Journal of Intelligent Manufacturing (1999) **10,** 587–601

Enterprise wide application integration platform for CIMS implementation

YUSHUN FAN, WEI SHI and CHENG WU

Department of Automation, Tsinghua University, Beijing 100084

Received March, 1998 and accepted August, 1998

Adoption of CIMS in manufacturing enterprises requires more advanced tools for application integration. This paper presents the system architecture and services of an integration platform, CIMS Application Integration Platform for Manufacturing Enterprises (MACIP). MACIP integrates a set of application software and application development tools to provide a complete system for CIMS implementation. It includes a communication system, a global information system, three domain application sub-integration platforms, an Internet interface and an operation management and control system. It is based on the Client/Server structure, and employs the object-oriented paradigm and agent technology. System openness, scalability and maintenance are ensured by conforming to international standards and by using effective system design software and management tools. MACIP can significantly reduce the complexity of CIMS implementation.

Keywords: Integration platform, CIMS, system architecture, object-oriented technology, application programming interface

Abbreviations

ACL:	Agent Communication Language
API:	Application Programming Interface
AD-API:	Application Dependent-API
AI-API:	Application-Independent-API
AM-API:	Application Management-API
APD-Tools:	APplication Development Tools
CIM:	Computer Integrated Manufacturing
CIMOSA:	CIM Open System Architecture
CIMS:	Computer Integrated Manufacturing Systems
CIN:	Control and Integration Net
CORBA:	Common Object Request Broker Architecture
CS:	Common Service
DCE:	Distributed Computing Environment
DXF:	Drawing eXchange Format
GIS:	Global Information System
IGES:	Initial Graphic Exchange Specification
IIS:	Integrating Infrastructure
IP:	Integration Platform
IT :	Information Technology
KQML:	Knowledge Query and Manipulation Language
LAN:	Local Area Network
LCS:	Local Communication System
MACIP:	CIMS Application Integration Platform for Manufacturing Enterprises
MIS:	Management Information System
MMS:	Manufacturing Message Specification
ODL:	Object Definition Language
ODMG:	Object Database Management Group
OMC:	Operation Management and Control
OMG:	Object Management Group
O-O:	Object-Oriented
OQL:	Object Query Language
OSI:	Open System Interconnection
PDM:	Product Data Management
RDB:	Relational DataBase
SIP:	Sub-Integration Platform
SNMP:	Simple Network Management Protocol
UML:	Unified Modeling Language
VMD:	Virtual Machine Device
WAPI:	Workflow API
WFMC:	Workflow Management Coalition

0956-5515 © 2000 Kluwer Academic Publishers

1. Introduction

Following the advent of mechanization and electronic automation, in the late 1980s manufacturing enterprises entered a new period of enterprise integration, which was marked by the acceptance of the Computer Integrated Manufacturing Systems (CIMS) concept. In the 1990s, new manufacturing paradigms, such as Concurrent Engineering, Agile Manufacturing and Virtual Manufacturing, have enriched the CIM methodology. These new advances have put more demands for CIMS integration technology and associated support tools more than ever before. One of these demands is to provide CIM systems with better software architecture, more flexible integration mechanisms, and powerful support platforms.

Within the implementation of CIM systems, integration is the most important technology. A complete integration paradigm includes the integration of data resources, the integration of application functions, and the integration of business processes. However, the complexity of manufacturing systems and the lack of effective integration mechanism arise problems for CIMS implementation. Some of these problems are:

(1) Lack of openness and flexibility. The integrated system is generally inflexible rather than open-structured, and it is difficult to incorporate new technology. Lack of flexibility brings difficulty to update when it is required with the evolving of enterprise requirement.

(2) Inconvenient and inefficient interaction between applications. This is caused usually by the heterogeneous platforms, non-standard data presentations and low-quality system management.

(3) difficulty in integration of a legacy information system. Many enterprises implement CIMS on existing Information Technology (IT), and wish to retain their old investment. But there lacks an effective mean of combining legacy resources into the new system.

(4) Long time for CIMS implementation. Without powerful application-oriented support tools, the whole implementation process is long and inefficient, lasting $3 \sim 5$ years or more. This leads to greater risk and higher expense for the enterprises.

(5) Inconsistency of user interfaces. In an integrated system, different user interfaces with similar function lead to a confused understanding and possible misuse of the system.

Computer software technology is evolving rapidly. Object-Oriented (O-O) paradigm has taken the predominant position in software development. Its well-known features of encapsulation, inheritance, and polymorphism have made the O-O based software easier to understand and simpler to build and maintain as compared to the traditional structured methods. At the execution level, old centralized system architectures have given way to the client/server structure and distributed architecture, which enables distribution of the application logic to several machines to match the real situation and to improve the system efficiency. From the base, the middleware technology is now flourishing quickly. Middleware technology aims at building an application system by integrating various components through so called software-bus, which is the defined standard and mechanism for component interaction. The introduction of Middleware enables software component integration at runtime. All these advances in software technology present a richer and more flexible means for CIM application development.

To meet the requirements enumerated above, and to take benefit of the new software technology, the Integration Platform (IP) concept has been proposed. IP is a complete set of support tools for rapid application system development and application integration in order to reduce the complexity of CIMS implementation and to improve integration efficiency. By providing common services for application interaction and data access, IP fills the gaps between the different kinds of hardware platforms, operating systems, and data storage mechanisms; it also provides a unified integration interface, which enables quick and efficient integration of different applications in various computing environments. At the application level, IP provides tools and application prototypes for each specific application domain to help the building of various CIM applications. Several IP products have emerged though their coverage is limited. One major reason is that, generally, they only provide an Application Programming Interface (API) for users, so that the user still need to spend much time in coding, compiling and integrating existing applications.

In the last 10 years, the China national high technology R&D plan has supported many research projects in developing CIMS application software. These software products cover the fields from decision making, production planning, to CAD/CAPP/CAM and shopfloor scheduling and control. Although these application products play an important role in enterprise management and production automation, their integration into one CIM system is still a difficult task. This situation leads to the demand for developing an IP to integrate them together to form a comprehensive solution for CIM implementation in Chinese enterprises.

In this paper we will present MACIP (CIMS Application Integration Platform for Manufacturing Enterprises), an Integration Platform for CIM implementation in manufacturing enterprises. The objective of MACIP is to provide a powerful and convenient integration environment, which consists of common services, standardized interfaces, integration mechanism, prototypes and tools.

Next, we review the evolution of IP technology, and summarize its current status. Section 2 introduces the architecture of MACIP. Section 3 provides a description of each of the MACIP components. Section 4 discusses in detail a key component of MACIP, the operation management and control module. The implementation issues are addressed in Section 5, including the description for the implementation environment, the realization of Communication System and Global Information System, Agent mechanisms of Operation Management and Control system, the realization of workflow management module, and Object Oriented development method used. In conclusion, some points for future study and the application prospect of MACIP are presented.

1.1. *The evolution of integration platform technology*

The Integration Platform has evolved through a number of stages. The early concept about IP was to regard it as an application programming support platform, which provided a common set of services for application integration through API. A typical structure of the early IPs is the System Enabler/ Application Enabler architecture proposed by IBM, shown in Fig. 1. Under such a structure, the IP provides a common low level set of services for the communication and data transfer (the System Enabler), and also provides application domain specific enabling services (the Application Enabler) for the development of application systems. Thus application developer need not start from coding with the operating system primitive services. The IP products which are built on this and similar structure including the DAE of IBM, BASEStar of DEC, Monitrol of Hilco, etc.. One of the disadvantages of the early IP products is that they only provide support for one or a limited number of hardware and operating systems, and the problem of heterogeneous distributed computation was not addressed. Also the released products often covered a specific domain in the enterprises, for example shopfloor control. These early IPs focused mainly on the support for the development of application software, and their support for application integration was rather weak.

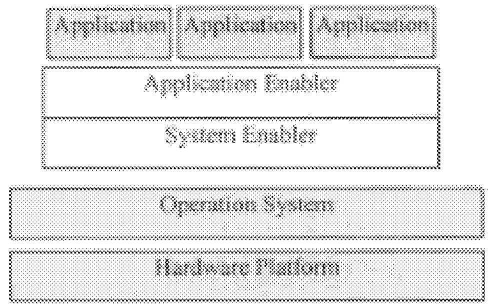

Fig. 1. IBM System Enabler/Application Enabler Architecture.

From the beginning of 1990s, IP technology moved into the phase of supporting wider application development and integration in an heterogeneous distributed environment. To meet this requirement several new concepts, paradigms, and specifications were introduced into the conceptualization, design, and development of IP, such as Middleware, Client/ Server architecture, Object-Oriented methodology, Open System Interconnection (OSI), Distributed Computing Environment (DCE), Common Object Request Broker Architecture (CORBA), etc. Based on these advances innovated IP structures have been proposed. An example is shown in Fig. 2, where the architecture is divided into several layers,

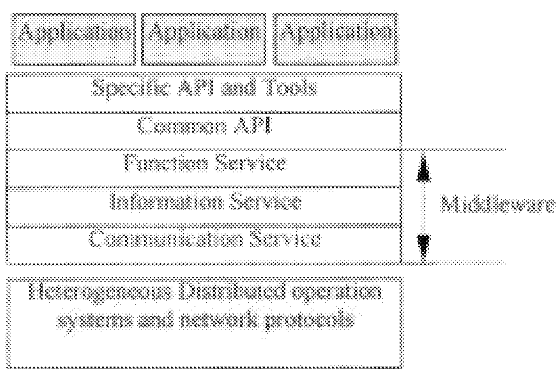

Fig. 2. A multi-layer IP structure.

communication layer, information management service layer and function service layer providing commonly used system-level services. These services form the middleware layer of IP, and the higher layers of IP are classified as general-purpose API, domain-specific API, and application development integration tools. The integration supporting area is extended from specific domain to the whole enterprise, including management, planning and manufacturing execution.

1.2. *The role of IP in CIM architecture*

IP play an important role in the enterprise CIM architecture. We discuss the issue in the framework set by CIM Open System Architecture (CIMOSA) (CIMOSA, 1993a). CIMOSA defines three orthogonal dimensions for CIM system evolution, i.e., derivation, generation, and instantiation dimensions. We examine the extent to which the IP supports each of these dimensions. From the derivation dimension, which consists of three levels: requirement definition, design specification, implementation description, the IP is concerned with the implementation level. It decides the execution description manner of enterprise functions and affects the design specification modeling. From the instantiation dimension with three levels: generic level, partial models, and particular models, IP provides standard building blocks for specific enterprise domains. For the generation dimension, IP directly deals with the function view, the information view, and also refers to the resource view, and the organization view.

From the CIMOSA life cycle view, IP supports the system building, operation and change phases in the life cycle. It also has an impact on the design phase since building CIM systems based on an IP requires changes in the system design paradigm. A complete IP acts both as an enterprise operational environment for real function execution, and as an enterprise engineering environment for system component building.

CIMOSA presents its Integrating Infrastructure (IIS) (CIMOSA, 1993b), which specifies standard common system services for general use. These services include business (process) integration, application integration, and physical system integration, We can see these are mostly system-level services provided by IP. We can conclude that IP covers main functions of CIMOSA IIS, and provides higher-level application-oriented services.

1.3. *Research trend*

Various IP technologies and frameworks have been studied, such as the ANSA Ware conducted by ANSA and Esprit/ISA project (Pleinevaux, 1994), the CCE-CNMA Esprit project (CCE-CNMA, 1995), the IBM CIM framework, the CALS-IIWG/EGP (CALS, 1993) enterprise integration study, etc. From the studies currently being carried out, and taking into account the requirement from enterprise applications and the advance of software technologies, several basic features of the IP development can be drawn. They are:

(1) providing support for CIM implementation throughout the life cycle, and the mechanisms for legacy system integration. IP provides the methods and the associated tools to support all phases in the CIM life cycle, one or more IP based CIM methodologies are presented to support CIM implementation, IP also addresses the problem of legacy system integration, which is a great problem within an enterprise.

(2) Supporting data integration, function integration, and process integration. IP will seek a solution to meet the rising demand for flexible function integration and process integration. Tools such as the application-oriented integration interface, the business process modeling and enabling system are provided or improved by future IP products.

(3) supporting standardization and open architecture, and providing integration infrastructure for component reusability.

(4) effective management of the platform. The safety, reliability, and integrity are of vital importance to an enterprise information system. The IP products should provide standard system administration tools to help the administrator efficiently manage the user accounts, data, software and physical components.

(5) adaptation to Internet/Intranet technology. In the near future global manufacturing will be feasible through the Internet, and now Intranets are being used to build information network within an enterprise. IP products should provide those key components, such as the firewall, the Web-base information server builder, etc. to support these kind of applications.

2. MACIP system architecture

In this section we discuss the MACIP. The MACIP project is designed to develop a research prototype of an application platform oriented to the new IP technology described above.

The design of an IP system architecture is closely related with its objectives. MACIP is an IP which supports not only the development and integration of applications, but also the operation of CIMS. The MACIP system architecture is presented in Fig. 3. It is a Client/Server structured, Object-Oriented platform with a high degree of flexibility:

MACIP consists of two layers; system enabling level and application enabling level. The system enabling level is itself composed of two functions; communication system and Global Information System (GIS). The primary function of these components is to allow for the integration of applications in a heterogeneous distributed computing environment, i.e., different operating systems, network protocols and database management systems. The communication system provides a set of services that allow for the transparent communication between applications. The global information system allows for applications to have a common means for accessing data sources in a variety of databases and filestores. These functions are implemented in the form of Application-Independent-API (AI-API). Application independence means that these functions are not designed for specific applications, but are general services for communication, data access and file management. Hence the system enabling level realizes the basic integration mechanisms for information and application integration.

The application enabling level, which utilizes the functions contained within the system enabling level is composed of three domain Sub-Integration Platforms (SIP); MIS SIP, CAD/CAM/CAPP SIP and shop-floor control SIP. Each SIP is designed according to the requirements of a domain application and provides functions for applications in the form of Application Dependent-API (AD-API). The AD-API functions are designed specifically to enable the quick and easy development of domain specific applications. These functions enable the complete integration of the application. Application Development Tools (APD-tools) are developed using the AD-API. Users can also develop applications using the functions provided by AD-API. Existing applications are integrated by modifying its data exchange interface using AD-API functions. An Internet interface is also included in the application enabling level interfaces, and provides the access to MACIP through appropriate Internet technologies.

An operation management system was also designed which uses AI-API functions to provide an Application Management-API (AM-API) for the users., Users use AM-API to develop management applications which manage the IP resources and coordinate the operation of different applications.

MACIP provides an Object-Oriented information

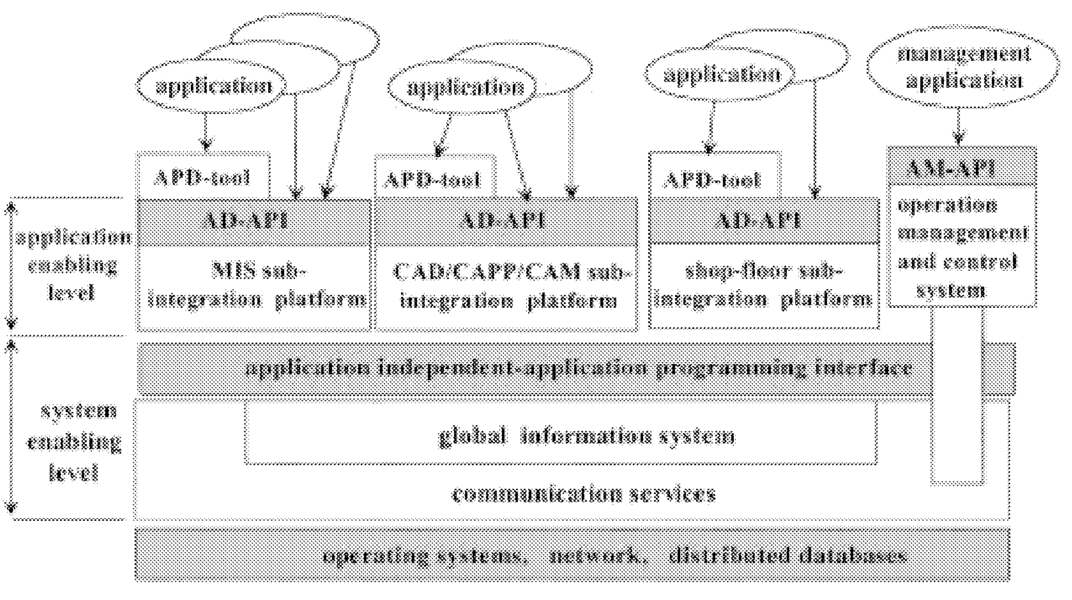

Fig. 3. MACIP system architecture.

presentation which is maintained by a GIS. GIS defines the object model for global information access and provides applications with basic object definition and object operation functions. Application can use these functions to build their information storage structures and access data through these structures. The basic object interface for O-O data handling is the database (logic) schema. The O-O model also regulates the transaction processing, such as carrying transactions across different databases, processes or threads; managing transient objects in transaction semantics, and concurrency control. The MACIP O-O information model conforms to the Object Database Management Group (ODMG) standard (ODMG, 1993), with extensions to include multiple database and file systems. Because the underlying database is still a relational database (RDB), the GIS model provides drivers to connect to different kinds of RDB. For the embracing of distributed file systems, the corresponding object interface is built based on the Communication System. GIS provides the user with an integrated view of all the information through schema integration and inter-operation between different RDBs and file systems. The MACIP O-O model is defined using the Object Definition Language (ODL) and operated using Object Query Language (OQL). The O-O model can also be embedded into various programming languages, such as C++, as dynamic linked libraries. The domain SIP and application use the basic GIS O-O model to construct application oriented O-O models. Figure 4 shows the GIS structure.

3. System functions

As shown in the system architecture, MACIP consists of the following function modules: a communication system, a global information system, three domain application sub-integration platforms, an Internet interface and an operation management and control system. We group these functions into three kinds: system, application and management.

The system functions are provided by the communication system and GIS. The communication system provides three kinds of services:

(1) basic communication services: data communication functions that are network protocols and operating systems transparent.

(2) common services: including naming service, message service, file access and management service, network management service, etc.

(3) network management: the Simple Network Management Protocol (SNMP) is used to manage network load, error handling and safety.

GIS provides applications with a unified information view and information services. Its major functions are:

(1) functions for integrating different information sources (RDBs and file systems).

(2) functions for the definition and maintenance of the global information model.

(3) functions for providing the transparent data access interface (GIS-API).

(4) functions for providing global sharing of information definitions and maintenance. These functions will be used by the users to define interface for existing information resources, then integrate the existing information system without changing the programs.

In MACIP, there are four groups of application functions: CAD/CAPP/CAM SIP, shop-floor control SIP, MIS SIP, and Internet interface.

Figure 5 shows the CAD/CAPP/CAM SIP structure. An engineering information management system was developed using the O-O product information sharing model to manage product design and process planning data. It also provides the integration interface with CAx application software and the interface with other application domains (MIS and shop-floor control). The integration of CAD with the CAPP system is achieved through a STEP based product data share model. Drawing eXchange Format (DXF) and Initial Graphic Exchange Specification (IGES) based interfaces are also designed to transfer two-dimensional design information between different CAD systems. To address the integration with the Product Data Management (PDM) system, an interface between the PDM software and GIS is designed in CAD/CAPP/CAM SIP. The interface is used to integrate PDM software with MIS (or MRPII) software to facilitate the integration of the whole design-production process.

Figure 6 shows the shop-floor control SIP structure. The objective of the shop-floor control SIP is to provide a flexible and agile structure for users to develop suitable shop-floor control and management

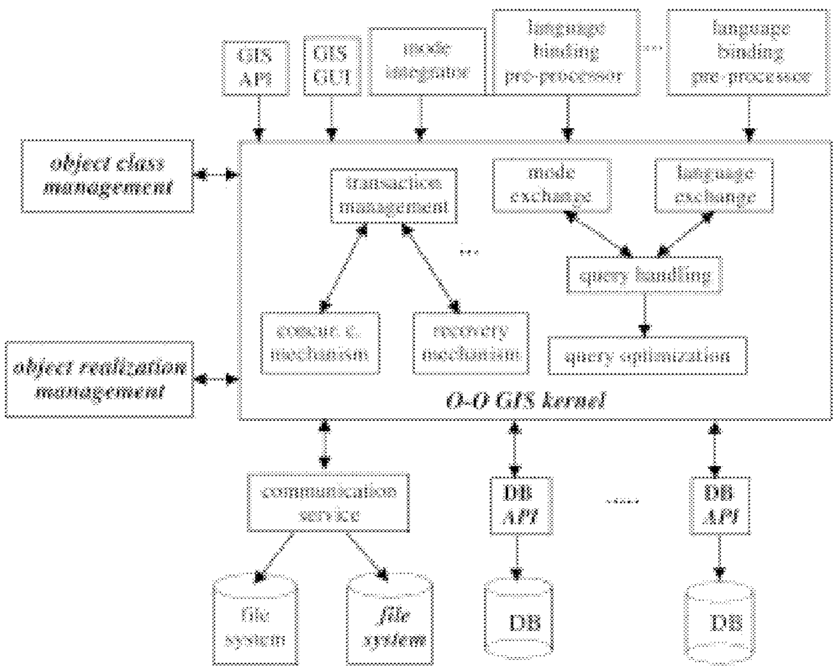

Fig. 4. GIS structure.

systems. Besides the data access services provided by GIS, this SIP includes a Virtual Machine Device (VMD) engine based on MMS (Manufacturing Message Specification) protocol. The VMD interface is used to integrate any MMS protocol compatible devices such as machine tools and transportation vehicles.

The shop-floor control SIP includes a control kernel that controls and coordinates all shop-floor control

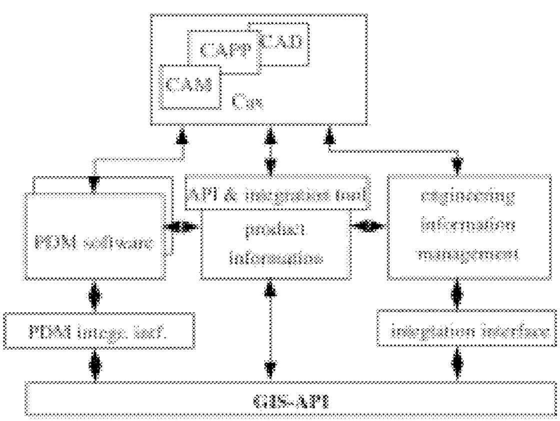

Fig. 5. CAD/CAPP/CAM SIP structure.

applications. The shop-floor control SIP also includes an integrated modeling tool based on Petri net modeling theory (Venkatesh and Zhou, 1998; Zhou and DiCesare, 1993) and has an object modeling interface. The graphical interface enables engineers to easily and quickly model the shop-floor system without prerequisite knowledge of Petri Net theory. An EXPRESS standard based neutral file sharing mechanism transfers modeling data between the Petri net and the objective model. The work-form data management is used to integrate shop-floor real-time data connected with the control of manufacturing devices. The development tool is used to develop shop-floor applications using the functions provided by the communication system and GIS. The control kernel, modeling tool, work-form management and the development tool together provide a user with a complete support system for developing shop-floor controllers.

The central function of MIS SIP is to provide an integration interface, the GISGlue, which takes the role of connecting GIS with various MIS applications. GISGlue addresses integration with different MIS applications: the most important is the integration of MRPII systems which handle the central control of the

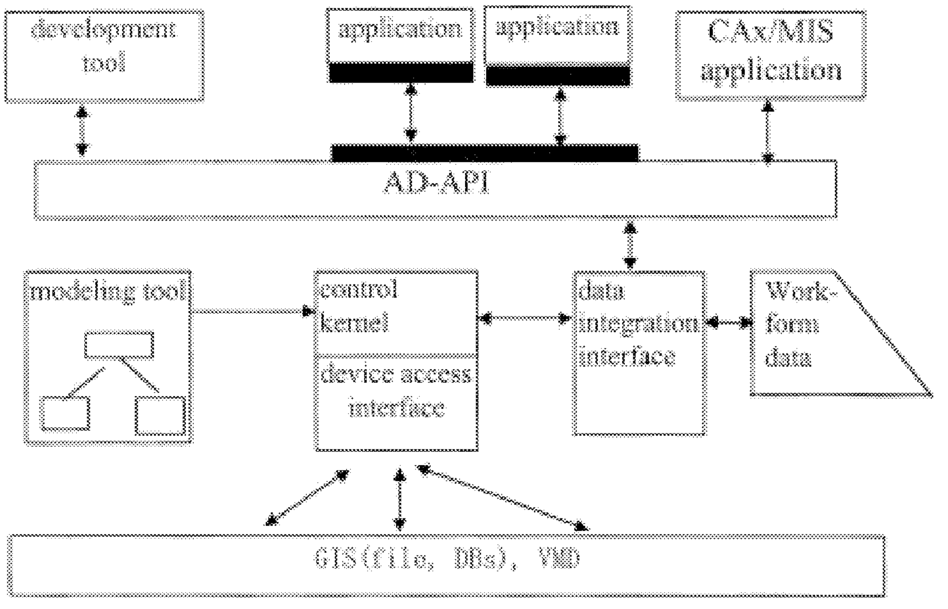

Fig. 6. Shop-floor control SIP Structure.

production planning. MIS SIP also enables the embodying of GIS data into several commercial document/report generation software. A MIS development tool is integrated in the MIS SIP. Based on this tool a MIS system can be built directly on MACIP, whose data is stored/retrieved through GIS.

The Internet interface aims to provide for Internet access to MACIP from both inside and outside. Its primary function is to build a Web access interface to GIS, so that applications can access GIS data through the Web. The access interface also includes the security control mechanisms which exclude any unauthorized data access. On the Web access interface, an internal Web-based enterprise information system framework is built using Web servers and browsers. It provides typical production data presentation to MACIP users. The information access request from outside is authenticated first by the security control mechanisms. If authenticated, it is then accepted through a special entrance.

MACIP management functions are realized by the operation management and control system which include:

(1) application and resource utilizing management, such as task running monitoring and control, network monitoring and management, resource (database, file system, device) management, system execution performance evaluation, etc.

(2) access and running control, including: MACIP configuration and tailoring, security management, urgent message broadcasting or notification.

(3) workflow management for process integration and application coordination.

4. Operation management and control (OMC) module

In this section we discuss the MACIP OMC Module. It is one of the components introduced above. Since it plays a significant role in the system management and integration, we will address it in some depth. In MACIP, the OMC module aims to provide the administrator with consistent function for the management of users, the application and the resource, and it is also in charge of the coordination of the application running on the integration platform. The OMC module covers three aspects: by imposing global management on the user accounts, data, resources, and the application access authority, it enhances the system's security and reliability; by monitoring the running application status and the resource utilization, it helps to improve the system execution performance; by equipping MACIP with process modeling tool and workflow enactment services, it can coordinate the interaction of related applications running on MACIP, and more importantly, it provides the user with ability for business process establishment and execution.

The OMC module is divided into two sub modules: operation management and workflow. The operation management module executes the system management function mentioned above. Its structure is shown in Fig. 7:

The functions of the operation management module are grouped in two classes: the application and resource utilizing management, and the access and running control.

The application and resource utilizing management include:

(1) task monitoring and control,

(2) network monitoring and management, load balancing, communication conflict detecting and solving, time delay monitoring, etc. This function is realized by utilizing the APIs provided by the Communication System,

(3) resource (database, file system, device) management,

(4) global system execution analysis, collecting and analyzing the running data, getting performance evaluation of each node, each database and applications.

The access and running control include:

(1) configuration, tailoring, and updating of MACIP components,

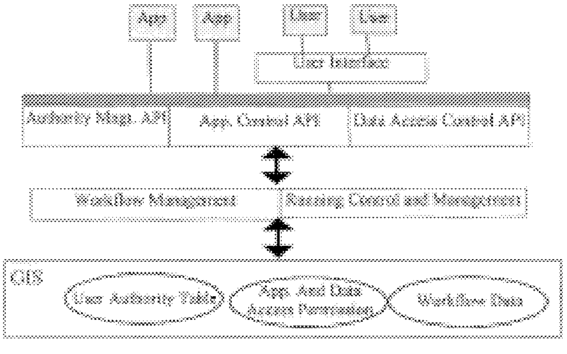

Fig. 7. MACIP Operation Management and Control Module.

(2) maintenance of the global safety architecture, management of the user authority, resource access permission control,

(3) urgent message broadcasting or notification,

(4) system backup/recovery,

As a mechanism for process integration, a workflow system has been introduced into various systems, such as the PDM system, the document management, etc. It provides users with the capacity to manage persons, applications and data in a dynamic process view. In MACIP we also provide a distributed workflow management module as a major component of process integration mechanism. The choice of placing the workflow management module in the OMC module was because: (1) it provides common business process service for all applications on MACIP, (2) it simplifies the workflow execution management, (3) the system management processes are those workflows mostly intensively executed in a CIM information system.

The workflow management module is comprised of two components: process modeling tool and workflow execution tool. The process modeling tool provides interface with workflow model builder, which abstracts logic constraint and information dependency from real world processes to construct computerized models. These computerized models are used as the templates for workflow instantiation at the execution phase. To be accessible on different platforms, the model uses a neutral language which can be interpreted everywhere, in the model there are also references to the resource type and organizational entities. These models are stored/retrieved through the GIS, and shared by a workflow execution component.

The workflow execution components perform the enactment of the processes. It generates workflow execution instances and manages its running through interaction with workflow participants (user, application). The execution module is divided into 3 parts: execution engine, supervisor facility and client side interface. The execution engine performs workflow execution operations, such as flow logic interpretation and flow progressing, workitem assignment, and user reply, etc.. It also maintains the state data for each workflow and other relevant data. In the distributed environment, the workflow execution engine is implemented by several coordinated agents that reside on different machines.

Another component of the workflow execution module is the supervisor facility. It provides the user with ability to initialize the workflow instances, allocate resources, monitor and control the workflow execution, and solve conflicts and problems. According to different requirements, the supervisor facility could be configured and assigned to different type of users, such as a system administrator, and process manager. The third part are modules that interact with workflow participants, including management of worklist, user interface, and interface with automated application. The user interface is placed in the MACIP user's general working interface for daily workflow operation. The interface with the automated application will be activated when the workflow execution component is interacting automatically (without human interference) with the application.

5. Implementation

We now discuss some implementation to gain a deeper understanding of MACIP.

5.1. *Development environment*

The prototype of MACIP is built on a multi-vendor heterogeneous computing environment as depicted in Fig. 8:

A TCP/IP based Local Area Network (LAN) is used to connect various kind of servers, workstations, and desktops. On the application layer, the MMS protocol is used in shopfloor control SIP. Different kinds of computers are included: a Shu-Guang server, a SGI Origin 200 server, a PC-LAN server and a number of PC desktops. The operating systems include: SGI/IRIX, Windows NT Server/Workstation, and Windows 95. Different RDB Management Systems (RDBMs) (Oracle, Sybase and SQL-Server) are installed to manage various kind of data. This configuration presents a rich heterogeneous environment.

Various MACIP components are distributed in this heterogeneous environment. For those common components such as Communication System, Global Information System, and Operation Management and Control tools, the relevant parts are installed on every node; the associated control modules, such as the OMC-Agents, reside on assigned servers and workstations, and their user interface is configured on every necessary computer. The SIPs and tools are installed on different computers according to their specific requirement, MIS SIP on the PC-LAN Server, which runs on the Windows NT and SQL-Server, CAD/CAPP/CAM SIP on the SGI Origin 200 server (having Oracle installed as DBMS), and Windows 95 desktops, Shopfloor Control SIP is built on two

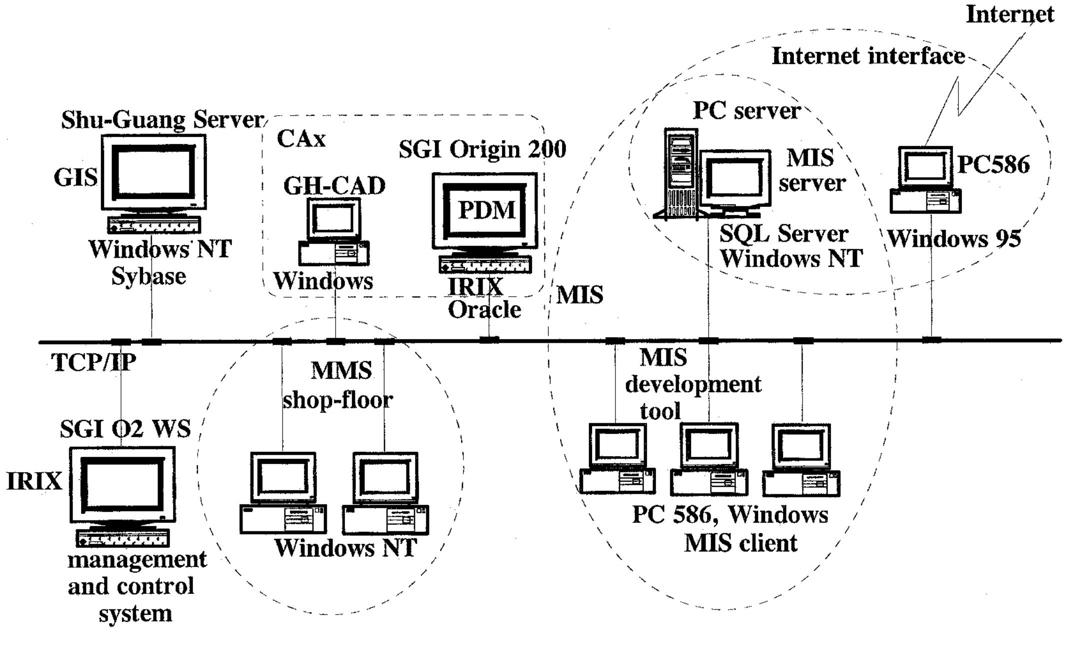

Fig. 8. The MACIP Develop Environment.

Windows NT workstations. The Internet interface is provided via a machine acting as Web-server. Different tools will be integrated into MACIP, and installed on machines they require.

The development environment of MACIP is diverse, such tools as C + + compiler, database management tools and client-side develop tools, GUI libraries, the Web-Server builder, are all used at different levels. The CASE environment for Object-Oriented analysis and design is also built. We use Rational Rose products together with PVCS as a Configuration Management system.

5.2. *The realization of the communication system*

The Communication System is realized in two layers: Local Communication System (LCS) and Common Service (CS). The LCS provides transparent data transportation between processes across heterogeneous platforms. Using commonly agreed data resentation format, typical communication session is: to apply for transportation port and system connection, send data, get response, destroy connection, release transportation ports. In order to provide more flexibility for CS functions, LCS provides two manners of communication: address-based communication and name-based communication.

The interactions with high-level components are carried out by CS. CS uses a client/server based architecture as shown in Fig. 9. The application request service to CS request handler. The request handler locates CS service provider and get the client application requested data. If the request handler and the service provider are on the same machine, they exchange data directly. If they are on different machines, communication is carried out through LCS.

The CS provides four types of services: naming, message, file access, and network management. In naming service, general mechanisms such as, name binding/resolving, name structure and its organization are implemented. Several name servers work in cooperation to complete the tasks. The name is organized in a name tree, and the naming manner is the same as the Internet naming string. In the name tree, all the step nodes represent the naming context, and the leaf nodes represent named entities. For the named entities, we do not interpret its meaning. Applications will do this work themselves. The message service presents application entities with a common message send/reply mechanism. The message data structure is as follows:

Sender	Receiver	Length	Content···

Fig. 10. Message structure of CS.

The content field is non-structured, whose definition is specified by high level modules according to a common definition rule. The file access service provides a common file property description and the lock mechanism on the base of different file system access control provided by the operating systems. For the file lock/unlock function, three kinds of persistent lock are built, read, write, read and write. File access service also realizes a basic file version control. The network mangement service is realized by integrating a commercial network management system. It fulfills the task of configuration, accounting, error handling and safety control based on the Simple Network Management Protocol (SNMP).

5.3. *GIS implementation*

As GIS is built utilizing RDBMS, the key point to be solved is how to transform the O-O data request to a

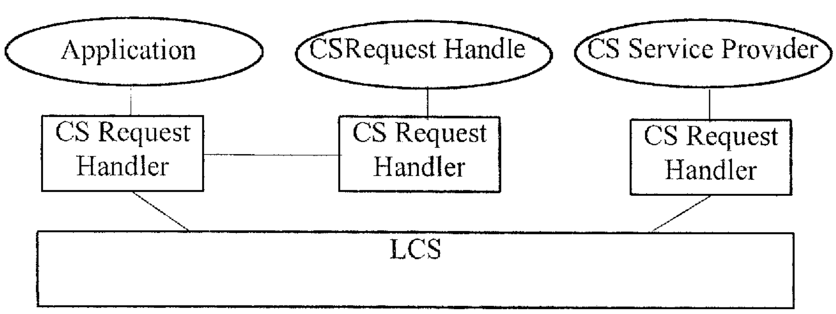

Fig. 9. Communication System Client/Server Architecture.

set of relational data requests and send these request to the appropriate RDBMS. In GIS this function is carried out by the GIS control module. When GIS receives the data request, its control module calls the language interpreter to translate the request into an API-format request script. A object/relation schema converter is called to convert the O-O request script to the relation-type schema, and finally these requests are sent to the associated RDBMS for processing. The management of GIS objects is carried out by GIS control module. Lower-level processing is completed using relational mode. This arrangement has two advantages: first the actual data access is still carried out by underlying RDBMS, so that the performance is not affected significantly. Second the object operation handling is confined within the control module, and is independent of lower level mechanisms. These features improve the flexibility for extension and future modification.

GIS provides object mechanisms for database access. In order to provide unique object access to file systems, a file system class library is implemented. The processing mechanism is similar to that of the database.

The handling of transactions is realized as active objects whose state and properties are modified in a transaction context. These objects contain or refer to persistent data objects and it is possible to request more than one active objects in a single transaction.

5.4. *The agent mechanism of the operation management and control module*

The OMC module is implemented through a multi-agent mechanism. Each agent controls the system resources and applications in the domain it maintains, usually a machine within the platform. All the agents cooperate to monitor and manage the whole system. An agent can be described as a software entity which functions continuously and autonomously in a particular environment, often inhabited by other agents and processes (Shoham, 1997), aims at simplifying the distributed computing and providing intelligent interoperability, and presenting a better user interface (Bradshaw, 1997). The construction of the OMC module through coordinated agents enhances its flexibility, and facilitates the efficient distribution of OMC parts.

In our OMC system, the OMC-Agent is defined as an autonomous entity which can execute the tasks of system operation monitoring, network management, user management, GIS management, and sharing devices management by automatically invoking proper applications. OMC-Agents are distributed on the nodes in MACIP. They communicate through an Agent Communication Language (ACL) to carry out their functions. The struture of these agents follows a peer-to-peer manner. Although a coordinating agent (a special case of agent) takes the extra charge of updating the global user authority table and the access authority table, it also notifies other agents of the current system status. Each agent controls the management and applications in the domain it supervises. These applications carry out the actual management operations when they are invoked by the agent. This structure facilitates the integration of various management functions. Users only need to associate the application tools to an agent, provide its capacity description to the agent's capacity description table, and register in the agent's service rule list, then its function can be exploited by the system through the request to that corresponding agent. The OMC-Agent relationship is shown in Fig. 11.

The OMC-Agent data consists of three parts: the capacity description, the status description and the service rules. The capacity description provides a formalized description of the services provided by the agent, and the constraints (manners, parameters, etc.) for requesting these services are set in the service rules. The agent's current status, such as the tasks in process and their state, is given by the status description.

The communication between agents is realized through an Agent Communication Language (ACL). ACL is an application level language, which allows for the specification of both syntactical and semantic aspects. Agents can use the ACL to query and request services from other agents. By standardizing the ACL format, various implementations of agents can negotiate and exchange knowledge freely, and the development of agents is simplified.

The ACL is implemented on the transport-level protocol provided by the communication systems. A specific port is used for the transfer of the ACL, because it may carry time-critical tasks. Our ACL is used only in the OMC module. Its semantics is rather simple. For the convenience of implementation, we built our own ACL. It may be more flexible to realize it as a subset of a standard ACL, such as Knowledge Query and Manipulation Language (KQML) (Finn, 1997).

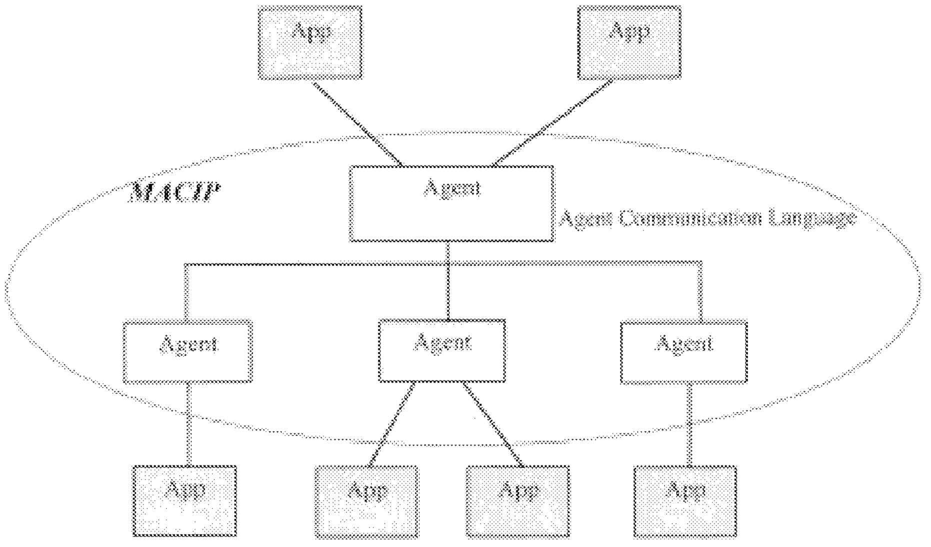

Fig. 11. OMC-Module Agent structure.

5.5. *The workflow management module*

The workflow management module is constructed with reference to WFMC workflow reference model (WFMC, 1994), which specifies the architecture of the workflow management system, its components, and their functions. The organization of components and some interfaces have been modified: the organization/ role data is managed by other MACIP components, and an interface is defined for all workflow components to access them; the automated invocation of applications is performed by the worklist handler. The direct interface between workflow enactment service and the applications are illustrated; for the invocation protocol, an Application Coordination API is built, which is a set of class libraries which can be called by the application program to connect to a worklist handler, and to send/receive task execution message; the interface between workflow supervisor and the workflow enactment service is based on the Workflow Management Coalition/Workflow API (WFMC/WAPI) (WFMC, 1996), the same as the interface between workflow client and the workflow enactment services. The realization of the workflow management module can be regarded as a simplification of WFMC reference model, as shown in Fig. 12.

An extended Petri Net language, the CIN (Control and Integration Net (Fan and Spar, 1997), has been adopted as the modeling language for workflow schema. CIN is a type of Colored Petri Net with extensions in the time transition, uncertainty arc, and procedure transition. Its semantics is rich, which leads to a simpler model description. It supports hierarchical and modular modeling, which helps to build nested workflow schema. The CIN model can be simulated and it is easy to perform control while it is executing. The basic modeling idea is: use transitions to represent workflow activity, use token and colors to identify the status and execution logic.

The instantiated workflow entities, such as process, activity, workitems, are designed as persistent objects, which are stored and retrieved through GIS. The

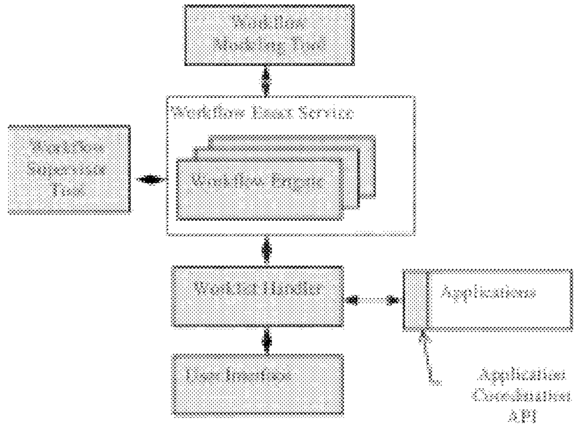

Fig. 12. Workflow Management Module structure.

communication between different workflow engines is realized through services provided by the Communication System. When the worklist handler carries out the invocation of applications, it requires the naming service provided by Communication System to locate the right applications. The workflow user interface is inserted into every user's GUI interface. The lower level processes such as the distributed transaction processing, the integrity of the workflow data, are handled by the GIS.

5.6. *Object oriented development*

O-O Analysis and Design methodology is extensively used in the design and development of MACIP. Booch method (Booch, 1991) is chosen as our main tool for analysis and design. From the implementation view, the method should be easy to use and be adaptive to environment, Booch method has its abundant semantics to model real world software systems, and its coverage of both the analysis and design phase made it a good choice.

The Booch method emphasizes two design processes at different levels (Booch, 1991), the macro process controls the steps of the general development process as conceptualization, analysis, design, evolution and maintenance. In the framework set by these steps, the incremental developing process, called the micro process, is carried out iteratively. It consists of such steps as identifying the classes and objects, identifying the semantics of classes and objects, identifying the relationships among classes and objects, and the implementation of classes and objects. In the analysis and design process, Booch notation is fully exploited to build our system architecture model, though we have made some modifications to meet our requirements. For example, MACIP is designed to be independent of supporting platforms. For the Booch notation of a process diagram, which is used mainly to illustrate the physical architecture of the system such as the hardware configuration, we modify its semantics to present the higher level supporting structures, such as a computing node with Operating System or DBMS installed.

The Rational Rose CASE tool family of Rational Rose Ltd. is used to support development work. The analysis and design tool used is Rational Rose C++, together with the configuration management systems as PVCS of Intersolv Ltd. They provide a parallel development environment for teamwork. For the document generation, Rational Soda document generator is used.

Recently Object Management Group (OMG) has adopted Unified Modeling Language (UML) as their standard for modeling. It is our intention in the future to convert our system models into UML notation, Since the UML and Booch method have a well defined mapping between their notation, this works should not be difficult.

6. Conclusions

In this paper a comprehensive study of IP technology has been presented. The system architecture, the function of its components are discussed. The implementation techniques for some of the modules are presented. The Object-Oriented database model, the agent-based inter-operation, and workflow methodology are used to reach MACIP objectives: the flexibility for application integration, the standard interface, and the open architecture. By providing various domain-specific SIPs, MACIP is more convenient to put into practice, and will help the CIM developers to speed and normalize their work. The preference to practicability will make it more acceptable to manufacturing enterprises.

The development of MACIP presents several new topics for study, for example:

(1) the scalability. If the platform components can be easily configured to fit in the requirement of enterprises with different scale and different purposes.

(2) Provide support both in methodology and tools for the successful application of MACIP. This will help the transformation of enterprise requirement to MACIP functions.

(3) In MACIP we have intentionally organized the enterprise information system into several related subsystems, this can be seen from the division of the application level component into several SIPs. This approach is different from traditional hierarchical view. We believe the flat architecture with several coordinated modules with proper autonomy will be more efficient when coping with manufacturing complexities. We are looking for means of introducing more autonomy into MACIP components.

Currently MACIP has integrated many application tools. There will be more tools to be integrated. We are building the interface of SDRC/Metaphase PDM system to our GIS, and extend its information model to meet the requirement from other SIPs as the MIS-SIP. Some CAx tools, such as the Pro/E, the GH-CAD, will also be integrated in CAD/CAPP/CAM SIP. For shopfloor control, we integrate a FMS workstation control and integration toolkit as well as a shopfloor scheduler. The MIS SIP will realize the interface for two MRPII systems: the LM-MRPII, and the JW-MRPII, it will also integrate RADISS, a MIS development tool, to provide developers with rapid MIS building capacity.

MACIP will be applied in real enterprise for case studies. The experience obtained from the case study will be accumulated and used to improve MACIP. When MACIP is finally relesed, we aim to apply it to various kinds of Chinese manufacturing enterprises to speed up the CIM system implementation.

References

Booch, G. (1991) *Object-Oriented Design and Applications*, Benjamin/Cummings, Rewood City. CA,

CALS-IIWG/EGP group. (1993) *Profile for Enterprise Integration*, PB93-203560.

CCE-CNMA Consortium. (1995) *CCE: An Integration Platform for Distributed Manufacturing Applications*, Springer-Verlag, Berlin.

ESPRIT Consortium AMICE. (1993) *CIMOSA: Open System Architecture for CIM*, Springer-Verlag, Amsterdam.

Section 2-4. (1993) *IIS Concepts & Structure*. ESPRIT Consortium AMICE. *CIMOSA: Open System Architecture for CIM*, Springer-Verlag, Amsterdam.

Pleinevaux, P. (1994) *Comparison of CCE, DCE, BASEStar and ANSAWare*, Esprit Project ''CCE-CNMA'' Management Committee Report.

Object Database Management Group. *ODMG standard*. fttp://www.odmg.org/

Venkatesh, K. and Zhou, M. C. (1998) Object-oriented design of FMS control software based on object modeling technique diagrams and Petri nets. *J. of Manufacturing Systems*, **17**(2), 118–136.

Zhou, M. C. and DiCesare F. (1993) *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer Academic Publishers, Boston, MA.

Yoav Shoham. (1997) An Overview of Agent-Oriented Programming. In: *Software Agents*, edited by Jeffrey M. Bradshaw, MIT Press, pp. 271–290.

Bradshaw, J. M. (1997) Introduction to Software Agents. In *Software Agents*, edited by Jeffrey M. Bradshaw, MIT Press, pp. 3–48.

Finin, T. *et al.* (1997) KQML as an Agent Communication Language. In *Software Agents*, edited by Jeffrey M. Bradshaw, MIT Press, pp. 291–316.

Workflow Management Coalition. (1994) *The Workflow Reference Model*, [WfMC1003].

Workflow Management Coalition. (1996) *Workflow Management Coalition Workflow Client Application (Interface 2) Application Programming Interface (WAPI) Specification*, [WfMC-TC-1009].

Fan, Y. and Spur, G. (1997) A Control and Integration Net Method for Manufacturing System. *Control Theory and Its Applications*, **14**(3), 306–312.