CHAPTER 6

AN INTEGRATING INFRASTRUCTURE FOR IAS

6.1 INTRODUCTION

CIM can be defined as the use of IT in Manufacturing Technology to favor communication, coordination and cooperation of human and other heterogeneous and distributed manufacturing system components (machines, applications and information systems) in order to improve overall productivity and efficiency. The new advances in Advanced Manufacturing Technology and IT have put more demands for CIM integration technology and associated supporting tools. One of these demands is to provide CIM systems with better software architecture, more flexible integration mechanisms, and powerful supporting platforms.

A complete integration paradigm includes three levels of integration (Spyros 1997) covering physical system integration, application integration and business integration leading to the total integration of the manufacturing enterprise, called as integrated automation system. This approach is applied in an integrated way to all activities from designing to delivery and after sales and uses various methods, means and techniques in order to simultaneously improve productivity, decrease cost, meeting with due dates, increase product quality, secure flexibility at local and global level in manufacturing system and involve every action. IIS provides a complete set of support tools for rapid application system development and application integration in order to reduce
the complexity of CIM system implementation and to improve integration efficiency. By providing common services for application interaction and data access, IIS fills the gap 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, IIS provides tools and application prototypes for each specific application domain to help in building various CIM applications. Several IIS products have emerged though their coverage is limited. One major reason is that, generally, they provide only an Application Programming Interface (API) for users, so that the user still needs to spend much time in coding, compiling and integrating existing applications.

The AMICE project defines the functionality of the integrating infrastructure services (ESPRIT 1993). IIS provides a structured set of system wide services thereby avoiding redundancy of function in the enterprise's systems. The IIS supports integration by providing homogenous access to system components. It also provides a unifying software platform to achieve integration of heterogeneous hardware and software components. Information systems at each level of the enterprise provide pieces of the overall picture of how a business functions. Information must flow automatically between systems and users must have fast and easy access to the right information regardless of where it resides. In this chapter an IIS for CIM implementation in manufacturing enterprise to develop an integrated automation system is presented.
6.2 THE EVOLUTION OF IIS TECHNOLOGY

Fundamentally, an information system is developed by creating a knowledge base and a set of user interfaces. With software development tools, information systems are developed through a modelling process by creating and manipulating information system models from which the resulting system is generated. The need to integrate information from different applications of an enterprise lead to the information integration concepts and later the development demands the use of a common information integrating infrastructure. The early IIS focused mainly on the support for the development of application software, and their support for application integration was rather weak. From the beginning of 1990s, IIS technology moved into the phase of supporting wider application development and integration in a heterogeneous distributed environment. To meet this requirement several new concepts, paradigms, and specifications were introduced into the conceptualization, design, and development of IIS, such as Middleware, Client/Server architecture, Object-Oriented methodology, Open System Interconnection (OSI), Distributed Computing Environment (DCE), Common Object Request Broker Architecture (CORBA), Component Object modelling (COM) and Distributed COM (DCOM) (Dhinesh et al 2002b) etc. Based on these advances, innovated IIS have been proposed. Figure 6.1 shows a multilayer IIS, which consists of communication layer, information management service layer and function service layer providing commonly used system-level services. These services form the middleware layer of IIS, and the higher layers of IIS 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.
The various integration approaches include: Technical approach using middleware, messaging, or file transfer technologies, Data integration which includes database replication, directory services, common data definitions and storage, Business Process integration think in terms of logical business processes, even when they cross the line of a particular application and User Interface approach which makes use of the web and portals which can unify the enterprise. For agility and reconfigurability, message passing distributed Multi-tier Application Architecture is needed.

However, the complexity of manufacturing systems and the lack of effective integration mechanism arise problems for CIM system 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 difficulties 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 CIM system on existing IT, and wish to retain their old investment. But they lack an effective mean of combining legacy resources into the new system.

(4) CIM system implementation is time consuming. Without powerful application-oriented support tools, the whole implementation process is long and inefficient. 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.

6.3 MOTIVATION FOR THE IIS DEVELOPMENT

For the design of such system Integration brokers, application servers and Web services provide integration capabilities. While integration brokers have historically provided messaging, intelligent routing, translation, transformation, adapters and, application servers have offered more in the way of reliability, scalability and transaction management and are evolving into full integration platforms by adding translation, transformation and some adapter capabilities. Web services are still evolving and there are still some pitfalls in
the standard in terms of scalability, reliability and management. Even though industries have more choices than ever for solving their integration problems, issues such as the flexibility for application integration, the mechanisms for legacy system integration, supporting standardization and open architecture, re-usability and effective management of the IIS require more tools and techniques to solve these issues. A possible solution to this problem is the development of conceptual framework or generic architecture for system integration, which allows the categorization of the various methods and consequently their conformities and discrepancies, can generate uniformity and mutual understanding. The IIS architecture facilitates a powerful and convenient integration environment, which consists of common services, standardized interfaces, integration mechanism, prototypes and tools for automating the development of IAS.

To optimize the operating characteristics of a manufacturing enterprise there exists major problems of unifying the goals and activities of individual processes, whilst retaining sufficient flexibility in the holistic systems formed. Here methods and tools are required which are capable of defining and constructing complex integrated systems in a manner which enables them to realize system-wide goals, where invariably these goals will continuously change with the characteristics of the consumer, supplier, labour and financial markets within which they operate. In particular, therefore, it is necessary to realize improved support for: (i) formally specifying the way in which the ‘entities’ (‘processes’, ‘computer systems’, ‘people’, ‘automated equipment’ and other ‘components’ of an integrated system) of a manufacturing enterprise should ‘inter-operate’. This needs to be achieved with sufficient realism and completeness to guide subsequent system implementation (Weston 1994a), and (ii) ‘enacting’ formal specifications so that system build and change
can be achieved in a structured and effective manner (Weston 1994b). This chapter considers ways of facilitating ‘integrating infrastructures (IIS)’, which themselves offer computational means of implementing and supporting the operation of integrated manufacturing systems. It is not the purpose of the IIS to contribute functionality, which will be contained within the inter-operating components of the systems, rather the purpose of the IIS is to facilitate an aggregation of that functionality by enabling and managing the required inter-operation. An IIS can thus be viewed as needing to satisfy the dual requirement of providing: (i) an appropriate set of integration ‘services’, which collectively underpin the runtime integration of a number of entities, i.e. it is required to offer services which enable inter-operation and (ii) a set of integration ‘tools’ which collectively define, manage and change associations formed between entities. When used in combination, the ‘services’ and ‘tools’ of an IIS can unify the various activities carried out in a complex manufacturing enterprise, whilst maintaining sufficient flexibility to allow such systems to evolve over a period of time i.e. an infrastructure can have marked benefits in terms of dealing with complexity and change. Important advantages stem from an inherent separation of ‘integration processes’ from ‘application processes’, where the former are concerned with accomplishing system inter-operation and the latter with realizing system functionality.

6.4 BASIS OF THE PROPOSAL

The issue has been discussed in the framework of CIMOSA (ESPRIT 1993). The aim of CIMOSA is to provide an open system architecture to support integration of CIM system components (ESPRIT 1993; Kosanke 1992). To this end, CIMOSA provides an Integrating Infrastructure and a Modelling Framework. The CIMOSA Integrating Infrastructure provides support for
system integration and application integration as a uniform platform made of a set of common services available to all users and functional entities on all nodes of a distributed system (Querenet 1992). The aim of the IIS is to transform the highly heterogeneous execution environment into a more homogeneous world and to ensure system interoperability on different vendor components. The four basic building blocks for EAI are: Communication model, method of integration, middleware, and services (William et al 2001). An enterprise should go through four stages to effectively implement integrated automation namely, (1) Initial Stage: Organisations must plan, learn and architect with the intent of achieving early success. (2) Architectural Stage: Organisation must create an architecture that can be understood by designers and developers and that can promote future reuse. (3) Organisation stage: Organisations must apply the results from the first two stages across the organisational expand EAI services. (4) Enterprise stage: The organisation has mastered EAI and is effectively employing and measuring the effectiveness of its EAI architecture.

6.4.1 CIMOSA Integrating Infrastructure

The CIMOSA Integrating Infrastructure is the enabling technology which makes it possible to execute the CIMOSA models i.e. to control and monitor day-to-day enterprise operations described in the model. The CIMOSA Integrating Infrastructure provides support for system integration and application integration as a uniform platform made of a set of common services available to all users and functional entities on all nodes of a distributed system (Querenet 1992). IIS provides a unified software platform to achieve integration of heterogeneous hardware and software components of the CIM system. The CIMOSA Integrating Infrastructure (ESPRIT 1993) is made of five entities as shown in Figure 6.2.
Business Entity: provides generic functions to control the enterprise operations according to the contents of CIMOSA model.

Information entity: provides generic functions for data access, data manipulation and data integration.

Presentation Entity: Provides generic functions to support the integration of enterprise components.

Common Service Entity: provides common services to other entities.

System Management Entity: provides generic functions to set up, monitor and maintain the information technology components.

![Figure 6.2 CIMOSA integrating infrastructure services](image-url)
6.5 THE DEVELOPMENT OF AN IIS

6.5.1 System Architecture of the IIS

The system architecture of the IIS is illustrated in figure 6.3. This is a Client/Server based, Object-Oriented Platform with high degree of flexibility. The primary functionality of the architecture is the integration of applications in a heterogeneous distributed computing environment with various operating systems, networking protocols and database management systems. The communication system provides a set of services that allow for the communication between various applications. The global information system allows the application to access the databases from various sources in a common way.

Figure 6.3 System Architecture of the IIS
In the application level various manufacturing applications such as CAD, CAM, CAPP, SFC, etc. are integrated together using the API. Since this architecture is a Client/Server environment, Remote Procedure Call (RPC) is utilized for application integration. Also, middleware technologies such as COM/DCOM, CORBA enable system integration in the application integration platform to share the remote information.

6.5.2 System modules

Application Programming Interface is a mechanism provided by an application to access its functionality or data. The ‘Application Service Interface’ facilitates access to the integration services provided by an integrating infrastructure. The ‘Application Event Manager’ provides a consistent mechanism for event management (such as arrival of data packets on a file descriptor, timer events, etc.). These functions can be used by the application code and are used by both the application service and integrating infrastructure interfaces. The ‘Application Code’ is provided by the application implementer, which encodes the functionality and behavior required.

Application Independent Services are general services for communication, data access and file management and are not designed for specific applications.

Middleware Technology is application-independent software that provides services that mediate between applications. A key technology that combine the functionality of an enterprise’s existing applications, commercial packaged applications, and new code by means of common middleware. Middleware Technology provides standard services for defining and
maintaining interfaces, simplifying architectures and securing applications. It provides the runtime environment to manage the requests between software components. Middleware enable the complexity of integrating applications by providing mechanisms by which applications can package functionality so that their capabilities are accessible as services to other applications, mechanisms by which applications can share information with other applications, mechanisms by which applications can coordinate business processes (Ron 2001). There are five basic type of middleware in the market today. They are Remote Procedure Calls, Database Access Middleware, Message oriented middleware, Distributed object technology, and Transaction processing monitors.

**Global Information System** allows applications to have a common means for accessing data sources in a variety of databases and file stores. These functions are implemented in the form of application independent.

**Communication Services** provides a set of services that allow the transparent communication between applications.

The internal functionality can be decomposed into four simple functional blocks as:

**Service interface** provides a consistent interaction mechanism for all integration services provided by the infrastructure. Various services currently offered include: establishing a logical (peer to peer) link to another application; sending data to an application; opening a file, etc.

**Runtime manager** controls all processes external to the infrastructure (i.e. applications and device drivers) and monitors any error conditions that
might occur within it and provides a power facility for debugging and maintenance of integrated systems.

Configuration manager maintains all internal system configuration data and external configuration files. An interface is provided to enable manipulation of configuration data, such manipulations are also possible via the use of the configuration services offered to applications.

Driver interface allows a variety of device drivers to interact with the infrastructure. Such device drivers accommodate both the diversity of functionality and range of communication protocols required to interact with system resources. Hence the device drivers allow an installable base for machines, software applications, databases and communication networks, within any given manufacturing organization, to be incorporated within integrated systems.

6.5.3 Common Services

Security provides a framework for authentication, access control, and encryption. Access control lists enable discretionary access across single or multiple security domains.

Performance, Availability and Scalability: The platform supports load balancing, fault tolerant, high availability option, native multi-threading and multi-processing, information catching, routing, audit, and high-speed disk logging. Distributed and federated architecture facilitates unlimited server scalability.
Manageability includes Web-based administration capabilities and native Simple Network Management Protocol (SNMP).

Reliability: Distributed Transaction Management Services coordinate activities across disparate systems so that critical business information is never lost or corrupted. Persistence provides recovery in the event of system or network failure using reliable data storage.

6.6 IMPLEMENTATION SCENARIO

An integration platform was developed based on this architecture to integrate the various application in the manufacturing enterprise, such as CAD, CAM, CAPP, SFC, MRP, ERP, and SCM as shown in Figure 6.4. A Knowledge Based System (KBS) was utilized to control the information flow and for decision-making. The knowledge base uses the multiagent concept in which the information agents can collaborate with each other for retrieving and integrating information to meet the user’s demand and the user also can interact with the agents to improve their behaviours or to change the team of agents in a flexible manner. To share the data between these applications Component Object Modelling/Distributed COM (COM/DCOM) technology is utilized.

This application is based on eXtensible Markup Language (XML), which improves the performance of Web based applications. This technology allows to access the data from the database server as an .xml file from the client machine. COM/DCOM technology is utilized to connect with the external server. The Component was developed by Visual Basic and converted into a dll (Dynamic Link Library) file. Then the dll file is registered in the server using
Regsvr32 command. These components can be invoked in any application
using the Active Server Pages (ASP).

![Enterprise application integration](image_url)

*Figure 6.4 Enterprise application integration*

### 6.6.1 Extension to Mobile Application

A method to reuse the existing Internet infrastructure and
transforming the plant information to wireless / mobile devices is developed
(Dhinesh et al 2002c). Figure 6.5 illustrates the way in which the
communication takes place between the mobile client and the server.
Figure 6.5 Mobile access from the server

**Origin server (wap-enabled):** This is an ordinary web-server in which web documents reside by adding Wireless Markup Language (WML) and Multipurpose Internet Mail Extension (MIME) technologies. Personal web server (PWS) has been used as a Wireless Application Protocol (WAP)-enabled server for storing WAP and ASP applications.

**WAP gateway server:** Mobile devices are communicated to the server via the WAP gateway. The gateway acts like an ordinary Internet client. It requests and receives documents from the server on behalf of the mobile device using HTTP.

**WAP-enabled device:** A cell phone that has a WAP browser, a microprocessor that uses WAP protocol, is a WAP-enabled device. The WAP protocol allows the mobile client to get connected to the concerned gateway server and access the information.

### 6.7 SUMMARY

This chapter describes the processes of developing IIS architecture for IAS to favour communication, coordination and cooperation of human and other heterogeneous and distributed manufacturing components of an
enterprise. Prebuilt and pretested components accelerate application integration, focusing on portal strategy. Integration services include visual tools for rapid application integration and new service creation, while foundation services provide valuable functions for managing users and service providers. Support services accelerate incorporation of key infrastructure functionality, and integrated tools facilitate portal deployment and management. While choosing an IIS, enterprise technology decisions include project scope, complexity, type of project (new application development or integration of existing enterprise entities), standards, skills, whether the project has a tactical or strategic focus, organizational scope. A static versus dynamic connection point refers to how fast the interface to an application changes. The architecture of a solution often relates to both its scope and its complexity and the rate of change that the integration code can be redone.