The Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility and the Virtual Enterprise

Proceedings of the Tenth International IFIP WG 5.2/5.3 Conference
PROLAMAT 98

2

Cooperative Information Management for Distributed Production Nodes[1]

Hamideh Afsarmanesh, Ammar Benabdelkader, and L. O. Hertzberger
University of Amsterdam, WINS - Department of Computer Science, Kruislaan 403 1098 SJ Amsterdam, The Netherlands
Phone: +31 20 525 75 12, Fax: +31 20 525 74 90
E-Mail: {hamideh, ammar, bob}@wins.uva.nl

Abstract

Advanced manufacturing environments nowadays involve a number of cooperating heterogeneous nodes, where each node supports a distinct activity, and their joint efforts and information exchange is of a complicated nature. There is a need to develop a strong interoperable distributed information management system to support these nodes with their information exchange and the management of large amount of data. Interestingly enough, different complex production environments share similar complexities. For instance, in production of potable water, that is the application described in this paper, similar to other production environments, there is a prominent need for coordinated control in order to achieve good product. Therefore, the information management strategies suitable for this production application can also properly support other applications. Existing systems for control and/or monitoring of the water production and water distribution are heterogeneous and of different levels of automation and reliability. The distributed/federated information management framework described in the paper supports the cooperation and information exchange among different nodes, and the activities involved in an intelligent water production network. Although, this paper mainly addresses the required interoperability and information management issues

[1] The work described here has been partially supported by the ESPRIT project 22.186 - WATERNET whose partners are: ESTEC, UNINOVA, SEBETIA, WBE, Univ. of Amsterdam, ADASA, Smas-Sintra, Univ. Politecnica de Catalunya, ALFAMICRO, and Univ. of Naples.

for the water production application, the approach described in this paper or a substantial part of it can be applied to any other manufacturing applications.

Keywords
Cooperative Databases, Modelling in Multi-Agent Environment, Water Production Networks, Federated/Distributed databases, and Schema Integration

1 INTRODUCTION

Manufacturing industries today require the cooperation of heterogeneous nodes. Each node performs a distinct activity and their co-working and information exchange are complex. What constitutes a node, depends on the level of complexity considered for the activity of the nodes, and the architecture defined for the cooperation network of the nodes. In principle, a wide range of activities may be considered to define one node, that can for example be as small as one CIM activity within an enterprise, or as large as the entire collection of activities performed by an enterprise within a Virtual Enterprise.

From the information management point of view, in order to support the requirements of advanced manufacturing environments, there is a need to develop a strong interoperable distributed/federated information management system to support cooperative heterogeneous nodes with their exchange and management of large amount of data.

Complex production environments share similar complexities. In today's water supply industries for instance as well as other production environments, there is a lack of global overview of the status of the production and of the water distribution system. Control of such systems is often carried out locally, based on the operators' experience. Typically, there is none or little coordinated control in order to assure a continuous supply, meet the quality standards, save energy, optimize pipeline sizes and reduce wastes.

This paper describes the example of water production, in which the information about the water characteristics, and production network status, are gathered in the remote units and studied at different stages of simulation, optimization, and quality check. Further, the proper plan and strategies for production and processing of untreated water are taken under the supervision of the control unit, and performed at the production plants. Existing systems for control and/or monitoring of the water production and water distribution are heterogeneous and of different levels of automation and reliability. As a result, these systems cannot be easily linked together in order to extract the needed integrated information (H. Afsarmanesh et al. 1997-b). The PEER federated information management system, as briefly described in the paper, provides a framework to support the cooperation and

information exchange among different sites, and for the different activities involved in an intelligent water production network (H. Afsarmanesh et al. 93; Tuijnman and Afsarmanesh 93).

In order to provide a better view of how the information is represented and exchanged between PEER nodes, several examples are provided in the paper. These examples show how the information that is stored for a given subsystem, can be imported and used by another subsystem within the cooperating network.

The remainder of this paper is organized as follow. Section 2 addresses the required interoperability and information management issues for heterogeneous and distributed applications. In Section 3 the analysis and design of the distributed water supply network is presented. Section 4 illustrates the information exchange and management of an intelligent water control network. A summary of the PEER federated integration system as well as a high level description of the WATERNET distributed architecture are provided in Sections 5 and 6. Section 7 describes, through several examples, the mechanisms used for implementing the WATERNET framework. Finally Section 7 concludes the paper.

2 MULTI AGENT ENVIRONMENT AND GENERAL APPLICATION REQUIREMENTS

A good System analysis begins by capturing the requirements of an application, and modelling the essential elements in its environment. To support the complex requirements of multi-agent environments and applications, the database systems 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' databases. In general nodes can be independent and self-serving with a large variety of data that they generate and handle (Camarinha-Matos 1997-b). Therefore, any assumption 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.

* The information accessible within a node can constitute the local, exported, imported, and integrated information that is further described in section 5 of this paper
* Within a query the user specifies his request through the integrated schema. Based on the definition of integrated schema, the query will be decomposed by the federated system to several sub queries each will be send to the proper remote database, then the local result will be merged with the remote results and presented as the final coherent response to the user request.

* For the imported/exported information, only the meta data (schema) should be stored locally at each node
* The exchange of information can be done using two processes:
  a) Free exchange of information, in which an agent is authorised to access the total information available on a remote node, in this case the complete local data is specified as exported information
  b) A limited exchange of information, in which at each remote node export schemas are defined in order to describe the parts of information that this node wishes to share with other agents(an export schema represents a part of the local information)

3 WATER CONTROL NETWORK

To support the complex network of components involved in the water management, a reference architecture needs to be developed. Within this Architecture the integration among components and their information exchange must be clearly represented. Some earlier publications address the problem of data representation and data modeling for the operational control of water distribution systems (Wang 1997; and Cembrano 1997), 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. In water supply and distribution network, typically the information is gathered, processed, and stored in geographically distributed sites called ***UNITS*** (Figure 1). In this network, every node and/or subsystem, e.g. supervision, simulation, optimization, water quality, machine learning, and remote unit, constitutes one such unit. Every unit serves a specific function in the integrated system and thus units are intrinsically of

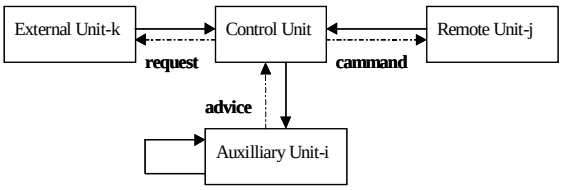

Figure 1: Logical units for the system architecture

different kinds (A. Benabdelkader et al. 1998; H. Afsarmanesh et al. 1998):

- ***Remote Unit:*** The Remote Units represent 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.

- ***Control Unit*** that performs some central supervision and control of the water supply and distribution system. The Control Unit functionalities includes storage and log of all sensored data, status polling, command sending, and alarm handling, in relation to the existing Remote Units. In some cases, there could be only one Control Unit in the network, but then some level of fault-tolerance needs to be implemented. However, it is also possible to have multiple Control Units distributed along the network. Usually, under the normal conditions, only the Control Unit makes the final control decisions in the system to modify the behavior of a Remote Unit. In this case, the Auxiliary Units can only ``***suggest***'' certain actions to the Control Unit, but the latter will make the final decision at the end
- ***Auxiliary Unit:*** The Auxiliary Units are units that complement the work of the Control Unit by other functionalities (some monitoring, tuning, and optimization of different functionalities of the system). 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 actions (commands or parameter modifications) in order to achieve a better performance.
- ***External Unit,*** which manages the information that is related to, but external to the water management system. For instance, the geographic and network maintenance information that could be provided by other information and service providers, for instance when requested by the Control Unit.

4 INFORMATION SHARING AND EXCHANGE AMONG NODES

The proper functionality of all units involved in the water control system depends on the *sharing* and *exchange* of data with other units (Figure 2). At the same time, the autonomy of some units needs to be preserved. For instance, the Control Unit is an autonomous unit, 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 nodes 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 (Spaccapietra, Parent 1994; M. P. Reddy et al. 1994).

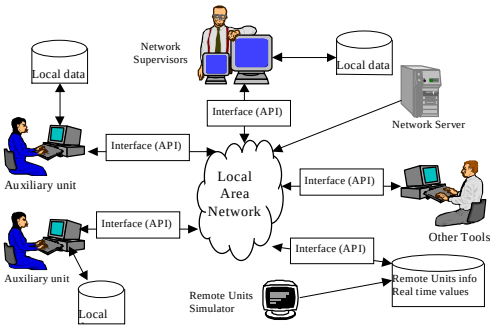

Figure 2: Water Management Environment

Figure 2 illustrates an example of a real water company in which all 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.

5 PEER FEDERATED INTEGRATION SYSTEM

To properly support the distributed information management among these units, each unit is augmented with a PEER layer that provides 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 (Figure 3): 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, using the PEER federated schema definition facilities. Consequently, through the federated query processing of PEER, from the users 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.
The ***PEER*** (H. Afsarmanesh 1993) federated information management system developed at the University of Amsterdam supports the management, and sharing and exchange of information in a network of loosely/tightly coupled nodes. Using

PEER, each node in the federation network can autonomously decide about the information that it locally manages, and which part of its local information it wishes to export and share with other nodes. Each node can import information that is exported by other nodes and then transform, derive and integrate (a part of) the imported information to fit its interest and corresponds to the local interpretation. PEER is a pure federated system; namely there is no need for a single global schema to be defined on the information to be shared by different nodes in the network, and there is no global control among the nodes.

The PEER integration infrastructure helps the human users in a cooperative team, by supporting their information integration at different levels of granularity, e.g. to support the global task, or among different activities and sub-activities.

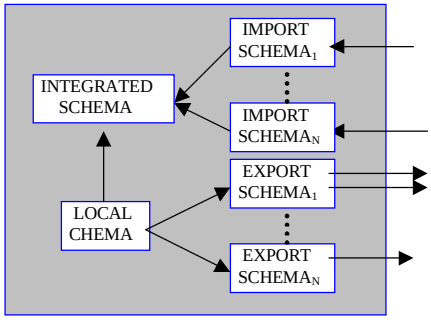

Figure 3: PEER schemas representation

In the PEER layer of a node, the information is then structured and defined by several kinds of `schemas' (Figure 3). The user needs to concentrate on:

1. The information that is available and stored locally in the node (LOC schema),
2. The information that the node needs to access remotely and so the node imports it (IMP schemas),
3. The information that is available locally and other nodes need to access, so the node exports it (EXP schemas),
4. The integration of the LOC information with the IMP information to create a coherent pool of information (INT schema) that is needed to be accessed by this node.

6 INFORMATION MANAGEMENT - USING PEER

This section describes the design of the distributed information management architecture, the modeling formalisms, and the information sharing mechanisms for the nodes (units) involved in water management. This design is based on the specific study and analysis of this application environment.

The architecture described here is a general 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.

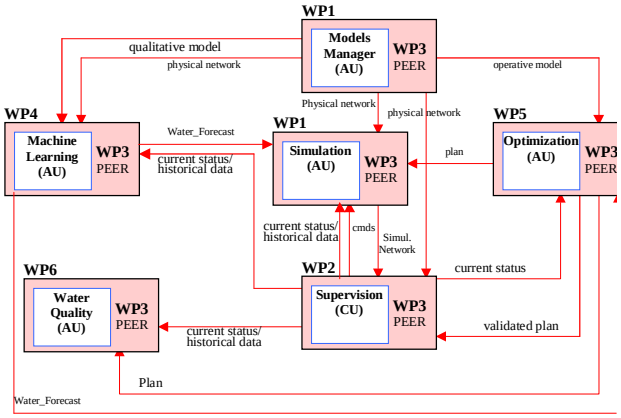

Figure 4: The water network architecture in terms of units

The ***PEER*** layer (also called 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 (H. Afsarmanesh et al. 1998).

The water cooperation architecture can be described using these units as its components. Figure 4 shows how the different units developed in the water production system can be modeled. The flow of information between the components is defined as export/import among units.

Through the PEER layer development for every unit, the following features are supported in the system:

- Integration and filtering of distributed data
- Support for local autonomy and heterogeneity
- Access to updated data with no redundancy of stored information
- Flexible support for potential network expansion with new functional units
- Resolution of differences in data structures, modeling approach, and naming in different nodes

7 IMPLEMENTATION APPROACH

In the general architecture of water production network, every component of this system, being a remote unit, a control unit, an auxiliary unit, or an extended unit constitutes a PEER node. In principal, one unit can either run on an individual

workstation or a PC, also several units can run on the same system (A. Benabdelkader 98).

The designed architecture and the interface tools that we have developed for the water management system are comprehensive enough to support different possible implementation strategies adopted in different water companies. Namely, it can support a wide range of companies. For instance, the case of a small water company can be properly supported, where all modules of the water distribution system run on a single system in the control room at the head quarter, and the remote units send all their collected data to this head quarter. At the same time, a medium to large size Water Company with many geographically distributed control sites can also be supported (A. Benabdelkader 98; H. Afsarmanesh 98).

At the development stage of the PEER layers, during the design of the schemas, the information managed by a node as well as its interfaces to other nodes is made explicit. However, once the PEER layer is developed, through the integration facility of PEER, its distributed schema management, and its distributed query processing, the physical distribution of information and the heterogeneous information representations among different nodes becomes totally transparent to the users.

Information integration in PEER is supported by a declarative specification using the PEER Schema Definition and Derivation Language called SDDL (H. Afsarmanesh et al. 1994). The SDDL language supports the integration, derivation, interrelation, and transformation of types, attributes, and relationships from the nodes' source schemas.

A prototype implementation of the PEER system is developed at the University of Amsterdam in the C language on SUN workstations and includes two user friendly interface tools, a Schema Manipulation Tool "**SMT**" and a Database Browsing Tool "**DBT**" (H. Afsarmanesh et al. 1994).

```
define_schema LOC

type NETWORK _NODES
        code_: STRINGS
        site_code_: STRINGS
        coord_x: REALS
        coord_y: REALS
        coord_z: REALS
        outflow: INTEGERS
type TANK subtype_of  NETWORK _NODES
        area: STRINGS
        low_level_alarm: REALS
        heigh_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: NETWORK _NODES
        end_node: NETWORK _NODES
        discharge: REALS
        broken: BOOLEANS
type PIPE subtype_of  NETWORK _DEVICES
        pipe_length: REALS
        pipe_diameter: REALS
type GROUP subtype_of  NETWORK _DEVICES
        power: REALS
type VALVE subtype_of  NETWORK _DEVICES
        open_time: DATE
        close_time: DATE

end_schema LOC
```

Figure 5-a: Simple network local schema (LOC) in "Control Unit" node.

The support of PC platform was not provided by the existing implementation of PEER. Due to the fact that most of the actual Waternet systems run on the PCs, the access to the PEER system from the PCs is required. Trough the development of two tools, the access from the PC platform to the PEER system (running on the SUN workstation) is supported. The PC Interface and The C interface are both developed in C and in Windows environment, and use the TCP/IP communication protocols (A. Benabdelkader et al. 98). The PC interface consists of a friendly and interactive user interface that supports both the low level and high level PEER commands to create types and attributes, add members (instances), and retrieve data. The C interface consists of a Dynamic Link Library (DLL) that supports the inclusion of PEER commands within other application programs (C, Delphi, Pascal, etc.) trough the TCP/IP protocols, it helps to modularize applications and supports the portability.

To provide a better idea about the information that is represented and exchanged between PEER nodes, a simple example is provided in Figures 5, 6, and 7. In general, these figures show how different pieces of information that is stored at a given node, can be imported and used by another different node. What is represented in these figures 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 Schema Definition and Derivation Language SDDL of PEER (H. Afsarmanesh 94).
Figure 5-a shows the LOC schema of the Control Unit CU. In order to export some information from 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 (Figure 5-b), contains network devices information in NODES (subtypes E_TANK and RES) 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.

```
Derive_schema EXP1 from_schema LOC
type NODES
        code_id: STRINGS
        code_site: STRINGS
        x: REALS
        y: REALS
        z: REALS
        flow: INTEGERS
type E_TANK subtype_of NODES
        hq2: STRINGS
        elevation: INTEGERS
        inflow: REALS
type RES subtype_of NODES
type DEVICES
        start_node: NODES
        end_node: NODES
        discharge: REALS
type PIPE subtype_of  DEVICES
        pipe_length: REALS
        pipe_diameter: REALS
type GROUP subtype_of  DEVICES
        power: REALS
type VALVE subtype_of  DEVICES
        open_time: DATE
        close_time: DATE
```

```
derivetion_specification
-------------------------------
NODES  = NETWORK_NODES@LOC
        code_id  = code_@LOC
        code_site  = site_code_@LOC
        x = coord_x@LOC
        y = coord_y@LOC
        z = coord_z@LOC
        flow = outflow@LOC
E_TANK = TANK@LOC
        hq2 = area@LOC
        elevation= weir_elevation @LOC
        inflow= inflow@LOC
RES = RESERVOIR@LOC
DEVICES = NETWORK _DEVICES@LOC
        start_node = start_node@LOC
        end_node = end_node@LOC
        discharge = discharge@LOC
PIPE = PIPE@LOC
        pipe_length = pipe_length@LOC
        pipe_diameter = pipe_diameter@LOC
GROUP = GROUP@LOC
        Power = power@LOC
VALVE = VALVE@LOC
        open_time = open_time@LOC
        close_time = close_time@LOC
end_schema EXP1
```

Figure 5-b: Simple network Export schema (EXP1) in "Control_Unit" node

```
define_schema LOC

type OPT_VALVE
        device_code: STRINGS
        site_code: STRINGS
        min_flowrate: REALS
        max_flowrate: REALS
        min_position: REALS
        max_position: REALS
type OPT_PIPE
        opt_coeff: REALS
type OPT_GROUP
        opt_coef: REALS
        min_pos: REALS
        max_pos: REALS
type OPTIMIZATION_NODE
        opt_name: STRINGS
        opt_address: STRINGS
```

Figure 6-a: Simple local schema (LOC) in Optimization node

```
Define_schema IMP7 same_as_export_schema
EXP1 from_agent Control_Unit
type NODES
        code_id: STRINGS
        code_site: STRINGS
        x: REALS
        y: REALS
        z: REALS
        flow: INTEGERS
type E_TANK subtype_of NODES
        hq2: STRINGS
        elevation: INTEGERS
        inflow: REALS
type RES subtype_of NODES
type DEVICES
        start_node: NODES
        end_node: NODES
        discharge: REALS
type PIPE subtype_of  DEVICES
        pipe_length: REALS
        pipe_diameter: REALS
type GROUP subtype_of  DEVICES
        power: REALS
type VALVE subtype_of  DEVICES
        open_time: DATE
        close_time: DATE
```

Figure 6-b: Simple imported schema (IMP7) in optimization

Other attributes have been exported, sometimes with different names or under different types (i.e. a subtype). Thus, through the attributes transformation, it is possible to export information with a different representation that is more simplified to better support other units' purposes. Figures 6-a and 6-b represent two schemas in the optimization unit, and in a specific its LOC schema and the IMP7 from the Control Unit (CU) respectively. Thus, 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.

```
define_schema INT from LOC, IMP7

type OPTIMIZATION_NODE
        opt_name: STRINGS
        opt_address: STRINGS
type NODES
        node_id: STRINGS
        node_site: STRINGS
        x: REALS
        y: REALS
        z: REALS
type ARCS
        start_node: NODES
        end_node: NODES
        discharge: REALS
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_coeff: REALS
type OPT_GROUP subtype_of ARCS
        power: REALS
        opt_coef: REALS
        min_pos: REALS
        max_pos: REALS

derivation_specification
-----------------------------------------------
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, GROUP@IMP7 ,
VALVE@IMP7)
        start_node = start_node@IMP7
        end_node = end_node@IMP7
        discharge = discharge@IMP7
OPT_VALVE = 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_coeff = opt_coeff@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
```

Figure 7: The INTegrated (INT) schema in Optimizaton

Finally, as shown in Figure 7, 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.

8 CONCLUSIONS

In this paper an open architecture for distributed/federated information management system was described. The design 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 described in this paper, 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. With the implementation architecture of the DIMS system, a small company can be properly supported, where all the modules run on a single system. While at the same time, a medium to large size company with many geographically distributed control sites can be supported. The PEER distributed/federated information management system was described and the rule of the PEER layer, its schema management, and query processing were briefly represented through the illustration of several schemas in different nodes of the Waternet network. These examples show how the information that is stored in a given subsystem such as the Control unit can be exported to other subsystems, even if they apply heterogeneous information representations. Further, it was illustrated how a second subsystem such as the Optimization can integrate the imported information with its local information, and therefore access both the local and remote information in a transparent way.

9 REFERENCES

A. Benabdelkader, H. Afsarmanesh (1998) PC and C interface to PEER. Technical Report Waternet-UvA-T3.2/98. Dept. of Computer science, university of Amsterdam, 1998.

B. Ulanicki, J. P. Rance, P. L. M. Bounds (1997) A systematic approach to Information integration in water company: International Conference on Computing and Control for the Water Industry, September 97.

C. M. Eastman, H Assal, T. Jeng - Structure of A Product Database supporting Model Evolution, CIB 180 August 95, USA

C. G. Wang (1997) Object-oriented modelling for the operational control of water distribution system: International Conference on Computing and Control for the Water Industry, September 97.

G. Cembrano, J.Quevedo (97) Optimization in water networks: the international Conference on Computing and Control for the Water Industry, Sept. 1997.

F. Tuijnman, and Afsarmanesh (1993) Management of shared data in Federated Cooperative PEER Environment. International Journal of Intelligent and Cooperative Information Systems (IJICIS), 2(4): 451-473.

H. Afsarmanesh, A. Benabdelkader, Cesar Garita (1998) DIMS schema description for modules in Waternet System. Technical Report Waternet-UvA-T3.1/98. Dept. of Computer science, university of Amsterdam, 1998.

H. Afsarmanesh; L.M. Camarinha-Matos (1997-a) - Federated Information Management for Cooperative Virtual Organizations, Proc. DEXA'97, 8th Int. Conf. on Databases and Expert Systems (LNCS1308, Springer Verlag), Toulouse, France, Sept 97.

H. Afsarmanesh; L.M. Camarinha-Matos, F. Martinelli (1997-b) - Federated Information Knowledge integration and Machine Learning in water distribution networks, In Proceedings of the IFIP/IEEE/OE International Conference on Integrated and Sustainable Industrial Production ISIP'97, pages 121-140, Chapman and Hall publication, Lisbon, Portugal, May 1997.

H. Afsarmanesh; M. Wiedijk (1994) The PEER user interface tools manual. Technical ReportCS-94-15, Dept. of Comp. Sys, Univ. of Amsterdam, Sep 1994.

H. Afsarmanesh; F. Tuijnman; M. Wiedijk; L.O. Hertzberger (1993) - Distributed Schema Management in a Cooperation Network of Autonomous Agents. Proc. of the 4th Int. Conf. on Database and Expert Systems Applications (DEXA'93), LNCS 720, Springer-Verlag, Sept 93.

L.M. Camarinha-Matos; H. Afsarmanesh (1998)- Cooperative Systems Challenges in Virtual Enterprises: Computing Engineering in Systems Applications, Tunisia, April 98.

Luis Manuel Camarinha-Matos, Fernando Martinelli (1997-a) Application of Machine Learning in Water Distribution Networks: An Initial Study, Workshop on Machine Learning Application in the real world; Methodological Aspects and Implications, Nashville, USA, July, 1997.

L.M. Camarinha-Matos; H. Afsarmanesh; C. Garita; C. Lima (1997-b) - Towards an Architecture for Virtual Enterprises, Proc. of the 2nd World Congress on Intelligent Manufacturing Processes & Systems, Budapest, Hungary, June 10-13, 1997.

M. P. Reddy, B. E. Prasad, P. G. Reddy, and Amar Gupta (1994) A methodology for Integration of Heterogeneous Databases, IEEE transactions on Knowledge and data Engineering, vol. 6, no. 6, December 94.

S. Spaccapietra, Christine Parent (1994) View Integration: A Step Forward in solving structural Conflics, IEEE transactions on Knowledge and data Engineering, vol. 6, no. 2, April 94.

10 BIOGRAPHIES

Hamideh Afsarmanesh is an assistant professor at the University of Amsterdam in the Netherlands. She has been involved and has directed the research in several European (ESPRIT, and DUTCH-HPCN) and American funded projects. At the WINS faculty, she coordinates the research in the area of Cooperative and Federated Databases, and Interoperable information management systems. She has served as the Program Chairperson in International Conferences and Workshops in the area of information management and expert systems.

Ammar Benabdelkader is a PhD student at the University of Amsterdam, Netherlands, since February 1997. He is working in the area of Information Management Architecture to support Multi-Agent Distributed application domains. His research focuses on the design and prototypical development of the

information management systems. In specific, the modelling constructs, and the mechanisms to support the tasks of supervision and distributed control, within the boundaries of the ESPRIT - Waternet project 22.186.