ODEFIX JAVA FRAMEWORK FOR DEVELOPING AND INTERFACING HYDROLOGICAL AND WATER MANAGEMENT MODELS - GENERIC COMPONENTS AND APPLICATION FOR WATER RESOURCES ALLOCATION

JEAN-CHRISTOPHE POUGET
UMR G-EAU / IRD - Maison des Sciences de l'Eau, BP 64501, Cedex 5
Montpellier, 34394, France

FABIEN BOUSQUET, PATRICK LE GOULVEN, DAVID QUARANTA, JEAN-CLAUDE REBATEL, DAVID ROLLAND
UMR G-EAU / IRD - Maison des Sciences de l'Eau, BP 64501, Cedex 5
Montpellier, 34394, France

OdefiX is a generic framework which provides Java software components to allow codeveloping and interfacing modeling environments for Integrated Water Resources Management decision support. OdefiX proposes generic interfaces. The goal is to be able to concentrate on models structures and functionalities and not on graphical user interfaces or data exchanges. The user interface allows to define models as structured objects and to facilitate navigation between these objects. The objects are visualized under various facets (description, definition, validity, results) as tables, charts or maps. The OdefiX core classes - composite definitions, associations, source data - are presented through the object-oriented design of HyD2006, the new version of HyD2002 modeling environment for testing water resources allocation.

INTRODUCTION

The water resources management covers various aspects (technical, economic, political and participative regulations), which can relate to 3 components: (a) the river basins, the bio-physical environment conditioning water resources with its space-time variability; (b) the uses, the human and socio-economic environment conditioning the water demands and their priorities; (c) the hydraulic projects, the infrastructures allowing to mobilize and share water resources. Decision Support Systems (DSS) for river basin planning and management need model integration (Yates et al. [9]). The object-oriented approach constitutes an efficient way to develop DSS and treat this integration (Fedra et al. [2]).

The object-oriented approach aims, through the development of generic components, to an iterative construction of software models which is based on reusability. This corresponds to a systemic approach which regards a system as an organized totality, whose interdependent elements can be defined only the ones compared to the others. The abstract description of the elements - objects - in classes makes it possible to formalize our perception of the systems and the phenomena which proceed there. The functions are
represented as forms of collaboration between the objects within the system. The evolution is supported by the approach of generalization, put into practice by the use of classes inheritance. The integration of static and dynamic aspects facilitates the decomposition and the composition of the studied systems (Meyer [4], Muller [5]).

We adopted the object-oriented approach to develop HyD2002, a modelling environment for testing water resources allocation, notably the strategic management of reservoirs system (Pouget et al. [6]). Through the HyD2002 Java code generalization, we have constructed the OdefiX (Object definition Extension) components to facilitate the development of software modelling platforms, such as: (a) HydroStruct, to study the river basins structuration (Cudennec et al. [1]); (b) ZonAgri, to represent regional agricultural activities (Pouget et al. [8]); (c) HyD2006, to extend allocation tests possibilities.

After some basics on water allocation models, we present the OdefiX user interface. The data structures are then exposed through the object-oriented design of HyD2006.

WATER RESOURCES ALLOCATION MODELS

The allocation models are used as tools to support decisions regarding: (a) planning, i.e. the definition of future projects to satisfy prospected needs and uses; (b) operational management, i.e. management rules to satisfy current objectives of an existing system.

Water projects have generally multi-purpose objectives: e.g. irrigation, water supply, fisheries, recreation, flood control, hydroelectric power, navigation,. These objectives have various priorities. Regarding reservoirs systems, water management implies the definition of strategies to take into account the hydrological variability.

Generic models, like MikeBasin (DHI software), Ribasim (Delft), HyD2002, allow the representation of various types of water projects by the interactive construction of topological diagram of water systems as nodes (reservoirs, consumptive and objective demands, junctions, etc) and hydraulic links (canals, pipes, natural arcs, etc).

Usually, these models allow to simulate the water resources allocation, at a time period varying from month to day, on natural inflows time series regarded as representative of the hydrological variability. The evaluation of system performances is based on intensity-duration-frequency analyses of the result variations compared to the objectives. One can study the frequency of failures occurrences (reliability), the extent of deficits (vulnerability) and the duration of shortages (flexibility).

The generic modelling environments differ by the representation possibilities, in term of objects types and management rules, and by the model calculation algorithms. These calculations are all based on the water balance conservation and the respect of transit constraints, but they can use or not numerical optimization. One may distinguish strategic models, which consist in managing the risk by adapting demands to available resources, and tactical models, which correspond to water distribution. The linking of strategic and tactical models appears significant in order to improve operational management and implies notably to treat explicitly time periods and the corresponding processes (Jamieson [3]).
ODEFIX GRAPHICAL USER INTERFACE

DSS for water resources management have to be easy-to-use, comprehensive tools, notably thanks to a user-friendly, graphically oriented interface (data manipulation, map representations, charts...), in order to support the discussions between specialists and managers (Fedra et al. [2], Yates et al. [9]). Various institutes related to water management developed proprietary software libraries to construct their models and DSS: Delft Tools (cf. www.wldelft.nl/soft/tools), DHI software (cf. www.dhisoftware.com), SEIB software (Yates et al. [9]), WaterWare project (Fedra et al. [2]), etc. Through the HyD2002 Java code generalization, we developed the OdefiX components, based on object-oriented structures, presented within the next section, in order to construct models without new Graphical User Interface (GUI) codes. The goal is to be able to concentrate on model structures and functionalities and not on GUI or data exchanges.

The user interface allows to define models as structured objects and to facilitate navigation between these objects. For instance, with the explorer (see Figure 1.), one can select the reservoirs list of a water resource system and add a new reservoir. This component can be defined with its structural characteristics (label, location, storage curve, inflow, etc) and management properties (objective storage, rule zones, etc).

The interface allows a map representation, with various information layers and at several scales. The use of GIS functionalities, and its integration with traditional water resources models, constitute an obvious strategy for the development of river basin management systems (Fedra et al. [2]). The map corresponds to the most natural
representation to facilitate the dialogues between thematicians and managers, and to synthesize and schematize information.

In the same way, the interface gives access to various temporal data: historical time series (rains, inflows, etc...) and time series related to evolution scenarios which are built. These time series data can be represented as tables, or as temporal charts, and exchanged with spreadsheets and databases.

More generally, each object can be visualized under various facets (description, definition, validity, results) as tables, and possibly as charts or maps. For instance, the water system definition views give access to data table with all components, as complete (flat view) or partial (database view) definitions. The validity view can report the definition problems, as warnings or errors. A water demand, which is not connected, constitutes an error, because one cannot simulate the water system operation. The results view gives access to composite objects resulting from the current definition. The results view of a water system proposes global indicators and references to various simulation results time series. Within the table, one can select an object, like a results chronicle, in order to detail this definition as table or chart. The fluidity of the navigation comes from the homogeneous definition of all objects. The next section details the definition classes.

ODEFIX COMPONENTS

Object architecture
The general architecture, presented in Figure 2, is defined by the regrouping of classes within packages, according the UML unified notation (Muller [5]). The relations between packages define the dependences and determine the stability of the architecture.

To create a new model application, like HyD2006, a new package is defined, with classes which are specializations of generic structures proposed within the odefix.data package. It contains several sub-packages, corresponding to main structures, detailed afterwards. All generic graphical user interface classes are grouped within the odefix.gui package. No new specific class is necessary. All packages depend on Java packages provided by Sun Microsystems (cf. http://java.sun.com) within the Java Development Kit.

Figure 2. General architecture.
Main classes
Within OdefiX, each object is described as a Definition. This class interface guarantees access to various properties (label, description, type, id, etc) and methods related to definition validity, ability to change, to chart, etc. A definition guarantees also to return table models corresponding to the multiple facets of an object seen within the previous section. The definitions are more or less complex according to the type of component. We distinguish 3 main types of definitions which will be detailed subsequently: (a) composite definitions, Composite<D extends Definition>; (b) association definitions, Association<D extends Definition>; (c) basic definitions, Data, which are associated with metadata describing their type. This DataType is itself a Definition.

The modelling environment ModellingSpace (see Figure 3) is the most complete composite definition. It is a specialization of a work space, which allows to handle documents of various natures. A Document is a composite definition which can be stored. Data persistence uses XML (Extensible Meta Language) formalism. In the case of the modelling environment, some documents are also models. A Model corresponds to a system whose operation can be simulated.

For instance, the HyD2006 modelling environment consists of the definition of: (a) a space environment, which refers to a cartographic document; (b) a temporal environment (various time series of inflows, rains, evaporations, etc); (c) global properties (generic priorities of water demands, etc); (d) simulations (simulations of a project state, simulations of evolution scenarios); (e) models, corresponding to references to water resources systems; (f) currently open documents.

![Figure 3. Main classes UML diagram.](image-url)
Composite definitions
We distinguish two kinds of composite definitions, \texttt{Composite\langle D extends Definition\rangle}:
(a) the non extensible ones, \texttt{ListComposite\langle D\rangle} and \texttt{SortedComposite\langle D\rangle}, which correspond to fixed definitions, based on respectively a list and a set of sorted definitions;
(b) the extensible ones, \texttt{XListComposite\langle D\rangle} and \texttt{XSortedComposite\langle D\rangle}. These definitions are normally strictly composite: a definition has only one composite parent, the removal of a composite will entail the removal of all its elements.

Within HyD2006, a water resources system is defined as a \texttt{ModelDocument (Model, because the system operation can be simulated; Document because the definition can be saved)} which corresponds to a predefined set of water components: reservoirs, junctions, consumptive demands, objective demands, artificial arcs, natural arcs, return arcs, etc. A \texttt{WaterSystem} is thus defined as a \texttt{ListComposite\langle WaterComponents\langle \rangle\rangle}.

\texttt{WaterComponents} is a \texttt{XSortedComposite\langle T extends WaterComponent\rangle} which is sorted by component labels. \texttt{Reservoirs}, for instance, corresponds to the composite definition, \texttt{WaterComponents\langle Reservoir\rangle}, in which new reservoirs can be added.

A \texttt{WaterComponent} is defined as a \texttt{ListComposite\langle Definition\rangle}, since structural properties and management properties are distinguished.

Association definitions
The concept of association is very significant within OdefiX. There are two main types of association: (a) \texttt{Reference\langle D extends Definition\rangle} defines a reference to a \texttt{Definition} which already exists within the modelling environment; (b) \texttt{Choice\langle D extends Definition\rangle} corresponds to a \texttt{Definition} which contains a definition chosen among a list of possible definitions. Contrary to \texttt{Reference\langle D\rangle}, the removal of \texttt{Choice\langle D\rangle} will entail the removal of the associated definition.

Within HyD2006, the upstream and downstream nodes of the arcs are defined as \texttt{Reference\langle T extends WaterNode\rangle}, the \texttt{WaterNode} class being a specialization of the \texttt{WaterComponent} class. A \texttt{Choice\langle Definition\rangle} is used to define the storage curve of a reservoir as 3 possible forms (formula of storage according to level, formula of surface according to level, tables of levels-storages-surfaces).

Data definitions and changes mechanisms
A \texttt{Data} definition corresponds to the primitive data of programming languages (boolean, integer, double, string, etc). We distinguish definitions with one value (\texttt{SimpleData}), an array of one dimension (\texttt{Data1D}) and an array of two dimensions (\texttt{Data2D}).

A \texttt{Data} is associated with a \texttt{DataType}, which specifies its basic type and various meta-data (label, description, format, unit, minimum, maximum, etc). The data capture of a out-of-bounds value notifies a warning problem during the validity treatment. This warning may involve a model error if one cannot simulate the model operation.

The problems management is thus related with the changes treatment. The adopted mechanisms for the treatment of definition changes allow to: (a) propagate these changes to involve other changes or updates (linked objects, GUI); (b) manage the date of change
to inform problems managers about validity checkings; (c) keep in list the changes with their properties to propose an undo-redo mechanism; (d) ensure the persistence of changes to manage alternatives of models.

With this previous mechanism, one can explicitly define a new water system as the evolution of existing systems with various additions and modifications. This is particularly significant for planning studies. So one can manage several versions of dependant systems. In the same way, one can define and store simulations with their main features and the list of changes on management properties for certain components.

The possibility to introduce formulas gives flexibility to define numeric values. The formulas can contain basic calculation operators and several functions: stochastic functions (random, normal distribution, etc), conditional functions (if-then-else), references to other data definitions, etc. This is notably interesting to define water management rules or prospected scenarios of agricultural activities (Pouget et al. [8]).

**Time management**
The `odefix.data.time` packages propose several classes to manage time explicitly. The `TimePeriod` class, which inherits of the numeric data class, has several specialized classes (`Date`, `Month`, `Year`, etc). The `SeasonalVariation` class corresponds to regular yearly variations. `RegularChronicle` allows to define a multiannual variation with a specified time period.

This time management object-oriented design facilitates the flexibility of systems (inflows, demands variations, etc) and simulations definition (linking strategic and tactical processes) and the frequency analysis of results time series.

**Map definitions**
The `odefix.map` packages propose several structures to manage 2D data and to offer certain functionalities of a GIS (Geographic Information System), as a representation by layers and thematic views. The graphical styles, defined within legends, are associated to data class with a specific data type. The data classes allow to define partitions in order to identify and analyze data.

The construction of topological diagram of water systems as nodes and hydraulic links is easier and more comprehensive using maps, with several information layers (rivers, cities, irrigation, etc). One can use these representations to analyze the spatial distribution of objectives satisfaction and to compare various simulations and/or projects alternatives. One can use graphical animations on inflows or results time series to analyze the spatio-temporal distribution.

**CONCLUSION**
This paper presents how the OdefiX components facilitate the development of modelling environments, avoiding notably new specific codes for GUI and model scenarios management. Our final goal is to be able to co-construct support decision tools
applications, where several models (allocation, hydrological, bio-physical, socio-economic, etc) can cooperate in order to answer efficiently to river basin problems (Pouget et al. [7]). The object-oriented approach and our generic interface facilitate this model integration. But a great challenge consists in adopting an open solution in order to allow collaboration with other models of various sources (freeware, open-source or proprietary model). To support the Water Framework Directive, the European research HarmonIT project (cf. www.harmonit.org) has developed the open modelling interface (openMI, cf. www.openmi.org), as a standard for model linkage in the water domain. So the next development step will integrate this interface within OdefiX components, in order to develop openMI compliant models.

REFERENCES