7th International Conference on Hydroinformatics
HIC 2006, Nice, FRANCE

MIGRATION OF "DYNAMIC CONTROL" TO JAVA

J-L DELTOUR
Société du Canal de Provence, Le Tholonet CS70064
13182 AIX EN PROVENCE cedex 5,France

O. GENTHON
Société du Canal de Provence, Le Tholonet CS70064
13182 AIX EN PROVENCE cedex 5,France

The Société du Canal de Provence conceived, more than thirty years ago, the "Dynamic Control" software, the theory of the software has been improved through several research programs [1][2][3][4]. This software has been used to control the Canal de Provence since 1970. It has also been implemented on other systems in the world. The migration towards the Java language makes it independent of the operating system and has the advantages of an object-oriented programming approach that reinforces its flexibility.

The article recalls first the basic principles of dynamic regulation. It reviews then the classes of objects that constitute the software. The review of the classes allows to describe the methods used in the software and to put forward the advantages of the object oriented approach. While taking advantage of computer developments achieved on previous projects, the software modules can easily be adapted to the specificities of new projects: modeling new physical classes of objects, adaptation of functional classes of objects…It is thus possible to take into account particular constraints and objectives of each project. Indeed, these are intimately linked to the technical and social context.

The software has been developed to replace the version used currently on the Canal de Provence. It can also be integrated to other systems using SCADA. The migration has been tested firstly in simulation mode. The new software is now controlling the "Canal de Provence".

INTRODUCTION

The Dynamic Control software has been written in Fortran language and run on OpenVms system. Like any other on-field application, in the course of time or at the time of its implementation on other canal sites, this software has been adapted in order to face new situations.

The Canal de Provence is completely automated and user oriented [5][6]. Water users can take the water freely with resorting neither to rotations nor to any sort of priority allocation. Maximum flow and delivery pressure are fixed by contract. This mode of water supply leads, on one hand, to very high water use efficiency (around 85 % between the volume charge to users and the volume taken from the reservoirs on the river). On the other hand it requires an effective water control in order to cope with a/ unscheduled demand and b/ the difficulty of controlling accurately discharge on canals.

A review of about 30 year's experience of automated canal management has been done and resulted in the decision to rewrite the existing software. When choosing the language, it appeared that an object oriented solution would be the most adapted, in order to represent correctly the structured and articulated actions or components of the system. Java was eventually selected as it fulfills the main objectives of the migration:

- Improvement of the software documentation, with an automatic update in case of modification of the sources: the documentation is automatically generated from Javadoc tags included in the software code.
- Great potential for evolution, so as to accept easily modifications and adaptation to a particular canal site: the language introduces an increased modularity through the concept of "Classes". As it is shown hereafter the notion of specialization enables to adapt some part of the software keeping the structure and main behavior of the software.
- Independence of the computer operating system: this main property of java reinforces the software perpetuity.

DYNAMIC CONTROL

The "Dynamic Control" software [1] automatically controls main canal and control structures. The control software performs simultaneously three different actions:

- Anticipatory action (demand forecasting and open loop control) : Depending on the type of off-take, flow forecasting can be generated either from a pre-established program, or by extrapolation of historical trends. This second method combines the discharge measured at the off-take and a running average computed over the preceding ten days.

  The forecasted discharges at check structures are then calculated by introducing a hydraulic delay from the check structure to the various off takes. Target volumes of the different pools are calculated from these forecasted discharges taking into account operating constraints (minimum and maximum water level). Between these two limits, the choice of the target volumes may follow different objectives, such as to minimize the response time of the canal, to minimize level variations, or to optimize the energy cost/benefit at pumping stations or turbines.

- Corrective action (feedback): Pool inflow and outflow cannot be perfectly balanced in practice. This results in volume variations in each pool which have to be counterbalanced by a corrective action. The volume of water contained by the canal is calculated in real-time from field sensors (level, discharge, gate opening) and canal modeling. This volume is compared to the target volume to compute corrective action through a controller (PI controller, pole placement controller, smith predictor or any other type of controller)[5][6].

- Coordination action: The corrective action can be different for two adjacent pools and introduces a discrepancy between inflow and outflow in a pool. Thus, it is recommended to mitigate this imbalance by carrying forward the corrective action from one pool to the other upstream pools. However, this action, called co-ordination action, is not mandatory as any imbalance can be corrected by another corrective action. Coordination actions speed up the control process and help maintain the pool volume closer to their targets.

SOFTWARE ORGANIZATION

An object oriented software is organized into several classes of objects. Each class defines on one hand data or "members" required to describe the object and on the other hand the collection of "methods" which can be used to carry out treatments. The renewed "Dynamic Regulation" is composed of classes which can be spread into 3 main packages:

- Topology package related to physical elements of the system
- Control package related to functional elements that define the operation rules and control logic of the system
- Utilities package for data acquisition and user interface.

We are not going to detail all classes of the application, let us simply choose some classes and describe their main characteristics and implemented functions. This will help the reader in understanding the advantage of java concerning software evolution capability.

The most general class of the software is called "CXMLObject". All other classes inherit from this one. It implements several general functions which are common to all the classes of the software (for example the building of an object through the reading of parameters in a file of XML format). In addition to inheriting behavior from "CXMLObject" specialized classes include new specific functions or data.

Topological package

The Topology is the first notion to be structured. All classes of this package inherit from the "CTopologicObject" class which adds to the "CXMLObject" the possibility of connection to sensors for physical measurement of object status. Below this package most general class, we find:

"Conduct" which inherits from "CTopologicObject" and add the possibility of connection to other "Conducts" located upstream or downstream. A "Conduct" is able to propagate its real discharge from upstream to downstream and its forecasted discharge from downstream to upstream. The way of modeling the propagation is defined in specialized subclasses.

"Segment" is a special type of "Conduct" dedicated to represent the basic geometrical canal element. It represents a linear section of canal. This class adds the possibility of calculation of volume of water in the canal and possesses eventually "Off/In Takes" modeled through a new class family.

"Control Structure" is a special type of "Conduct" which represents a structure on the canal through which the system is able to measure and adjust the discharge. If a more detailed description of the "Control Structure" is needed, it can be divided into several "Passes" as "Gates" or "Pumps".

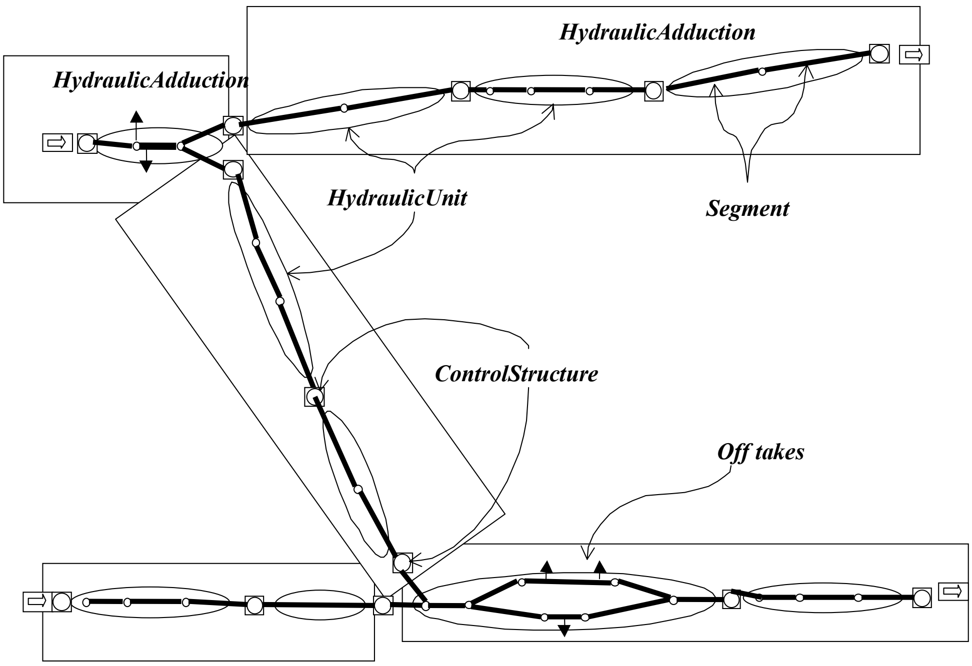

Figure 1. Topological graph of a canal system

A “CRamification” is required if a “Conduct” is connected to more than one other “Conduct” upstream or downstream.

Control package

The basic class of this package is the “HydraulicUnit” which is a specialized “CXMLObject” made up of “Segments” and limited upstream and downstream by “ControlStructures”. The “HydraulicUnit” controls the status of its segments through action on upstream “ControlStructures”. This class includes functions required to implement operational rules and control logic. Initially two different “HydraulicUnit” had been developed:

- The “ControlledVolumeHydraulicUnit” is able to maintain the volume of water in the canal close to a target volume which depends on canal discharge.
- The “PumpingStationOptimizationHydraulicUnit” manages the stored volume so as to reduce pumping costs (minimizes pumping duration during day time and maximizes it during night time).

The possibility of specialization offered by the java language enabled us to easily enhance the capability of the software by adding:

- “ControlledLevelHydraulicUnit” to control a level on one hydraulic unit.
- “ControlledDischargeHydraulicUnit” to control the discharge at the downstream end of the hydraulic unit.

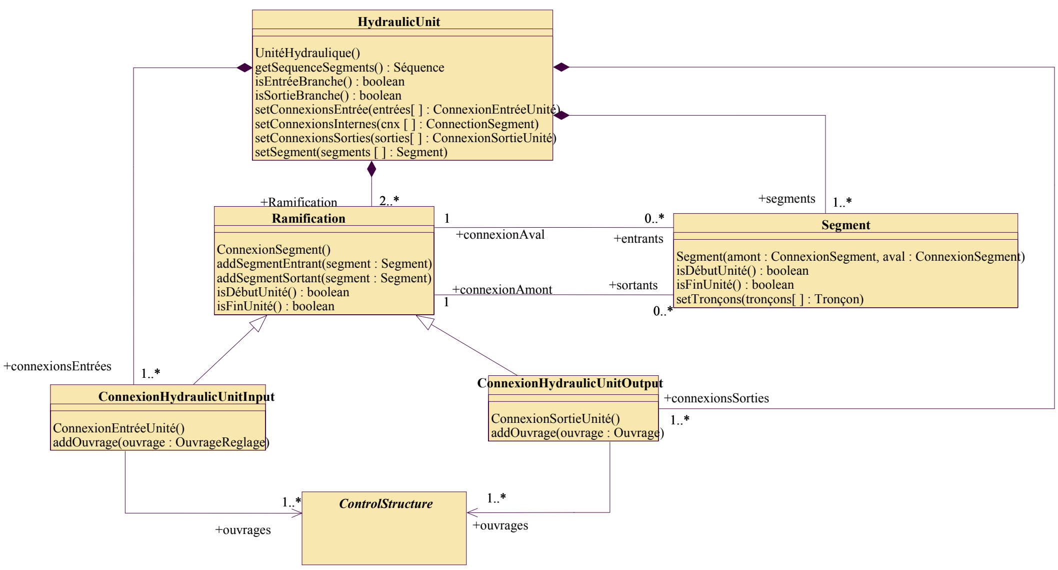

Figure 2. The hydraulic unit object

“HydraulicAdductions” are composed of several synchronized “HydraulicUnits”. This class gives the opportunity to share deficits or excess among several “HydraulicUnits” or to implement a full MIMO controller (Multiple Input Multiple Output controller).

Utilities package

The utilities package consist of

- Classes devoted to data acquisition as “CommunicationLink” through which the software can be connected to supervisory software. The software offers now two mode of communication: exchange of files or communication through sockets. One “CommunicationLink” manages several “Sensors” or “Commands” which are able to get/send information from/to field equipment.

- Classes devoted to user interface as
    - “IHM” devoted to graphical interfaces to parameter the software and adapt it to the system (create and link objects) or to modify operational parameters (set points, mode of operation).
    - “DataLogging” to memorize all events and actions performed by the software.
    - “ControlReport” which creates at each control time step one report presenting system status and commands sent to the field.

IMPLEMENTATION ON CANAL DE PROVENCE

The migration has been undertaken in three steps:

- We chose the language in 2002 on the basis of a pilot project during which the main part of the architecture was defined by using the Unified Modeling Language (UML) [9] method, and the compatibility between java constraints and the project tested.
- The software migration take actually place from 2003 to 2004, the development was performed by a software company (TRANSICIEL) with a tight collaboration with SCP staff .
- In 2005, the “Canal de Provence” system has been numerically modeled using Sic software [10]. This CEMAGREF software can simulate canals behavior with all operational structures, off takes and perturbations. The control software has been tested with the numerical model playing the role of the canal. We tested several operational scenarios (predicted or unpredicted modifications of demand, failures of sensors …) before operating the software on the “Canal de Provence” itself in September.

The software is running in real-time since that time, and has faced successfully several operational situations like:

- Discrepancy between forecasted and actual demand
- Failure of communication links
- Failure of sensors
- Operation of the canal in specific conditions for maintenance works.

The two graphs hereafter compare the target volume and the measured volume on one reach of canal, during a 3.2 m^3/s discharge increasing. In the first case the discharge was withdrawn as scheduled, in the second case it was taken with a 3 hours delay. It can be seen that the software behave rightly in both cases.

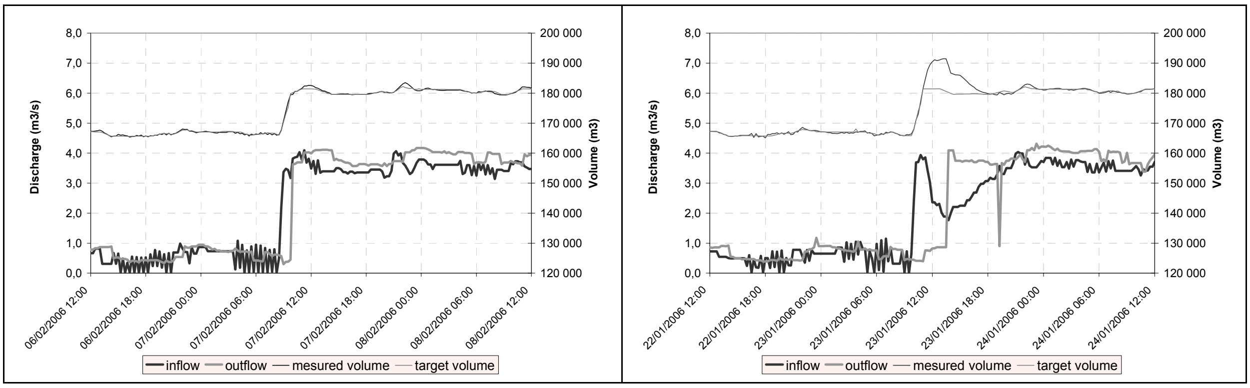

Figure 3. Reaction of the software to predicted and unpredicted variations

CONCLUSION

We have described the logic of the application. It can be noted that the general structure is rational, easy to maintain and to adapt to particular canal sites. The advantages of the structure appear at least at two levels:

- in the software itself which is structured from general classes to specialized ones (this avoid to develop the same behavior several times depending on the component of the software it refers to),
- Along the time it is possible to adapt the behavior of a class and create new classes, limiting the modifications to what is strictly required and taking advantage of the parent class.

These possibilities are linked to a well conducted initial analysis phase, required to define the appropriate structure of the program. This analysis also allows saving development time when evolutions are to be implemented, and to reinforce the reliability of the new development as it is mainly based on already tested parts.

It can also be noted that the controller while being essential is a well isolated and small part of the whole software. It can be interchanged, and, for the same canal, it can be different from one hydraulic unit to another. At the present time, a library of controllers exist which includes PI,PID, Pole Placement three order controller, Smith Predictors, Internal Model approach. The success of the implementation of this software on the "Canal de Provence" required several safety procedure to be able to operate in real-time even in case of external devices failures (communication links, sensors..). User interfaces are also essential to avoid any error in parameters input.

REFERENCES

[1] Rogier D., Coeuret C., Brémond J., "Dynamic Regulation on the canal de Provence", *Proc. . Planning, Operation, rehabilitation and automation of irrigation water delivery systems"*, Portland, Oregon, (July 1987), –pp 180-191.

[2] Deltour J-L, "Application de l'automatique numérique à la regulation des canaux (proposition d'une méthodologie d'étude)", PhD thesis, Institut Mécanique de Grenoble, (1992)

[3] Sanfilippo F., "Application de l'Automatique multivariable à la régulation des canaux – comparaison avec une approche monovariable", PhD thesis, Université Claude Bernard – Lyon I, (1997)

[4] Viala Y., "Intégration d'une commande multivariable pour la régulation des canaux d'irrigation – Application à la branche d'Aix Nord du Canal de Provence", PhD thesis, SCP – CEMAGREF – ENGREF, (2004)

[5] Rousset P., Coeuret C., "La télégestion des canaux d'irrigation en Provence ", 22émes journées de l'hydraulique, Société Hydrotechnique de France

[6] Deltour J-L, "La régulation dynamique du Canal de Provence : un procédé sûr et adapté à la distribution d'eau à la demande ", Revue de l'Elelectricité et de l'Electronique (REE), pp 65-70, (avril 2003)

[7] Deltour J-L, Sanfilippo F., "Introduction of Smith Predictor into Dynamic Regulation ", ASCE Journal of Irrigation and Drainage engineering, 124(1):pp47-55, (Jan/Feb 1987)

[8] Viala Y., Malaterre P-O., Deltour J-L, Sanfilippo F., Sau J., "A multivariable approach for the command of Canal de Provence Aix Nord water supply subsystem", Proc. Water rights and related water supply issues USCID conference, Salt Lake City, Utah, USA (2004)

[9] Booch G., Jacobson I., Rumbaugh J., *"UML user guide"* . Addison Wesley (1999)

[10] *"Sic, User's guide and Theoretical Concepts"*, Cemagref Publications, (1992)