The International Journal of Flexible Manufacturing Systems, 16, 183–199, 2004

An Infrastructure for Integrated Automation System Implementation

K. DHINESH KUMAR* dhinesh123@yahoo.com
CSI Institute of Technology, Thovalai, TamilNadu, India

L. KARUNAMOORTHY karun@annauniv.edu
Department of Mechanical Engineering, Anna University, Chennai, India

H. ROTH hubert.roth@uni-siegen.de
Institute of Control Engineering, University of Siegen, Germany

T. T. MIRNALINEE mirna_32046@yahoo.com
CSI Institute of Technology, Thovalai,TamilNadu, India

Abstract. Evolution of advanced manufacturing technologies and the new manufacturing paradigm has enriched the computer integrated manufacturing (CIM) methodology. The new advances 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 support platforms. In this paper, we present an integrating infrastructure for CIM implementation in manufacturing enterprises to form an integrated automation system. A research prototype of an integrating infrastructure has been developed for the development, integration, and operation of integrated CIM system. It is based on the client/server structure and employs object-oriented 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.

Key Words: CIM, enterprise application integration, integrated automation, integrating infrastructure, open platform

1. Introduction

Computer integrated manufacturing (CIM) can be defined as the use of information technology in support to manufacturing technology for effective 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 set of demands is to provide CIM systems with better software architecture, more flexible integration mechanisms, and powerful support platforms. In this paper, we present an integrating infrastructure for CIM implementation in manufacturing enterprise to build an integrated automation system.

*To whom correspondence should be addressed.

A complete integration paradigm includes three levels of integration (Tzafestas, 1997) covering physical system integration, application integration, and business integration leading to the total integration of a manufacturing enterprise that has been referred to as an integrated automation system. This approach is applied in an integrated way to all activities from design to delivery and after sales, and uses various methods, means, and techniques in order to simultaneously improve productivity, decrease cost, meet due dates, increase product quality, secure flexibility at local and global level in a manufacturing system, and involve every actor.

Integrating infrastructure (II) provides a complete set of supporting 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, an integrating infrastructure fills the gap between different kinds of hardware platforms, operating systems, and data storage mechanisms; it also provides a unified integrating interface, which enables quick and efficient integration of different applications in various computing environments. At the application level, II provides tools and application prototypes for each specific application domain to help building various CIM applications. Several II products have emerged although their coverage is limited. One major reason is that, generally, they only provide an application programming interface for users, so that the user still needs to spend much time in coding, compiling, and integrating existing applications.

The AMICE (European computer integrated manufacturing architecture) project defines the functionality of integrating infrastructure services (ESPRIT, 1993). II provides a structured set of system-wide services, thereby avoiding redundancy of functions in an enterprise. II 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 different applications and users must have fast and easy access to the right information regardless of where it resides. In this paper, we present an II for CIM system implementation in manufacturing enterprise to develop an integrated automation system.

2. A literature review and the evolution of II technology

A review of literature on database frameworks, knowledge-based systems, enterprise ontology, enterprise application integration, and integrating infrastructure for CIM system implementation is presented in this section.

Database frameworks

Kim and Nof (2001) propose two models for the design of collaboration frameworks for distributed CIM data activities. Deng, Xu, Wang, and Wang (2001) has dealt with integrating techniques for multi-databases in CIM enterprises and describe the schema structure and client/server-based system architecture of a multi-database for CIM integrated platform. Brown (2001) describes the nature of data transfers from the “shop floor to top floor automation" and has stated that an efficient and future-proof means of data transfer from factory floor to enterprise is an ultimate goal of a manufacturing information system. But issues related to interfacing and interoperability have not been adequately addressed.

Knowledge-based systems (KBSs)

Woss, Wagner, and Retschitzegger (1999) illustrate how rule-based management of product data allows data exchange between several CIM component to be enabled and possibly automated. However, the maintenance of knowledge-bases and their needed redesign when incorporating new application systems has not been fully addressed. Harkins and Antanies (1999) has stated that an integrated plant information system converts data into knowledge and when combined with a decision support system and advanced controls, provides means of turning knowledge into profit.

Brown (1999) presents an integrated information model with knowledge management as its focus, but does not address logical rules for decision-making. Delic, Douillet, and Dayal (2001) argue that real-time decision support systems are complex, because they must combine elements of several different types of technology, such as enterprise integration, real-time systems, workflow systems, knowledge management, data warehousing, and data mining.

Enterprise ontology

Ontologies are used to specify the tasks addressed by the components of enterprise engineering systems (Gruninger and Fox, 1994). According to Guarino and Giaretta (1995), ontology is a modeling view of knowledge engineering in the field of artificial intelligence and is a research of constructive modeling process of an objective reality. Uschold and King (1995) highlight knowledge-sharing and reuse as primary advantage gained from ontological applications. Such an ontological application is essential to achieve faster agreement on conceptual models of systems and more consistent use of terms and concepts throughout the development and implementation of II. Fox and Gruninger (1997) introduced ontologies as a basis for modeling enterprises. Ontologies enable a precise specification of enterprise structure and the application of this structure to the problem of designing an enterprise whose processes are integrated with its organization structure and behavior (Gruninger, Atefi, and Fox, 2000). It should be noted that the development of ontology for enterprise models has been used as the basis for the development of II in the CIMOSA project (ESPRIT, 1993).

Enterprise application integration (EAI) and integrating infrastructure (II)

ERP systems were introduced into companies to solve various organizational problems, and to provide an integrated infrastructure. Autonomous and heterogeneous applications co-exist in companies with ERP systems and integration problems remain. EAI is a new class of integration software that leads to the development of strategic business solutions by securely incorporating functionality from disparate applications. EAI could be the solution to ERP's integration problems (Themistocleous, Irani, O'Keefe, and Paul, 2001). Weston and Coutts (1994) identified the properties of integrating infrastructures to highlight features of existing integrating infrastructures, with reference to enabling system integration on an enterprise-wide scale. Puschmann and Alt (2001) argue that the need to integrate the packaged software applications, such as enterprise resource/business planning, supply chain management, customer relationship management, and electronic commerce systems with each other as well as with existing or legacy business applications for a standardized integrating architecture for the flexible implementation of new business processes across different organizations and applications.

To share information, an infrastructure is needed (Shaw, 2000). At present, manufacturers look for middleware systems that can be customized to meet their specific needs, with scalability and flexibility to allow deployment at a pace to suit each individual user (Divins, 2000). Nguyen (2000) deals with an EAI-based modeling scheme of e-business solutions for an enterprise and across enterprises. Yasuhiko, Tsujimoto, Yamada, and Yamamoto (2002) propose multiple character-agents interface (MCI) as an information integration platform, where multiple animated life-like characters interact with each other and with the user to retrieve and integrate information from the Internet. The MCI makes the process open to the user and allows collaborating with the character-agents.

The database-related literature does not suggest an optimal database storage and retrieval system. The literature related to KBS insist that KB is a requisite for CIM-based IAS that falls short of the required precision in highlighting the logical flow and controlling the information. The literature under EAI and II concerns their development for manufacturing enterprises. This literature on CIM systems deals with highlighting different islands of automation and associated parameters, which are impediments in the optimization of manufacturing processes and as a pre-requisite for the implementation of CIM-based IAS. Moreover, the literature paves the way to the development of II for the implementation of an integrated automation system.

An information system can be developed by creating a set of user interfaces using an object-oriented technique. With software development tools, information systems are developed through a modeling process by creating and manipulating information system models. The need to integrate information from different applications of an enterprise leads to the information integration concepts and later to the development of a common information integrating infrastructure. Early forms of II have supported the development of application software, but their support for application integration was rather weak. From the beginning of the 1990s, II technology moved into a phase of supporting wider application development and integration in a heterogeneous distributed environment. To meet these requirements, several concepts, paradigms, and specifications, such as middleware, client/server architecture, object-oriented methodology, open system interconnection, distributed computing environment, common object request broker architecture, component object modeling, and distributed COM were introduced for the conceptualization, design, and development of II (Dhinesh, Karunamoorthy, Roth, and Mirnalinee, 2002b). Based on these advances, a new form of II is proposed by the authors. Figure 1 shows a multilayer II consisting of: communication layer, information management service layer, and function service layer. Collectively these layers provide common system-level services. These services form the middleware layer of the II. The higher layers of the II are classified as general-purpose

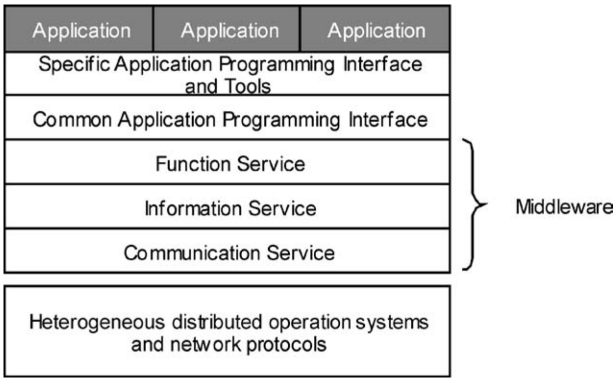

Figure 1. A multi-layer II.

API, domain specific API, and application development integration tools. II extends the integration from specific domains (including management, planning, and manufacturing execution) to the entire enterprise.

The various integration approaches include a technical approach using middleware, messaging, and file transfer technologies. Data integration includes data base replication, directory services, common data definitions, and storage. Business process integration requires thinking in terms of logical business processes, even when they cross boundaries of a particular application, and a user interface approach makes use of the web and portals to unify applications in an enterprise. For flexibility and reconfigurablity, we need a message passing in distributed multi-tier application architecture.

However, the complexity of manufacturing systems and the lack of an effective integration mechanism raise problems for CIM system implementation. Some of these problems are:

(1) Lack of openness and flexibility. An integrated system is generally inflexible rather than open structured and it is difficult to incorporate new technology. Lack of flexibility makes it difficult to update when it is required with evolving enterprise requirements.
(2) Inconvenient and inefficient interaction between applications. This is caused usually by 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 information technology, and wish to retain their old investment. But there lacks effective means of combining legacy resources into a new system.
(4) Long time for CIM system implementation. Without powerful application-oriented supporting tools, the implementation process is time consuming 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 functions lead to a confused understanding and possible misuse of the system.

3. Motivation for the II development

It seems that to satisfy the evolving market need, manufacturing facilities need to integrate plant automation systems with enterprise business system (Dhinesh, Karunamoorthy, Roth, and Mirnalinee, 2002c) in order to bring their manufacturing facilities under control and to satisfy the time dependent supply-chain requirements of global e-commerce (which is still evolving). Integrated automation is an essential success factor in automation, controls, and information systems (Dhinesh, Roth, and Karunamoorthy, 2002a). The development in the area of integrated automation is strongly influenced by standardization efforts in control engineering, computer engineering, and information technology, which follows the efforts in ISO and IEC standards. Integrated automation concepts now provide cost-effective answers to production requirements in virtually all areas of industry to eliminate uncertainties, introduce greater reliability, better information flow, and increased responsiveness. Integrated automation leads to reuse of existing objects, such as architecture, database, frameworks, applications, hardware, software, design specifications, resources, information, project plans, procedures, and policies, and is based on the integration of different functions (Siemens, 2001) into a single optimized model. Integrated automation (IA) provides an information aggregation and decision-support infrastructure that delivers information to every user from senior executives, production-line workers, sales officers, suppliers, customers, and partners. IA capabilities enable firms to turn environmental trends (rapid market changes, increasing complexity, declining opportunities, and economics of sale) into opportunities for gaining a competitive advantage. The easy availability of production and field data facilitate quick product design changes, resulting in better engineering, and quality.

For the design of such systems, technologies such as integration brokers, application servers, and web services provide integration capabilities. Integration brokers have historically provided messaging, intelligent routing, translation, and transformation (Zahavi, 2001). More recently, application servers have offered more in the way of reliability, scalability, transaction management, and are evolving into full integration platforms. Web services are evolving and there are some pitfalls in web standards in terms of scalability, reliability, security, and management. Even though industries have more choices for solving their integration issues, effective management of II require more tools and techniques to solve these issues.

A possible solution to this problem is the development of a conceptual framework or generic architecture for system integration, which allows categorization of the various methods and consequently their conformities and discrepancies that can generate uniformity and mutual understanding. The II 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, here remains major problems of unifying the goals and activities of individual processes, while retaining sufficient flexibility in the holistic systems formed. Methods and tools are required that are capable of defining and constructing complex integrated systems, which enable them to realize system-wide goals, where invariably those goals can 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 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 do inter-operate and enact formal specifications so that system building and change can be achieved in a structured and effective manner. This needs to be achieved with sufficient realism and completeness to guide subsequent system implementation (Weston, 1994). This paper consider ways of facilitating integrating infrastructures, which themselves offer computational means of implementing and supporting the operation of integrated manufacturing systems.

4. Basis of the proposal

We discuss the development of II on the basis of the framework of CIM open system architecture (CIMOSA) (ESPRIT, 1993). The aim of CIMOSA is to provide open system architecture to support integration of CIM system components (ESPRIT, 1993; Kosanke, 1992). To this end, CIMOSA provides an integrating infrastructure and a modeling framework (Vernadat and Zelm, 1993). Enterprise application integration uses middleware technologies, such as remote procedure call, database, message oriented middleware, distributed object technology, and transaction processing monitors to reduce the integration issues. II supported by CORBA and COM/DCOM can be used as the underlying infrastructure for EAI. Services are not part of a core in EAI building blocks, but aid the developer in implementing a solution. The four basic building blocks for EAI are: communication model, method of integration, middleware, and services (Ruh William, Maginnis, and Brown, 2001). An enterprise should go through four life stages to effectively implement integrated automation (1) initial stage: organisations must plan, learn, and architect with the intent of achieving early success, (2) architectural stage: organisations must create an architecture that can be understood by designers and developers, and to promote future reuse, (3) organisation stage: organisations must apply the results from the first two stages across the organisation and expand EAI services, (4) enterprise stage: the organisation has mastered EAI and is effectively employing and measuring the effectiveness of its EAI architecture.

4.1. CIMOSA integrating infrastructure

The CIMOSA integrating infrastructure is an enabling technology that makes it possible to execute 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). The aim of the II is to transform a highly heterogeneous execution environment into a more homogeneous world and to ensure system interoperability amongst different vendor components. It provides a unified software platform to achieve an integration of heterogeneous hardware and software components of a CIM system. The CIMOSA integrating infrastructure (ESPRIT, 1993) as shown in Figure 2 consists of five interacting sets of services. These set being called entities. They are:

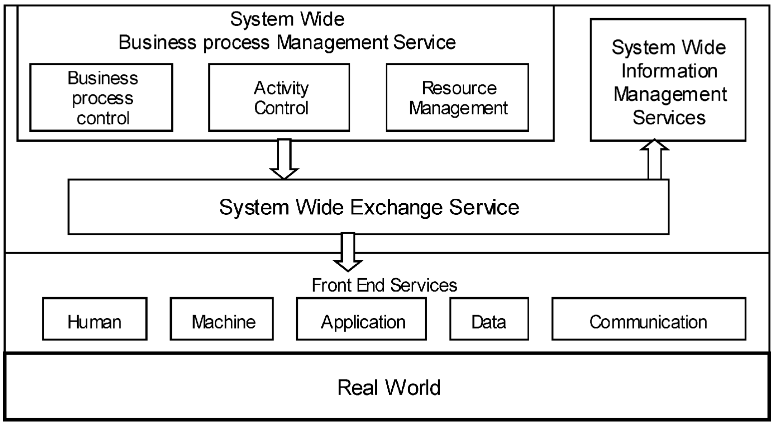

Figure 2. CIMOSA integrating infrastructure services.

- Business entity provides generic functions to control the enterprise operations according to the contents of a CIMOSA model.
- Information entity provides generic functions for data access, data manipulation, and data integration.
- Presentation entity provides generic functions to support an 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.

4.2. Role of semantic web technology and web services in enterprise integration

The emergence of the web, and its underlying Internet technologies, is fundamentally a key factor in the emergence of EAI and to the changes in the IT environment. Web has become a major computing platform and a uniform interface for sharing data; many organizations have found interest in delivering information through the Web, both in Internet and intranet environments (Atzeni, Merialdo, and Mecca, 2001). Web provides universal connectivity, and a platform for business competition. Using the Web infrastructure, intranet supports intraorganizational business processes. Extranets on the other hand connect enterprises to their partners and the Internet links the enterprises to their customers, and partners globally.

Tinham Brian (2000) has stated that rethink your business on the web for integrating their internal islands of information technology and their external supply chains to build a responsive, inclusive foundation before venturing into e-business. Web today is in a stage of maturation lending extensive scope for integration of diversified applications wherein it is involved. The web provides the ability to integrate the underlying information systems and business processes quickly (Shaw, 2000). Applications such as remote maintenance, remote testing, and remote control of automated systems become possible using web technology. To build the web applications, an enterprise needs to remain competitive and should master EAI.

5. The development of an II

In order to have a tactical or strategic focus and organizational scope, while choosing an II**,** enterprise technology decisions should include project scope, complexity, type of project (new application development or an integration of existing enterprise entities), standards, and skills. For the implementation of CIM systems, integration of different applications of the enterprise has to be squarely addressed. A complete integration paradigm includes the integration of data sources, an application function, and business processes. A component-based system supports elements conforming to certain standards and allows instances of elements to be plugged into a framework (Szyperski, 1998) that improves interoperability and modularity. Middleware technology aims at building an application system by integrating heterogeneous components and enables component integration at runtime. Middleware can be defined as "a software layer that provides a programming abstraction as well as masking the heterogeneity of the underlying networks, hardware, operating systems, and programming languages" (Coulouris, Dollimore, and Kindberg, 2001). This advancement in technology paves the way for more flexible II development. II hides the location, storage, and physical placement of information and resources, and manages the link with the underlying communication infrastructure. The development of an II consists of designing the system architecture, defining the functionalities for inter-operation, and aggregation.

5.1. II system architecture and design decisions

The system architecture of an II is illustrated in Figure 3. This is a client/server structured, object-oriented platform with a high degree of flexibility. The primary functionality of the architecture is to integrate applications in a heterogeneously distributed computing

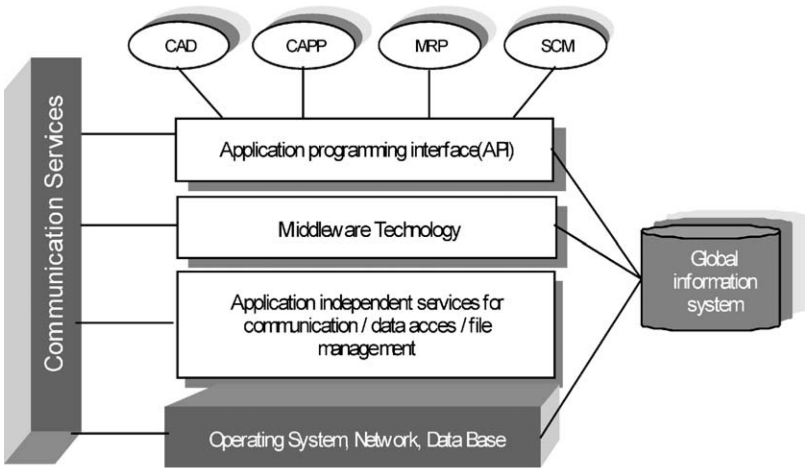

Figure 3. System architecture of an II.

environment, with various operating systems, networking protocols, and database management systems. The communication system provides a set of services that underpin communication between various applications. The global information system allows applications to access databases in a common way.

At the application level, various manufacturing applications, such as CAD, CAPP, MRP, and SCM are integrated using an application programming interface. Since this architecture conforms to a client/server environment, a remote procedure call is utilized for application integration. Also, middleware technologies such as COM/DCOM and CORBA enable system integration in the application integration platform to share remote information.

The system design allows the integration of application in a heterogeneous distributed computing environment such as different operating systems, network protocols, and database management system. Integration services incorporated include visual tools for rapid application integration and creation of new services, while foundation services have been made to provide functions for managing users and service providers. Support services have been adopted to accelerate infrastructure functionality. Integrated tools are made to facilitate portal deployment and management. Communication service and global information systems are application-independent services that are not designed for specific applications, but for general services. Thus this architecture enables information integration and application integration. Application-dependent API is used for quick and easy development of domain specific applications.

II is developed using DCOM technology (Microsoft, 2000), as middleware and a knowledge-based system for decision-making. DCOM supports platform dependent and language independent applications. It enables components to communicate directly over a network in a reliable, secure, and efficient manner. It is designed to operate across network transports including Internet protocols such as http and enables local inter-process communication with a special network protocol, when a client and its component reside on the same machines.

Knowledge-base is used as a superior tool for distributing decision-making expertise and provides the ability to capture critical business knowledge. This knowledge has been used in existing applications and made available in an enterprise for decision support. It is made to use a rule-based technique for decision making in the form of structured if-then statements. A KBS generates a file with an updated list of processes, their current status ready for execution, and an HTML file containing the status of the system for viewing with a web browser and links to the other web pages as output. The adaptability of web technology in the overall process has been structured to allow an application to be quickly distributed to all applications.

5.2. System modules

Application programming interface (API) is a mechanism by which functions and modules can be accessed by the application. The application service interface utilizes a suite of *C* runtime functions that facilitate access to the integration services provided by the integration infrastructure. The application event manager provides a consistent mechanism for event management (such as the arrival of data packets on a file descriptor). 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 developer, so as to encode specific functionality and required behavior. 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. Middleware technology provides standard services for defining and maintaining interfaces, simplifying architectures, and securing applications. It provides the runtime environment to manage requests between software components.

Middleware provides a means of coping with 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, and mechanisms by which applications can share information with other applications, mechanisms by which applications can coordinate business processes. There are five basic types of middleware in the market today. They are remote procedure call, database access middleware, message oriented middleware, distributed object technology, and transaction processing monitors. Global information system allows applications to share data sources available in a variety of databases and file stores. These functions are implemented in an application independent form. Communication services provide services that allow transparent communication between various applications.

Rather than specifying the functionality of an application, the purpose of an II is to facilitate an aggregation of that functionality by enabling and managing the required inter-operation using suitable interfaces. An II can thus satisfy the dual requirements by providing an appropriate set of integration services that collectively underpin the runtime integration of a number of entities, (i.e., it is required to offer services which enable inter-operation) and a set of integration tools which collectively define, manage, and change the associations formed between entities. When used in combination, the services and tools of an II can unify the various activities carried out in a complex manufacturing enterprise, while 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 adaptability. 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.

The internal functionality of independent and dependent API can be decomposed into four simple functional blocks as follows. A service interface provides a consistent interaction mechanism for all integration services provided by the infrastructure. Examples of services currently offered includes establishing a logical (peer to peer) link to another application, sending data to an application, and opening a file.

A runtime manager controls all processes external to the infrastructure (i.e., applications and device drivers), monitors any error conditions that occur within it, and provides a powerful facility for debugging and maintaining integrated systems.

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

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

6. Implementation scenario

An integration platform was developed based on the architecture described in this paper. This integration platform has subsequently facilitated integration of various applications commonly deployed within a manufacturing enterprise as shown in Figure 4. The information system and flow of information between different functions/applications include: product information, such as product or part identification, part list, NC data, CAD data, engineering data, and configuration management information; process informations, such as work plans, manufacturing processes, activities, and operations; resource information such as machines, humans, and applications; production planning information, such as plans,

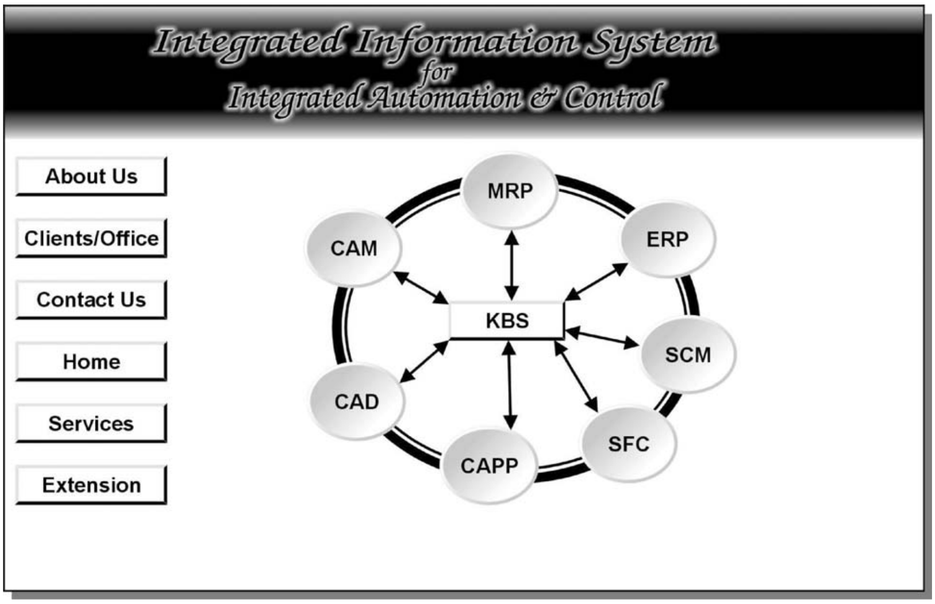

Figure 4. Enterprise application integration.

schedules, and inventories; production control and shop floor control information, such as equipment status, labor allocation, quantities ordered, and quantities stored; and administrative informations, such as personnel data, supplier data, customer data, cost, and orders. A knowledge-based system was utilized to control information flows and for decision-making. The knowledge-base is built using a multiagent concept in which the information agents can collaborate with each other for retrieving and integrating information such that user's demands are met. Also by such means, users can interact with the agents to improve their behavior or to change how a team of agents interoperate in a flexible manner. Data sharing between these applications is achieved using component object modeling/distributed COM technology.

This application is developed based on the use of eXtensible Markup Language (XML) that improves the performance of Web-based applications. This technology allows accessing data from database servers as an .xml file from a client machine and DCOM technology is used to access the component from the external server. The component was developed in Visual Basic and converted into a dynamic link library (dll) file. Then the dll file is registered in the server using a Regsvr32 command. To enhance the reusability, these components were invoked in any application using the active server pages.

6.1. Requirement specification and information flow

The system administration and IT service modules reside on a Windows NT server. The enterprise objects and file repositories can exist in different databases residing on multiple devices, typically on a server. This framework uses web servers and the HTTP protocol for client-server communication. HTML and XML are used as the data format for the transfer of data and wide-area level security that can work over the Internet or intranets and support the concept of intermittent (connectionless) communication between clients and servers. In the current platform implementation, active server pages and Microsoft Internet information server are used as an integral part of the architecture, via MS XML parser for parsing XML data, and the use of MS NT security mechanism.

6.2. Component-based solutions and re-use

Conventional approaches to integration, by suites of integrated but fundamentally separate tools, result in a complex patchwork of code that must be carefully stitched together, and are becoming increasingly brittle and indecipherable over time. Business objectives can be obscured and lost in the translation from business process to technical implementation.

Component-based integration simplifies development and maintenance by re-use. Business processes, sub-processes, process flows, data translations, data queries, human workflow steps, and other integration elements are grouped, and encapsulated into re-usable components. A port is specified for each component, defining how other components may interact with it as a service. By decoupling the component's function from other components, the port allows the service-based component to be re-used and changed more easily without breaking some other element of the integration solution.

6.3. Distributed artificial intelligence and software agents

Artificial intelligence enables the development of intelligent systems capable of making decisions under conditions of uncertainty. Intelligence is a property that enables a system to operate effectively when available information is inadequate. Information can be considered as a resource for reducing risks associated with decision-making, for accumulating knowledge, and for organising the world. The agent (a software entity that assists people and can act on their behalf) provides versatile mechanisms of interoperation and co-ordination, and flexibility in assembling communities of autonomous service during development time and runtime. The concepts of artificial intelligence and software agents were applied for effective information flow and decision making between various applications. Human-oriented user interfaces provide a conceptually natural means of interacting with multiple distributed components. They provide integral support for the construction of multimode interfaces, including the handling of user's requests expressed in a natural language, treat users as privileged members of an agent community, and support collaboration (simultaneous work over shared data and processing resources) between users and agents. Realistic software engineering minimize the effort required to create new agents, to wrap existing applications, encourage reuse of both domain-independent and domain-specific components, and minimize platform and language barriers to the creation of new agents, as well as wrapping of existing applications.

6.4. Extension to mobile application

A method for enabling reuse of the readily available Internet infrastructure and for transforming plant information via wireless/mobile devices was developed. Figure 5 illustrates the manner in which communication is enabled between a mobile client and the server.

- Origin server (wap-enabled): This is an ordinary web-server in which web documents reside by adding WML extensions and multipurpose Internet mail extensions. A personal web server has been used as a wireless application protocol-enabled server for storing wireless application protocol (WAP) and active server pages (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 hyper text transfer protocol (http).

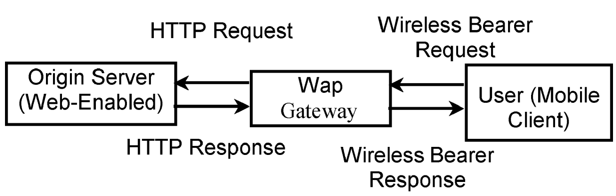

Figure 5. Mobile access from the server.

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

6.5. Performance issues

The working of the II has improved the performance on different areas like resource sharing, process incompatibility, synchronization of time, quality, prescribed cost, web compatibility, and customer adaptability. Also it reduces transaction hurdles, operational risk, and maintenance risks of the enterprise application integration.

These issues have been addressed on the following modes. All resources have been brought into a common platform and the distribution to several application areas has been made through system gearing. Information flow between different applications are controlled and coordinated by a synchronised mode of operation using the knowledge-base. Quality control checks are incorporated in multi-layers of the system and suitable control warning alarms in the form of error signals are made to warn/stop. To improve cost, the development time has been controlled by modular development. Software interfaces have been in-structured for web compatibility. Identifying and linking suitable interfaces have handled transaction hurdles. Operational and maintenance risks have been overcome with complete backup software/offline storage as modules. Issues in data-sharing have been overcome by distributed database management system.

The platform supports load balancing, fault-tolerance, high availability, multi-threading, multi-processing, information caching, routing, and high-speed data logging. Distributed architecture facilitates scalability. Authentication, access control, and encryption incorporated in this application provide more security. Web-based administration capabilities and native simple network management protocol provide manageability. Distributed transaction management services coordinate activities across disparate systems, so that critical business information is never lost or corrupted and improves reliability. Persistence provides recovery in the event of system or network failure using reliable data storage.

It has been identified that performance in areas of resource sharing, process enactments, information synchronisation, and performance quality has increased. Data/information sharing effectiveness has a high level of satisfaction among all application units. Incorporation of web-facility leads to online real-time performance, with a high degree of compactness, convenience, and cost. Interactions among different applications showed higher dynamics. Operational risks and transactional risks were reduced.

7. Conclusions

The integrating infrastructure proposals of the authors build upon a unique combination of component-based object technology, web technology, and knowledge-based developmental methods. An implemented version of this II is capable of improving the integration of business processes, because its underlying web technology will facilitate cross-enterprise information sharing through interconnectivity and integration. This paper conveys the essence of the proposed integrated infrastructure development and implementation in the context of integrated automation system. A future goal of the authors is to reuse these ideas and to integrate and create a virtual network of networked enterprises to enhance reconfigurability and agility.

References

Atzeni, Paolo, Merialdo, Paolo, and Mecca, Giansalvatore, "Data-Intensive Web Sites: Design and Maintenance," *World Wide Web*, Vol. 4, pp. 21–47 (2001).

Coulouris, G., Dollimore, J., and Kindberg, T., *Distributed Systems: Concepts and Design*, 3rd edition, Addison-Wesley, Wokingham, UK (2001).

Delic, K. A., Douillet, L., and Dayal, U., "Towards an Architecture for Real-Time Decision Support Systems: Challenges and Solutions," *Proceedings of the International Database Engineering and Applications Symposium*, IDEAS 2001, pp. 303–311 (2001).

Deng, S. C., Xu, X. F., Wang, N. B., and Wang, G., "Integrating Techniques of Multidatabase in CIMS Enterprise," *Computer Integrated Manufacturing Systems*, Vol. 7, No. 1, pp. 11–15 (2001).

Dhinesh Kumar, K., Roth, H., and Karunamoorthy, L., "Critical Success Factors for the Implementation of Integrated Automation Solutions with PC Based Control," *Proceedings of the 10th Mediterranean Conference on Control and Automation,* Instituto Superior Tecnico, Lisbon, Portugal, pp. 1–10 (2002a).

Dhinesh Kumar, K., Karunamoorthy, L., Roth, H., and Mirnalinee, T. T., "A Modular Approach for the Implementation of Integrated Automation," *Proceedings of the 18th National Convention of Mechanical Engineers and National Conference on Emerging Trends in Mechatronics for Automation*, Rurkela, Orisa, India, pp. 49–59 (2002b).

Dhinesh Kumar, K., Karunamoorthy, L., Roth, H., and Mirnalinee, T. T., "Web Based Real-time Mobile Access to Integrated Automation," *Proceedings of the 7th Seventh International Conference on Control, Automation, Robotics, and Vision,* Nanyang Technological University, Singapore (2002c).

ESPRIT Consortium AMICE, *CIMOSA: Open System Architecture for CIM*, 2nd edition, Springer-Verlag, Berlin (1993).

Fox, Mark S. and Gruninger, M., "On Ontologies and Enterprise Modeling," *International Conference on Enterprise Integration Modeling Technology 97*, Springer-Verlag, Berlin (1997).

Gruninger, Mark and Fox, Mark S., "The Role of Competency Questions in Enterprise Engineering," *Proceedings of the IFIP WG5.7 Workshop on Benchmarking—Theory and Practice*, Trondheim, Norway (1994).

Gruninger, M., Atefi, K., and Fox, M. S., "Ontologies to Support Process Integration in Enterprise Engineering," *Computational and Mathematical Organization Theory*, Vol. 6, No. 4, pp. 381–394 (2000).

Guarino, N. and Giaretta, P., *Ontologies and Knowledge Bases Towards a Terminological Clarification, Towards Very Large Knowledge Bases*, IOS Press, Amsterdam (1995).

Harkins, B. and Antanies, J., "Integrated Information Management System: Turning Knowledge into Profit," *Process Control News,* Vol. 19, No. 4, p. 1 (1999).

Kosanke, K., "CIMOSA—A European Development for Enterprise Integration. Part 1: An Overview," *Enterprise Integration Modeling*, C. Petrie (Ed.), The MIT Press, Cambridge, MA, pp. 179–188 (1992).

Microsoft: http://www.microsoft.com/isapi (2000).

Nguyen, Thang, N., "Modeling EAI-Based E-business Solutions," *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 1, IEEE, Piscataway, NJ, pp. 286–291 (2000).

Querenet, B., "CIMOSA—A European Development for Enterprise Integration. Part 3: Enterprise Integrating Infrastructure," *Enterprise Integration Modeling*, C. Petrie (Ed.), The MIT Press, Cambridge, MA, pp. 205–215 (1992).

Ruh, William, A., Maginnis, Francis, X., and Brown, William, J., *Enterprise Application Integration*, John Wiley and Sons, New York (2001).

Shaw, Michael J., "Information-Based Manufacturing with the Web," *The International Journal of Flexible Manufacturing Systems*, Vol. 12, No. 7, pp. 115–129 (2000).

Siemens Automation and Drives Group, available at http://www.siemens.com/automation (2001).

Szyperski, C., *Component Software: Beyond Object-Oriented Programming*, Addison-Wesley, New York (1998).

Themistocleous, M., Irani, Z., O'Keefe, R., and Paul, R., "ERP Problems and Application Integration Issues: An Empirical Survey," *Proceedings of the Hawaii International Conference on System Sciences*, p. 278 (2001).

Tinham, Brian, "Rethink Your Business on the Web," *Manufacturing Computer Solutions*, Vol. 6, No. 3, pp. 4–5 (2000).

Tzafestas, Spyros, G., *Computer Assisted Management and Control of Manufacturing Systems*, Springer-Verlag, London (1997).

Uschold, M. and King, M., "Towards a Methodology for Building Ontologies," Technical Report (TR-183), Artificial Intelligence Applications Institute, The University of Edinburgh (1995).

Vernadat, F. and Zelm, M., "Advanced Modeling Approach to CIM Systems," *Advances in Factories of the Future, CIM and Robotics*, M. Cotsaftis and F. Vernadat (Eds.), North-Holland, Amsterdam, pp. 77–89 (1993).

Weston, R. H., "Requirements and Trends in Manufacturing Systems," *Advances in Manufacturing Technology VIII*, K. Case and S. Newman (Eds.), Taylor and Francis, *10th National Conference on Manufacturing Research*, Loughborough, UK (1994).

Weston, R. H. and Coutts, I. A., "Model Enactment Based on Use of the CIM-BIOSYS Integrating Infrastructure," *Proceedings of International Conference on Automation, Robotics, and Computer Vision*, Singapore, pp. 322–326 (1994).

Woss, Wolfram, Wagner, Roland, and Retschitzegger, Werner, "Rule-Based Integration of Business Oriented and Technical Components in CIM Systems," *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 26, No. 3, pp. 249–266 (1999).

Yasuhiko, Kitamura, Hideki, Tsujimoto, Teruhiro, Yamada, and Taizo, Yamamoto, "Multiple Character-Agents Interface: An Information Integration Platform where Multiple Agents and Human User Collaborate," *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, Bologna, Italy (2002).

Zahavi, R., *Enterprise Application Integration with CORBA*, John Wiley and Sons, New York (2001).