CHAPTER 2

Information systems for distributed production

Bernd E. Hirsch, Zbigniew K. Marciniak, Thorsten Kuhlmann
BIBA Bremen Institute of Industrial Technology and Applied Work Science at the University of Bremen - Germany

The authors intention is to demonstrate how the principles of the federated database systems and the mediator approach can be applied for mastering the software systems integration problems in the distributed production management. This chapter gives an insight into the Institute's internal research on the integration mechanisms for tailoring autonomous and heterogeneous systems in a multi-site production environment. This research combines results of several CEC ESPRIT and national projects and is focused on integration of already existing, independent software systems - mostly large commercial products. An integration approach chosen is based on both tightly and loosely coupled federations with mediators as building blocks for virtual interfaces. As a data exchange mechanism a Common Object Request Broker Architecture is applied.

2.1. INTRODUCTION AND PROBLEM ADDRESSED

Nowadays the amount of stored data is becoming extremely voluminous, therefore an acceptable quantity of alternative choices should be presented in a clear and compact form to the end-user. This means, the existing data has to be pre-processed towards goals relevant for the decision-support.

However, there are many different aspects which have to be considered while making a decision. The required information[1] is stored in various places (geographical distribution), different kind of sources (heterogeneous databases

1 We understand the information as syntactically and semantically interpretable data.

R. Soenen et al. (eds.), *Advanced CAD/CAM Systems*

and/or applications) are involved, and - most important - the information representation is different, too. There will be mismatches of data caused by different levels of abstraction. In many cases, the processing of these mismatches can be made in an algorithmic way but sometimes sophisticated AI mechanisms will be needed for this purpose (logical relevance is present but the ontology of data sources is very different; e.g. PPC systems and CAD in design dept.). There may also be the case that a human decision is required or even where the matching is not possible.

This multiple-paradigm for data processing (algorithmic processing, AI modules, human-machine conversation etc.) leads to a partitioned architecture of a complex, i.e. including many heterogeneous data sources, information systems. The partitioning of large and huge integration systems is also the only way to keep them under control and to maintain them. The availability of those modularized systems is higher as well.

Currently, new techniques make this approach more attractive than ever before: parallel computing and neural networks. Given, the problem can be broken down into small basic, logically independent operations. The usage of parallel computers will bring a significant performance growth. Also some kinds of neural networks may help to e.g. automatically recognize the optimal compositions of virtual interfaces.

This chapter deals with distributed production management systems in which the information logistics is nearly the most important factor for success. The chapter addresses complications arising during implementation of CIM techniques and novel production organization paradigms (e.g. lean production) in a distributed manufacturing environment [HIR 91]. The most important is to support multi-site production by improving the information flow within an enterprise as well as within the entire consortium. In order to face the demand of customer-specific product versions, it is necessary to support customer participation in the product development process.

The above-mentioned generic problems require flexible business and manufacturing process integration mechanisms for tailoring relevant but geographically distributed and strongly heterogeneous information systems.

The main objective of the chapter is to design an integration system reference architecture for distributed CIM applications in a distributed production environment. This architecture should allow the connection of existing large-scale software systems, which can be implemented in organizational, technological and technical areas (e.g. bid preparation, PPC, CAD, CNC, SFC, after-sale support etc.) of various enterprises. The tailoring platform should support unlimited information exchange, providing data distribution transparency and systems autonomy to the users of integrated systems/applications. This will create required IT facilities to perform the concurrent and simultaneous engineering in a lean production organization profile.

2.2. CONTEXT AND BACKGROUND

Nowadays, industrial enterprises face new challenges as a result of market globalization for industrial products. In some branches e.g. heavy industry (like shipbuilding or steel production) this globalization process started in the mid sixties, but now almost every product competes with numerous equivalents coming from abroad.

Therefore, more and more enterprises suffer under growing product development costs caused by increasing product complexities, demand for decreased maintenance costs and increased product quality and reliability. Due to additional customers requirements like rapidly reduced delivery times and the requirement of sophisticated after-sale services, the overall situation is becoming critical. Hence, many enterprises establish production consortiums in order to minimize time-to-market, R&D and manufacturing costs and to maximize production flexibility. This helps in the above-mentioned areas but creates new problems in the information logistics.

Since the new information management technologies have been applied, the advantages of the proper usage of these techniques have become obvious. Quick response to market requirements now often depends on the ability to perform deep rearrangements in production processes including the subcontractors involved. The questions on how to perform the successful transfer of information, when and where it is required, and - above all - what information has to be exchanged are the real challenge.

With respect to the above-mentioned information logistics problem in intra- as well as in inter-organizational structures, BIBA was and is involved in a number of ESPRIT projects focusing on different internal and external integration aspects. Based on the ideas and results of these projects, BIBA develops its own integration approach of information systems for distributed production. Below an overview of four main projects.

2.2.1. MARITIME

Objective of the ESPRIT Project 6041 MARITIME (Modeling And Reuse of Information over TIME) is to establish an IT architecture for the integration and coordination of projects working with time dependent information used during the whole life cycle of complex and multifunctional products in particular ships. This IT architecture is primarily based on concurrent neutral product data definition methodology (ISO standardization project STEP) improving the sharing, exchange and reuse of product data within distributed, temporary consortia of individuals and organizations participating in the engineering process [MAR 92].

Technically the project pursues the following individual goals:

1 Develop a life cycle product type model for maritime products focusing on those elements of information which are exchanged between different organizations during the product life cycle.

2 Develop methodology for extracting relevant subsets from neutral product

databases in order to provide the transfer of digital information between dissimilar systems, based on different systems architectures.
3 Develop a prototype presentation component to accompany neutral product data exchange.
4 Develop an Object Based Architecture and Information Modeling Environment in order to support system integration and coordination. Furthermore, this allows an information base evolving dynamically during the product life-cycle, without any concern about which organization is handling the information at any given time.

The logical MARITIME integration approach is shown by figure 1. There exist three types of elements [MAR 93]:
1 The applications
2 The information repository and its models. The models used to illustrate the information form the basis for defining the contents of the repository. The repository contains only entry points for the local/native data according to the object structure model.
3 The integration services which control and distribute requests/replies to support data sharing and exchange and tie applications and the Information Repository together.

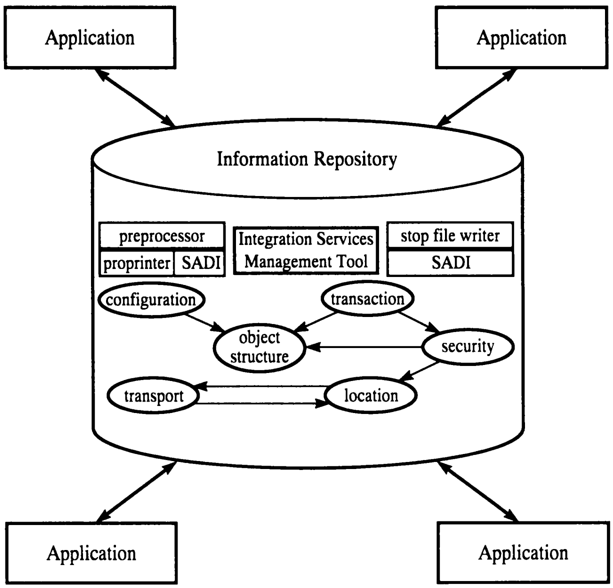

Figure 1. Maritime integration approach

Whereas the modelling work is almost closed and has led to the definition of several application protocols in STEP, the software development of the integration services is at the beginning.

2.2.2. DECOR

The main objective of the ESPRIT project 8486 DECOR (DEcentralized and COllaborative Production Management via Enterprise Modeling and Method Reuse) is to create an integrated toolbox for the development of multi-level, distributed, hierarchical and autonomous decision making systems in the domain of production management. The aim is not to develop new production management applications but to integrate existing solutions and apply them within a problem-solving framework allowing their distributed coordination across multiple software and hardware platforms. The coordination framework itself will provide a solution along both the computational (i.e. the processing level) and the behavioral (i.e. the problem solving level) level of distributed management [DEC 93].

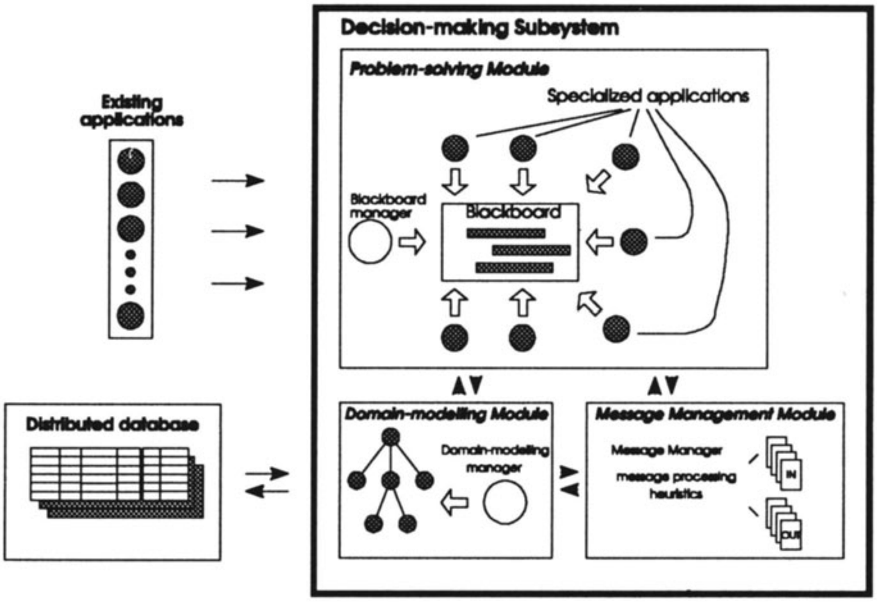

Figure 2. DECOR integration solution

The package of software tools for the construction of DECOR-like distributed management systems will be a layered and modular collection of object-oriented tools. Categorically, the tools should support the implementation of three classes of management knowledge:

1 Domain knowledge covering the representation of all the physical and conceptual entities in the factory (Domain Modeling Module),

2 Problem-solving knowledge defining the actual production management procedures and problem-solving rules (Problem Solving Module), and

3 Communication knowledge establishing coordination links within a distributed set of loosely coupled decision-making subsystems (Message Management Module).

The PSM uses the blackboard framework as a control metaphor. The blackboard approach has its power in organizing problem-solving by means of multiple independent knowledge modules, possibly based on different problem solving methods, and knowledge representation techniques. The blackboard model is generally structured in such manner that it supports state-of-the-art, opportunistic production management methodologies based on multiple knowledge sources as well as more monolithic methodologies. The model constitutes a framework which supports integration of heterogeneous production management software.

Message communication and the interpretation of messages form a high-level problem-solving protocol. The specification of this protocol is open, to allow customization to end-users' needs, future growth, and heterogeneity in the management network. The lower level of the network control problems consisting of the adaptation of a reliable message transmission mechanism including widely accepted low-level protocol. The high level problem solving protocol will be based on the structure and syntax of the message standard EDIFACT. The DECOR problem-solving message mechanism is able to respond to the following questions: When to send messages to other nodes; what are the contents of the messages and how to react to incoming messages.

The DMM models the underlying production data and knowledge relevant to the particular DMS in question and also acts as an interface to a distributed database which stores this information. This means the DMM provides a rich, "deep" model of the underlying manufacturing facilities and processes. It combines a multilevel representation of the objects and their relations in the decision domain with an active representation of the various constraints on these objects. The DMM delivers both the MMM and the PSM with the required domain knowledge.

Since the DECOR project started in December 1993 conceptual work has done exclusively at the moment. First prototypes of each module will be available at the end of this year.

2.2.3. MUSYK

The main objective of the ESPRIT project 6391 MUSYK (Integrated MUlti-level Planning and Control SYstem for One-of-a-Kind Production) is the development of an integrated concept for project/production planning and control. This covers the various control levels within a CIME environment and within multi-site production facilities. Apart from developing different coordination modules to support the planning, control and harmonization of engineering and production activities, the approach integrates the information archive as a generic means. In particular, the following aims have been achieved by the information archive [MUS 92].

1 The information archive is the main integrating element of MUSYK. It provides a means for integrating a large variety of heterogeneous information system components. The consideration of existing architectures fulfills the objective of portability and inter-changeability as well as re-usability of the system components.

2 The integration of various control levels will be done by joining the rough planning level with the production area control level via event-driven control circuits. This requires a common "virtual" data platform such as the information archive.

3 The information archive provides also all necessary information to prepare data dynamically for different planning and control levels. This facilitates shorter planning cycles and an improved production flexibility.

The use of the information archive with respect to integration is shown by figure 3 [MUS 93]. The archive embraces the product/process model of exchanging data described in EXPRESS schemata. This allows the automatic generation of pre- and post processors for writing and reading STEP physical files. These files contain the data to be exchanged.

This approach was realised prototypically by the integration of production coordination modules developed within the project with an existing PPC-system and a shop floor control system.

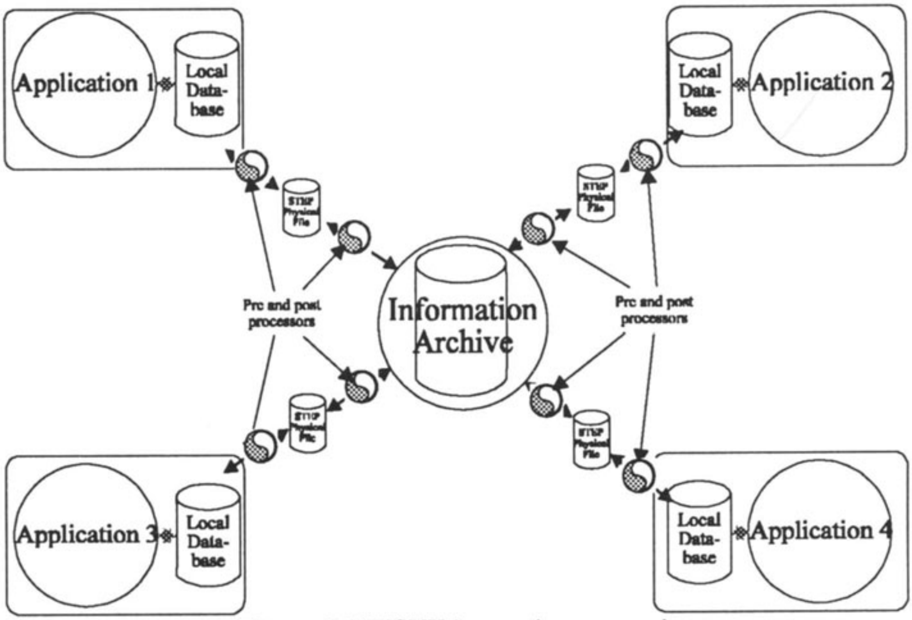

Figure 3. MUSYK integration approach

2.2.4. CMSO

Objective of the ESPRIT project 2277 CSMO (CIM for Multi Supplier Operations) is to optimize inter-organizational operations and logistics chains through means of advanced intelligent communications systems and by logistics coordination procedures, recognizing the requirements for technical electronic data interchange

(EDI) to support the product development and product support processes in the automotive industry. Main results with respect to integration aspects are:

1 EDI Reference Model
The CMSO approach in EDI applies a mechanism to bundle all interorganizational communication functions conceptionally in one EDI communication server, termed CMSO-Box. In this Box all functions are located which are related to the analysis, preparation and transmission of EDI messages. In order to structure the required EDI functions a multilayer reference model was introduced [CMS 90]. The services were structured into five sub-layers ranging from communication support functions to CIM applications on the highest level. This reference model was the basis for the development of a commercial EDI handler.

2 Technical Information Management System (TIMS)
With regard to the inter-organizational product development (between manufacturer and supplier or sub-supplier, respectively), TIMS optimizes and coordinates the information flow and its control between the associates. Basically, the development process subscribes design, planning, control and quality assurance functions which are supported by the integrated applications. Thus the system consists of a development coordinator, neutral interface selection, order database manager, temporary database manager, and a customer database manager [CMS 91].

2.3. REQUIRED FUNCTIONALITIES

The functionalities of the integration systems are defined depending on the subjects of integration. There will be different functionalities specified for integrating the distributed banking systems (e.g. for credit-authorization services like Visanet with 22.000 member banks) and for integrating the sales and marketing management systems of word-wide operating companies (e.g. Coca-Cola) [ALT 92]. The distributed production management systems have also some specific requirements which make their integration different from others. The most important characteristics of these systems are as follows:

1 geographically distributed systems/applications in a multi-site production environment and, in special cases, support of site-oriented temporal manufacturing (e.g. construction industry)

2 support of simultaneous and concurrent engineering processes beyond enterprise borders

3 autonomous systems in different enterprises (or enterprise domains) with different responsibilities at different sites (e.g. customers, sub-contractors, classification societies, etc.)

4 independent applications - usually developed by different software firms - in one enterprise domain (e.g. various CAD Systems used for different areas of construction like volume-based CAD for the development of engine-blocks or surface-based ones for chassis design)

5 heterogeneity (concerning systems ontologies[2], SW and HW platforms, networks)
6 complex systems i.e. many, possibly also distributed, modules (e.g. machine data collection systems)
7 exchange of large and huge amounts of product and process data
8 real-time operation mode of some of the integrated systems (e.g. CNC, machine data collection)

To support integration of autonomous applications the integration system has to be neutral to the subjects which are to be integrated. This means, that original functionality of the integrated components cannot be influenced by the integrating system. The integration must ensure safety of integrated applications and security of exchanged information as well. Here both the data access classification mechanisms and the reliable transaction management have to be considered. The data access classification allows the controlling of information access rights for members or their groups in the integrated software environment. This issue includes also cryptography aspects. The next aspect of the data security is the reliable transaction mechanism. Since there are various systems (concerning their ontology, complexity, respond times, etc.) in an integrated environment, the distributed transactions across networks (LANs as well as WANs) will have to be carefully managed in order not to violate the information being retrieved or modified. Another functionality concerning managed information is high data accessibility and reliability. Also the data and application distribution transparency has to be provided. The entire integrating system has to have an open architecture in order to enable flexible configuration and extension by adding new member applications. In order to support the demand-driven information exchange an event-oriented process control management has to be applied.

2.4. APPROACH CHOSEN

To fulfill the requirements arising while introducing CIM technologies in a distributed production environment, we have to apply proper IT which can support us in basic, though crucial, aspects of software integration in general. Therefore, the integration approach we introduce here is based on the following general principles:
1 federated system architecture with neutral data models
2 object-oriented work flow modeling language
3 virtual processors and interfaces build upon mediators
4 opened set of functional operating modules (external, specialized services of the integration platform e.g. standard converters like EDIFACT, STEP but also AI-applications for e.g. pattern recognition)

2 The term 'ontology' - originally introduced in AI research - presents the primitive concepts, terminology and interpretation paradigm of a domain of discourse.

5 CORBA as underlying network-wide data exchange mechanism

On one hand our approach combines the object-oriented modeling (EXPRESS-G, EXPRESS) with CORBA architecture to achieve thoroughly consistent design and implementation[3] of the low-level integration platform. On the other hand, the federated databases combined with the mediators architecture provide new functionalities concerning the data security and information processing in large-scale environments. Last but not least, the modern enterprise modeling methods (e.g. object-oriented work flow as enterprise function modeling instrument) as well as application of EDIFACT standard business data interchange protocols and STEP product data models ensure practical exploitation of results at industrial sites.

Below we first introduce our general integration philosophy based on federation principles. Further, we discuss application of mediators as integration actors. Finally we focus on the CORBA as underlying integration mechanism.

2.4.1. Federation as general integration philosophy

According to the definition provided by Sheth and Larson, a federated database system[4] (FDBS) is a collection of co-operating but autonomous component database systems (DBS). Since in our implementation context are many various information sources, which not necessarily have to be '*a database*', we will extend the scope of the federation. We define the *federated information system* (FIS) as a collection of co-operating but autonomous components, whereas any kind of information source (any sort of DBMS e.g. EDB, simulation systems, schedulers, real-time data collectors, reference systems, etc.) can be a federation member. These components may be integrated to various degrees and can participate in more than one federation. In general, other FISs can be members of a federation. We also introduce special kinds of FIS members, which provide control and co-ordination of the federation components: *federation government systems* (FGS). Further the entire configuration of FIS and its FGS will be called *federated information management system* (FIMS).

A key characteristic of a federation is the co-operation among independent systems. In terms of a FIMS, it is reflected by controlled and sometimes limited integration of autonomous member systems/applications. The integration of member systems may be managed either by their users or by the FGS together with the administrators of integrated applications. The degree of integration depends on the users needs and desires to participate in the federation and share their data. The organization paradigm applied here is actually a generalization of a *server-client* model, whereas a member can be a *server* and a *client* at the same time.

The co-operation requires integration which implies the information exchange.

---

3 We use Orbix - the Iona Technology Ltd. implementation of the ODMG'93 CORBA standard.

4 The term *federated database system* was coined by Hemmer and McLeod in 1984. Since its introduction the term has been used for several different but related DBS architectures.

Thus, it is necessary to discus all aspects associated with information management in general. As presented in the figure 4, there are two possibilities to partition the information pools, which allow strict control over external access.

The first one relays on the definition of various exported schemata by a federation member. These schemata will afterwards be integrated in different federation schemata. Thus, members of one federation will not have an access to information available in the other federation. This principle is usually applied in loosely coupled federations with many federation schemata. The second possibility is to define different external schemata of the federation schema which will be addressed (i.e. accessible) to different members or their groups. This implies the active role of the FGS since this system is in charge of the central management of the entire federation. This approach is used in tightly coupled federated information systems (TCFIMS), but can also without any exceptions be applied in loosely coupled ones (LCFIMS).

In our distributed production context we feel that both types of federations will be established. The more centralized solution - the tightly coupled federations - will be more likely found inside an enterprise, while the loosely coupled federations will probably cover inter-organizational integration. The reasons for this are obvious.

Because of their specific features, the TCFIMS ensure complete control over all federation members. Since they support *pull* and *push* principles, it is possible to trigger required members activities directly (to start processes like e.g. material just-in-time procurement, re-scheduling of a customer order, reprogramming a CNC machine etc.). The well-recognized transaction management in tightly coupled FDBS ensure the consistence of the stored information. This is of vital importance during simultaneous and concurrent engineering processes.

The LCFIMS principally supports only information retrieval processes. They do not directly support *push* principles and cannot manage transactions. From the other point of view, they have significant advantages which make them interesting for inter-organizational integration purposes. The most important are the following:

1 Neutrality to the integrated federation members.
The crucial aspect of this feature is, that original members functions (like e.g. designing, manufacturing and assembly controlling, order tracking in the local environment etc.) do not depend on the federation. This means that short-term disturbances (e.g. in terms of hours) in the functioning of the federation will not influence the member's original functions.

2 Failure tolerance.
During logical aggregation of information retrieved from federation members, the missing data (e.g. caused by errors during query processing) will have minor influence on the result (given the failure rate is low), although the uncertainty will arise. The degree of uncertainty will of course increase by a growing errors rate.

3 Management of radical changes in federated schemata.

Such fundamental changes take place when e.g. one member abandon the production consortium, thus leaves the federation.

4 Information security in the sense of inter-organizational data exchange.

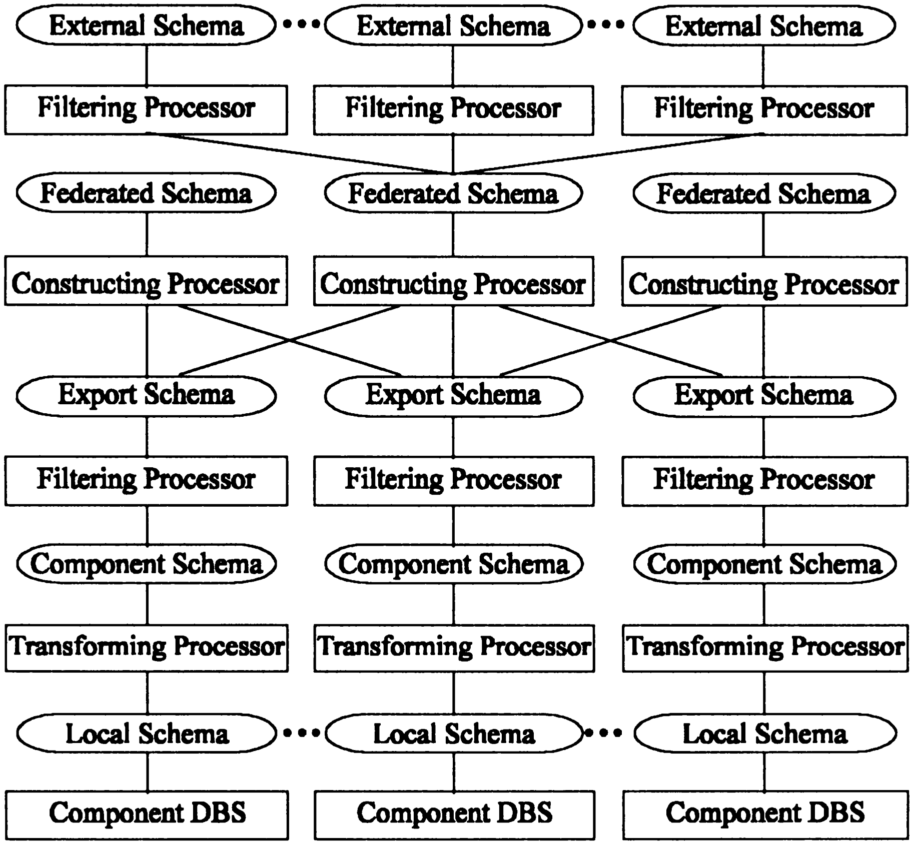

Figure 4. Federated Information Systems

The existence of multiple federated schemata based on different exported schemata of federation members (here enterprises) allows partitioning of the entire information. For example, product data of a ship can be partly exported by a shipyard into various export schemata. There can be one schema for classification society and another one for engineering bureaus. There will also be exported schemata for the customer and suppliers involved. All these schemata, according to the demands, will be integrated in different federated schemata (like e.g. shipyard-suppliers FIMS, shipyard-customer FIMS, shipyard-classification society FIMS etc.). Moreover, it may be possible to define different external schemata of a particular federated schema e.g. there can be one external schemata of the shipyard-suppliers FIMS for supplier A and another one for supplier B.

2.4.2. Mediators as integration actors

In this section we discuss a novel approach to information processing which is applicable for large systems integration purposes. This approach bases on relatively small processing modules - mediators (figure 5). The principals of the mediators are thoroughly described by Gio Wiederhold (see Section 'Bibliographic notes').

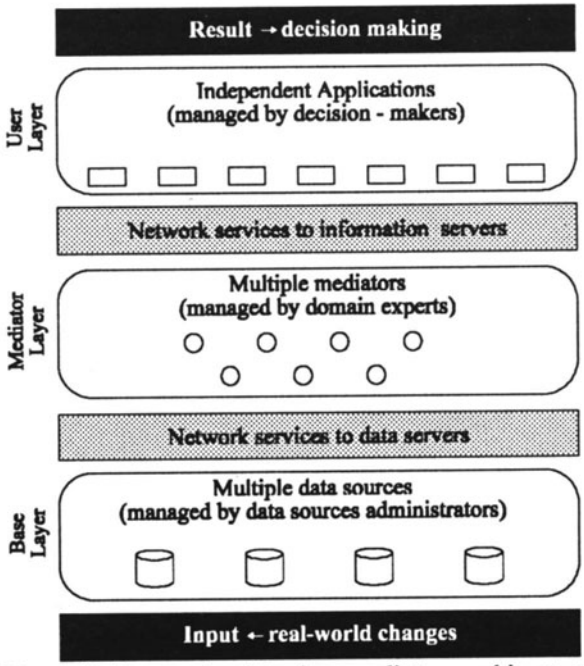

Figure 5. Integration based on mediators architecture

The very characteristic feature of mediators is that they remain simple and are constrained by a single ontology. Furthermore, we postulate an additional requirement to them, which is essential for our approach: they have to be able to communicate with each other, whereas this cannot be limited to one address space or one machine. The communication has to be network-wide supported i.e. mediators on different network nodes must be able to exchange data (heterogeneity and distribution transparency).

Actually, there are different logical layers of tasks performed by mediators. Thus there are different mediating layers. One layer deals with basic data manipulation like focusing, pruning, fusion, and reduction. The other one is placed higher and addresses control operations associated with effective resource management. These resources should be understood very generic and can mean:

1 software resources like other mediators or large integrated applications, etc.

2 hardware components like simple printers or massive parallel machines at some network nodes, etc.

3 information resources like small documents or huge product data models (e.g. ship data model) etc.

Thus, we can present mediators as programs providing specialized services, which modify information very selectively. The information processing presents a number of specialized operations (provided by relevant mediators) on an *input set* of data resulting in a new state of this data (*output set*) which can afterwards be used either as *input set* for other processing or, if the processing has been completed, can be forwarded to the application (in our context: to a federation member).

This processing paradigm is similar to both the blackboard approach and the agent systems. However, there are important differences between mediators and agents as well as blackboard frameworks.

The simplified blackboard framework is shown in the figure 6. In the classic blackboard approach there is a number of specialized AI modules called Knowledge Systems (KS, normally domain experts), which operate on data gathered on one common 'blackboard' [NII 86].

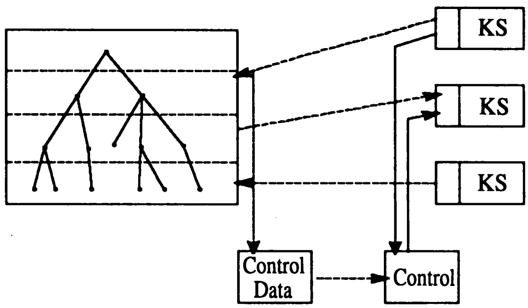

Figure 6. The blackboard framework

The co-ordination of the KS is usually performed by Control Modules, which actively react to the current state of the blackboard. They decide about the order of the working experts. They also estimate whether the result has been reached or not. Hence, the key characteristics of the blackboard systems are the existence of the 'blackboard' and control modules.

The information processing based on mediators requires neither a common data pool (blackboard) nor control modules for task distribution or synchronization. The mediators work on one or more input sets of data and act according to the process definition. The control functionality is incorporated in the process model. The required task synchronization is performed in neutral integration mechanisms (e.g. DTS - *distributed time service* in DCE) which are the implementation platforms for mediators. Therefore, the application of these integration mechanisms is of such importantance for our approach.

Also, the estimation of whether the current state can be recognized as a result does not take place in the mediator approach. There is no particular result to achieve, but an output set has to be produced according to process description. This means, nothing analyzes the output set according to its logical meaning. The mediators approach differs from the blackboard one in the scale of components involved, too. The mediators are small specialized programs whereas the blackboard components are usually AI modules which normally are quite complex. Thus, the maintenance of mediators is much easier and cheaper.

The mediators have many more similarities to the agents. The agent systems - a research area of DAI (distributed artificial intelligence) - also act independently as autonomous modules. The agent systems are principally networks of relatively autonomous problem-solving modules which work together towards the common goal.

An agent is a module that includes knowledge about one specific domain. A standard model of an agent defines three layers: the problem layer (with an inference processor), the task layer (with a task processor) and the interaction layer (with an interaction processor). The co-operation of agents is based on message exchange.

The mediators differ from agents mainly in their complexity and their self-acting possibility. Since the agents are problem-solving modules, they are *de facto* AI modules. Hence an agent most probably is a very complex system. Moreover, the agent's ability of interacting is higher than by mediators in the sense of unforseenable requirements. As a matter of fact, agents decide themselves about the interaction they need. Mediators do not have this feature. They interact according to a given process pattern (Figure 7).

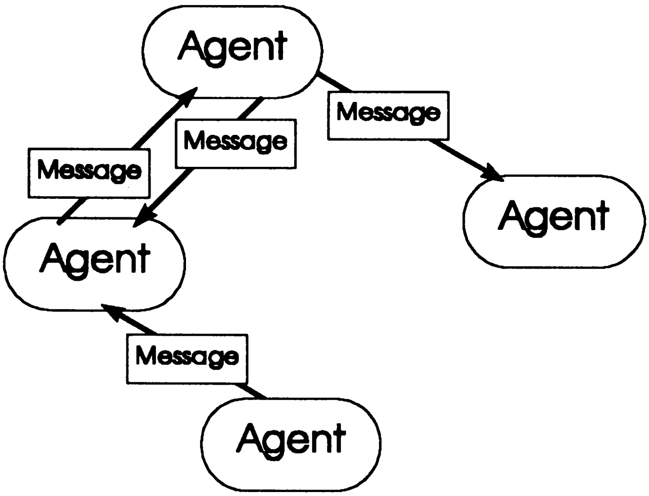

Figure 7. Interaction principals in the agent systems

2.4.3. CORBA as underlying integration mechanism

The discussed integration approach relays on the low-level data exchange mechanism which is designed according to the CORBA standard provided by OMG (Object Management Group). The OMG defined the information management paradigm which models 'real world' by representation of program components called *objects*. This paradigm enables faster application development, easier maintenance, reduced program complexity and reusability concerning mainly (but not only) the implementation of distributed information systems. Therefore, the CORBA integration mechanism enables to retrieve information provided by *objects* across the borders of a single program by any other program in a network. [OMA 92]

Figure 8 presents the general architecture of the CORBA. There are principally three basic components: client, object implementation and object request broker.

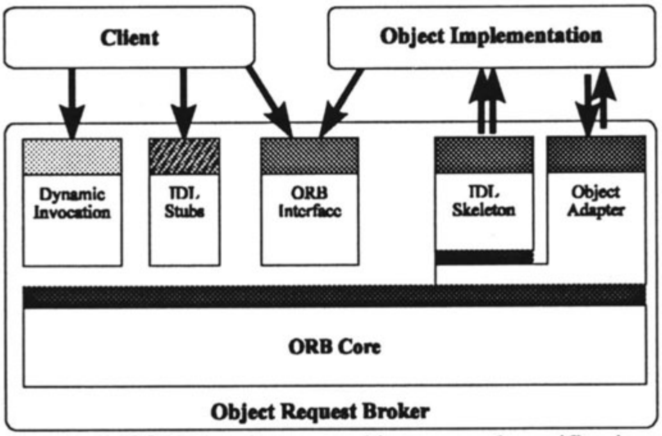

Figure 8. CORBA reference architecture and specification

The clients are all those applications (represented also as objects) that operate on desired objects i.e. invoke methods provided by these objects. The object implementation provides the semantic of the object, usually by defining data for the object instance and code for the object's methods. The object request broker is that part of the CORBA architecture that takes over all tasks required for the communication between clients and object representations [COR 92].

2.4.4. Bibliographic notes

There is a number of related publications that describe particular components of our approach or provide basic knowledge about concepts applied in this paper.

The mediators, their role and proposed architecture of systems based on them are presented in [WIE 91], [WIE 92a], [WIE 92b], [WIE 93].

The architecture of federated databases was first proposed by Hammer and McLeod in 1980[5] and is further described in [HML 85]. Sheth and Larson in [SHE 90] did more specific research and described the Five-Level FDBS Architecture. They have presented also the ANSI/SPARC Three-Level Schema Architecture. The mapping processors are described here as well.

The more general scope on distributed databases is provided in many interesting publications. The transaction management aspects in the distributed databases were first discussed in [GLI 85] and later in [NCW 93]. A new method for selectivity estimation (multi-level grid file: MLGF) - crucial aspect for query optimization - in distributed databases is presented in [WHA 91]. A brief review of work on multidatabases is presented in [BRE 90]. A very interesting case study on the heterogeneous DB for production use is provided in [GOM 90], [JAK 88], [RAF 91]. The problems arising while mapping different schemata are discussed in [SIE 91], [SCI 92]. In the [CSR 91] is a proposal for management of semantic metadata with the help of an Intelligent Information Dictionary which may be very interesting for solving the difficulties arising while mapping different schemata.

In our approach the object-oriented databases play a very important role. The [BUK 92] provides a short research overview on the OO multidatabases like (CIS, DOM, EIS/XAIT, FBASE, OIS, Pegasus and ViewSystem). The [BER 91] introduces concepts on OODBMS.

Essential for studding the OODBS architectures is getting familiar with ODMG (Object Database Management Group) documents. The ODMG has published the first standard ODMG-93 [ODM 93]. This will probably be the turning point in the progress of commercial OODBMS products, since it provides a standard data model (based on OMG *object model*), data modeling language (*object definition language*), query language (*object query language*) and bindings to programming languages (first C++ and C). This ensures future portability which will increase the acceptability and endorsement of the OO approach in the DB area. These are the most important factors for future vendors and customers since they ensure investments in this new technology.

2.5. REFERENCE MODEL AND PROPOSED ARCHITECTURE

As stated before, the large and huge integration systems have to be scaled down in order to be controllable. Therefore, we have specified a small set of modular components we need to establish a federal information system.

Our integration system is a network-wide and network-independent communication software functioning as a shell between the existing systems/applications. We would like to present it as a universe of different kinds of - more or less connected - objects, whereas everything can be an object. Enterprises, domains, applications, databases and even particular data are objects

---

5 M.Hammer and D.McLeod, On database management system architecture, in Infotech State of the Art Report, vol. 8: Data Desugn, Pergamon Infotech Ltd., 1980

for us.

We can principally divide our entire model of the system into three logical areas (see figure 9):

1 federation framework
2 mediators shell
3 CORBA-based integration platform

These three areas actually represent three different views on our integration approach: general organization paradigm (federation), novel interfacing philosophy (mediators) and technical realization view (CORBA). Thus, when focusing on one particular component of the system we will always find aspects of these views.

The concepts of each area are described below, but as an example, we can present a federation processor (e.g. constructing processor) which does not exist as a module or program (we will use further a term *virtual processors*), but is actually built from a number of specialized mediators. These mediators exist *de facto* as services at the CORBA-based integration platform.

Note that our system is highly dynamic. Complex integration operations (e.g. schema mappings) will be not supported by one specialized program. According to temporal needs they will be constructed from many basic, low-level services. As a matter of fact, our *virtual interfaces* consist of ingredients (existing services - mediators) and receipts (information processing models).

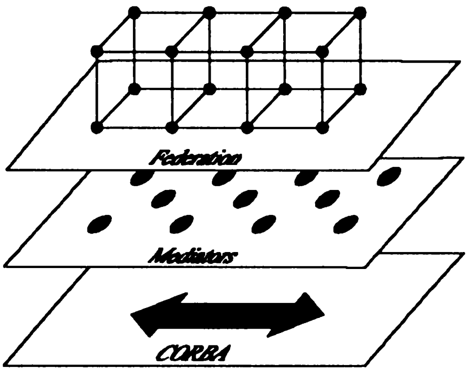

Figure 9. Three logical integration areas

Since we want to keep our integration system flexible (rearrangements of existing connections must be possible) and expandable (new features, services etc.), we apply a modular design. The modular architecture allows step-by-step development and enhancement of the system by exchanging or adding new modules. Moreover, every component is designed and implemented in an object-oriented manner in order to support the manageability and maintenance of the entire system.

2.5.1. Model of the integration platform

The discussed integration approach relays on the low-level data exchange mechanism which is designed according to the CORBA. This integration mechanism enables to retrieve information provided by *objects* across the borders of a single program by any other program in a network. Principally we have the following components:

1 *Information Objects*

They are special kinds of objects that can be exchanged between any applications that are *Stations*. An *Information Object* is identified by a *type* (can be the class name or any other string, and must be unique in the network) and a *name* (must be unique in the network within this *type*). Every *type* and every provided *Information Object* can have a *description* that is visible for all *Stations*.

2 *Stations*

The *Stations* are the applications which are able to export (make available) and import (access via referenced pointers) *Information Objects*. The *Stations* that need information from other *Stations* are able to either copy the values of another *Stations* object, or to call a method of this object and retrieve just the result of this method call. We use a name *Station* for any process that exchanges information and we do not limit it only to mediators. Mediators belong to this category as well the special, additional low-level network services, which are components of the underlying integration mechanism itself.

3 *Control Servers*

They manage interaction between *Stations* that provide or request information. The *Control Servers* keep information about all offered *Information Objects* and their *types* in the network. They know where a particular object is offered, but - except of registration and error cases - they are invisible: Stations just request and provide objects without worrying *how* they are distributed or *where* they are found. There can only be one *Control Server* at a machine, because *Control Servers* are distinguished by their host names (machine's id). If more *Control Servers* are started at other network nodes, they will automatically get all the information the already existing *Control Servers* have. Our *Control Servers* are actually typical Object Request Brokers with additional functionality needed for creation of virtual interfaces and processors.

2.5.2. Mediating shell model

The major objective of the mediating layer is the logical integration of the federation members. Thus, the mediating *shell* has the following main functions in the entire integration system:

1 to join the federation members (i.e. systems/applications) with the CORBA-based integration platform,
2 to support federation members in defining their own exported schemata and their specific events which have to be recognized by the integrating system,
3 to support the FGSs by creating and managing the federation schema and - if required - external schemata,
4 to support federation members with standard services like browsing in the federation schema, scheduling of required activities, messaging.

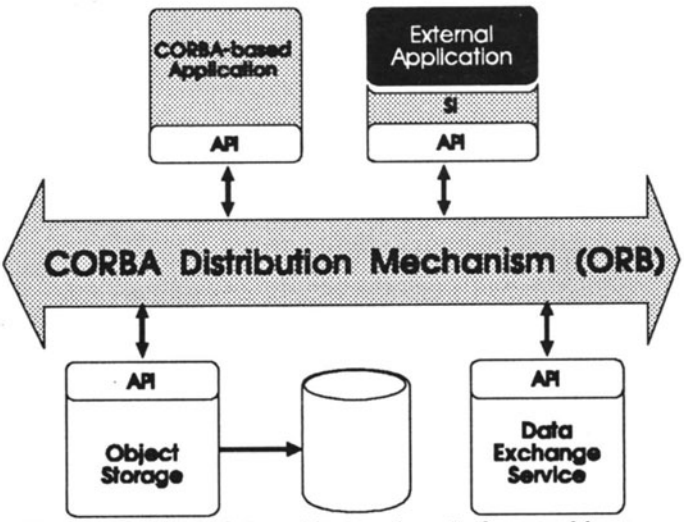

Figure 10. CORBA-based integration platform architecture

The *shell* uses meta-information stored in the common distributed data repository which is in turn provided by the CORBA-based integration platform.

The mediating *shell* itself consists of:

1 mediators pools
2 work flow managers (builders, testers, executive managers)
3 mediators maintenance tools (builders, browsers, configuration analyzers)

The mediators pools (figure 11) consist of many mediators which are related to each other according to their functionality e.g. atomic mediators, schema constructing processors, STEP or EDIFACT translators, vectored semantic analyzers etc.

The work flow managers provide functionalities for building, testing and executing the information processing patterns. These patterns define which mediators should be used in which order to produce the desired data output set. Hence, it is a kind of process description. We use an object-oriented work flow model based on state diagrams for objects involved. These managers are required for constructing virtual federation processors and virtual application interfaces.

The mediators maintenance tools support the users with standard functionalities like browsing in unknown mediator pools (e.g. in order to select required mediators), building composed mediators from atomic ones and analyzing the existing unknown composed mediators (e.g. in order to exchange their components).

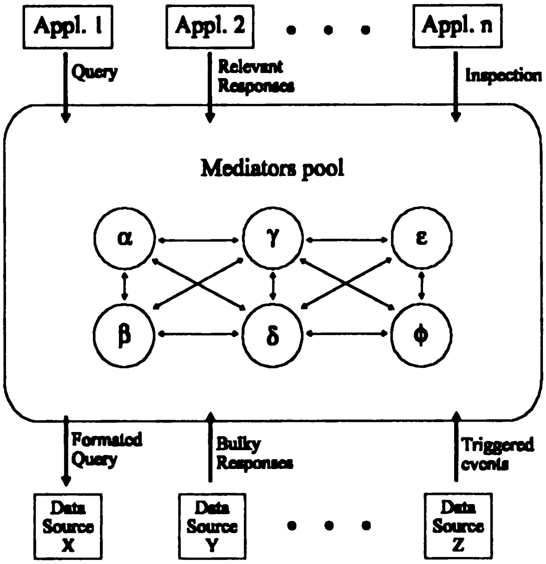

Figure 11. Mediators pool

Principally we distinguish between two general types of mediators:

1 atomic mediators (mediators which for performing their own functionality do not need other mediators),
2 composed mediators (mediators which involve other ones to perform their own functionality).

The mediators perform different kinds of tasks, thus there are different logical layers where they resist:

1 basic manipulation layer (focusing, pruning, fusion, and reduction of information),
2 resource management layer (supporting the usage optimization of accessible SW and HW resources),
3 direct interfacing layer (highest level of tasks, these mediators are 'visible' for applications - application interface mediators - and/or users - user interface mediators).

2.5.3. Reference model of the entire federation

As stated before the entire model of the integration system is based on three basic elements: the federation as organization paradigm, the mediating layers and the CORBA common object request broker architecture.

The federation as organization archetype allows integrated applications of selective information partitioning regarding other federation members. This ought to be seen as an integration framework where different members can take desired locations for obtaining corresponding privileges and limitations. At this stage of system design it is irrelevant how the principles of this system organization will be implemented. In some environments, there may be implemented architectures with missing components (e.g. no external schemata since the entire federation schema is accessible). We can also anticipate extended architectures with additional schemata and processors implemented for very specific needs. Nevertheless, the clue is to provide the federation principles in order to support the real needs (concurring the information security) of the users.

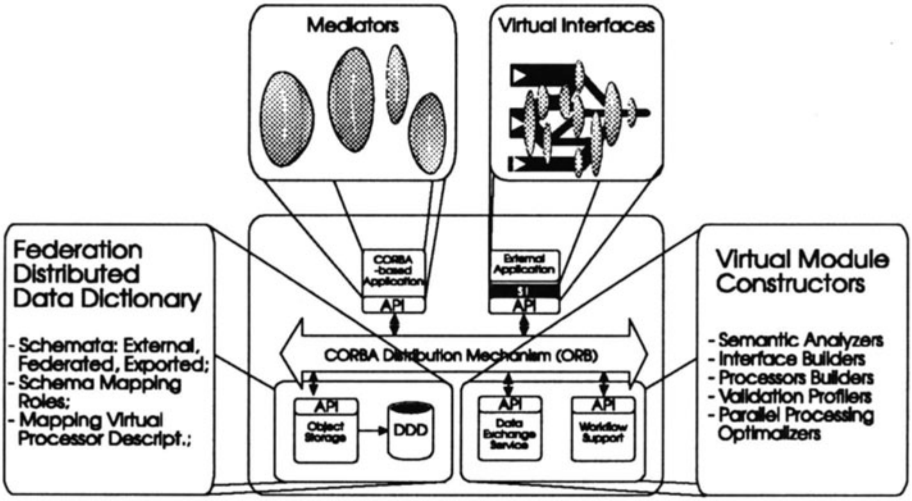

Figure 12. System model

The mediating layers consist of specialized modules which take over all automatical integration tasks. These tasks are performed by *virtual processors* and *interfaces* which are built from basic elements - mediators. The building blocks are accessible to the federation users, especially FGSs, and can be used for different purposes simultaneously. Since they are implemented on CORBA, they can communicate across entire network and are not bound to any particular application. The mediating layers receive requests from federation members, analyze them, perform required operations and finally produce results. Since both push and pull principles are supported, the integrated applications can retrieve and send information as well as trigger activities. This feature is crucial if one wants to use the federation for controlling purposes, which in our CIM context is always the case.

The lowest level in our integration architecture is the CORBA integration platform. The main purpose of it is to support mediators and other services with features that are required for network-wide data exchange. The platform consists of a number of components which are necessary for it to function.

The CORBA itself provides a development platform for network-wide operating modules which communicate with each other regardless of their localization and their differences in implementation platforms. It is the suitable integration mechanism, as it offers SW/HW independence, distribution transparency and object-oriented data management including all its advantages. Additionally, the progress of the OMG standardization affords allows to expect standard bindings for many programming languages and storage mechanisms.

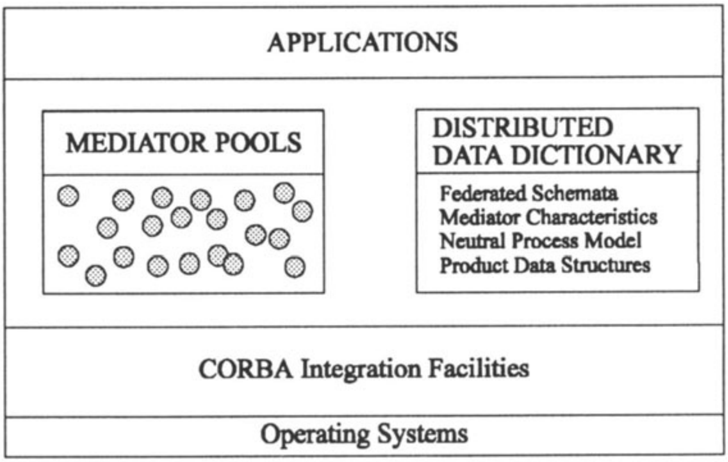

Figure 13. System architecture

2.6. EXPECTED ADVANTAGES

The integration system described in this paper has a number of advantages in comparison to standard approaches based on direct application interfaces or common databases. The most interesting benefits are briefly described below.

High flexibility of application integration is ensured by the applied mediating approach. Relaying on many specialized mediators, which offer functionalities in different generalization levels, very many different *virtual processors* and *interfaces* can be built by partly using the same mediators. The functionality of each virtual module will be determined by the applied information processing pattern i.e. by a combination of mediators.

The **data security** support in the sense of partitioning of information pools for different external users, is in federated systems one of the most important topics.

Therefore - concerning particular interests of enterprises co-operating in distributed production environment - loosely coupled federations are the optimal solution to support their organizational autonomy. Also the low-level context of data security (i.e. particular data access and exchange mechanisms like e.g. RDBMS queries) can be ensured because the implemented common object request broker architecture provides required mechanisms (e.g. filters for information exchange monitoring and control, standard OOP features, etc.).

Enhanced system availability for inter-organizational integration is ensured by adopting of loosely coupled federations whereas short-term disturbances within the integration system have a minor impact on the members involved.

The **HW resources usage flexibility** will be enhanced because the mediators approach does not bind mediators to any network node. Moreover, we foresee duplication and migration of mediators for reuse purposes. This means that the load of network nodes can be controlled by migrating the mediators used for integration purposes from one computer to another.

Lower costs and simplified system maintenance is provided since there are no complex and specialized application interfaces. The approach based on standard data formats (STEP, EDIFACT) excludes multiple interfacing (there are canonical data models in the exported, federated and external schemata of the federation). The mediating approach simplifies the maintenance of these small programs. The people responsible for special domains, will have to take care only of relevant mediators (small modules) and will not have to worry about the way the mediators are used (in high-level virtual modules).

2.7. CONCLUSION AND FUTURE RESEARCH

The presented integration approach is currently being developed and implemented at the BIBA institute. The system architecture introduced in this paper is very promising since it enables usage of modern facilities such as parallel computers and neural networks. As any multiple-paradigm approach with partitioned architecture, our mediator-based integration platform is well suited for multiprocessor environment. Especially because the mediators are highly independent from each other and are not bound to any particular application, the implementation in a parallel computing environment is the next logical step to be made. The possible application of neural networks may improve the overall performance as well. The goal would be to support automatic setting of optimal information processing patterns for virtual processors based on e.g. vectored analysis of semantic models of involved mediators and tailored applications.

Hence, future research will focus on the following major topics:

1 implementation of the mediators pool on the parallel virtual machine build on workstation clusters,

2 development and implementation of the distributed workflow execution engines, which will be able to accept *ad-hoc* changes in processing models,

3 definition and development of semantic description models which will

describe mediators as well as integrated applications,

4 automatic setup of processing mediator models for virtual interfaces which will allow the on-line demand-driven creation of interfaces from existing mediators according to the particular task to be performed.

However very flexible, the integration platform is not directly designed to support financial applications. Enterprise's non-technical domains dealing with banking, personal and public relations are - because of their very specific requirements arising from money-exchange principles [MUL 91] - excluded from the scope.

2.8. BIBLIOGRAPHY

[ALT 92] STEVEN ALTER, Information Systems - A Management Perspective, Addison-Wesley, 1992

[BER 90] ELISA BERTINO,LORENZO MARTINO; Object-oriented database management systems: concepts and issues; 0018-9162/91/0400-0033 IEEE, April 1991;

[BRE 90] BREITBART Y.; Multidatabase Interoperability; Sigmod Record, Vol. 19, No. 3, September 1990;

[BUK 92] BUKHRES OMRAN A., ELMAGARNID AHMED K., MULLEN JAMES G.; Object-Oriented Multidatabases: Systems and Research Overview; 1992

[CMS90] CMSO Consortium (Eds.): Communication ArchitectureModelling; D4; ESPRIT project 2277; Bremen 1991

[CMS91] CMSO Consortium (Eds.): Technical Information Management System; D13; ESPRIT project 2277; Bremen 1991

[COR 92] The Common Object Request Broker: Architecture and Specification, OMG Doc. 91.12.1, 1992

[CSR 91] CAMMARATA STEPHANIE, SHANE DARREL, RAM PRASAD; IID: An Intelligent Information Dictionary for Managing Semantic Metadata RAND National Defense Research Institute, 1991

[DEC 93] DECOR Consortium (Ed.): Technical Annex; ESPRIT Project8486, 1993

[GLI 85] GLIGOR VIRGIL D., RADU POPESCU-ZELETIN; Concurrency control issues in distributed heterogenous database management systems; Distributed Data Sharing Systems, F.A. Schreiber and W. Litwin (editors), Elsevier Science Publishers B.V. (North-Holland), 1985;

[GOM 90] GOMER THOMAS, GLENN R. THOMSON, CHI-WAN CHUNG, BARKMEYER EDWARD, CARTER FRED, TEMPLETON MARJORI, FOX STEVEN, HARTMAN BERL; Heterogeneous Distributed Database Systems for Production Use; ACM Computing Surveys, Vol. 22, No. 3, Sep. 1990

[HML 85] HEIMBIGNER DENNIS, MCLEOD DENNIS; A Federated Architecture for Information Management; ACM Transactions on Office Information Systems, Vol. 3, No. 3, July 1985, P. 253-278.;

[HIR 91] HIRSCH BERND E., THOBEN K.-D. (Editors); New approaches towards 'one-of-the-kind' production; IFIP WG 5:7, Proceeding of International Working Conference 12-14 Nov. 1991 Bremen, Germany

[JAK 88] JAKOBSON G., PIATETSY-SHARPIRO G., LAFOND C., RAJINIKANTH M., HERNANDEZ J.; CALIDA: A System for Integrated Retrival from Multiple Heterogenous Databases; Proceedings of the third International Conference on Data and Konwledge Engineering, p. 3-18, Jerusalem, Israel, 1988

[MAR 92] MARITIME Consortium (Eds.): Technical Annex Part II; ESPRIT Project 6041, 1992

[MAR 93] MARITIME Consortium (Eds.): MARITIME Reference Architecture, D1401; ESPRIT Project 6041, 1993

[MUL 91] MULLENDER SAPE J., Acounting and resource control, in Distributed Systems, p. 133-147, ACM Press, 1991

[MUS 92] MUSYK Consortium (Eds.): Requirements Definition of the Coordination Modules; D41; ESPRIT project 6391; Bremen 1992

[MUS 93] MUSYK Consortium (Eds.): Abstracted Requirements and Design for Methods of the Archives; D31; ESPRIT project 6391; Rijswijk 1993

[NCW 93] WOLFGANG NEJDL, STEFANO CERI, GIO WIEDERHOLD; Evaluating Recursive Queries in Distributed Databases; IEEE transactions on knowledge and data engineering, p. 104-140, Feb. 1993/Vol. 5, No. 1;

[NII 86] NII H. PENNY; Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures; The AI Magazine, Summer 1986;

[ODM 93] R.G.G. CATTELL (Editor); The Object Database Standard: ODMG-93; 1993;

[OMA 92] Object Management Architecture Guide, OMG TC Doc. 92.11.1, Sep. 1992

[RAF 91] AHMED RAFI, DE SMEDT PETER, DU WEIMIN, KENT WILLIAM, KATABCHI MOHAMMAD A., LITWIN WITOLD A., RAFII ABBAS, SHAN MING-CHIEN; The Pegasus Heterogeneous Multidatabase System; IEEE Computer, p. 19-27, December 1991;

[SCI 92] SCIORE EDWARD, SIEGEL MICHAEL, ROSENTHAL ARNON; Context Interchange using Meta-Attributes; Information and Knowledge Management CIKM-92; Proceedings of the ISMM International Conference Baltimore, MD, USA November 8-11,1992

[SHE 90] AMIT P. SHETH, JAMES A. LARSON; Federated databases: architecture and integration; ACM Computing Surveys, p. 185-228, September 1990;

[SIE 91] SIEGEL MICHAEL, MADNICK STIART E.; A Metadata Approach to Resolving Semantic Conflicts; Proceedings of the 17th International Conference on Very Large Data Bases, Barcelona 1991;

[WHA 91] KYU-YOUNG WHANG, SANG-WOOK KIM, GIO WIEDERHOLD; Dynamic maintenance of data distribution for selectivity estimation ; Report No. STAN-CS-91-1388, Dept. of Computer Science, Stanford University, September 1991

[WIE 91] GIO WIEDERHOLD; The role of artificial intelligence in information systems; ISMIS '91, Methodologies for Intelligent Systems, 6th International Symposium, Charlotte, USA, October 16-19, 1991, Springer-Verlag,

[WIE 92a] WIEDERHOLD GIO; Mediators in the Architecture of future Information Systems; IEEE Computer, p. 38-49 March 1992;

[WIE 92b] WIEDERHOLD GIO; Intelligent Integration of Diverse Information; Information and Knowledge Management CIKM-92; Proceedings of the ISMM International Conference Baltimore, MD, USA November 8-11,1992

[WIE 93] WIEDERHOLD GIO; Intelligent Integration of Information; ARPA-SISTO and Stanford University, ACM 0-89791-592-5/93/0005/0434 ,March 15, 1993