wand.uk.net*] and the "Consortium for Decision Support" (CoDeS) project - both funded under the Engineering & Physical Science Research Council's "Sustainable Urban Environment" Programme by EPSRC, UK government and industrial collaborators.

7. REFERENCES

Atkinson, R.M., M.S. Morley, D.A. Savic and G.A. Walters, GAnet: The Integration of GIS, Network Analysis and Genetic Algorithm Optimization Software for Water Network Analysis, in *Hydroinformatics 98*, V. Babovic and Larsen, L.C. (eds.), Balkema, Rotterdam, 357-362. 1998.

Densham, P. J. Spatial Decision Support Systems. In: *Geographic Information Systems: principles and application*, Maguire, D.J., Goodchild, M.S. and D.W. Rhind (eds.), 403-412, Longman, Harlow, U.K., 1991.

Goldberg, A. and D. Robson, *Smalltalk-80: The Language*, Addison-Wesley, Reading, U.S.A., 1989.

Makropoulos C., D. Butler and C. Maksimovic, A Fuzzy Logic Spatial Decision Support System for Urban Water Management. *Journal of Water Resources Planning and Management - ASCE*, *129*(1), 69-78, 2003.

Makropoulos, C. *Spatial Decision Support for Urban Water Management*. PhD Thesis, Imperial College London, UK, 2003.

Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs. 3rd edition*, Springer-Verlag, 387pp, New York, 1996.

Morley, M.S., R.M. Atkinson, D.A. Savic and G.A. Walters, GAnet: Genetic algorithm platform for pipe network optimisation, *Advances in Engineering Software*, *32*(6), 467-475, 2001.

Morley, M.S., R.M. Atkinson, D.A. Savic and G.A. Walters, OpenNet: An application-independent framework for hydraulic network representation, manipulation and dissemination, 4th Hydroinformatics Conference of the IAHR, Iowa City, U.S.A., 2000.

Murphy, L.J., A.R.Simpson and G.C.Dandy, *Pipe Network Optimization using an Improved Genetic Algorithm*, Research Report R109, Department of Civil and Environmental Engineering, University of Adelaide, 51pp, 1993.

Rossman, L.A. *EPANET User's Manual*. United States Environmental Protection Agency, Cincinnati, U.S.A., 1993.

Savic, D.A., Single-objective vs. Multiobjective Optimisation for Integrated Decision Support, In: *Integrated Assessment and Decision Support*, Rizzoli, A.E. and A.J. Jakeman (eds.), Proceedings of the First Biennial Meeting of the International Environmental Modelling and Software Society, Lugano, Switzerland, *1*,7-12. 2002.

Savic, D.A. and G.A. Walters, Genetic Algorithms for the Least-cost Design of Water Distribution Networks, *Journal of Water Resources Planning and Management - ASCE*, *123*(2), 67-77, 1997.

Schaake, J. and D. Lai, *Linear Programming and Dynamic Programming Applications to Water Distribution Network Design,* Report 116, Department of Civil Engineering, MIT, 66pp, Cambridge, U.S.A. 1969.

Walters, G.A. and D.A. Savic, Optimal Design of Water Systems Using Genetic Algorithms and Other Evolution Programs, keynote paper in *Hydraulic Engineering Software V Vol. 1: Water Resources and Distribution*, W.R. Blain and K.L. Katsifarakis (eds.), Computational Mechanics Publications, Southampton, UK, 19-26, 1994.