Chapter 3

WATERNET: Intelligent Supervision and Control in Heterogeneous and Distributed Application

This chapter describes the design and implementation of the Waternet integrated/federated environment, which allows the coupling of distributed, heterogeneous, and autonomous subsystems in the water distribution/management system [ABH 98a, ABH 98b]. The Waternet ESPRIT IV project (No. 22.186) aims at developing an evolutionary knowledge capture and an information management system. Waternet supports the control, optimal operation, and semi-authorized decision making for drinkable water distribution networks.

The main goal of the Esprit project Waternet is stated as Knowledge Capture for Advanced Supervision of Water Distribution Network. It involves: the development of several different subsystems and their integration into a coherent environment, in which they can easily access and exchange the information they need. In order to support the requirements of advanced distributed control and management of water distribution networks, there is a need to develop a strong interoperable information management system. The interoperable information management system supports the cooperative heterogeneous subsystems with their exchange and management of large amount of data.

The Development of the Waternet information management system is based on the use of the PEER federated system. PEER provides basic means for information integration and interoperation among heterogeneous systems without the need for data redundancy or replication at multiple sites. The information management framework for Waternet, presented in this chapter, is augmented with the development of adapters serving the need for system flexibility and openness. Therefore, adapters facilitate the insertion of new components to the system. Our contribution to the Waternet project consisted of:

- Analysing the requirements of the Waternet subsystems (units) and the design of the database structure and communication mechanisms for their managements of information. Waternet units included: supervision, simulation, optimization, water quality management, and machine learning. These other units in the Waternet system were
developed by other partners of this project. The complete list of partners included: ES-TEC, UNINOVA, SEBETIA, WBE, University of Amsterdam, ADASA, Smas-Sintra, Universitat Politecnica de Catalunya, University of Naples, and ALFAMICRO.

Development a strong integration system for distributed information management to properly support the data exchange and information sharing among the different units of the Waternet system.

Our design and development of the innovative integration architecture and framework achieved within the Waternet project, present an implementation prototype of a federated environment, which allows different subsystems of a typical water company to work in collaboration. Regarding the information management approaches presented in chapter 2, the Waternet framework can be considered as an implementation prototype of the node-to-node federation, in which import/export schemas are well defined and node autonomy is preserved. The designed approach although addressing the Waternet requirements in specific, is generalized enough to be applied to other complex application environments, that involve the interoperability among heterogeneous and autonomous subsystems.

3.1 Introduction

Water supply industries nowadays lack a global overview of the status of the production and the water distribution system. Distinct functionalities required in this industry, e.g. optimization, water quality, etc. are supported by independent, heterogeneous, and autonomous subsystems. Each subsystem performs its specific activity, but their co-working and complex information exchange needs to be properly supported. Typically, there is none or little coordinated control in order to assure a continuous supply of water with a better quality monitoring, minimize the costs of exploitation, meet the quality standards, save energy consumption, optimize pipeline sizes, and reduce wastes. Furthermore, these systems are heterogeneous and of different levels of automation and reliability.

The main focus of the WATERNET project is two-fold: (1) the development of several subsystems performing the necessary functionalities (i.e. the supervision, the simulation, the machine learning, the models manager, the optimization, the remote unit, and the water quality); and (2) the integration of these subsystems into a coherent environment, in which the subsystems can easily access and exchange the information they need from the other subsystems, in order to function properly.

The integration/interoperation architecture designed for WATERNET, involves the development of Distributed Information Management System (DIMS) for every subsystem, that provides all mechanisms necessary for such interactions among the subsystems. Considering the fact that the DIMS plays the role of the interlocutor/integrator among all other subsystems, its implementation must reflect the inter-operation requirements specific to the design of WATERNET system and its subsystems. The two main requirements to be considered involve:

1. The need for provision of the information produced by every subsystem, for access by any other subsystem.

2. The general requirements of "openness and flexibility" to support the WATERNET system life cycle.

Subsystems in Waternet cooperative environment are independent and autonomous modules developed by different individual partners within the community. The best approach
3.2 Water Environment and General application requirements

Water industries today require the cooperation of heterogeneous subsystems (also called units in this document). Each subsystem, e.g. optimization, water quality, simulation, supervision, machine learning, etc. performs a distinct function and their co-working and information exchange are of complicated nature. In principle, a number of activities may
be assumed by every subsystem. Clearly, the number of units and the complexity of every system depend on the size and functionalities of the water industry. In Europe, water industries constitute a wide range, for example as small as a company where all modules run on a single system that is located in the control room of its headquarter, or as large as a water company with geographically distributed control, processing, and distribution sites.

Independent of the size, water companies today lack a global overview of the status of production and of the water distribution network. Control of such systems, is often carried out locally, based on the operators' experiences. Typically, there is none or little coordinated control, that is needed to assure a continuous supply, meet the quality standards, save energy, optimize pipeline sizes and reduce wastes [URB 97].

A good System analysis begins by capturing the requirements of an application, and modeling the essential elements in its environment. To support the requirements of complex applications, such as Waternet, the information management system must deal with heterogeneous sources of information from geographically distributed sites. The interconnection among the cooperating nodes is established through a variety of wide area (WAN) and local area (LAN) networks, in which a node (agent in the community) may need to access, in run time via remote queries, one or several sources of information in other nodes' data sources. In general nodes can be independent and self-serving with a large variety of data that they generate and handle. Therefore, any assumption in the direction of centralization and replication of data or unification of data descriptions (schema) in different nodes is unrealistic. Namely it is preferred to have no global schema or redundant storage of data in the network.

Subsystems involved in water management system are heterogeneous and geographically distributed. In this network, every subsystem constitutes one such unit. Every unit serves a specific function in the integrated system and thus units are intrinsically of different kinds [BA 98b, BAG 98].

Through the analysis of the existing water supply and management networks [BAG 98] we have identified and classified the heterogeneous and distributed water management subsystems into four categories of units, namely: Control Unit, Remote Unit, Auxiliary Unit, and External Unit (see Figure 3.1):

1. The Control Unit performs some central supervision and control of the water supply and distribution system. Usually, under the normal conditions, the Auxiliary Units can only “suggest” certain actions to the Control Unit, but the latter will make the final decision.

2. The Remote Unit represents the concept of a site where the information is gathered from a set of sensors and control devices.
3.2 Water Environment and General application requirements

3. The **Auxiliary Unit** is a unit that complements the work of the Control Unit with other functionalities. Examples of Auxiliary Units include the machine learning unit, simulation unit, optimization unit, and water quality monitoring unit. These units will read the information from the Remote Units and/or the Control Units, and give the proper feedback in terms of certain suggestive actions (commands or parameter modifications) in order to achieve a better performance.

4. The **External Unit** is a unit that typically functions outside the water management system, but needs to be accessed to provide some information and/or services. Examples of external units can include: the geographic information systems that could be provided by some services provider, the companies that can perform water network device maintenance, and the centers that can collect users complaints.

3.2.1 Water Network Structure and Management

Existing systems for control and monitoring of water production and distribution are heterogeneous and of different levels of automation and reliability. In such a cooperative environment, the proper functionality of all subsystems involved in the water control system depends on the sharing and exchange of data with other subsystems. Typically, direct connection between these subsystems is none existent or at best exists as point-to-point, where the exchanged information for example consists of documents, phone calls, and electronic mails. Some earlier publications address the problem of developing an infrastructure and/or mechanisms to support the systematic sharing and exchange of information [URB 97, Wang 97, CQ 97], but the suggested solutions still lack a coherent environment to provide a global overview of the status of water production and the water distribution, an integration strategy for the considered subsystems and their activities, and support for the openness and flexibility requirements [AWH 94].

Figure 3.2 illustrates an example of a real water environment as designed and validated by the Waternet partners. In such an environment, the Waternet units mentioned above (control unit, remote units, and auxiliary units) are interconnected through a communication Intranet network via an Application Program Interface (API) that ensures data exchange and data security. Each unit (node) in the system has the full autonomy on its local data, can export a part of its local information, and can import some information that is exported by other nodes.

Following are the different kinds of subsystems that are considered necessary for the development of the WATERNET system [BAG 98]:

1. **Remote Unit Subsystem:** remote unit represents the concept of a site where the information is gathered from a set of sensors and control devices, and some local control is executed. Every remote unit keeps track of the local information of the site (basically device information, status readings, alarm events, and commands) and is able to handle some local events by itself.

2. **Supervisory Subsystem:** supervisory element performs some central supervision and control of the water supply and distribution system. In some cases, there could be only one supervisory subsystem in the network, but then some level of fault-tolerance needs to be implemented. However, it is also possible to have multiple supervisory systems distributed along the network. Usually, under the normal conditions, only the supervisory system makes the final control decisions in the system to modify the behavior of a Remote Unit. In this case, the other units (e.g. simulation, optimization,
etc. described below) can only "suggest" certain actions to the supervisory system, but the latter will make the final decision at the end.

The functionalities of the supervisory system include:

a- **Planning**: planning is a daily process that allows a supervisor at the control headquarter to define the set of actions (production plans) to be taken for the coming hours, days, or weeks. The production plans are executed at fixed schedules and according to the set of pre-defined steps of each action. Planning also includes the set of parameters to be kept in the network and their range of values. In order to achieve a good water supply, two sets of results, described below, can be used: the water demand forecast generated by the Machine Learning subsystem and the optimized strategy proposed by the optimization subsystem.

b- **Controlling**: controlling is the main regulatory task of the water management network, it focuses on three operations: choosing a plan and making it the practical strategy for remote units, manipulating the devices, and adjustment of set points. Such a process allows to recognize failures in the system, to identify the non-optimized operations, and to take the proper recovery actions.

c- **Monitoring**: monitoring is an important process in water distribution network since it watches at the system by reading values of all devices and checks if everything runs properly (parameters should be within the defined range limits, default values, rules checking for pressure, level, flow, etc.). The monitoring system can monitor all remote units for the company regarding their actual running status and collected information, periodic readings of network devices, alarms, and
the sensor values about the pressure, flow, and quality for all remote units. The monitoring interface supports the browsing of Network Current Status, Historical Data, and Graphic Display of the device information and statistical analysis.

d- **Alarms Handling:** once an alarm is detected it will be presented to the system supervisor at the control room, who shall then make an expert decision on how to react properly. In this case the knowledge/rules extracted by the Machine Learning subsystem (described below) can be used in order to help the supervisor to react properly, taking the suggested rules into consideration [CM 99].

3. **Simulation Subsystem:** simulation assists the operator if he/she decides to look forward in time (e.g. for a few hours) to spot potential problems that can develop if the network is not monitored aggressively. Finding such problems can be supported through the use of the most up-to-date consumption forecasts for the network. In this case, the simulation process looks at what will happen during the next hours, with the goal to spot the eventual problems before they actually develop and occur. A set of simulated network results will be produced by this subsystem and presented to the supervisory system.

4. **Machine Learning Subsystem:** that complements the work of the supervisory subsystem by other functionalities [CM 97]. Two activities are supported by the machine learning subsystem.

   a- The **Knowledge Extractor** process that uses the network model information and the historical data for knowledge and rules extraction, in order to be used by the supervision system for an advanced monitoring of the system.

   b- The **Water Demand Forecasting** process that needs to extract some knowledge to be used in forecasting future water demand, and giving information on how the network is evolving. Its objective is to predict the water consumption for a region of the network in the near future [CM 99].

5. **Optimization Subsystem:** optimization in the water networks refers to the optimized operational strategies for the elements controlling water transfer in the network such as the pumps or valves related to cost, quality, etc. The optimized operational strategies are based on the forecast of future demands over the time horizon using a simplified model of the water network’s dynamic behavior.

6. **Water Quality Subsystem:** quality in the water management comprises a large set of parameters, however the most important are related to the quality of supply (pressure, flow, continuity, etc.) and biological characteristics. The quality monitoring process gets actual values of the sensors for quality measurement from the supervision system and then it generates a list of possible abnormal situations for which a set of alarms will be generated and presented to the supervisor at the control room.

7. **Other External Subsystems:** external subsystems may run outside the water management system, while they are needed to be contacted in order to provide information/services necessary for water distribution. For instance, the geographic information systems and/or the water network maintenance systems that can be contacted by the supervisory subsystem or others, when their information/services are required.
3.3 Information Management Approach

In water supply and distribution network, typically the information about the water characteristics and network devices is gathered in remote units and processed at different stages of network simulation, network behavior learning, strategy optimization, and water quality checking. Furthermore, the proper planning and strategies for water management and processing untreated water are achieved under the supervision of the system supervisor.

Units involved in the water control network (e.g., supervision, simulation, optimization, machine learning, and water quality) function properly if and only if they can access the information produced by other units. Therefore, the sharing and exchange of information among subsystems must be properly supported. While the proper independence and autonomy of the units needs to be also preserved. For instance, the control unit, or an external unit, are autonomous units, while the remote unit has only partial control over its functionality and takes orders from the Control Unit. Similarly, the heterogeneity of information representation in different units and its varied classification needs to be supported. In general, the same piece of information is viewed differently by two units, and different levels of details can be associated with it [URB 97, RPR 94, SP 94]. The database schemas involved in the definition of the Waternet subsystems are described in details [BAG 98].

Some earlier publications have addressed the problem of data representation and information modeling for the operational control of water distribution systems [Wang 97, CQ 97]. However, they mostly lack a comprehensive approach that involves the entire set of components and their activities, and takes into consideration the distribution and evolution of the system. In general, subsystems are independent and self-serving, with a large variety of data that they generate and store. Therefore, any assumption of centralization, replication, or unification of data descriptions in different subsystems (through one global schema) is unrealistic. It is preferred to have no centralized global schema or redundant storage of data within the entire network.

3.3.1 The Waternet Architecture

In order to support the complex information management requirements in water environments and their applications, we have designed a comprehensive architecture for the Waternet system. Within this Architecture, presented in Figure 3.3, the integration among components and their information exchange is clearly defined and represented. Some earlier publications address the problem of data representation and data modeling for the operational control of water distribution systems [Wang 97, CQ 97]. But, they mostly lack a comprehensive approach that involves the entire set components and their activities, and takes into consideration the distribution and evolution of the system.

The information management architecture of the water network illustrated in Figure 3.3 is a general and comprehensive architecture that supports the autonomy and heterogeneity of information representation in all sites involved in the water management. In practice, the case of every water company is different and may require only a subset of this comprehensive and open architecture. Here the purpose is to define an architecture that is capable of handling an advanced federated control network.

The DIMS (Distributed Information Management System) is augmented with every unit in the network in order to support the capability of information sharing with other units in the Waternet network in a transparent way. Therefore, the DIMS layer ensures the runtime access to information stored in other subsystems (via remote queries). The DIMS
information management is supported via the federated schema facilities and the federated query processing of PEER. Moreover, DIMS extends the PEER approach with the data adapters mechanism, in order to better support the information exchange between the PEER system and the specific information systems used at each unit in Waternet (see section 3.5.1).

![Diagram of Information Management Architecture for the Water Network in Terms of Units](image)

The federated schema definition facilities provide the capability of information sharing with other units in a transparent way. This also implies that every such unit can handle different kinds of information: the information which is going to be stored locally, the sharable information for public access, and the information which needs to be imported from other units. Consequently, through the federated query processing, from the user point of view the access to physically distributed information along the network is the same as a local access. In general, the Control Units and Auxiliary Units may retrieve the information from other different sites, and are in some cases able to provide suggestions/strategies to improve the behavior of the Remote Units and Control Units. Every unit will read the information, which is required from the other units, at the time that is needed. Therefore, the information that is accessed from other units is always up-to-date and there is no repetition of information among the units in the network.

### 3.3.2 Simple Scenario for Subsystems interaction

The WATERNET system operation requires a real cooperative environment in terms of the integration and the exchange of information between different subsystems. In order to give the reader an overview on the complexity of the interactions between the WATERNET subsystems, for data sharing and some results validation, a simple scenario for the process involved in developing an optimized strategy is presented in this section.

The cooperative work required to develop an optimized strategy can be considered as "a part of the bigger cooperative environment required every day", to identify many operations to be carried out the next day. As depicted in Figure 3.4, the cooperative process needed for the simple scenario involves the optimization, machine learning, simulation, water quality,
and the supervision subsystems. The steps involved in the execution of the scenario are described below:

- **First**, in order to generate a *management plan* for the next day operation of the network, the *Optimization* asks the *Supervision* for the network devices information; asks the *Machine Learning* for the forecasting results; and asks the *Models Manager* for the operative model. The management plan generated by the optimization subsystem, primarily consists of a sequence of commands to be performed at specific times on the network, e.g. opening a valve at 2 AM, stopping a pump at 5 PM, and so on.

- **Second**, the *Simulation* and the *Water Quality* subsystems are invoked by the optimization. These two subsystems must access the *generated plan* information from the *Optimization* subsystem, perform some processing and give their feedback about the consequences of the *generated plan* on the ability of the system to support the proposed plan and/or how this plan affects the quality of the water.

- **Third**, the *Optimization* subsystem needs to access both the *Simulation* and the *Water Quality* subsystems, in order to check their evaluation results of its earlier generated plan and in order to decide either to recommend the plan to the *Supervision* as an *optimized plan* or to reject it. If the plan is rejected, the whole process described above needs to be restarted to develop and test a new plan. Otherwise, the plan will be approved and presented to the supervisor for acceptance. If accepted by the supervisor, the plan will be loaded at the remote units by the system supervisor at the control room; otherwise it will be canceled and re-planing starts again.

![Figure 3.4: Simple Scenario for Subsystems Interaction in WaterNet](image)

**3.4 Distributed Information Management System (DIMS)**

In the general architecture, as presented in Figure 3.3, every component of the WATERNET system being a remote unit, a control unit, an auxiliary unit, or an external unit, constitutes a PEER node. In principal, one unit can either run on an individual workstation (or PC), or several units can run on the same system. The PEER system and the development of the PEER federated layer for DIMSs are further described in this section.
3.4.1 The PEER Federated Layer

The PEER system provides an environment for the cooperation and information exchange among different nodes in a network, where every node is composed of one server process and may consist of several client processes. The federated schema management and the federated query processing of PEER [AWT+94] support the sharing and exchange of information among nodes, without the need for data redundancy and/or creation of one global schema. Therefore, the problems of data consistency, referential integrity, and updates propagation are eliminated.

The federated schema management of PEER organizes four different kinds of schemas for every node (Figure 3.5). The local data at the node is defined by the schema called LOC. Every node can create other several schemas called exported schemas (EXP), to represent a part of its local schema (LOC); and only authorized users at remote nodes can access data of this node through some EXP schemas. The authorized nodes can import the EXPs of other nodes that will be called imported schemas (IMP). The imported schemas (IMPs) are then merged with the LOC to build the integrated schema (INT) for the node. Hence, every node in the federated community can access both its local and the remote information (from other nodes) through its INT schema, as if all the data is local information. At the same time, the physical and logical distribution of information becomes completely transparent to the users. The four kinds of schemas for the subsystems are defined using the SDDL (Schema Definition and Derivation Language) of PEER [TA 93]. Several examples of these schemas defined for the WATERNET subsystems are included and described in earlier publications [ABH 98b, BAG 98].

![Diagram of PEER Federated Layer](image)

Figure 3.5: PEER Federated Layer Representation

The PEER layer development for every unit, supports the following features:

- Integration and filtering of information accessed from a set distributed units.
- Support for local autonomy and heterogeneity at every unit.
- Access to updated data with no redundancy of stored information.
• Flexible support for potential network expansion with new functional units.
• Differences in data structures, modeling approaches, and objects naming in different nodes are solved through the definition and derivation of the integrated schema.

### 3.4.2 Schemas Management in WATERNET Using PEER

To provide a better idea about the information that is represented and exchanged between PEER nodes, simple examples are provided below. In general, these examples show how different pieces of information that are stored at a given node, can be imported and used by another different node. What is represented in these examples is a small part of the PEER schemas developed for certain units in the implementation of the Waternet system. The definition of the schemas in these figures is based on the use of the Schema Definition and Derivation Language SDDL of PEER [ATW'93. AWT'94].

Table 3.1 shows a part of the LOC schema of the Control Unit CU. In order to export some information from the CU to the auxiliary Units (or in general, to any other unit), one or more export schemas need to be defined at the CU node. The export schema EXP1 defined at CU (Table 3.2) contains network devices information in Nodes (subtypes E_Tank and Reservoir) and Devices (subtypes Pipe, Group, and Valve) types. This information is derived from the LOC schema of CU and due to the preference of the autonomous CU node, it is defined rather differently than in the corresponding LOC schema type definition. For instance, the type Head_Dep(ending) is not exported in EXP1 from the local schema of CU, as well as several attributes such as low_level_alarm and high_level_alarm in Tank and broken in Network_Devices.

```plaintext
define_schema LOC
  type NETWORK_NODES
    code, site_code: STRINGS
    coord_x, coord_y, coord_z: REALS
    outflow: INTEGERS
  type TANK subtype_of NETWORK_NODES
    area: STRINGS
    low_level_alarm, high_level_alarm: REALS
    weir_elevation: INTEGERS
    inflow: REALS
  type RESERVOIR subtype_of NETWORK_NODES
  type HEAD_DEP subtype_of NETWORK_NODES
    coef_discharge: REALS
  type NETWORK_DEVICES
    start_node, end_node: NETWORK_NODES
    discharge: REALS
    broken: BOOLEANS
  type PIPE subtype_of NETWORK_DEVICES
    pipe_length, pipe_diameter: REALS
  type GROUP subtype_of NETWORK_DEVICES
    power: REALS
  type VALVE subtype_of NETWORK_DEVICES
    open_time, close_time: DATE
end_schema LOC
```

Table 3.1: Simple Network Local Schema in Control Unit Node.

Other attributes have been exported, sometimes with different names or under different types (i.e. a subtype). Thus, through the attribute transformation, it is possible to export information with a different representation that is more simplified to better support other
units’ purposes. Table 3.3 and Table 3.4 represent two schemas in the optimization unit, and in a specific its LOC schema and the IMP7 from the Control Unit (CU) respectively.

<table>
<thead>
<tr>
<th>Derive_schema EXP1 From_schema LOC</th>
<th>Define_schema IMP7 same_as_export_schema EXP1 from_agent Control_Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODES = NETWORK.NODES@LOC</td>
<td>type NODES</td>
</tr>
<tr>
<td>code_id : STRINGS</td>
<td>code_id, code_site: STRINGS</td>
</tr>
<tr>
<td>code_site : STRINGS</td>
<td>x, y, z: REALS</td>
</tr>
<tr>
<td>flow : INTEGERS</td>
<td>flow: INTEGERS</td>
</tr>
<tr>
<td>E_TANK subtype_of NODES</td>
<td>E_TANK = TANK@LOC</td>
</tr>
<tr>
<td>hq2 : STRINGS</td>
<td>hq2 = area@LOC</td>
</tr>
<tr>
<td>elevation : INTEGERS</td>
<td>elevation = weir.elevation@LOC</td>
</tr>
<tr>
<td>inflow : REALS</td>
<td>inflow = inflow@LOC</td>
</tr>
<tr>
<td>RES = RESERVOIR@LOC</td>
<td>RES = RESERVOIR@LOC</td>
</tr>
<tr>
<td>DEVICE = NETWORK.DEVICES@LOC</td>
<td>DEVICE = NETWORK.DEVICES@LOC</td>
</tr>
<tr>
<td>start_node : NODES</td>
<td>start_node = start_node@LOC</td>
</tr>
<tr>
<td>end_node : NODES</td>
<td>end_node = end_node@LOC</td>
</tr>
<tr>
<td>discharge : REALS</td>
<td>discharge = discharge@LOC</td>
</tr>
<tr>
<td>PIPE = PIPE@LOC</td>
<td>PIPE = PIPE@LOC</td>
</tr>
<tr>
<td>pipe_length, pipe_diameter : REALS</td>
<td>pipe_length = pipe_length@LOC</td>
</tr>
<tr>
<td>GROUP = GROUP@LOC</td>
<td>GROUP = GROUP@LOC</td>
</tr>
<tr>
<td>power : REALS</td>
<td>power = power@LOC</td>
</tr>
<tr>
<td>VALVE = VALVE@LOC</td>
<td>VALVE = VALVE@LOC</td>
</tr>
<tr>
<td>open_time, close_time : DATE</td>
<td>open_time = open_time@LOC</td>
</tr>
<tr>
<td>close_time = close_time@LOC</td>
<td>close_time = close_time@LOC</td>
</tr>
<tr>
<td>end_schema EXP1</td>
<td>end_schema IMP7</td>
</tr>
</tbody>
</table>

Table 3.2: Simple Network Export Schema (EXP1) in Control Unit Node

<table>
<thead>
<tr>
<th>define_schema LOC</th>
<th>Define_schema IMP7 same_as_export_schema EXP1 from_agent Control_Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>type OPT_VALUE</td>
<td>type NODES</td>
</tr>
<tr>
<td>device_code : STRINGS</td>
<td>code_id, code_site: STRINGS</td>
</tr>
<tr>
<td>site_code : STRINGS</td>
<td>x, y, z: REALS</td>
</tr>
<tr>
<td>min_flowrate : REALS</td>
<td>flow: INTEGERS</td>
</tr>
<tr>
<td>max_flowrate : REALS</td>
<td>elevation: INTEGERS</td>
</tr>
<tr>
<td>min_position : REALS</td>
<td>inflow: REALS</td>
</tr>
<tr>
<td>max_position : REALS</td>
<td>PIPE = PIPE@LOC</td>
</tr>
<tr>
<td>type OPT_PIPE</td>
<td>pipe_length, pipe_diameter: REALS</td>
</tr>
<tr>
<td>opt_coeff : REALS</td>
<td>GROUP = GROUP@LOC</td>
</tr>
<tr>
<td>type OPT_GROUP</td>
<td>power: REALS</td>
</tr>
<tr>
<td>opt_coeff : REALS</td>
<td>open_time, close_time : DATE</td>
</tr>
<tr>
<td>min_pos : REALS</td>
<td>end_schema LOC</td>
</tr>
<tr>
<td>max_pos : REALS</td>
<td>end_schema IMP7</td>
</tr>
<tr>
<td>type OPTIMIZATION_NODE</td>
<td>type NODES</td>
</tr>
<tr>
<td>opt_name : STRINGS</td>
<td>code_id, code_site: STRINGS</td>
</tr>
<tr>
<td>opt_address : STRINGS</td>
<td>x, y, z: REALS</td>
</tr>
<tr>
<td>NODES = NETWORK.NODES@LOC</td>
<td>flow: INTEGERS</td>
</tr>
<tr>
<td>code_id = code@LOC</td>
<td>elevation: INTEGERS</td>
</tr>
<tr>
<td>code_site = site_code@LOC</td>
<td>inflow: REALS</td>
</tr>
<tr>
<td>x = coord_x@LOC</td>
<td>PIPE = PIPE@LOC</td>
</tr>
<tr>
<td>y = coord_y@LOC</td>
<td>pipe_length = pipe_length@LOC</td>
</tr>
<tr>
<td>z = coord_z@LOC</td>
<td>GROUP = GROUP@LOC</td>
</tr>
<tr>
<td>flow = outflow@LOC</td>
<td>power: REALS</td>
</tr>
<tr>
<td>discharge = discharge@LOC</td>
<td>open_time, close_time: DATE</td>
</tr>
<tr>
<td>end_schema EXP1</td>
<td>end_schema IMP7</td>
</tr>
</tbody>
</table>

Table 3.3: Simple Local Schema (LOC) | Table 3.4: Simple Imported Schema (IMP7)
The Optimization Unit will access some up-to-date and current information from the Control Unit through the import schema IMP7. An import schema (IMP) always has the same structure as the definition of its corresponding export schema (EXP) at its origin. For every import schema, the name of the node and the name of the export schema from that node are specified.

Finally, as shown in Table 3.5, the Optimization Unit will define an integrated schema, derived from its local schema and other imported schemas, such as IMP7, using some types and maps derivation operators (e.g., union, restrict, substract, rename, threading, etc., from the SDDL language of PEER). However, for simplicity reasons, here the examples only show the UNION operation. The INT schema represents the proper database view for the optimization applications and optimization programs. Once the integrated schema is defined and created, the user at the Optimization node can formulate his queries against this global and complete schema that represent an overview of all the information accessible from this site. Based on the integrated schema definition, when a query arrives, it will be decomposed into several sub-queries, each related to a different remote node where the needed information is available. The result from different remote queries will then be merged with the local one and then presented to the user as a coherent response for his request.

```
Define_schema INT from LOC, IMP7
Type OPTIMIZATION_NODE
  opt_name : STRING
  opt_address : STRING
Type NODES
  node_id : STRING
  node_site : STRING
  x, y, z : REAL
Type ARCS
  start_node : NODES
  end_node : NODES
  discharge : REAL
Type OPT_VALVE subtype of ARCS
  open_time : DATE
  close_time : DATE
  min_flowrate : REALS
  max_flowrate : REALS
  min_position : REALS
  max_position : REALS
Type OPT_PIPE subtype of ARCS
  pipe_length : REALS
  pipe_diameter : REALS
  opt_coef : REALS
Type OPT_GROUP subtype of ARCS
  power : REALS
  opt_coef : REALS
  min_pos, max_pos : REALS
```

```
OPTIMIZATION_NODE = OPTIMIZATION_NODE@LOC
  opt_name = opt_name@LOC
  opt_address = opt_address@LOC
NODES = UNION (E_TANK@IMP7,RES@IMP7)
  node_id = code_id@IMP7
  node_site = code_site@IMP7
  x = x@IMP7
  y = y@IMP7
  z = z@IMP7
ARCS = UNION (PIPE@IMP7,GR0UP@IMP7,VALVE@IMP7)
  start_node = start_node@IMP7
  end_node = end_node@IMP7
  discharge = discharge@IMP7
OPT_VALUE = UNION (OPT_VALVE@LOC,VALVE@IMP7)
  open_time = open_time@IMP7
  close_time = close_time@IMP7
  min_flowrate = min_flowrate@LOC
  max_flowrate = max_flowrate@LOC
  min_position = min_position@LOC
  max_position = max_position@LOC
OPT PIPE = UNION (OPT PIPE@LOC,PIPE@IMP7)
  pipe_length = pipe_length@IMP7
  pipe_diameter = pipe_diameter@IMP7
  opt_coef = opt_coef@LOC
OPT_GROUP = UNION (OPT GROUP@LOC,GROUP@IMP7)
  power = power@IMP7
  opt_coef = opt_coef@LOC
  min_pos = min_pos@LOC
  max_pos = max_pos@LOC
end_schema INT
```

Table 3.5: The Integrated (INT) Schema in Optimization
3.5 Extended Integration Approach

An important outcome of the DIMS integration approach is that any subsystem in WATERNET can develop its application programs without the need of knowledge about the format, structure, and/or location of the data produced somewhere else in another subsystem.

Figure 3.6 represents the main integration architecture of WATERNET based on PEER, in which each subsystem within the WATERNET is augmented with its DIMS (a PEER-based federated layer). Within the DIMS layer, the information sharing and data exchange is supported through the integrated schemas. Using the provided mechanisms, users and application programs in a subsystem can specify the queries for data retrieval or data insertion through the subsystem’s integrated schemas. The defined queries can be specified online by human users, using the on-line PC-interface, or within an application program using the programming languages-interface. Once a query arrives, it will be decomposed into several sub-queries. The query decomposition is based on the definitions in the integrated schema at the DIMS layer. Each sub-query is then sent to the proper remote subsystem. Finally, the local partial-result for the query is merged with the remote partial-results, and produces the complete result to the query, that will be presented to the end-user and appears just the same as if it was handled completely local at this node. The approach described above allows a complete integration of the data stored in different subsystems in a transparent way, while preserving access security issues, and the execution of concurrent transactions.

![Figure 3.6: Basic Integration Architecture](image)

As represented in Figure 3.6, an application program within a subsystem can then simply receive its input from (and similarly send its output to) the PEER database server. However, considering the specific characteristics that define every application domain, this implementation architecture may not sufficiently support all the requirements within the WATERNET environment. In specific, to support the water management applications and the wide variety of subsystems within the configuration of different water companies, in our requirement analysis stage we have identified the need for a more “open and flexible” architecture [BAG 98]. For instance, from time to time different subsystems (mostly pre-existing and some commercial, e.g. new simulation software) may need to be added to (or removed from) the WATERNET system, in order to better support the specific needs of the company. Even as a product, some of the existing subsystems may need to be disconnected from WATERNET, and/or replaced by other existing or new commercial products that run in the company. Using the federated architecture and approach as described above, these alterations within the subsystems require that subsystem developers must have database
language expertise to properly add/remove/replace the subsystems to the federated architecture. For instance, the knowledge of PEER database language commands is mandatory, to generate the appropriate PEER commands to be included within any application program written in a subsystem in order to develop the interaction between a new subsystem and its DIMS. A similar problem arises when a unit decides to use as input (for its application programs) some other resources outside the DIMSs that may be available from external applications or databases. Hence, there is a need for an open and flexible integration architecture. Under the influence of this “openness” requirement, we have extended the integration architecture of the DIMS to also include the “Adapter” (or data adapters) components, described in section 3.5.1.

This extensible integration approach supports the system flexibility and application programs modularity for the WATERNET subsystems. The extended approach, as depicted in Figure 3.7, through (a) preserves the main properties for cooperative working in multi-agent environment, such as the data-location transparency, access security, transactions concurrency, etc. (similar to the architecture described in section 3.4), but additionally through (b) with the adapters, supports the openness requirement. Among other features, the adapters support adding/removing new subsystems within the WATERNET system that can be developed independently from the WATERNET project. Using the adapters, an application can receive its input either from the remote DIMS or from external application. Similarly, the generated output (in addition to storing it in DIMS) if needed can also be stored locally in a storage facility (or another simple database system) and made available to external applications that may not even be allowed to access and retrieve information stored in different WATERNET DIMSs. Clearly, within the WATERNET system, the data of a subsystem stored within its DIMS can always be accessed by other WATERNET subsystems through the DIMS to DIMS interconnections.

![Figure 3.7: Extended Integration Architecture](image)

The adapter framework supports the following:

a- Provides the storage of the exact output of the application programs in every subsystem within its DIMS layer. In fact, a module called pre-processor takes the output of every subsystem's program in the exact format that it is produced, (being a set of values, or a record, etc.), reformats it according to the object definitions in the subsystem’s LOC schema and stores it in the DIMS.
b- Supports every subsystem’s (e.g. supervision’s) access to the data stored in other subsystems (e.g. optimization, machine learning, and others) in the exact format that is required by every application program. In fact, a module called post-processor provides access to imported data through the DIMS for every application program but changing its format to the exact format as desired to be read by the required application program (e.g. supervision’s).

Therefore, the pre-processor and post-processor (in Figure 3.8) together provide the access to and from the PEER database for every application program in every subsystem. This mechanism in turn supports the modularity and autonomy of nodes within the cooperative community, while also supporting their desired specific application-program-dependent input/output formats for data.

### 3.5.1 Data Adapters Supporting Openness

The Adapter framework, as represented in Figure 3.8, provides flexibility and openness, and facilities for the development of application programs. In other words, the programs can read/write their data in the most convenient way to them. For every subsystem the adapters constitute a set of dual pre-processor and post-processor components, where each pair supports the input/output of one of its application programs.

![Figure 3.8: DIMS Layer – Federated Data Process using Adapters](image_url)

Considering the above clarification, the DIMS integration architecture makes the development process of every subsystem (as well as adjustment to other environment configurations) very convenient, as it proved itself in practice during the development phase in the WATERNET project. Namely, every subsystem developed its application programs completely independent of the others, and it was enough to just specify to the DIMS developer the desired format for the input and output of those programs and not being concerned with how this data is produced by others. For instance, a program in machine learning subsystem produces as output "a file" for which the record format represents: "$r_1, r_2, r_3, r_4, r_5, r_6$". Meanwhile, for instance, a program in the simulation subsystem, reads its input from "a file" with the record format: "$r_3, r_5, n_7, r_2, r_1, d_1$", while here the "$r_3" need to be imported from the DIMS of the machine learning subsystem, "$n_7" needs to be imported from optimization subsystem, and "$d_1" is a computation result using different imported and local values. At the last stage the imported information and other values need to be re-arranged according to the record format required by the simulation application program.
3.5.2 The WATERNET System Implementation

The architecture designed for the WATERNET system is comprehensive enough to support different possible implementation strategies adopted in water companies. Namely, it can support a wide range of companies. For instance, it can support on one hand the case of a small water company where all modules of the WATERNET system run on a single system in the control room at the headquarter, and the remote units only send their collected data to this headquarter. At the same time, it can support a medium to large size water company with many geographically distributed control sites, even if different modules of the WATERNET system, for instance, the forecasting, machine learning, and water quality management each run on different sites and are connected only through the communication network. The PEER federated database system [TA 93], is used as the base for the implementation of the information management in the WATERNET project and supports the communication and interoperability among these subsystems. However, the PEER system was extended to better adjust to both to the specificities of the water management environment and the specific development strategies of different subsystems in WATERNET.

Some extensions enhanced the portability of the PEER federated system. For example, the development of two interfaces: the on-line PC interface and the programming languages interface for PEER [BA 98b]. Considering the facts that PEER is Unix based, while most WATERNET subsystems are developed and run on PCs, the on-line PC interface developed for PEER, allows a user to interact with any database within the cooperative community in order to check, retrieve or update the information for which he/she has gained the appropriate access rights. The Programming Languages Interface includes the necessary functions that allow programmers to develop their own programs, while interfacing with PEER through several different applications programs written in C, C++, Pascal, etc.

3.6 Conclusion and Discussion

In this chapter, a general approach for the design of an open and flexible architecture for the integration between different WATERNET system units, and the mechanisms used for their implementation were presented. The implementation of the designed architecture for the WATERNET framework is based on the PEER federated information management system, since it properly supports the cooperation and information exchange among different nodes involved in an intelligent cooperative environment. To better support the "openness and flexible" requirements in water management environments, the implementation architecture of the DIMS was extended to include the adapter framework. In addition to the main properties provided by the PEER federated layer in the DIMS implementation, the extensions with adapters provide among other features: (1) support for the systems expansion (addition, removal, or replacement of subsystems), (2) the adjustment to subsystems evolution (new/modified application programs), (3) the use of external media (resources from external application) as the input information, and (4) the storage of generated output in a different media, in order to be made available to external applications that may not even be allowed to access the information stored in different DIMSs within the community.

The development of the Waternet project has provided a good environment for implementing a prototype of a federated environment. In this environment an open architecture for distributed/federated information management system was designed. The development of the federated DIMS system for Waternet was based on the detailed study and analysis of the water network production environment. The designed architecture and the approach de-
scribed in this chapter, or a substantial part of it, can be applied to any other manufacturing and production application domain, in which several heterogeneous and some autonomous nodes need to cooperate and exchange their information.

### 3.6.1 Major Characteristics and Benefits of Federated Approach in Waternet

The federated schema management and federated query processing mechanisms of PEER, in addition to the adapters framework presented in the sections above, provide a flexible, and an open environment for the development of a strong water management system. Following characteristics resulted in this environment represent the major benefits gained from the approach taken in the project for the design and implementation of the DIMS.

- **System openness**, so that different modules can be added to/removed from the WATERNET system, as needed, in order to support the specificities of different water companies. This characteristic strongly supports WATERNET as a flexible product, since in order to install the WATERNET system in a company, some of its subsystems may need to be disconnected from this product, and/or replaced by other existing products that already run in the company.

- **No need for the development of a single global schema for all Waternet subsystems** (being centralized or distributed).

- **No need for data redundancy/duplication among the subsystems** (no data transmission unless needed). As a result, the problems of data consistency, referential integrity, and update propagations are eliminated.

- **Complete transparency of logical/physical distribution of information among the nodes in the network**, to the end user.

- **Retrieved data is always accessed directly from its origin and as a result it is always up to date.**

- **The WATERNET development environment has become totally flexible.** In fact, all subsystems continued developing their functionalities and application programs, while simultaneously the gradual and dynamic development of the DIMS adjusted itself to their extensions and modifications.

### 3.6.2 Contribution to GFI$_2$S

The designed architecture and the implemented approach described in this chapter can be applied to many other cooperative environments, in which several heterogeneous nodes need to interact and exchange their information. Several of the developed aspects and lessons learned, during the design and implementation of the Waternet system, are in fact conforming the base for the design of the GFI$_2$S system described in chapter 6. In specific, the DIMS architecture of Waternet contributes to GFI$_2$S at two levels:

- **At the local system level of GFI$_2$S, we adopted the approach of developing data adapters of Waternet and developed the Local Adaptation Layer component (LAL) serving the system openness.** In Waternet, similar to many other systems, the role of an adapter is to adapt and translate data from one application to another. In GFI$_2$S, we take this approach, but further extend it such that the role of an adapter is to
also include the local resources specifications. Such specifications include among other features the access rights to the data. When a query is submitted by an external user/application, the Local Adaptation Layer (LAL) of GFI₂S only delivers the data to authenticated users. In addition, data is delivered according to the local specifications for users access rights and their corresponding visibility levels. Therefore, the LAL in GFI₂S controls the access to the local system and preserves its autonomy (see section 6.2.1).

- At the Node Federation Layer (NFL) of GFI₂S, we adopted the federated schema management of Waternet (local, import, export, and integrated schemas), in tackling the fundamental schema management challenges. Additionally, in GFI₂S, we extend the definition of schemas by augmenting by the specification of the federated resources and the semantic descriptions. Such specifications provide a better understanding of the exchanged information and facilitate the development of schema integration mechanisms (see section 6.2.2).

Furthermore, we must also admit to the fact that during the development of different federation components of the Waternet system, we faced two major obstacles, which made the development of the system requiring a lot of efforts. On one hand, the knowledge and expertise of the PEER system was needed at the development phase of the various Waternet subsystems. On the other hand, development of the adapter components was specific to every legacy system used at the components. Therefore, along with the development of the Waternet System, for each new module, adapters needed to be developed.

In chapter 6, these two issues will be addressed in the design and development of the integration architecture of GFI₂S.

- To solve the first issue, related to specific languages, we suggest the use of standard languages for data definition and information access (e.g. ODL, SQL, and OQL). These standard languages are widely understood by a large community, thus, less effort will be needed when defining federated schemas and accessing data through them.

- To solve the second issue, we use middleware and standard solutions for information exchange, and communication protocols, serving the need and requirements of openness and extensions. The use of middleware solutions play an important role in reducing the number of intermediate tools, unifying the access to shared information, and facilitating the interoperation process among heterogeneous units.