sima

BACKGROUND STUDY

REQUISITE ELEMENTS, RATIONALE, AND TECHNOLOGY OVERVIEW FOR THE SYSTEMS INTEGRATION FOR MANUFACTURING APPLICATIONS (SIMA) PROGRAM

Editors
Edward J. Barkmeyer
Theodore H. Hopp
Michael J. Pratt
Gaylen R. Rinaudot

NIST
UNITED STATES
DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY

NISTIR 5662

U.S. DEPARTMENT OF COMMERCE
TECHNOLOGY ADMINISTRATION
National Institute of Standards and Technology

NISTIR 5662

Background Study

Requisite Elements, Rationale, and Technology Overview for the Systems Integration for Manufacturing Applications (SIMA) Program

Editors:
Edward J. Barkmeyer
Theodore H. Hopp
Michael J. Pratt
Gaylen R. Rinaudot

Contributors:
Neil Christopher
Shaw Feng
Simon Frechette
Al Jones
Mark Luce
Kevin Lyons
Chuck McLean
Stephen A. Osella
Steven Ray
Bradford Smith
Evan Wallace
Peter Wilson

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Gaithersburg, MD 20899

September 1995

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arati Prabhakar, Director

TABLE OF CONTENTS

EXECUTIVE SUMMARY

To stay competitive in today's business environment, many companies are introducing advanced technologies, particularly computer-based applications, into their manufacturing operations. Often, however, technical barriers prevent the successful integration of these applications into a single, smoothly operating system.

NIST's Manufacturing Systems Environment (MSE) project aims to address this problem by taking a comprehensive systems engineering approach to the integration of systems for design, manufacturing engineering and production, three activities which together are referred to as *product realization.* The project's central purpose is to provide industry with open architectures and interface specifications that will facilitate the implementation of efficient integrated product realization systems built from commercially available software packages. MSE is part of the Systems Integration for Manufacturing Applications (SIMA) program initiated by NIST in 1994 as part of a new Federal government initiative on High Performance Computing and Communications (HPCC).

The first phase of the 5-year MSE project will be devoted to gathering information and developing models for a broad range of manufacturing activities. During the second phase, one or more integrated systems will be constructed, and the project will concentrate on interface development and testing. In the third phase, the results will be made available to U.S. industry through workshops, training materials, electronic data repositories, and pre-commercial prototype systems available to potential vendors for testing and evaluation. Throughout the project, MSE personnel will develop and maintain strong collaborations with industry, other research institutions, and standards organizations.

This report—which documents the findings of a background study of industry needs in the area of manufacturing systems integration—defines the initial MSE project focus in detail and sets the technical direction for project efforts. The report also describes the principal types of product realization software applications now in use, reviews efforts within manufacturing industry towards developing integration solutions, discusses relevant research trends in the field of product realization, and surveys standards which may be applicable to the work of the MSE project. A brief review is also given of some potentially relevant supporting technologies from the realm of information technology.

PROJECT SCOPE

The domain of the MSE project is the design and manufacture of electromechanical products, though electrical and electronic components will be considered to exist already and merely require assembly into the final product. The project will focus on the four manufacturing processes—machining, injection molding, die casting, and sheet metal stamping and pressing— accounting for the great majority of mechanical parts made today. However, in a time of rapid technological change, it will be important to give some consideration to less commonly used manufacturing methods that may become dominant in the future.

The three major areas of product realization are design, manufacturing engineering and production. Figure 1 in Chapter 1 shows the activities, information flow and control factors of SIMA's model of manufacturing. The design function starts with the requirements for a new product. Its output is a completely documented specification of that product, including a geometric description.

Traditionally, once the design phase is complete, the resulting product specification becomes the input to the manufacturing engineering phase, which essentially determines how the product is to be made. There is, however, a trend toward *concurrent engineering,* in which some activities in these two phases are carried out in parallel to shorten the overall product realization cycle. It is important that the results of the MSE project be compatible with this mode of operation. The planning methods used in manufacturing engineering vary widely, depending on the processes to be used in making the product, though there are some unifying features. The input to planning activities includes not only a specification of the product to be made but also details of the manufacturing resources available for the task. Here, interfaces to large databases will be very important. Another important manufacturing engineering activity involves the use of computer simulations to verify the feasibility of the plans generated, without taking production equipment out of service to perform test runs.

Production software systems handle activities such as scheduling of jobs on the shop floor, requirements planning, inventory control, shop-floor monitoring, job tracking, and tool management. These activities determine the timing and sequencing of the mix of products being manufactured at any one time and the allocation of resources to their production. Computer simulation plays a key role here as well, particularly in the optimization of job scheduling through the use of statistical experiments performed on the computer. Simulation technology will be very important to the MSE project, because resource limitations will restrict the number of production scenarios that can be implemented in actual demonstration systems.

The integration of software systems supporting the whole range of activities will require the provision of automated links between many modules having diverse characteristics. The specification of such interfaces, and of a supporting architecture for them, will be a primary objective of the MSE project.

Product quality considerations pervade the whole spectrum of product realization activities, and the drive towards higher quality will be a unifying influence throughout the work.

THE IMPORTANCE OF STANDARDS

Several different approaches have been used by industry in efforts to build integrated product realization systems, and a survey carried out as part of the project has shown that they suffer from a common problem: a lack of standards that may be used to achieve interoperability between commercially available software products. One goal of the MSE project is to foster the development and use of such standards, especially in the following areas:

- Standard neutral data formats for the exchange of engineering and production information
- Standard application program interfaces (APIs) for accessing the internal functionality and internal data repositories of commercial systems
- Use of third-party, off-the-shelf, conversion and "wrapper" software to provide an application with a standards-compliant interface to other systems.

The MSE project will monitor standards development, provide active support where appropriate, and cooperate with vendors to incorporate standards into commercial products. Specific areas where standards are needed are advanced product modeling (including parametric, variational, and feature-based methods), tolerances data, process and production plans, and production resource and inventory information.

As a general policy, the SIMA program will look to U.S. industry to drive the standards process. Individual groups working within the SIMA MSE project will therefore take the lead in initiating development of new standards only if industry clearly expresses a need for the standards and provides support for their development. Otherwise, the involvement of the MSE project will be limited to established industry-led standards activities.

Where appropriate standards already exist, the project will make use of them. It will also draw upon detailed architectures and information models proposed by various research consortia, where these prove to be valuable. Otherwise, the MSE project will identify needs for new standards, make significant contributions to emerging product realization standards activities, and support the broader acceptance and standardization of industry-developed specifications subject to the guidelines stated in the last paragraph.

The MSE project will also utilize any general-purpose information technology standards that are potentially relevant to its aims. In many cases it will be possible to use corresponding off-the-shelf commercial products, enabling the project to concentrate on the engineering and manufacturing application concerns. Such standards and products will be identified, evaluated, and in some cases used. However, it is not expected that the project will identify the need for new developments in information technology or general-purpose information technology standards.

The effective use of standards will require the developers of integrated systems to agree on:

- What information each system needs, and what information it generates
- What requests each system will respond to, and what functions it will perform
- What information exchanges will occur, when, and what systems will be involved
- How the information is to be exchanged

The specification of the functions a system will perform, and the information it needs and provides, is termed a *system architecture.* The specification of what information exchanges will occur, and how they will be achieved, is termed an *interface specification.* The MSE project must therefore define system architectures and interface specifications that permit integration ofthe component systems of design, manufacturing engineering, and production systems.

SYSTEM IMPLEMENTATIONS AND TEST-CASE PRODUCTS

A key part of the MSE project will be the implementation of demonstration systems to show the feasibility of the proposed integration solutions. The project will use SIMA's Advanced Manufacturing System and Networking Testbed (AMSANT) as a demonstration site for this purpose. It is hoped that demonstrations of *distributed* integration can also be given, using the facilities of AMSANT to network to the sites of remote industrial collaborators. The demonstration systems will serve the purposes of focusing MSE project activities and providing an environment for integration experiments with specific technical goals. However, their most important role will be in technology transfer, for communicating MSE achievements and the benefits of integration to a wide audience.

Test-case products will be selected at various times during the MSE project lifetime, to focus attention on real manufacturing problems. Generally, these will be representative of widely used electromechanical products, giving rise to significant problems from the integration point of view, while still being feasible within the scope and resources of the MSE project. Test case products will be chosen whose components require a range of different manufacturing processes.

Generally, electromechanical consumer products satisfy these requirements. Workshop tools, kitchen appliances, and certain items of recreational equipment are likely candidates. It is anticipated that formal relationships (i.e., Cooperative Research and Development Agreements or CRADAs) will be set up with companies providing details of test-case products. Collaboration between industry and NIST is planned as an important aspect of the MSE project. Industry support will be essential in developing realistic test-case scenarios for testing project results.

Throughout the project, it will be vital for MSE personnel to be constantly aware of a broad spectrum of relevant research and development efforts by software vendors, universities and manufacturing companies. New research results and software products will constantly be reviewed, and updates made to the MSE project system architecture and interfaces to reap the maximum benefit from significant technical advances.

STRUCTURE OF THIS DOCUMENT

The report is divided into three parts. Part I provides an overview of the MSE Project, including a discussion of its scope, objectives, and implementational strategy. Part II is at a more technical level, and provides detailed discussion of the current state of progress and research directions in the integration of manufacturing software applications. Part III reviews the information technology aspects of systems integration. Throughout the document, the potential role of standards is emphasized, and two appendices give details of many of the standards of potential interest to the project.

SUMMARY OF PRIMARY RECOMMENDATIONS MADE

It is almost axiomatic that the MSE project should determine the most suitable methods, software tools and standards for use in developing its approach to an effective integration strategy. Apart from this, the major recommendations made in the technical chapters of the document, which frequently reinforce each other, are as follows:

1) The MSE project should engage in liaison/collaboration with

- ***Manufacturing industry (to establish integration needs, to find examples of successful integration strategies, to obtain product data for implementation scenarios, to seek support for development of new standards)***
- ***Software vendors (for cooperation in developing system interfaces)***
- ***Standards bodies (to keep in touch with and contribute to ongoing standards development)***
- ***Universities (to foster MSE-related research projects and gain access to new ideas relating to systems integration)***
- ***Other related NIST projects (to use their results in enhancing the MSE deliverables)***

2) The project should concentrate on short-term (0 to 2 years) and medium-term (2 to 4 years) approaches to integration, while also monitoring emerging possibilities for more advanced integration techniques in the longer term.

3) The project should note a strong industrial need for standardized information bases, for the storage of manufacturing plans and resource data. Creation of a database of standards information is also recommended.

4) The study has shown that the early stages of design are currently not well supported by computer aids. It is therefore recommended that encouragement be given by all appropriate means to research efforts directed towards the computer

support of product planning, functional and configuration design.

5) It is recommended that the project should identify a suitable framework for the development of integrated manufacturing system architectures, and that a single reference architecture should be developed within it. This should allow a flexible approach to the specification of engineering architectures, permitting the testing and comparison of different detailed integration methods.

PREFACE

This report documents the findings of a three-month background study—conducted by the Manufacturing Systems Integration Division (MSID) of the National Institute of Standards and Technology (NIST) in 1994—of industry needs in the area of manufacturing systems integration. The study addressed integration needs, existing technologies, and emerging standards in order to identify technical obstacles faced by industry in developing integrated manufacturing systems. Many of these integration problems will be addressed by NIST as part of the Systems Integration for Manufacturing Applications (SIMA) program (see box).

The background study was conducted for the Manufacturing Systems Environment (MSE) project element under SIMA. The MSE project[1] focuses initially on integration problems related to the design and production of electromechanical products. This document defines the initial MSE focus in detail and sets the technical direction for MSE project efforts.

The study was led by a 5-member MSE project management committee, and involved some 16 members of the MSID technical staff. Information was obtained through literature surveys, interviews with manufacturing experts from government and industry, and visits to manufacturers and vendors. The report also includes independent observations and findings by technical staff members with expertise in information technology and manufacturing systems.

This document aims to provide an understanding of the scope of the manufacturing systems integration problem, and to establish a basis for the SIMA MSE work. It is also intended as a useful overview for software developers, vendors, system integrators, and users of manufacturing software applications. Additionally, it provides a rationale for developing collaborative efforts between NIST, industry, other government agencies, research organizations, and standards-setting bodies.

The report is organized into three parts. Part I, consisting of Chapters 1, 2, and 3, is an overview of the MSE project. Chapter 1 introduces the project, explains the thinking behind it, and discusses in general terms what it will accomplish. Chapter 2 discusses the technical scope and objectives of the project. Chapter 3 discusses plans for implementing and testing MSE systems.

Part II, consisting of Chapters 4 through 8, reports the study findings concerning manufacturing software applications. Chapter 4 describes the principal types of manufacturing software applications, as determined by the study. Chapter 5 surveys commercial, off-the-shelf software products for these applications. Chapter 6 describes manufacturing software development activities currently under way in industry. Chapter 7 discusses related research trends, and Chapter 8 discusses standards pertinent to manufacturing applications.

SIMA AND THE HPCC

In 1994, NIST initiated the Systems Integration for Manufacturing Applications (SIMA) program as part of a new Federal government initiative on High Performance Computing and Communications (HPCC), which is described in the High Performance Computing Act of 1991 and Senate bill S.4, "The National Competitiveness Act of 1993."

NIST's program for FY 1994 and beyond is included under the Information Infrastructure Technology Applications (IITA) category of the HPCC initiative. The objectives of the program are: (1) to accelerate the development and deployment of HPCC technologies required for the National Information Infrastructure (NII), and (2) to apply and test these technologies in a manufacturing environment. Ultimately, these technologies will radically transform America's manufacturing environment, allowing individual companies to interact electronically as part of a "virtual enterprise" to produce world-class products for the 21st century.

The SIMA program will focus on technologies and standards that can improve computer systems integration and networking as applied to manufacturing. The program, which involves all eight NIST laboratories, emphasizes both product data exchange (for manufacturing) and electronic data interchange (for electronic commerce).

---

[1] Although it is referred to in this document as "the MSE project," the Manufacturing Systems Environment effort comprises several individual projects, as explained in Chapter 1.

Part III, consisting of Chapters 9, 10, and 11, deals with the information technologies supporting the integration of manufacturing software applications. Chapter 9 describes in some detail the process used to design an integrated system, and discusses the modeling methods, architectures and frameworks providing support for that process. Chapter 10 reports on interfacing mechanisms that may be used in the actual implementation of an integrated system. Chapter 11 surveys information technology standards in the areas addressed in Part III.

PART I: PROJECT OVERVIEW

Part I is an overview of the Manufacturing Systems Environment project. Part I is organized as follows:

CHAPTER 1: INTRODUCTION TO THE MSE PROJECT

This chapter provides background information about the SIMA project, and in particular those of its activities which fall under the major subheading of Manufacturing Systems Environment (MSE). The remaining chapters of this document deal exclusively with these activities, which will from here on be referred to collectively, for the sake of brevity, as the *MSE Project.* Section 1.1 discusses the challenges facing U.S. industry in advanced manufacturing, and the need for a systems approach to the introduction of new technologies. Section 1.2 discusses the potential contribution of manufacturing systems integration—the focus of the MSE project—towards meeting these challenges. Section 1.3 places the MSE project within the wider context of the SIMA program as a whole. Additional background information on MSE and supporting SIMA projects can be found in [1], Technical Program Description Systems Integration for Manufacturing Applications (SIMA).

1.1 THE NATIONAL CHALLENGE OF ADVANCED MANUFACTURING

To stay competitive in today's manufacturing environment, many companies are introducing advanced technologies, particularly computer-based applications, into their businesses. This is motivated by a belief that advanced technologies such as computer-aided design, manufacturing, and engineering (CAD/CAM/CAE)—combined with effective resource management and improved work force training and education—can greatly improve a company's competitiveness and profitability. There are numerous examples supporting this belief, yet on the other hand many cases exist where the expected benefits of advanced technology have not been realized by industry. Some examples of expectations and how they may go unrealized are shown in the table below.

So companies today find themselves between Scylla and Charybdis. They must accept either the costs and risks that accompany the introduction of advanced technologies or the risk of being driven out of business by competitors who are using those technologies to reach the market quicker, with better products at lower prices.

Despite the risks, businesses are increasingly choosing to introduce advanced technology. In 1993, for instance, the worldwide market for CAD/CAM/CAE software applications increased 5 percent to $16.5 billion, and it is expected to continue growing strongly.

The potential benefits of applying information-based systems to manufacturing are recognized by the Office of Science and Technology Policy in its August 1994 report *Information Infrastructure Technology and Applications* (IITA). The report describes advanced manufacturing capabilities needed to support Vice President Gore's National Challenges for the National Information Infrastructure:

> *"American manufacturers seek to recapture world leadership and respect. The specific technical goals are to exploit lean manufacturing (e.g., greater efficiency and lower cost), flexibility (e.g., variation in production runs to allow for*

Advanced technology is expected to . . .	**Expectations are often met, but . . .**
raise the quality of manufactured products	while introducing advanced technology usually results in a more *consistent* level of quality, it has not always *improved* quality
improve the productivity of people	people are less productive with new technology until they gain the training and experience to use it effectively
improve the productivity of systems	introducing new technology in one area can create incompatibilities with other business and manufacturing systems, resulting in loss of overall productivity
lower manufacturing costs	the direct costs of advanced technology are high, and indirect costs, such as those associated with training and systems re-engineering, can be even higher
reduce time from product conception to market	limitations of the interfaces between advanced information systems can restrict their range of interaction, leading to the inability of a manufacturing enterprise to react effectively to change

> *consumer preferences), and agility (e.g., supporting small production runs, rapid retooling, and exploitation of . . . electronic commerce services)."*

The report further describes how advanced information technologies might be used in manufacturing:

> *"In addition, companies will be able to band together to jointly manufacture goods. This will require rapid tailoring and composition of shared information services such as inventory control, work scheduling, and product delivery. Potential machine tool vendors and other manufacturing support companies will be willing to provide simulations of new process-control and planning software to enable companies to test before they buy. In addition, software specialty companies will provide access to powerful computer-aided design tools that are currently too expensive for purchase by small companies. This is economically viable because a small company can access both the software and the human expertise that lies behind it, through coordinated on-line consulting services."*

This description clearly identifies key technical elements of advanced manufacturing:

- Shared information services across enterprise functions
- Simulation of processes and plans
- Direct system access to engineering and analysis software tools
- Direct system access to human expertise compiled in knowledge bases

Feasible and cost-effective implementation of these elements will require more than the development of new technologies. The liabilities and risks of these advanced technologies must be reduced by applying a comprehensive systems approach to the integration of new technologies into existing manufacturing systems. To be applicable on a national scale, a systems approach should include:

- Industry-accepted models of system functions and interfaces
- Testbeds for experimenting with high-risk technologies and interoperability
- Prototype implementations demonstrating feasibility and cost-effectiveness
- National and international standards codifying proven practices

The MSE project will develop such a systems approach, focusing on advanced information technologies for manufacturing, and will disseminate the results through workshops, training materials, electronic data repositories, and other mechanisms.

1.2 PROGRAM BACKGROUND

BACKGROUND

The National Institute of Standards and Technology (NIST)'s program is part of the multi-agency High Performance Computing and Communications (HPCC) initiative as described in the High Performance Computing Act of 1991 and the Senate bill S.4, "The National Competitiveness Act of 1993." NIST's program for FY 1994 and beyond is included under the Information Infrastructure Technology Applications (IITA) category of the HPCC initiative. The objectives of the program are: (1) to accelerate the development and deployment of HPCC technologies required for the national Information Infrastructure (NII) and (2) to apply and test these technologies in a manufacturing environment. Ultimately, these technologies will radically transform America's manufacturing environment, allowing individual companies to interact electronically as part of a "virtual enterprise" to produce world-class products for the 21st century.

The program will focus on technologies and standards that will improve the systems integration function in manufacturing. NIST will perform appropriate activities in the areas of flexible computer-integrated manufacturing (FCIM) with emphasis on both product data exchange (for manufacturing) and electronic data interchange (for electronic commerce) standards that are part of the overall vision for 21st century manufacturing. The infrastructure technologies beingdeveloped will serve as an enabler for such manufacturing paradigms as Agile Manufacturing, Concurrent Engineering, and the Virtual Enterprise. The centerpiece of these activities will be a model facility at NIST, the Advanced Manufacturing Systems and Networking Testbed (AMSANT). Researchers nationwide will use the AMSANT facility to research and develop methods for applying HPCC technology to manufacturing. Besides the technology development, important functions of the program include improving the process for developing the key manufacturing interface standards and providing a technology transfer mechanism for getting the program results to industry.

NATIONAL INFORMATION INFRASTRUCTURE (NII)

The National Information Infrastructure (NII) is designed to promote a seamless web of communications networks, computers, databases, and consumer electronics that will put vast amounts of information at users' fingertips. Development of the NII can help unleash an information revolution that will change forever the way people live, work, and interact with each other. The NII is the platform of information technology resources upon which industry, government, and academia can integrate their information functions.

The HPCC/IITA initiative supports the key areas of research and development and systems integration to demonstrate prototype solutions to National Challenges starting from the advanced technology level moving through higher level of user capabilities to the ultimate user level in National Challenge projects. The IITA consists of four elements: (1) National Challenges are fundamental applications that have broad and direct impact on the Nation's well-being and competitiveness, (2) Information Infrastructure Services provide the underlying network-capable building blocks upon which the national Challenges can be constructed, (3) Intelligent Interfaces will bridge the gaps between users and the future NII, and (4) System Development and Support Environments will provide the network-based software development tools and environments needed to build the advanced user interfaces and the information-intensive National Challenges themselves.

The NIST program is concerned with one specific National Challenge, Advanced Manufacturing:

> Supports work in advancing manufacturing technologies through the use of HPCC capabilities in design, production, planning & quality control, marketing & user services. A key element is the development of the infrastructure necessary to make the process and product information accessible over the information highway to both enterprises and customers. Research areas include concurrent engineering, protocols for electronic exchange of product data, electronic commerce for manufacturing, virtual design technologies, etc.

Implementation of the NII concept for manufacturing will allow such capabilities as: (1) customer to “custom design” products, (2) companies to form alliances needed to produce new products (i.e., Agile Manufacturing), (3) small to medium size companies to interact with large companies for bidding on products (i.e., the Virtual Enterprise), (4) software system brokers to"rent" sophisticated manufacturing systems tools, and (5) rapid access to manufacturing knowledge by the product designers which will enable enterprises to use concurrent engineering practices.

1.3 THE BENEFITS OF MANUFACTURING SYSTEMS INTEGRATION

The MSE project is predicated on the belief that systems integration is key to the effective use of advanced information technology in manufacturing. Integration may help a company in several ways:

Knowledge and information: The quality of the decisions made by a company's managers and engineers depends on their knowledge and judgment, and on the information available to them when making the decisions. Decision-makers need access to accurate and timely information. In modern manufacturing facilities, many decision processes are supported by manufacturing software packages, and different decision-makers use different software packages designed for different functions. However, information transfer between packages is often inadequate or inaccurate. Integration of these packages into systems can significantly improve the availability, consistency and accuracy of the information delivered to the decision-makers and thus enhance the quality of their decisions.

People: The systems approach defines the principal functions of systems and standardizes the form of—and in many cases the access to—the principal information units. This reduces learning time for people who have to deal with multiple systems or new systems. Moreover, systems integration eliminates the need for human involvement in non-value-added activities such as copying, reorganizing, and reinterpreting information passed between different software systems. This releases skilled personnel for the value-added activities that further the interests of the company.

Systems: The systems approach defines the information interactions among all components of the system, so the impact of changes in one component can more easily be evaluated, and the overall system can be adapted to make the best use of new technologies. For example, improving the capability of a machining process without changing the economic models used in tolerance synthesis during product design may simply drive up manufacturing costs for the product. Such problems can be avoided through the use of integrated systems models.

Process management: Integration of design engineering, manufacturing engineering, and production software systems improves the availability of production-related information and constraints to designers, resulting in better design decisions. It also improves the availability of design-related information and constraints to production engineers, resulting in better production decisions. The major saving is in the number of engineering iterations required before production can commence.

Metrics and diagnostics: Systems integration creates the paths by which automated flow ofinformation among systems takes place. This allows (without requiring) automated capture of the timing, source, and content of information transfers without special actions on the part of engineers and managers. Such captured data enable the evaluation of performance metrics for engineering and manufacturing systems. The data also provide valuable support in backtracking and diagnosing problems in the product realization process.

An activity model shown in Figure 1 defines the elements of SIMA's manufacturing model and identifies the major flows of information and controlling functions.

1.4 OVERVIEW OF THE SIMA PROGRAM

Details of the broad background of the SIMA Program as part of the federal government HPCC initiative were given in the Preface and also in section 1.2. As stated there, all eight NIST laboratories are involved in the program. By the time it is completed, NIST will have developed, tested, validated, and demonstrated multiple integration methods, tools, and technologies for integrating product and process-related activities in the product realization process. The intention is to use commercially available, state-of-the-art software systems wherever possible, and to utilize existing or emerging infrastructure technologies and standards to provide means for the construction of modular, open, reconfigurable, intelligent integrated systems.

The STandard for Exchange of Product model data (STEP: see Chapter 8) is considered to be the key standard for integration activities, although many other interface standards are potentially useful within the program. The STEP effort is expected to accelerate the evolution of concurrent engineering, support electronic commerce, and enable business partners to share sophisticated digital product data as easily as paper drawings are shared today. If it lives up to its promise, STEP will be one of the most influential standards that has ever been developed in the field of industrial automation.

Three major environments have been defined for SIMA program activities. These environments were defined as a result of a joint NIST/Industry workshop on defining systems integration ne eds for manufacturing. The workshop conference report [2] recommendations helped focus the SIMA program activities around three major needs. They include Standards Development needs, Technology Development needs and Technology Transfer needs. The SIMA program followed the workshop recommendations by creating three program environments which focus on the various needs in each area. The program environments are:

- Manufacturing Systems Environment (MSE),
- Standards Development Environment (SDE), and
- Testbed and Technology Transfer Environment (TTTE).

Each environment includes projects appropriate to particular NIST roles in support of the HPCC initiative, and addressing the major technology and standards issues outlined in the IITA programreport referred to in Section 1.1. As mentioned earlier, the present document covers work performed in the Manufacturing Systems Engineering Environment of SIMA, referred to in what follows as the MSE Project. Both this and the other two major SIMA components are outlined below in order to place the MSE work in its wider context.

1.4.1 THE SIMA MANUFACTURING SYSTEMS ENVIRONMENT (MSE)

The major focus areas of MSE are the development of information models, infrastructure technologies and interface protocols to support systems integration, and the application of HPCC to the design, planning and production activities of the product realization cycle. The chosen product domain is that of electromechanical products, though many of the MSE deliverables will have relevance to the integration of manufacturing applications in other domain areas. Little more will be said under this sub-heading, since the MSE scope, domain and methodology are described in detail in the remainder of this report.

1.4.2 THE SIMA STANDARDS DEVELOPMENT ENVIRONMENT (SDE)

The SDE objectives are:

- to assist industry in implementing voluntary consensus standards relevant to computer integrated manufacturing (CIM),
- to facilitate industrial testing of new applications of advanced manufacturing systems and networks,
- to facilitate efforts to develop and test new data exchange standards utilizing HPCC technology, and
- to accelerate industry deployment of consensus standards.

Within SDE there is a general theme of providing effective support environments for the development of standards as well as facilitating harmonization across a broad spectrum of standards supporting many diverse aspects of enterprise integration.

The following are the primary SDE focus areas:

- Conformance Testing - development of methods for verifying the compliance of vendor-developed products with existing standards
- Application Protocol Development Environment - creation of an environment providing tools and information to facilitate the creation of STEP application protocols by standards developers
- Standards Documents Repository - creation of a logically structured framework withinwhich the development of standards can take place
- Information Modeling - development and use of mechanisms for the representation of the information required in specifying standards
- Standards Methodologies - development and use of methodologies that ensure the generation of useful and unambiguous standards.

1.4.3 THE SIMA TESTBEDS AND TECHNOLOGY TRANSFER ENVIRONMENT (TTTE)

The TTTE objectives are as follows:

- to develop a technology transfer infrastructure for the exchange of manufacturing information using HPCC technology,
- to develop, in collaboration with industrial partners, prototype information services that could eventually be commercialized,
- to develop services providing document searches and retrieval of government and other research reports relevant to the standards-making process,
- to establish communication channels for a network of researchers and implementors of manufacturing technologies,
- to serve as a demonstration site for the use of industrial technology suppliers and users,
- to serve as the interface to a network of technology development testbeds across the United States, and
- to serve as the interface to one or more information dissemination organizations.

1.5 TIMESCALE AND RESOURCES OF THE MSE PROJECT

The MSE project is planned for five years. During that time, it will develop through three distinct phases. Phase I includes analysis of integration problems and design of integration solutions. Phase II includes implementation and testing of integration solutions through prototypes and demonstrations. Phase III includes promotion of results through standards and industry implementations.

During Phase I, lasting two years, the project will gather information on new technologies that support integration, develop information models and interface specifications for a broad spectrum of manufacturing applications, and acquire CAD/CAM/CAE software systems representative of those used by industry. The objective will be to understand current integration limitations andto develop an engineering architecture that defines the technologies and standards to be used in Phase II to support MSE integration demonstrations. Activities in Phase I will allow the examination of integration requirements across a wide range of engineering activities, so that the results will be useful to many sectors of industry.

During Phase II (years 3 and 4) an integrated system will be constructed, and integration solutions will be tested in collaboration with industry through computing and communication facilities provide by the NIST AMSANT (see below). Phase II demonstration activities will be more narrowly focused due to a number of factors, including resource constraints on the range of commercial packages that can be included in the integrated system, the capabilities of those packages, their level of conformance to standards such as STEP (STandard for the Exchange of Product model data), the availability of human resources, and the choice of product domain. The specific domains of interest may be influenced by collaborations with industry.

The third and final phase of the MSE project (year 5) will concentrate on the dissemination of integration solutions developed in Phase II, in order to promote the implementation of these solutions as future standards.

MSE will make use of SIMA's Advanced Manufacturing System and Networking Testbed (AMSANT), which will serve as a site for demonstrations and for testing high-risk technologies and interoperability by industrial technology suppliers and users. AMSANT is a distributed set of testbeds linking the computing resources of researchers at NIST and in industry in order to promote collaborative development of integration solutions. AMSANT will provide high-speed communications, software development tools, information repositories, and physical laboratory space in order to accomplish MSE goals. Researchers from across the United States will use the AMSANT facility to research and develop methods for applying HPCC technology to systems integration problems affecting manufacturing.

1.6 COLLABORATION AND TECHNOLOGY TRANSFER

Throughout the MSE project, strong collaborations with industry, other research institutions, and standards organizations will be developed and maintained. Prototype systems and interface specifications will be communicated to appropriate standards organizations. Results will be made available to U.S. industry through workshops, training materials, electronic data repositories, and pre-commercial prototype systems that can be installed by potential vendors for test and evaluation. NIST will distribute standards reference data, technical information, and product designs via digital library technologies.

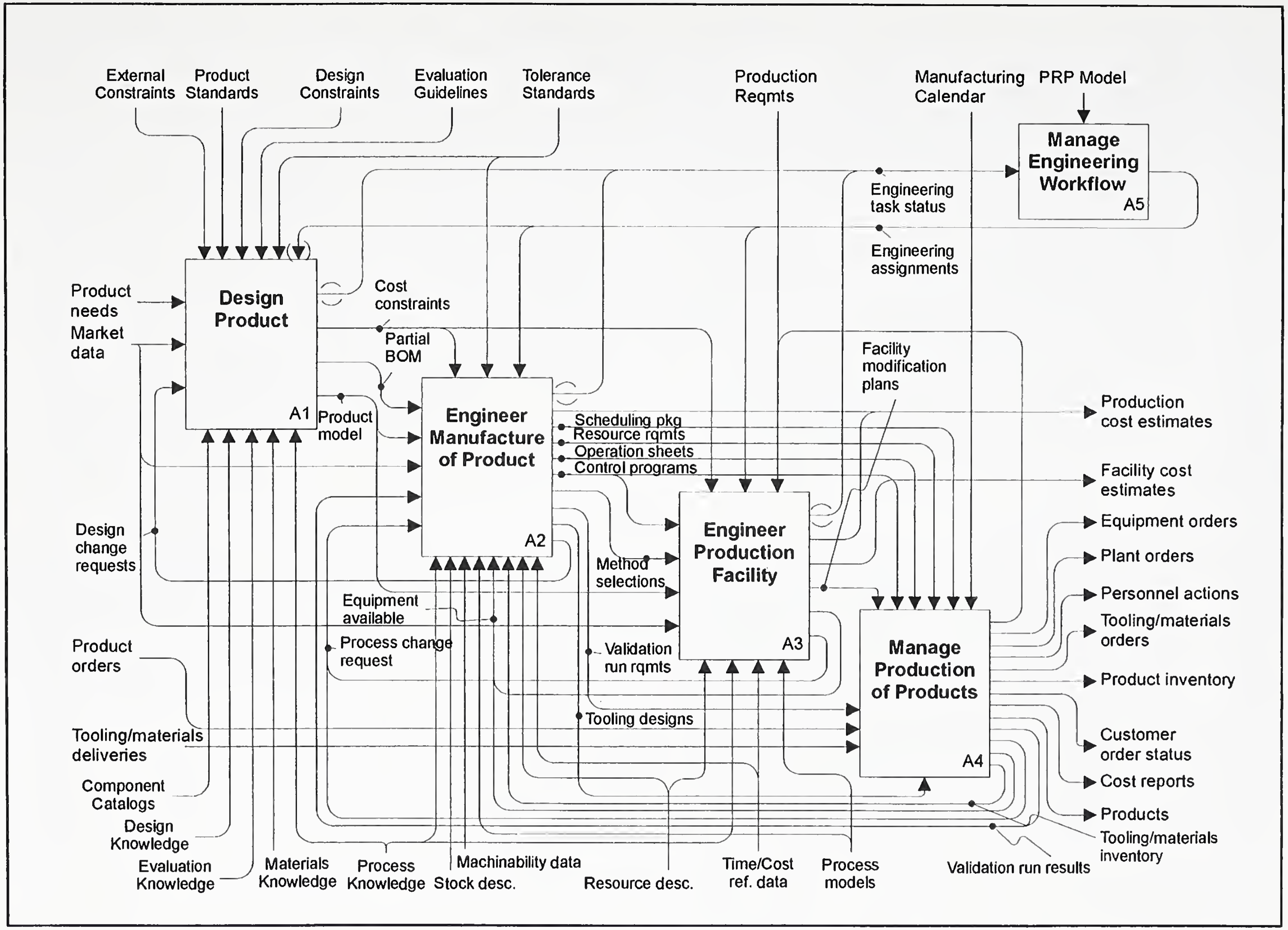

Figure 1. SIMA Manufacturing Activity Model

CHAPTER 2: OBJECTIVES AND SCOPE OF THE MSE PROJECT

The product realization process is defined as:

> *The process by which new and improved products are conceived, designed, produced, brought to market, and supported. The process includes determining customers' needs, translating these needs into engineering specifications, designing the product as well as its production and support processes, and operating these processes.*[2]

The development of computational aids supporting specific aspects of the overall process has led to what are sometimes called "islands of automation." The MSE project is concerned with building bridges between the islands by integrating specialized software systems that support various parts of the product realization process.

The MSE project will focus on the technical aspects of that process, in the hope that they will be integrated with the more management- and business-oriented aspects in the future. This chapter briefly outlines the activities covered by the MSE project and explains how the product domain and scope of the project were decided.

2.1 OBJECTIVES

The overall objective of the MSE project is to provide industry with open architectures and interface specifications that will facilitate the implementation of CIM (Computer-Integrated Manufacturing) systems built from commercially available software packages.

This will be achieved through the use of a formal systems approach to the specification and implementation of integrated manufacturing systems. A suite of generic models and specifications will be developed for the information and processes within the scope of the MSE project. These will be validated by the implementation and demonstration of one or more integrated systems based on commercially available software.

The models and specifications will provide guidance to software developers and vendors on making their packages easy to integrate within larger heterogeneous systems. They also will be used to identify critical areas for—and provide technical contributions to—the development of new standards. An information repository containing the MSE models and specifications, as well as other related models, specifications, and software, will be made publicly available on the Internet to serve the needs of system developers, system integrators, and the research and development community.

2.2 OVERVIEW OF THE SCOPING ACTIVITY

An important activity in Phase I of the MSE project has been the further refinement of the following initial broad guidelines:

- The domain of the MSE project is discrete manufacturing, limited to cutting, forming, and mechanically assembling component parts.
- The scope of the project is limited to the design engineering, manufacturing engineering, and production phases of the product realization process.

A primary intention has been to identify a body of work that is appropriate for NIST to undertake, that advances what has been done before, and that will have a high payoff in terms of value to industry. The primary aims in establishing the scope have been to ensure that:

- The project is feasible in terms of current resources and available technology
- The results are "upgradable" to accommodate future developments in engineering and information technology
- The domain is relatively self-contained, i.e., it has limited interfaces with other organizational activities
- The results are able to fit readily in the context of a larger enterprise system when appropriate

The budget and human resources available for the MSE project also were taken into account in determining its scope.

Since product realization processes are not the same for different types of products, it has been necessary to decide on a product domain appropriate for the MSE project. Boundaries also have been defined for the range of product realization activities covered and the aspects of information systems technology taken into account. These decisions and their rationales are detailed in the following sections.

2.3 SCOPE OF PRODUCT DOMAIN

The manufacturing processes specified in the guidelines quoted above are appropriate for mechanical products. But many familiar artifacts such as kitchen appliances are electro-mechanical in nature, and since a significant sector of U.S. industry manufactures such prod-

---

[2] Improving Engineering Design: Designing for a Competitive Advantage, National Research Council, 1992.

ucts, electromechanical products have been included in the MSE project domain. In order to adhere to the guidelines, that domain will exclude the design and production aspects of electrical and electronic components; such components will be treated only as additional parts to be assembled into the final product. This rules out consideration of systems oriented specifically toward electrical or electronic design and manufacture, and avoids possible conflict with other national research programs.

A further reason for excluding electronic products is that design and manufacturing integration in this area is currently more advanced than it is for mechanical products. In the field of microelectronics, for example, it has been possible for some years to proceed in an integrated manner from logic design, through chip layout and functional simulation, to manufacturing planning. This is far from the case in the domain of mechanical products, where geometric problems are more severe and the range of engineering activities is much wider and more diverse.

Even with the above provisos, electromechanical products represent an extensive domain, including the manufacture of parts by

- Machining
- Injection molding
- Die casting
- Sheet metal stamping and pressing

These account for the great majority of mechanical parts made today.

The MSE project will concentrate on these four processes. However, in a time of rapid technological change, it will be important to give some consideration to less commonly used methods that may become dominant in the future. Some of these are similar to the processes listed (e.g., other types of molding and casting, forging, etc.). Less traditional methods include powdered metal forming, filament winding and other composite techniques, electro-discharge machining (EDM), and specialized sheet metal forming processes such as stretch-forming and shot-peening. Solid free-form fabrication processes, including stereolithography and selective laser sintering, currently are used only for prototyping, but may in the foreseeable future be used in production processes. The MSE project will monitor the progress of these emerging technologies; in Phase I however, attention will be restricted to noting their specialized information requirements.

Part materials will include metals, thermoplastics, and composites, the last being increasingly important in the manufacture of aircraft, cars, sporting goods, and medical equipment. Products within the chosen domain may also require materials conditioning operations such as heat treatment, and material finishing operations such as painting or anodizing.

There also is a domain-related aspect to product assembly. Aside from the purely geometrical positioning and orientation of mated parts with respect to each other, this concerns the actual fabrication of large parts from smaller components by joining techniques such as riveting, bonding, or welding. Some of these techniques have been automated, and so are included within the scope of the MSE project.

2.4 SCOPE OF PRODUCT REALIZATION ACTIVITIES

In order to make the project reasonably self-contained, it will cover the following product realization functions:

- Design engineering
- Manufacturing engineering
- Production

Activities covered under each of these headings are listed in the following subsections and described in more technical detail in Part II.

2.4.1 DESIGN ENGINEERING ACTIVITIES

The design engineering function starts with the requirements for a new product. Its output is a completely documented specification of that product, including a geometric description.

Although the design process can be subdivided in several different ways, for the purposes of this report it is divided into four phases, each with its own output:

- Product planning (Output: design problem specification)
- Functional design (Output: functional decomposition of design)
- Configuration design (Output: layouts; assembly/subassembly structure; materials specification; preliminary cost estimates; safety, maintainability, and environmental considerations, etc.)
- Detail design (Output: detailed drawings or product models; analysis results; detailed cost estimates; nonfunctional prototypes from stereolithography; etc.)

Computer aids for the first two phases are largely nonexistent today, and they are therefore excluded from the scope of the project. Commercially available rule-based systems are currently used in industry for some aspects of configuration design. Conventional geometry-based CAD systems as well as newer feature-based and constraint-based systems are used for detail design. All these types of design systems fall within the scope of the MSE project.

It must be noted that a high proportion of design activity in industry is not design from scratch, but rather modification or redesign of existing products for improved performance or lower production cost.

A more detailed list of design engineering activities within the MSE project scope includes:

- Layout design
- 2-D drafting, including specification of dimensions and tolerances
- 3-D modeling of parts and assemblies
- Selection of standard parts from catalogs
- Application of company-specific or standards-related design rules
- Materials selection
- Generation of bills of materials
- Performance of structural, vibration, and thermal analyses, using finite element (FE) or other techniques
- Other special purpose functional simulations
- Design optimization
- Management of product data
- Generation of design documentation
- Use of rapid prototyping for design verification

The above is merely a list of functions and is not intended as a system decomposition. These functions are fully described in Part II.

Design is not a one-time-only process; as noted above, redesign for product improvement is common, while design changes also are made in response to feedback from manufacturing engineering, as is shown in the next section.

2.4.2 MANUFACTURING ENGINEERING ACTIVITIES

Traditionally, once the design engineering phase is complete, the resulting product specification becomes the input to the manufacturing engineering phase, which essentially determines how the product is to be made. There is a current trend toward *concurrent engineering*, in which some activities from these two phases are carried out in parallel to shorten the overall product realization cycle. It is important that the results of the MSE project are compatible with this mode of operation.

The following manufacturing engineering activities fall within the MSE project scope:

- Process planning
- Cost estimation
- Tooling design and planning
- NC (Numerically Controlled machining) program generation
- Inspection planning
- Assembly planning
- Simulation of process plans
- NC program verification

These activities (which will be discussed in more detail in Part II) fall under the two main headings of planning and simulation. As will be shown below, planning activities may give rise to subsidiary product realization cycles.

Process planning is the task of determining a set of manufacturing operations—and a sequence for those operations—that will result in the satisfactory manufacture of a product. Usually, givena comprehensive set of manufacturing resources, there are many possible manufacturing plans, and a company seeks to find the one that is near-optimal in terms of either cost or manufacturing time.

There are two basic approaches to this problem. Variant process planning methods essentially edit existing plans for similar products, and rely on some form of coding to measure similarity between parts. Generative methods create plans from scratch; they need access to information on available manufacturing resources, but make no prior assumptions about the part itself.

Variant systems are used widely in industry, and most rely heavily on human interaction. Generative systems are increasing in popularity, since they provide greater flexibility and are potentially easier to integrate into larger systems. Since commercial process planning systems of both types are available, both approaches will be included in the MSE project scope. The use of variant planning will require some means for part coding.

There are many examples of feedback from manufacturing engineering into design engineering. For instance, one of the important outputs of process planning is a detailed estimate of production costs; a high estimate may lead to a request for redesign so that the product can be produced more cheaply.

Other planning activities include NC, assembly, and inspection planning. In the case of machined parts, NC planning specifies detailed machining strategies, from which control programs are generated to drive machine tools used in the manufacturing process itself. Inspection planning defines strategies for the inspection of manufactured parts to ensure that they meet their original design specifications. Assembly planning finds sequences by which parts can be assembled into subassemblies and full assemblies. Assembly planning is also relevant to product maintenance and repair, since they usually involve the reverse process of disassembly followed by reassembly.

Most manufacturing processes have specific requirements regarding tooling. For example, machining may require the use of one or more fixtures to hold the part while it is being processed, while injection molding requires the creation of a mold. Thus, once a part has been designed and the manufacturing process specified, an additional product realization cycle may have to be completed to generate the tooling requirements for the main production process.

It is possible to use the output from many types of planning activities to drive graphical simulations of the operations they specify. In an industrial context, this provides a valuable means of checking plan validity without the expense of taking production equipment out of service to carry out real tests on the shop floor.

Simulation also will provide a means for extending the range of activities covered in the MSE project. For example, resource limitations will restrict the number of production methods that can actually be implemented in SIMA demonstration systems. But simulation will allow the MSE project to investigate the integration of other methods. Simulation is therefore essential in the MSE project.

2.4.3 PRODUCTION ACTIVITIES

It is in the production domain that the actual manufacturing processes occur. These are specified during the manufacturing engineering phase, which generates data essential for their control. However, further control data are generated by other activities within the production domain. This is partly because resources usually need to be shared in manufacturing a range of products rather than just a single product.

One boundary to the activities covered by the MSE project was set in the design domain, between functional and configuration design. A reasonably clear-cut boundary also has been identified in the production domain. The activities in the Finance and Administration sector of Figure 2.1 are either not computerized or are not intimately linked to the product realization cycle. The most direct links are those between the production activities and Finance, Procurement, and Distribution. These are relatively narrow information channels, which will be given sufficient consideration during the MSE project to ensure that results may be applied in a wider organizational context in the future.

Production systems handle the following activities:

- Production scheduling
- Production control
- Materials requirements planning
- Manufacturing resource planning
- Inventory control
- Shop floor monitoring
- Job tracking
- Tool management
- Production simulation
- Physical shop-floor processes

Production activities determine the products to be manufactured at any one time, the order in which they are produced, and the allocation of resources to their production; they also ensure the quality of manufactured products. These activities are further discussed in Part II, with the exception of the last; although the software systems controlling and simulating them are within the MSE scope, the physical processes themselves (including machining, materials handling, assembly etc.) are not included.

Simulation also plays several important roles in the production domain, primarily in the optimization and verification of production plans. As in the manufacturing engineering domain, simulation will provide a valuable means for extending the effective scope of MSE project through the use of virtual rather than real production facilities.

2.5 SCOPE WITH REGARD TO INFORMATION SYSTEMS TECHNOLOGY

An integrated product realization system consists of many software modules, each of which may be regarded as an information system in its own right. Examples include CAD systems, planning systems, resource and materials databases, and scheduling systems. Each of these may generate information, store information, acquire information from other systems, or pass information on. Usually, the individual modules will be distributed over a range of hardware platforms. To make these components work together effectively, it is necessary to allow them to share information and to make use of each others' capabilities.

This requires the developers of an integrated system to agree on:

- What information each system needs, and what information it generates
- What requests each system will respond to, and what functions it will perform
- What information exchanges will occur, when, and what systems will be involved
- How the information is to be exchanged

The specification of the functions a system will perform, and the information it needs and provides, is termed a *system architecture*. The specification of what information exchanges will occur, and how, is termed an *interface specification*. The MSE project must therefore define system architectures and interface specifications that permit integration of the component systems of design engineering, manufacturing engineering, and production systems. The scope of this activity encompasses:

- Functional models—the specification of required functions, inputs, and outputs
- System architectures—the assignment of those functions, inputs, and outputs to specific systems
- Information models—the specification of the semantics of shared or exchanged information
- Exchange formats (file formats, message formats, database schemas)—the specification of the form in which the information is exchanged
- Protocols—the specification of the rules for the exchanges
- Interface specifications—detailed specification of the form of function requests and responses, including the nature of the information exchanged

In the context of protocol and interface specifications the primary focus of the MSE project will be on *application protocols* (APs) and *application programming interfaces* (APIs). These are the protocols and interfaces allowing direct communication with the manufacturing application software modules. Additionally, the MSE project will need to identify communications and networking protocols to serve as media for the exchanges.

A major part of the work in this area will stem from the differences between "ideal" and commercially available product realization systems in terms of architectures and interface specifications. One possible approach will be to embed each commercial system in a "wrapper" that makes it *appear* to comply with the ideal specifications for communicating with other systems. In some cases this will involve converting the native internal information formats of the embedded systems into the chosen ideal formats. Where possible, the MSE project will collaborate with the developers of commercial systems in overcoming these problems.

There are several existing and emerging standards for manufacturing information, systems, functions, and exchanges, and the MSE project will make use of these where appropriate. In addition, detailed architectures and information models proposed by various research consortia may prove valuable. In these areas, the MSE project will identify needs for new standards, make significant contributions to emerging standards activities, and support the broader acceptance and standardization of industry-developed specifications whose use is judged to be beneficial.

There also are numerous existing and emerging general-purpose information technology standards that are potentially relevant to the MSE project. In many cases it will be possible to use corresponding off-the-shelf commercial products, enabling the project to concentrate on the engineering and manufacturing application concerns. Such standards and products will be identified, evaluated, and in some cases used. However, it is not expected that the project will identify the need for new developments in information technology or general-purpose information technology standards.

2.6 QUALITY CONTROL

Quality considerations pervade the whole of the product realization cycle, and will be considered in more detail in various sections of Part II. The drive for product quality is a unifying influence throughout the domain of the MSE project. The term "quality" is used in both a narrow and broad sense. In its broadest interpretation, it refers to the responsiveness of an institution to societal needs. For a manufacturing enterprise, quality means meeting the needs of customers (end users, in particular) in regard to price, delivery date, and fitness for use. The most common broad view of quality is fitness for use, excluding price and delivery dates from the domain of quality.[3] "Fitness for use" is always taken to mean fitness as perceived by the *user* of a product or service. The evaluation of fitness for use by the producer or supplier is essentially irrelevant to the definition of quality (but not, of course, to the implementation of quality).

For products, fitness for use can be defined in terms of four classes of quality characteristics:

- *Quality of design* is a composite of three elements: identification of what constitutes fitness for use; conceptual design of a product fit for use; and development of a detailed product specification that will meet all users' needs.
- *Quality of conformance*—the extent to which the product conforms to the detailed specifications. This is the narrow sense of quality, and may include variables such as technology, personnel skills, and management of the production process.

---

[3] The definitions used in this section are taken primarily from: Juran, J.M., *Quality Control Handbook*, McGraw-Hill, 1974.

- *The utility of a product over time*—a category including traditional measures of availability, reliability, and maintainability. It also may include usability—the extent to which a product is convenient and foolproof for users.
- *Field service*—the user's ability to secure continuity of use after purchase.

Another characteristic, *manufacturability,* is often considered a quality characteristic. However, this is concerned with whether a design can be manufactured with a given set of resources. It therefore relates to the quality of design from a company's internal viewpoint, and is not closely related to fitness for use of the resulting product by the customer.

It is clear from the above discussion list that the product specification—the material properties, geometric configuration, tolerances, finish requirements, etc.—is a *substitute* for the real goal: fitness for use.

It is important to understand the basic terms of quality. The *Quality Control Handbook* [4] defines the *quality function* as "the entire collection of activities through which we achieve fitness for use, no matter where these activities are performed." It also defines *quality control* as "the regulatory process through which we measure actual quality performance, compare it with standards, and act on the difference." (Note that product inspection is only one part of quality control.) Finally, *quality assurance* is defined as "the activity of providing, to all concerned, the evidence needed to establish confidence that the quality function is being performed adequately." If quality control is comparable to accounting, quality assurance is analogous to the financial audit.

The MSE project is concerned with the quality function to the extent that quality activities are within the MSE project scope as described earlier in this chapter. Thus, in design engineering, the quality of the specifications resulting from embodiment and detail design are within the MSE project scope, while quality control of the conceptual design is not. The quality of process specifications developed during the manufacturing engineering phase are all within scope. Similarly, quality control of production activities includes the quality of production plans, production control strategies, etc.

All major quality objectives are cross-functional in nature. The MSE project will identify and address a multitude of integration issues regarding the implementation of the quality function. Some of the most obvious opportunities for improving implementation of the quality function involve communication between the activities in the scope of the MSE project. A number ofexamples can be identified. Design engineering must represent both functional and nonfunctional characteristics, and must identify the difference. (Typically, the Design Department must be a party to any waiver of functional requirements, but need not be for nonfunctional requirements.) Designers must in turn receive enough data to select tolerances, which involves a tradeoff between fitness for use and manufacturing costs. In theory, the designer should perform a formal tradeoff analysis; in practice, this is seldom done, often because of a lack of data or resources to model downstream effects. (This can result in the undesirable situation of unrealistic tolerances being loosely enforced.) Finally, manufacturing engineering must have access to process capability data to design the manufacturing processes for a product.

The MSE project will need to analyze these and other quality considerations in defining a systems approach to manufacturing. All aspects of MSE project work—data requirements, database architectures, functional models, and other elements—are affected by quality control issues.

---

[4] Juran, *op.cit.*, Chapter 2.

Chapter 3: Project Implementation

The major focus of the MSE project is research and development of integration solutions. The implementation of manufacturing systems is not included in that focus, *per se*. However, because system implementations can serve a number of MSE project goals, they will be an important part of the project. Part of the background study therefore involved an analysis of implementation issues in order to support implementation demonstrations planned for Phase II of the MSE plan. This chapter discusses the study findings in the areas of implementation scenarios, selection of test-case products, selection of system components, and plans for implementing scenarios in Phase II of the MSE project plan.

3.1 Implementation Scenarios

Systems implementation can serve three purposes: to demonstrate the feasibility and benefits of MSE project results; to carry out engineering experiments with specific technical goals; and to focus MSE project activities in general. These reasons may conflict in terms of their implementation requirements, and careful attention must be given throughout the MSE project to identify leveraging opportunities.

Demonstrations will be essential for communicating project results to a broad audience of industry, government, and academic visitors, who are likely to visit the MSE project on a yearly basis. They must be designed to emphasize the impact of MSE accomplishments and to point out optimal future directions for the project. The demonstration systems must also be sufficiently polished that their objectives are not obscured by technical minutiae. Test-case products must be chosen to highlight the technologies used to integrate design engineering, manufacturing engineering, and production functions.

The MSE project will carry out engineering experiments to identify and test solutions of specific systems integration problems. Major issues in MSE are interface definitions, testing the feasibility of standards, and systems interoperability. The implementation environment must support these experiments as efficiently as possible. Infrastructure systems—network communications, workstations, operating systems, etc.—should support open, distributed processing to enable these experiments. Test-case products must also exhibit manufacturing problems being addressed by the MSE project.

Developing integration solutions useful to industry requires the MSE project to focus its demonstrations on real-world manufacturing scenarios. The implementation of these scenarios must include software systems and test cases provided by industry collaborators actively working on problems in design engineering, manufacturing engineering, and production. Multiple scenarios will be defined and evaluated in Phase I, in order to devise a scenario that can be supported by the resources planned for Phase II.

3.2 Selection of Test-Case Products

Test-case products will be selected many times throughout the MSE project. This section identifies the criteria to be used in this selection and identifies candidate products. The choice of specific test-case products will be the subject of future MSE project tasks.

Within the product realization process cycle, the nature and level of detail of the information used by different activities varies. Use of a common test-case product throughout the cycle will provide insight into the completeness and correctness of the shared information and its convenience of use in each of the manufacturing activities. The product data will be chosen to encompass all essential information requirements in each domain within design engineering, manufacturing engineering, and production, so that no application is limited by another's unique requirements.

Generally, test-case products should be representative of the mechanical parts manufacturing industry. They should pose challenging manufacturing problems, yet be feasible within the scope and domain of the MSE project. Specific product selection criteria are discussed further below.

3.2.1 Product Perception

To serve as demonstrations, test-case products should be familiar to the public, present problems of significance to MSE's customers, and have relevance for a broad spectrum of the U.S. manufacturing industry. Manufacturing problems associated with the products should relate to integration issues rather than to processing technology or other issues unrelated to the scope of the MSE project.

Generally, consumer products are most likely to satisfy these requirements. Workshop tools, kitchen appliances, and certain recreational equipment are likely candidates.

High-technology products run the risk that the processing requirements (e.g., extremely tight tolerances) may distract attention from the manufacturing integration issues.

Similarly, defense-related products do not meet these requirements. They are viewed as high-technology, complex, extremely expensive to fabricate, and highly specialized. These products can pose very interesting and highly complex engineering and manufacturing problems, but their choice could also raise controversial side-issues, again shifting attention away from integration problems. In addition, the applications, tools, and integration methods used in the defense industry often differ from those used in commercial industry due to federal guidelines, standards, and contract restrictions. Defense-related industries in any case are the focus of other national research programs.

The following list of product categories consistent with the selection criteria were identified in the study:

- Functional subsystems (e.g., automotive brake system, aircraft landing gear)
- Health care (e.g., glucometer, wheelchair, orthopedic devices)
- Hobby (e.g., radio-controlled vehicles, camera)
- Home entertainment (e.g., VCR, camcorder, CD/tape players)
- Household appliances (e.g., major appliances, small appliances)
- Office equipment (e.g., computer equipment, furniture, printers)
- Recreational (e.g., jet ski, bicycle, roller blades)
- Tools (e.g., home workshop, industrial, garden)

3.2.2 PRODUCT TECHNICAL CHALLENGES

Ideally, test-case products should pose significant research challenges in all MSE domains. Design engineering research, for example, will benefit from a choice of test-case products having alternative designs so as to allow study of different design scenarios, such as design from scratch versus redesign. Manufacturing engineering research will benefit from test-case products belonging to common product families, allowing study of variant design and variant planning. Production research will benefit from a diverse mix of test-case products, allowing a study of production planning, scheduling, and resource allocation issues.

All MSE projects have recommended that test-case products be assembled from several components. The components of each product should represent various fabrication processes such as machining, near net shape formation (i.e., sintering, castings), and plastic injection molding, in order to test process-specific information requirements. It is also desirable to use a whole product rather than a functional subsystem or assembly within a larger product, so that "uninteresting" components also get considered in the testing. The use of test-case products with multiple components will provide a higher level of project input with respect to integration issues cutting across the major activity areas.

3.2.3 PRODUCT MANUFACTURING CHARACTERISTICS

Several manufacturing characteristics affect the feasibility of producing test-case products within the resources of the MSE project. The following characteristics were identified in the background study as well as in recommendations from project participants.

Size/weight: The size and weight of a product is important in that the processes used to manufacture and assemble the product have limitations. Size limitations usually are defined in terms of a "working envelope." This defines the space that a product may occupy at each machine or assembly work station without adversely affecting machine or assembly operations. The weight of a component or assembly also can dictate whether special lifting equipment or specialized machining equipment is required. Based on project input, the work envelope of test-case products should be restricted to approximately 0.1 cubic meter to maximize the number of manufacturers that meet the machining and assembly characteristics.

Complexity: The complexity of a product is a function of the number and diversity of the engineering and production activities required for its realization. The following are thecomplexity factors considered and the corresponding recommendations:

- Number of components: preferably between 20 and 50, with a maximum of 250
- Complexity of the components (i.e., features and tolerances): low to moderate for 90 percent of components
- Number of processes required to fabricate the product: between 5 and 10
- Types of processes required to fabricate the product: varied, and the more the better
- Use of high-technology processes (i.e., company proprietary processes): none required, but preferably at least one and no more than two
- Mix of manufactured vs. procured components: require both, with a minimum of 25 percent of the components manufactured in-house

- Special tooling and fixturing requirements: preferred for at least one operation
- Labor skills required: medium

Manufacturability/Assemblability: Manufacturability and assemblability of components are functions of material selection, manufacturing resources, assembly equipment, tooling, fixturing, and tolerances of manufacturing and assembly features. To achieve cost and quality goals, the product design must take into account manufacturing and assembly processes. Test products should not be extraordinarily difficult to make, because that would tend to divert the research away from integration issues. On the other hand, test products should be such as to make manufacturing engineering considerations important in some of the design choices.

3.2.4 PRODUCT AVAILABILITY

Product availability means that the product is either readily available at low cost (e.g., less than $150) or can be easily fabricated. Ready availability would facilitate using an actual product and its components, along with associated CAD models, to demonstrate project results. Using such "concrete" examples would help communicate and clarify key project objectives. On the other hand, the proprietary nature of some products may be a barrier to their use as test cases in the absence of collaboration by the manufacturer.

Two MSE projects have expressed a desire to have some key components of the test-case products fabricated at NIST. The ability to make a test product in-house at NIST depends largely on the processes involved in designing the part. Facilities and equipment to cast, forge, and extrude are not available within NIST.

3.2.5 MARKET CONSIDERATIONS

The volume of production can drastically affect the entire product realization process. A product may be designed quite differently if only one is made, as compared to 100,000. Likewise, manufacturing plans and production requirements may vary significantly with market volume. Test-case products should therefore reflect a range of market requirements to better address these issues.

Special reliability and safety restrictions, such as commonly apply to health-care products, create additional design requirements and also affect the requirements for process control and inspection during production. Such products therefore make good test cases for these aspects of manufacturing systems integration.

3.2.6 RECOMMENDATIONS

Based on consideration of all the factors described above, the background study identified three classes of candidate test-case products, as follows.

Household: A hand-held hair dryer could be a suitable example. This product is familiar to the public. Although it is viewed as low-tech, its design and manufacturing processes meet program guidelines. Due to the product's intended use and the environment in which it is used, strict safety regulations have to be met regarding shock hazard and noise levels. Since hair dryers are produced in high volume, manufacturing process optimization is important to minimize the component and assembly costs. Products from multiple manufacturers are available, which could support the evaluation of alternative approaches to design and manufacturing.

Recreational: A bicycle could be an example in this class. This product is also very familiar to the general public. Bicycles are manufactured by many companies in volumes ranging from very low (exotic racing bicycles) to high (bicycles for the general public). The complexity and number of components and processes match the program guidelines.

Tools: Here a home workshop drill could provide a suitable example. This product addresses all the areas listed for the household product, but is more advanced in terms of its technology content, using tighter tolerances and a more complex assembly.

3.3 IMPLEMENTATION PLANS

The real-world manufacturing scenarios developed in Phase I of the MSE project will lead to planning and implementation of demonstration systems during Phase II, i.e., in years three through five. This will be undertaken in collaboration with industry in order to ensure that proposed integration solutions support realistic product realization situations. Implementations will make use of the NIST AMSANT facilities in order to perform remote experiments with industry collaborators through the use of advanced networking and communication systems. Integration demonstrations will consist mostly of design to production simulations because of limited access to production resources. The specific topics of industry involvement, the demonstration facility, and simulation systems are discussed further below.

3.3.1 INDUSTRY INVOLVEMENT

Industry involvement is an important part of the MSE project. It is for this reason that actual manufactured products will be chosen as test cases, rather than artificial "benchmark products." One benefit from industry participation may be the provision of additional manufacturing support information for the product, such as a bill of material, engineering specifications, process plans, and assembly plans. However, if the chosen product requires industry-supplied knowledge and information, consideration must be given to any conditions that may be imposed on its use and dissemination. The effects on the MSE project of restricted access to proprietary information will need to be carefully evaluated.

Even with limited industry involvement, it would be possible to select a test-case product based on (but not identical to) an actual industry product. This would require significant effort by MSE project staff in creating and validating the product design and its supporting information, however.

Collaboration between industry, other government agencies and NIST is seen as an important, ongoing part of the MSE implementation plan. It is vital that the project remain aware of the broad spectrum of industry efforts to improve the design and fabrication of electromechanical products. It is also important for the MSE project to be aware of other government agency sponsored programs developing supporting technologies, and addressing similar integration problems, in order to eliminate redundant efforts and to leverage technology results supporting systems integration. Specific external programs will be targeted for SIMA participation, and new candidate products and functional subsystems will be reviewed as appropriate. If new product and process technologies are found to be superior to the solutions developed within the MSE project, existing MSE products and technologies will be replaced. MSE implementation scenarios must be realized in conjunction with existing programs whose plans include pilot demonstrations of real-world problems.

It is anticipated that formal relationships (i.e., Cooperative Research and Development Agreements or CRADAs) will be set up with the companies providing test-case products, and with programs developing supporting technologies. Additional collaboration mechanisms such as the NIST Industry Fellows Program will be used to provide NIST staff with the opportunity to work in an industrial setting. These industry collaborations will provide industry validation for the proposed integration solutions developed under the SIMA program.

3.3.2 DEMONSTRATION SYSTEM AND FACILITY

The MSE project will include continuous and discrete-event simulations in design, planning, and shop-floor production activities as part of its implementation plans to demonstrate systems integration. Demonstrations of actual parts production will be limited to what can be accomplished using existing resources in the NIST workshops, and/or the facilities of an industrial collaborator. Inevitably, this will limit the real-world scope of the integration demonstrations. Heavy emphasis will therefore be placed throughout the program on the provision of simulation capabilities so that virtual demonstrations are possible over a wide range of part domains and engineering activities.

Virtual demonstrations will be facilitated by the advanced computing and communicationcapabilities implemented within the AMSANT. The AMSANT will serve as the primary test bed for project demonstrations, technology evaluations, and external collaborations, throughout the five-year duration of the MSE project. AMSANT will provide the MSE project and its collaborators with the ability to perform integration demonstration between multiple sites.

PART II: MANUFACTURING SOFTWARE APPLICATIONS

Part II is a study of industry needs, current technology and research trends in the integration of product realization systems, with recommendations for consequent directions in the MSE project. Part II is organized as follows:

CHAPTER 4: ENGINEERING AND PRODUCTION APPLICATIONS

Chapter 2 described in broad terms the activities and technologies that fall within the scope of the MSE project effort. This chapter expands on those topics to provide explanations of the activities and supporting technologies that will be used in later chapters of Part II.

As discussed in Chapter 2, the manufacturing process can be divided into three stages: design engineering, manufacturing engineering, and production. The output of the design engineering stage is a detailed specification of the part to be manufactured. This becomes the input to the manufacturing engineering stage, which results in a detailed specification of *how* the part is to be manufactured. This specification in turn becomes the input to the production stage, which determines when and where the part will be made, and then proceeds to manufacture it. Other activities such as sales and marketing, product support and financial management may exchange information with these activities, and may have a significant impact on the decisions made, but they are outside the scope of this report.

Although these stages are described separately, in practice their activities overlap and interact in most manufacturing organizations. It is now generally accepted that the deliberate planning of such interaction—leading to what is known as *concurrent engineering*—is beneficial in improving product quality and reducing the time from design to production. On the other hand, the nature and timing of the concurrent engineering interactions, and also of the engineering and production activities themselves, vary considerably from one organization to another, and even from one "shop" to another within the same organization. An important difference among shops, for example, is whether the manufacturing engineering activities are viewed as closely coupled to the design process, or closely coupled to the production process, or neither. It is a subgoal of the MSE project to model all the interactions that may need to be generally supported by manufacturing software products.

4.1 DESIGN ENGINEERING

As stated in Chapter 2, product design can be broken down into four phases:

- Product planning
- Functional design
- Configuration design
- Detail design

For the purposes of the MSE project, however, product realization is considered to begin at the point where major software tools begin to be employed, namely at the end of the functional design phase, when a functional specification for the product is available and the configuration design phase begins.

The activities actually undertaken in the design process vary considerably according to the natureof the product and the commitment of the company to the use of computer aids. Where families of essentially similar and fairly simple products are concerned, it is sometimes possible to encapsulate the basic design principles in a few equations or design rules. These may then be used to drive the detail design process in such a way that the designer only has to enter values for certain key dimensions or other parameters to enable the design system to generate a complete specification of the product.

A similar situation holds for modular products, built from standard subassemblies either manufactured by the company or bought in from external suppliers. In this case design is largely a matter of bringing together specifications of the appropriate components to meet a customer's requirements; detailed drawings and costs can often be generated in a matter of minutes.

In both the above situations, considerable preliminary work is necessary to develop new software systems or to configure existing ones for the intended specialized applications. In these cases, it has been found that the configuration phase and much of the detail phase of design is so standard that it can be programmed into a system. As a result, little further design work in the traditional sense needs to be done.

On the other hand, the design of a new passenger aircraft can require the individual design from scratch of many thousands of completely new components, and can extend over a period of several years, even with extensive use of computer aids. The overall process involves the extensive use of analysis and simulation in arriving at an optimal design solution meeting all the constraints imposed by conflicting requirements on payload, range, fuel economy, safety, noise generation, price, and operating costs.

No brief overview can do justice to the wide spectrum of possible approaches to design engineering. The remainder of this chapter will assume a "traditional" breakdown of the process into component tasks, as is still common in companies manufacturing a diverse range of non-modular products. Even then, however, it is not easy to draw precise boundaries between the two design phases of interest to the MSE project. It is important to bear in mind, therefore, that the following is an "averaged" account of the process—actual practice in any particular company will almost certainly differ from what is described below.

The brief descriptions of the four design phases given in Chapter 2 will be expanded in the following sections to provide a basis for more detailed discussion of the activities occurring in each phase.

4.1.1 PRODUCT PLANNING

Essentially, product planning clarifies the design task to be addressed. Product planning may be stimulated by the desire to improve upon an existing product, or by the identification of a new market opportunity. The latter may be stimulated in turn by new technological developments, as was the case with the pocket calculator in the 1960s and more recently the mobile telephone.

The questions arising at this stage are of a very broad nature: What is the purpose of the new product? What market sector is it aimed at, and what therefore should it cost? What will be the size of the market, and how many product items should be produced? The output of this phase is a set of constraints on the work of the next phase. In particular, the intended functionality of the product is defined, and limits are imposed on its development and production costs.

4.1.2 FUNCTIONAL DESIGN

Functional design is concerned with how the desired functionality can be achieved in the new product, subject to the constraints imposed at the product planning stage. There may be several solutions to this problem, possibly using different physical principles. An example of a design choice at this level is the decision whether a new aircraft will be powered by jet engines, turboprops, piston engines, or some new and exotic form of propulsion.

Initially, design choices are made at a high level, but each choice leads to a new set of design problems at a lower level, which must be solved in turn. The process is therefore one of successive refinement; at each level, design possibilities are either rejected or followed down to lower levels of problem decomposition. Each new level poses a set of functional problems to which technical solutions must be found by the designers. This results eventually in a set of viable possibilities for achieving the desired functionality while satisfying the design constraints. The possible designs are then evaluated against each other in terms of estimated production cost, estimated performance, or some combination of these and other criteria. An optimal choice of design results from this comparison.

4.1.3 CONFIGURATION DESIGN

Whereas the functional phase of design is concerned with a functional decomposition of the intended new product, the configuration phase maps the functional elements of the design onto (in the case of the MSE project domain) mechanical and electromechanical systems and subsystems providing the required functionality. This phase therefore covers the layout of assemblies and subassemblies. Once again the process is one of decomposition from higher to lower levels, and some iteration between levels may be necessary to obtain acceptable results. As in the previous phase, the result is a set of possibilities from which an optimal choice must be made. At this stage it is possible to make more accurate estimates of cost and performance.

4.1.4 DETAIL DESIGN

In the detail design phase, the chosen configuration design is fully documented. Detailed drawings or product models are created for all the components to be manufactured in-house for the new product, and any standard components to be brought in from outside are specified. Once the detailed part designs are available, it is possible to perform various computer-based analyses to determine whether the desired product functionality will be achieved. If not, a design iteration will be necessary.

As mentioned in Chapter 2, few computer aids are currently in use for product planning and functional design. The scope of the MSE project therefore includes only configuration and detail design, which use numerous computer-based tools. Some of the more important of these are reviewed in the following sections.

4.1.5 CAD SYSTEMS

Historically, the first interactive graphical CAD systems were 2-D drafting systems. These provided a means for generating traditional design drawings more quickly than was possible using manual methods. The major time-saving resulted from the use of automated techniques for generating drafting symbols, for copying other recurring combinations of geometric elements, and for generating assembly drawings from previously created part drawings.

Many smaller industrial companies are still using systems of this kind, which often run on PCs.

The next major development came in the early 1970s with the introduction of the 3-D *wireframe model*. This represents the shape of an object as a set of edges in three dimensions; its primary significance is that it provides a unified model of the object rather than several partial models, as in the case of a traditional engineering drawing with its three orthogonal views. One immediate advantage of the wireframe representation is that the computer can automatically generate drawings of the object from any point of view, using any projection chosen by the viewer. Wireframe systems are extensively used by industry today.

Most currently available wireframe CAD systems also allow the attachment of surfaces to the edge-based model, which enables the use of realistic shaded surface renderings. The geometry available generally includes complex doubly curved surfaces such as NURBS (non-uniform rational B-splines), whose use was pioneered in non-graphical systems developed in the 1960s, mainly in the aircraft industry.

An even more advanced tool for representing product shape is the *solid modeler*, which brings together the advantages of the wireframe and the surface modelers in an optimal way. Like the enhanced wireframe model, the solid model contains information concerning all the faces of the object, including the surfaces on which they lie and the edge curves forming their boundaries. Such systems also create *topological* data, recording the interconnections between the faces and edges in the model. This information is now generated automatically and verified internally by the system, which can also automatically compute the volume, mass, and moments of inertia of the object. Most major CAD systems today have solid modeling capability, although this technology has only recently become widely used in industry.

During the 1970s, it was believed that the existence of a complete computer model of an object's geometry would allow the easy automation of many engineering activities downstream from design such as, for example, process planning. Unfortunately, during the 1980s this proved not to be true. Consequently, further developments in CAD systems continue to be made, with several different but related thrusts now starting to converge. The aim is to generate not merely a solid model (i.e., geometry alone) but a *product model*, containing the additional engineeringsemantics needed for automation and integration.

Some of the major areas of new development in CAD modeling are briefly summarized below. Further details are given in Chapter 7.

Parametric modeling: Here the intention is to create product designs in which the dimensions are not fixed, but can be varied for purposes of modifying the design or generating different members of the same family of products. This capability has existed in a limited form for several years.

Variational or constraint-based modeling: This is related to parametric modeling, but is more powerful. It allows the designer to specify constraints on elements of the design, such as "these two plane surfaces are parallel," or "Circle A is concentric with Circle B." These constraints are usually related to the intended functionality of the product, and once defined they are required to hold when any design modifications are made. The implementation of this capability is giving rise to many technical problems, but several CAD systems currently provide at least limited 2-D constraint modeling.

Feature-based modeling: A *feature* (or more fully a *form feature*) is a local geometric configuration on the surface of a manufactured part that has some engineering significance. Design features are related to the intended functionality of the product; examples include cooling fins, gear teeth, and holes for bearing housings. Other product realization activities may have different feature-based views of the same part. For instance, features for machining processes are simply volumes of material that must be removed, such as holes, pockets, or slots. Research has shown that form feature information provides the "natural" input required for manufacturing and other applications. However, it has proved difficult to generate this information automatically from shape representations used by the kind of purely geometric solid modelers described above. For this reason, many CAD systems are now providing facilities for "design-by-features," though few currently have any means of generating manufacturing feature models automatically from design feature models.

The most significant aspect of CAD system evolution is the increasing potential for interpretation of the model by the computer. The manually produced drawing was intended exclusively for human interpretation, whereas the design systems of the future will generate information that will directly drive automated manufacturing engineering processes. Although good progress has been made, much work remains to be done in this field. Some current research issues will be discussed in Chapter 7.

In addition to the geometry-based graphical CAD systems of the kind discussed above, a variety of other types of systems provide additional support for the design process. Some of these are discussed briefly below.

4.1.6 ENGINEERING ANALYSIS

Engineering analysis tools provide additional support during the design process. They aid designers by calculating information about functional behavior, production cost and other matters related to design optimality. In particular, it is desirable for the designer to verify that the chosen design will meet functional or environmental requirements by simulating its behavior under operational conditions. Off-the-shelf software tools are available for this purpose, providing structural analysis, vibration analysis, thermal analysis, flow analysis, and other still more specialized capabilities. Both static and dynamic (time-dependent) computations may be performed using most of these systems.

Many engineering analysis packages use *finite element* (FE) approximations and use a problem formulation in terms of a large set of linear (or sometimes non-linear) equations describing the physics of the situation to be analyzed.

Knowledge-based analysis is based on a different approach, using off-the-shelf inference engines and expert knowledge bases. Interfaces to the design system convert its internal data into "facts" accessible by the inference engine. These facts are then used, together with design rules and other information in the knowledge base, to deduce important assertions about the characteristics, quality, and functionality of the design.

4.1.7 COMPUTER SUPPORT OF CONFIGURATION AND DETAIL DESIGN ACTIVITIES

The configuration design phase typically encompasses the following activities:

- Design layout
- Assembly structure and component definition
- Materials specification
- Invocation of design rules
- Preliminary cost estimates
- "Make-or-buy" decisions
- Identification of design concerns: safety, maintainability, environmental impact, etc.

The detail design phase is characterized by activities such as:

- Development of detailed drawings and product models
- Engineering analysis: structural, vibrational, thermal, fluid flow, etc.
- Design optimization
- Tolerance specification
- Complete materials specification
- Generation of bill-of-materials
- Production of non-functional prototypes (via stereolithography, etc.)

Configuration design is most naturally a top-down process, starting from an initial idea of theassembly layout of the overall product. This idea is pursued to lower levels of detail, the top-level view being refined as the process proceeds, until the level of individual components is reached. Unfortunately, this way of working is not well supported by existing CAD systems, which are more suited to the design of individual parts followed by the creation of models of assemblies in a bottom-up manner.

Some systems provide limited support for the schematic modeling of layouts, mainly by extending the use of their existing geometry definition capability, but no CAD tools are known that permit effective use of the top-down approach. This problem arises because of the historical origin of CAD systems in the detail rather than the configuration design area, and it provides a major example of poor compatibility between product realization activities and the systems available for their support.

Other aspects of configuration design are better supported. For example, the optimal choice of materials may require use of extensive databases of materials and their properties. Additionally, knowledge-based systems can provide guidance on company-specific design rules and issues such as manufacturability of parts and assemblability of components. Early estimates of the production costs of a new product are often made using rule-of-thumb techniques based on past experience and approximate measures of product complexity. Such methods may be implemented using either a rule-based approach or by implementing add-on computer code using facilities provided by a conventional CAD system.

The effort involved in programming knowledge-based systems for use within a particular design context can be very significant, but the rewards can be great, as in the design automation examples mentioned at the beginning of this chapter. Such systems cannot be regarded as general-purpose design systems, and at present they are used mainly in narrowly focused areas of design.

Since existing CAD products were developed originally to support detail design, the match between activities and systems is better in this area. It is comparatively easy today to create drawings and/or computer models of parts having highly complex geometry. The parametric, constraint-based, and feature-based capabilities currently under development will not only speed up the detail design process, but will also enable easier interfaces with other modules of an integrated system. Once parts are fully defined, they may be used to build assembly models. One current shortcoming of CAD systems relates to engineering tolerance data. Many systems allow this to be associated with the drawing or model in a human-interpretable manner, but the effective automatic use of this type of information has yet to be demonstrated.

It is in the detail design phase that most engineering analysis takes place. As mentioned in the previous section, FE software is widely used for this purpose. Two current problems are that the interface between CAD and FE analysis is only partially automated, and that the results of the analysis are almost exclusively human-interpretable. As a result, setting up the computational model can be a lengthy and tedious task, and there is no automatic feedback of analysis results into the design process. The optimization of designs with respect to functionality and cost isessentially an iterative process requiring repeated analysis and interpretation. Design optimization can therefore be very labor-intensive.

One other widely available form of engineering analysis system provides a means for modeling kinematic assemblies and providing dynamic simulations of their motion.

The design of many products requires special-purpose analysis techniques, and it is often necessary for companies to write the software for this purpose themselves. One very important form of analysis is production cost estimation. It is not usually possible to do this accurately without planning the actual production process in detail, in terms of company-specific resources and costings. This activity therefore belongs primarily to the manufacturing engineering stage of product realization. However, it is possible that a feature-based design system might provide feedback to the designer concerning the cost associated with the manufacture of individual features as they are added to the product model. This would enable a certain level of cost optimization during detail design.

4.2 MANUFACTURING ENGINEERING

Manufacturing engineering includes the following activities:

- Process planning
- Tooling design
- Assembly planning
- Inspection planning
- Tolerance allocation
- Cost estimation
- Control program generation
- Simulation and verification

These activities are briefly discussed in the following sections.

4.2.1 PROCESS PLANNING

Process planning is the specification of a detailed sequence of manufacturing operations for conversion of material in some raw state into a finished part as specified by the design process. Initially, the raw work-piece geometry must be defined. Then, for machined parts, the overall task is often broken down into two subtasks. The first subtask, called *macro-process-planning*, identifies the sequence of operations to be performed and the types of manufacturing resources needed for them. The second subtask, called *micro-process-planning*, specifies the sequence of operations to be performed on each machine type There is no universal agreement as to the precise boundaries between these two phases.

Knowledge of the machines and processes to be used allows automatic calculation of processing time for the part on each machine. Company-specific knowledge about equipment depreciation costs, operating costs and personnel costs then enables the overall manufacturing cost to be calculated accurately. This essential function of a process planning system enables production costs to be kept in check; if they are found to be too high, then some redesign of the part may be necessary.

Other aspects of process planning for machined parts concern the specification of fixtures to hold the part in different set-ups during the machining operations. At the micro-planning level, speeds, feeds, and cutting depths need to be specified for each machining operation. There are complex interactions between all these activities, and the generation of an acceptable plan may require several iterations.

Computer-aided process planning (CAPP) systems are commercially available to provide assistance in some or all of these functions. Most such systems do not provide full automation of this activity; in fact many engineers strongly resist the idea of process planning automation, taking the view that there can be no substitute for human knowledge and experience in this area.

CAPP systems exist in two forms, using methods known as *variant* and *generative*. The variant method is based on the editing (usually manual) of a previously existing process plan for a similar part. This requires some mechanism for retrieving process plans from a database, based on some criterion for part similarity. To this end, various *parts classification* and *coding* schemes have been devised, which attempt to characterize the geometric and other properties of a part in terms of a string of alphanumeric characters. These may carry information, for example, about overall size, basic shape (rotational/prismatic), and various types of manufacturing features occurring on the part. Coding refers to the attachment of this characterization to the part model, and classification to the method used to identify similarities based on the part code. These techniques are widely used in the context of group technology (GT), a means of achieving efficiency by grouping together on the shop floor the machining resources involved in the manufacture of families of similar parts. These techniques also form the basis of case-based reasoning approaches to design and planning currently being researched. (See Chapter 7.)

Variant process planning systems are well suited for use with GT, but are inflexible; they assume that the factory configuration changes slowly or not at all. Nevertheless, most CAPP systems currently in industrial use are of this type.

In principle, generative process planning systems automatically generate process plans from computer models of parts and information drawn from databases of available manufacturing resources. Most generative systems require the input of a part model expressed in terms of manufacturing features. Currently, the feature data may be identified by the user from a drawing and manually input to the system, or the features may be manually identified on an existing CAD part model. At present, very few systems are capable of recognizing manufacturing features automatically from a part model, although this is an active area of research and development (see Chapter 7). The plan is generated from knowledge of the part material, feature dimensions, andother relevant parameters, making best use of the available manufacturing resources.

The content of a generative CAPP system resource database is highly company-specific, and means are provided for populating the database and for making extensions and deletions as resources change. In particular, information must be available concerning machine tools and their capabilities in terms of part sizes handled and accuracy characteristics, and also concerning cutting tools that may be used with them. Planning logic also varies between companies, and it should be possible to modify this as desired. The user is provided with a means for manually overriding decisions made by the system if necessary.

Access to machinability information for specified part materials is also needed, since this has a bearing on the choice of feeds and speeds used in machining. Other manufacturing methods have similar requirements. For example, injection molding and sheet metal stamping require databases of standard mold components and standard bending and side-action accessories, respectively.

Generative planning is more flexible than variant planning. Changes in manufacturing resources are taken into account automatically as soon as the relevant databases have been updated. Additionally, it is possible to generate multiple versions of a plan, so that when a machine in use breaks down, the manufacturing process can branch to an alternate path using another comparable machine.

An activity sometimes associated with manufacturing processing is in-process inspection to ensure that the part being produced meets its design specification. If this capability is to be used, then an inspection plan must be generated concurrently with the process plan. However, inspection is often deferred until manufacture is complete. This matter is further discussed in Section 4.2.4 below.

4.2.2 TOOLING DESIGN

The tooling requirements for any parts manufacturing process give rise to additional design problems. For example, machining often requires that a fixture be designed to hold the part in one or more setups during material removal processing. The fixtures usually are made of standard components, although special-purpose fixturing devices sometimes are necessary. If standard components are used, the automation of fixture design requires a database of available components.

Another example of design problems caused by tooling requirements is in injection molding, which demands the prior manufacture of the injection die. The design of die surfaces is derived from that of the molded part, with allowances made for shrinkage of the part during cooling. Other parts of the die assembly usually are built from standard components, which requires access to a database of available components. The die design may also be optimized by simulating the flow of molten plastic in the die cavity. This allows, for example, the determination of gating layouts to avoid undesirable characteristics in the molded part.

Tooling requirements also affect the design of cast and forged parts, which usually are designed in their final form, taking into account any machining of the part after it is originally formed. The design of dies must be based on the shape of the part before it is machined, however, and the overall design process must therefore provide some means for adding material to the part as initially designed, to allow for machining. As with injection molding, shrinkage occurs as the cast or forged part cools. This must also be considered when deriving the die shape from the shape of the designed part.

A last example of tooling requirements influencing design is sheet metal pressings such as those used for car bodies. These suffer from a phenomenon known as "springback," an elastic deformation that occurs after the part is removed from the press, which results in part geometries that do not conform to the shape of the press. As with castings, forgings, and moldings, the design of the press tool may be derived from that of the desired part. Again, however, the design must also account for the unwanted deformation.

Design requirements arising from tooling considerations such as the four listed above are within the scope of the MSE project. The above examples exhibit the requirement in integrated systems for process-specific databases, CAD system enhancements, analytical tools, and simulation programs.

4.2.3 ASSEMBLY PLANNING

In planning for manufacturing processes other than machining, the generic technology described in Section 4.2.1 is still applicable, though the details of the processes involved differ, and different knowledge bases need to be used. Assembly planning is usually a manual activity at present, though there is the possibility of using the same basic approach for micro-planning as is used for machining. An assembly plan may then be used to drive automated assembly equipment.

4.2.4 INSPECTION PLANNING

Inspection planning at the micro level may follow similar lines to the feature-based micro-process-planning used in machining planning. For post-manufacture inspection, using, for example, a coordinate measuring machine (CMM), the equipment may be programmed to operate automatically in much the same way as a numerically controlled machine tool. The programmer specifies a sequence of points at which the measuring probe should contact the part; sub-sequences of the overall sequence are usually related to individual features of the part in much the same way as machining operations were prescribed for the generation of those features.

At the macro level, inspection planning calls for quite different strategies because the goals of inspection are to gather information rather than to transform the workpiece. Information may be gathered either to feed back to process control (as with statistical process control, or SPC) or to provide feed-forward data (for rework planning, part quality control, etc.). Occasionally, one inspection activity serves both functions. Inspection planning decisions are based on the expectedutility of the information to be gathered. Measurements for process control may be made concurrently with the process (in-process measurement), by interrupting the process to check the part (process-intermittent measurement), or after the part has been removed from the process setup (post-process measurement). Each type of measurement, and each strategy for using the resulting data, makes use of special knowledge of measurement methods. In addition, inspection planning must concern itself with lot sampling and other statistical issues not applicable to machining planning.

4.2.5 TOLERANCE ALLOCATION

In the manufacturing engineering context, tolerance allocation is the reinterpretation of the designer's functional tolerances in manufacturing terms. It involves the distribution of overall required tolerances between the different manufacturing operations to be used. This process contributes towards the determination of the minimum-cost plan that will achieve the specified design tolerances. This topic is discussed further in Section 7.2.1.4.

4.2.6 COST ESTIMATION

To shorten response time and improve accuracy when responding to a bid request, many companies have built automated packages for estimating manufacturing costs. These packages also are used in "make-or-buy" decisions for component parts and assemblies. Off-the-shelf, general-purpose, spreadsheet packages are often used to develop databases, but larger firms typically develop their own packages to reflect their own business rules and cost experience.

Cost estimation packages generally are linked to the process planning system (when there is one), since most of the cost is in the manufacturing process. It is not uncommon, however, for the cost estimation package to be linked to the design system, either directly to the CAD system or to a closely attached feature analysis package. In such cases, the cost estimation system first performs a crude "macro" process identification operation (see

Section), and then does a cost estimate based on those processes.

Cost estimation usually includes analysis of material, time, and resource requirements.

4.2.7 CONTROL PROGRAM GENERATION

More common in industrial use than automated process planning is the generation of control programs for automating manufacturing equipment—particularly machines used in cutting, forming, welding, and materials handling. The control program specifies a sequence of operations, each having a tool/end-effector to be used and a "tool path" to follow. Additional parameters under the guidance of the control program might include the position of the tool, its turning speed, rate of motion along the path, etc. Automated measurement equipment has similar requirements for motion planning, although certain machining parameters (such as cutting tool geometry and rotational speed) are replaced with sensor characteristics and operating parameters.

CAD/CAM systems typically output tool paths for NC machining in a standard format called *cutter location data (CLdata).* The file containing this data is known as a *CLfile,* and it requires further processing to give a control program tailored to the capabilities of a particular target machine. This *post-processing* function may be performed by a separate program, or by the machine tool controller itself. Coordinate measuring machines (CMMs) and vision systems can be programmed in a manner similar to that used for NC motion programming, although with special provisions for communicating tolerance data and inspection results.

4.2.8 SIMULATION AND VERIFICATION

When a manufactured part is especially complex, the process plans and control programs for cutting, shaping, and assembling the part may themselves be quite complex. Typically, automated and even interactive systems will construct these plans and programs as a sequence of smaller operations, each dealing with a particular feature. Verifying that the whole process can be performed successfully with a particular machine—without damaging fixtures or other process equipment—can be difficult.

The long-established solution to this problem is to actually run the program on the manufacturing machine, while an operator observes the process (with one hand on the emergency stop switch) and then inspects the resulting part. This technique requires scheduling both the machine and the machinist for one or more verification runs, however, which takes them away from other work.

A common modern solution is to develop software that simulates machine paths and part (de)formations, given a particular process plan and control programs. In simple cases, changes to the machine and to the part are displayed graphically as the simulation takes place, so that a human planner can identify potential problems while watching the simulation. More complex systems might perform the simulation as part of an automated manufacturability analysis, in which case the software itself identifies problems, thus saving the planner's time.

Such systems may also capture the geometry of the part as realized by the simulated process, compare it to the real part's design geometry, and identify significant differences. Assessing part functionality and quality, however, usually requires the planner or designer (or both) to examine the results. Still, the use of simulation software minimizes the production equipment and personnel required for these purposes, with only one—and sometimes no—verification runs being required on the shop floor.

Simulation systems usually are linked to the operations planning and control-program (NC) generation software. Some vendors provide these simulations with (but unfortunately also in many cases on)[5] the controller, so that the verification can be performed using production interfaces for the control programs. The more advanced process simulation systems are almost always company-developed and often require access to some stored form of design geometry and tolerance information as well.

4.3 PRODUCTION

Production systems deal with the actual manufacture of products, and with the planning, preparation, scheduling, and delivery of material, equipment, and human resources for themanufacturing process.

The production systems of a manufacturing enterprise include:

- Materials requirement planning (MRP I)
- Manufacturing resources planning, lot sizing, and time phasing (MRP II)
- Job routing and scheduling
- Manufacturing control
- Tooling management and preparation

In a larger sense, they also include:

- Materials and inventory management
- Equipment and facilities management

---

[5] Meaning that the controller of the tool itself must be used to perform the simulation, making it unavailable to perform manufacturing tasks during that time.

- Human resource management

Production systems also include other financial and administrative systems such as cost accounting, procurement, warehousing, and shipping and receiving. It is difficult to draw a clear line between the technical production activities and these other activities of the manufacturing enterprise.

Production planning differs significantly among manufacturing shops. "Job shops" or "small batch" manufacturers generally have a fixed set of equipment and a variable workload. For them, the planning process is largely a matter of scheduling the job mix to get optimal throughput from a given facility. For "large batch" manufacturers, on the other hand, it is cost-effective to engineer the production facility to make a particular product mix. For these enterprises, the planning process involves choosing the right combination of factory organization and job mix.

In the production area, the MSE project will not be concerned with large-batch problems: namely, facility engineering, job mix selection, and "line balancing." The discussion of production processes presented here is therefore oriented toward the job shop.

4.3.1 MATERIALS REQUIREMENT PLANNING (MRP I)

Materials Requirement Planning systems take into account existing and predicted orders for products, along with existing and planned manufacturing capacities and inventories, to produce an optimal set of "orders" or "production quotas" that define an overall plan for the output of the manufacturing facility for periods of several months. The plan includes internal production of components, and possibly special tooling, needed for the final products. The plan takes into account lead times for acquiring stock materials, tooling, fixtures, and component parts from external suppliers; produces a schedule of the required procurements; and creates an overall production schedule consistent with these acquisitions.

In large production facilities, a major component of MRP I is "capacity planning" or "capabilityplanning"—identification of the types and quantities of machine resources and human resources that will be needed to produce a postulated job mix with some statistical variation. This type of production planning activity is considered to be outside the scope of the MSE project.

4.3.2 MANUFACTURING RESOURCE PLANNING (MRP II)

Manufacturing Resource Planning systems regroup orders and production quotas into optimal lot sizes, taking into account tooling and setup requirements. The system monitors inventories, equipment, and human resources, and considers requirements and lead times for internal tool and fixture building. From this information, it determines what lots can be manufactured, and when, in terms of available resources. It then creates a list of production jobs, each of which is eligible for on-the-floor scheduling and initiation at some future time.

The MRP-produced production plan identifies jobs to be performed—tools and fixtures to be built, quantities of products to be manufactured, etc.—over some period of time. Specific jobs are said to be "released" to the shop when the requisite materials, tools, and fixtures are available.

4.3.3 RESOURCE ALLOCATION AND SCHEDULING

The complete process plan for a part specifies both a macro-plan or "routing sheet"—a sequence of processes to be performed by different types of machines—and one or more micro-plans or "operations sheets"—sequences of operations making up the processes to be performed on one machine. *Scheduling* is the process of assigning the processes (or "job steps") on the routing sheet for a given job to specific machines at specific times. This can be done in two ways:

- *Predictively* ("job push"), by identifying the expected availability of each machine over some period of time (usually a day or a week) and placing specific job steps for specific jobs in the available time slots, while maintaining the required sequence of processes from the process plans
- *Reactively* ("demand pull"), by finding the next useful job step for a machine to perform when the machine completes a previous task and becomes available

Predictive scheduling is usually done for a collection of "released" jobs (those for which machines, tooling, and materials are available) all at one time. It can take three possible approaches:

- The *in vacuo* approach, where the plan is optimized in terms of all manufacturing resources known to the system (as if the facility were initially idle)
- The *status quo* approach, where only currently available resources are considered, the remainder being already employed on other tasks
- The *globally optimal* approach, where existing jobs are rescheduled if necessary to achieve the best overall use of resources

In each case, estimates of the time required to perform the operations on the corresponding operations sheet and for the lot to travel between machines are used to produce a feasible schedule. In some systems, particularly those in which the handling time is a significant fraction of the processing time, delivery of workpieces, tools, and materials is planned and scheduled just like the processing operations. In others, the delivery systems are assumed to operate reactively and to have infinite capacity, so that planning considers only the standard estimate for traveling time.

Reactive schedulers, on the other hand, are informed directly of events on the shop floor: task completions, stops, equipment failures, and recoveries. When a machine becomes available, the scheduler compares activity in the facility to established goals and the current released job list, and determines the "best" process (job step) for the available machine to perform. It then schedules the machine to perform that operation. In such systems, it is very difficult to plan materials deliveries in advance (unless all jobs require a common collection of materials), which means that materials handling also is reactive. But even in facilities where the processing is scheduled predictively, reactive scheduling often is used for material handling systems.

Either type of system may make use of mathematical methods (such as linear programming), statistical methods, discrete-event simulations, or heuristic algorithms in choosing the best schedule. In material handling systems, heuristic scheduling rules are commonly called "dispatch algorithms."

4.3.4 Shop-Floor Monitoring and Control

A shop-floor monitoring system tracks the true state of the facility—the state of each machine, the identity of its operator(s), the job and step/process being performed on the machine, and the state of that process. This requires communication with controllers, machine operators, and supervisors.

The shop-floor monitoring system is the link between scheduling systems and control systems. In a predictive scheduling system, this interaction is usually limited to keeping shop supervisors informed of any variations between the schedule and the actual state of operations on the floor. In theory, this information could be fed back to the scheduler, which could then modify or improve the schedule. But only very experimental systems are able to do anything like this. By comparison, shop-floor monitoring in some form is an integral part of a reactive scheduling system.

Identifying the state of a manufacturing station, and the job and step it is performing, usually requires either a manually updated information base or direct communication with the controller software. Any of several standard protocols may be used to communicate with equipment controllers. There is no requirement for the equipment subsystem to be totally automated, but it does have to be able to communicate.

A *totally automated system* is a manufacturing system controlling one or more physical processes,receiving control information and reporting its status through direct communication with other systems, and requiring no human assistance in performing the physical process except in unusual circumstances.

Partially automated systems do some or all of the above, but require human interaction as a normal part of the process.

An *adaptive control system* is a control system that monitors the process, the product, and the environment and alters the process control parameters so as to maintain product quality. The control system may be totally or partially automated. A distinction is often made between *adaptive controllers*, which actually control and modify process parameters, and *reactive schedulers*, which only control the timing and distribution of tasks.

The function of *job tracking* is to log the location and state changes of jobs, workpieces, tools etc. on the shop-floor. It is sometimes considered to be a part of shop-floor monitoring, though here the primary focus is on machine activity, while in tracking systems it is on the location of objects. Some control systems (or their operators) can report the job they are working on, with the obvious implication that the materials for that job must be at that station. But many tracking systems use independent devices such as bar-code readers to identify the objects themselves as they move through particular control points, including not only processing stations, but also material handling and storage systems.

4.3.5 Production Simulation

Production simulation (as distinct from process simulation—see Section 4.2.8) is a means of analyzing the behavior of a whole shop or facility in response to a production scenario, with a statistically predictable set of related events and perturbations. The nature and degree of simulation differs significantly from case to case, depending on the intent of the simulation.

The most common production simulations are used to generate or validate production plans or schedules. In these simulations, the primary measurement is the predicted throughput of the facility and its correlation to planned throughput. Other beneficial outputs may include estimates of personnel and lead-time requirements, as well as machine use and maintenance requirements. In the MSE project, such simulations are considered to be part of the MRP system or the scheduling system, as appropriate.

Production simulation also can be used for an entirely different purpose, namely to validate the software that runs (a portion of) an automated facility—automated control, dynamic scheduling, and dispatch software—in a virtual production environment. In these cases, all that is simulated is the actual manufacturing operation of the equipment, the occurrence of unusual physical events, and possibly the lapse of time. Such systems use all of the facility's scheduling, control, and dispatch software, so that the simulated production run actually exercises all the computer code to be used, performs all the communications, and responds to all real and simulated events. Simulated production runs thus enable systems developers to identify and correct software errors,bottlenecks, and other anomalies that will arise during real production runs, without consuming time and resources in the production facility itself.

4.3.6 RESOURCE MANAGEMENT

Resource management includes management of tools, materials, equipment, and personnel.

Tool management systems track inventory and orders for tooling components and other consumable materials, and track tool and fixture building orders. They also bind tools to job assignments, and track the location, usage, and wear of tools on the shop floor. The latter is primarily a job accounting problem, but it also affects productivity, cost, and response time when dealing with expensive tooling and fixtures.

Equipment management deals with the acquisition, installation, and maintenance of manufacturing machines and other major pieces of equipment. Acquisition and installation of equipment is a facility management concern outside the scope of the MSE project. Equipment maintenance, however, may be a concern, to the extent that it represents both scheduled and unscheduled activity that affects shop-floor activity, scheduling, and even manufacturing engineering (some workpieces require processing that exceeds the tour-of-duty of certain machines).

4.4 TOLERANCES AND THE CONTROL OF PRODUCT QUALITY

Product quality is an important factor in all stages of the design/manufacturing cycle. As mentioned in Section 2.6, quality has many aspects including, for example, aesthetic appearance and reliability in use. The achievement of product quality is one of the primary aims during the design phase, but quality-related decisions taken there also have a significant impact on the manufacturing engineering and production phases. This may be illustrated by considering the effect of the functional tolerances specified during detail design. These are intended to ensure that the parts of the product can be assembled and that the resulting assembly provides the desired functionality.

During manufacturing engineering, the designer's functional tolerances must first be reinterpreted in the context of the manufacturing process to be used, and the available dimensional freedom allocated between the different operations making up the overall process. The resulting accuracy requirements must then be matched with the process capabilities of available manufacturing resources. For machined parts, the selection of setups and fixtures is related to tolerance datum specifications, and these therefore have an influence on the sequencing of operations. Surface finish and other tolerance information also affect the choice of feeds and speeds for machining. Thus, quality requirements, expressed in terms of tolerance specifications, have important implications regarding the choice and control of manufacturing processes and parameters. Similar influences arise whatever the choice of manufacturing method.

During the production phase, inspection processes are used to check whether manufacturing tolerances are being achieved. If they are not, then either the processes or the process parameters will need to be changed to rectify the situation.

In a fully integrated system, tolerance information should be generated at the design stage and used automatically throughout the manufacturing process. It is unlikely that the MSE project can meet this ideal, however, since it would require the use of computer-intelligible tolerance information, which is not currently available in commercial systems. This is an area of current research interest, and there may be significant developments in the automatic treatment of tolerance information during the lifetime of the project. In the meantime, since tolerance data are necessary for planning and inspection, the project will have to handle them in some non-ideal manner.

CHAPTER 5: COMMERCIAL SOFTWARE FOR MANUFACTURING APPLICATIONS

This chapter analyzes the commercial software products currently used in the mechanical parts manufacturing industry for design and manufacturing engineering. It also identifies technical "voids" and recommends criteria for selecting and installing commercial technologies for MSE project demonstration. The methods used for this analysis were a computer product literature search, telephone interviews with vendors, and analysis of market research reports.

One purpose of the MSE project is to define the activities supported by a given set of software applications and model those activities as a reference architecture which defines the scope of applications addressed by the MSE project. We therefore analyzed commercial software products which support activities in the three main topic areas defined in chapter 2. A follow-on report which models the activities in the three main topic areas as a reference architecture, along with a mapping of software applications to the activities defined in the reference architecture is planned for FY95. The commercial software reviewed in the three topic areas include the following:

- DESIGN ENGINEERING
  - Computer-aided design (CAD) systems
  - Product data management systems
- MANUFACTURING ENGINEERING
  - Computer-aided process planning (CAPP) systems
- PRODUCTION
  - Production scheduling systems
  - Production simulation systems

Due to limited resources for the study, other important product areas were not included in the analysis. These included applications for engineering analysis, materials requirement planning, statistical process or quality control, cost estimation, and equipment control and sensory systems.

The characterization includes general system functions, special functions provided by some products, and available integration mechanisms. Market information includes price range, market size, and market size trend. The survey of technological barriers identifies key issues faced by integrators in getting software products in the area to interoperate with other systems.

5.1 COMPUTER-AIDED DESIGN (CAD) SYSTEMS

CAD systems provide a means of developing, recording, and managing drawings or other forms of product models. They are commonly used in four major application areas: mechanical design, electronic design, cartography, and architectural design. Mechanical design applications havebeen the primary market for CAD products, with 49 percent of the total usage, according to a Dataquest survey [3]. Mechanical CAD systems cover shape modeling, materials specification, documentation of part functions, and assembly layout.

5.1.1 GENERAL CAD SYSTEM FUNCTIONS

Mechanical CAD systems use computers to aid in generating product models of mechanical parts—including specifications such as materials, features, tolerances, surface conditions, etc.—in an electronic format. General CAD capabilities are:

- Model creation, editing, and viewing
- Component and subsystem layout
- 2-D drafting (includes setting dimensions and tolerances)
- 3-D geometric modeling (solids, wireframes, boundary representations)
- Annotations—tolerances, surface finish, materials, etc.
- Design documentation
- Assembly modeling
- Surface blending (for creating rounded edges and corners)
- Surface fitting (for reverse engineering)
- Journalization, and version and revision control

As discussed in Section 4.1.5, some CAD systems support special design capabilities:

- Parametric design
- Variational design
- Feature libraries and feature-based design
- Representation of design knowledge and decision rules
- Group technology coding

Many CAD systems are advertised as CAD/CAM (computer-aided design and manufacturing) systems, because they also directly support certain manufacturing engineering operations closely related to part geometry:

- Tool path generation for numerically controlled machines (2- to 5-axes)

- Tool-surface or fixture collision checking and avoidance
- Path preparation for use by coordinate measuring machines (CMMs)
- Generation of stereolithography (STL) files for rapid prototyping systems

In addition, some CAD systems provide mechanisms for linking to engineering analysis packages (e.g., finite element, tolerance, dynamic, and kinematic analyses).

5.1.2 CAD Software Market

More than 1,000 CAD products are available on the market, sold by more than 200 vendors. Theprice of the software ranges from under $1,000 to over $75,000. Systems at the lower end of that range have more limited capability and generally run on personal computers. Higher-end systems have integrated modules for design, analysis, and NC machining capability, and run on state-of-the-art computer workstations.

Major users of mechanical CAD systems are the aerospace, automotive, capital machinery, electronic and consumer goods, and tool and die industries. According to Frost and Sullivan,Inc. [2], the market for CAD/CAM products has been growing steadily due to: (1) competitive pressure on mechanical product manufacturers, (2) more functions being added to the software, (3) CAD system hardware becoming cheaper and more powerful, and (4) CAD systems becoming more flexible and open to users.

CAD systems provide the product design data that drive downstream activities such as process planning and production planning. The CAD market is currently much larger than markets for other manufacturing software products, however.

5.1.3 Future Challenges to CAD Software Technology

The conventional CAD system is an information capturing system that depends on the human design engineer to provide all the intelligence in the design process and to acquire independently most of the other information that affects design decisions. Future CAD systems will provide linkage to massive information repositories useful to the design process, such as parts catalogs, materials databases, etc. They also will either incorporate or provide linkage to more and more sophisticated automated engineering analysis systems. In addition, they will incorporate or provide linkage to sophisticated—and in many cases industry-specific—design advisory systems. It is also likely that a future CAD system may be linked to an engineering workflow management system, to assist the designer in managing the administrative aspects of the design process.

Such links are not possible today, because:

- The few online parts catalogs, materials databases, and workflow managers that exist require the use of system-specific interfaces. Thus, a CAD system written to work with one such system usually would not work with another.
- Many of the interfaces to catalogs, databases, advisories, etc., are intended for humans, rather than for other programs. Having the CAD system emulate a human operator when linking to such systems is extremely difficult.
- Different engineering analysis systems require different forms of product description, often containing different information. Currently, some of the necessary information is captured by CAD systems only in human-interpretable form. Computer-interpretable formats must be developed for the representation of such data, together with automated methods for the generation of appropriate analysis models.

Standards for the exchange of product descriptions are emerging from the ISO STandard for the Exchange of Product Data (STEP) effort (see Chapter 8). To what degree they will simplify the interfaces to engineering analysis systems is not yet clear.

Similarly, it is expected that Electronic Data Interchange (EDI) standards may assist in the access to parts catalogs and perhaps other databases. But standards for these applications have not yet emerged.

5.2 Product Data Management Systems

Product data management (PDM) systems have emerged as separate applications distinct from CAD software. They are referred to by some vendors as design data management systems. These software products have recently become available from third-party vendors who themselves do not market CAD system products. Such systems are in essence databases providing facilities for the management of product-related information.

5.2.1 General Product Data Management Functions

Product data management systems are designed to manage all product-related information for design and manufacturing engineering functions. The services they provide include:

- Product data capture, maintenance, and retrieval

- Support for approval processes and release procedures
- Configuration management, version control, and change management
- Access control and security
- Interfaces to systems that are product information providers or consumers

Information created or maintained by such systems includes configuration management data, development/revision history, part specifications, CAD drawings and models, finite element models, engineering analysis results, process plans, NC machine programs, etc. These systems are rapidly evolving into general data management systems handling a wide range of production-related information additionally.

5.2.2 PRODUCT DATA MANAGEMENT SOFTWARE MARKET

Since this is a new field, most PDM products are lumped together with CAD systems or with general-purpose database systems in market surveys. At the time of writing, there were around 15 independent vendors of systems specifically described as product data management (PDM) systems, but many more CAD systems which were being "extended" to provide the PDM features. This is clearly a rapidly expanding market.

5.2.3 FUTURE CHALLENGES TO PRODUCT DATA MANAGEMENT SYSTEMS

Two major difficulties limit the use of PDM systems:

- The absence of any standard for the interface to the PDM system for automatic insertion or retrieval of product information sets. While most PDM systems will accept IGES or STEP files as the form in which the data is transferred, the transfer must be human-controlled.
- Differing business practices for review, signoff, and change management require the customer organization to specify these procedures. There is no standard language in which such procedures can be documented, and there is disagreement on what those procedures might be, so that a given PDM product may not be able to support the practices of a given organization.

5.3 MANUFACTURING EXECUTION SYSTEMS

Together, process planning and production scheduling systems make up a category of software known as manufacturing execution systems [4]. This category also includes software for shop-floor monitoring, control, and quality management.

A change in the fundamental philosophy of manufacturing from "push" to "pull" systems (see Section 4.3.3) and from "just-in-case" to "just-in-time" techniques has led to the emergence of process planning and production scheduling systems. Until recently, most manufacturers still created their process and production plans on paper and distributed those plans on the shop floor. Manufacturing execution software provides manufacturers with tools to bridge the gap between CAD/CAM systems and shop-floor production.

Recent surveys have shown that use of manufacturing execution systems in the aerospace/defense field is decreasing due to a reduced demand for weapon systems, but worldwide use in all other manufacturing sectors (automotive, consumer electronics, capital equipment) is increasing rapidly. This is due to the perceived ability of these systems to improve productivity and quality whilst reducing production time. In particular, intense worldwide competition among car manufacturers, has led them to upgrade and modernize their manufacturing systems in an attempt to boost their market share.

5.3.1 COMPUTER-AIDED PROCESS PLANNING (CAPP) SYSTEMS

CAPP systems are used primarily for preparing instructions for producing machined parts, based on design specifications. CAPP systems are important for three reasons: (1) They ensure conformance of the process plan to an established process-plan development procedure, thus improving the probability of first-time success; (2) They increase productivity in generating process plans; and (3) They can formally analyze producibility of a design and identify problems with specific design features, thus assisting in concurrent engineering of a part. The CAPP software market is relatively small compared to the CAD/CAM market, but the metal parts industry is becoming increasingly aware of the importance of CAPP systems.

5.3.1.1 GENERAL CAPP SYSTEM FUNCTIONS

Process planning systems use product design and manufacturing resource data to generate instructions for transforming raw material into a desired product. This includes selecting tools and machines, choosing stock materials, configuring fixtures, determining processing parameters, and defining operations and sequences. Generic process planning technology can be used for machining planning, material forming processes, assembly planning, and inspection planning.

Most CAPP systems are able to:

- Create, modify, and view process plans
- Retrieve data from databases of manufacturing resources (e.g., machines, cutters, machining parameters)
- Identify raw material requirements
- Classify, code, and retrieve data for group technology
- Automatically calculate processing time (time standards)

CAPP systems also include the following data and knowledge resources:

- Process capability data
- Standard sets of data on machining operations
- Built-in understanding of specific products (mechanical, chemical, electronics, pharmaceutical)

5.3.1.2 CAPP SOFTWARE MARKET

Currently, CAPP systems are highly customized to fit specific needs. Their sales totaled about $6 million in 1993, with some 15 vendors providing more than 20 products to the market.

CAPP software ranges in price from $4,000 to $100,000 per copy. Products at the high end of that price range are highly sophisticated and customized, and are usually provided with a database of manufacturing resources and capabilities.

Many companies and universities in the United States and elsewhere are currently developing CAPP systems.

5.3.1.3 FUTURE CHALLENGES TO CAPP SOFTWARE TECHNOLOGY

Integration with CAD systems is vital for CAPP systems if they are to thrive and provide benefits to manufacturers. Technological barriers to this integration exist in three areas:

- Because CAPP is an evolving technology, manufacturers need up-to-date information on its capabilities and benefits, as well as on how to select, install, and use a system effectively. Industry awareness of CAPP is a key issue for vendors.
- A set of standard product definitions is needed so that CAPP and CAD systems canexchange product information.
- A framework for establishing mechanisms for interoperability of CAD and CAPP systems needs to be developed.

In addition, there are no standard forms for the resource information bases needed. Either the CAPP vendor or the user must build the local resource "catalog," which significantly delays the installation of the software in many organizations.

5.3.2 PRODUCTION SCHEDULING SYSTEMS

Production scheduling has three aspects: project scheduling, job shop scheduling, and assembly line balancing. Project scheduling typically relates to one-time-only jobs. Job shop scheduling involves scheduling a variety of jobs performed by a set of machines in a process flow with the highest possible throughput. The technique of assembly line balancing aims for maximum use of assembly lines (of multiple workstations), which requires real-time scheduling to handle machine down time. Production scheduling systems help shop-floor planners determine when to start each job and when it will finish at each workstation.

5.3.2.1 GENERAL PRODUCTION SCHEDULING SYSTEM FUNCTIONS

Production scheduling software performs the following functions:

- Manufacturing capacity planning
- Finite capacity production scheduling
- Forward, backward, manual, finite, infinite, and/or network scheduling capabilities
- Real-time rescheduling (due to unexpected machine down time, changes in product demand, or operational descriptions)
- Scheduling based on make-to-order, assemble-to-order, and engineer-to-order requirements
- Determination of parts routing
- Inventory location and lot control
- Rough-cut capability planning
- What-if analysis of hypothetical production situations

Other functions include shop-floor dispatching; work order tracking; documentation, editing, and reporting of schedules; and production simulation.

Production scheduling systems may be stand-alone products, not directly linked to any other manufacturing software systems or information bases. However, they are sometimes part of larger production management systems, for example MRP II systems, though these in turn are usually of a stand-alone nature.

5.3.2.2 PRODUCTION SCHEDULING SOFTWARE MARKET

Approximately 60 vendors are selling production scheduling software at the present time. A large number offer their products for minicomputers and UNIX-based workstations. The price of the software ranges from $3,000 to $90,000 per copy. When hardware such as sensors and control devices is included, some systems cost as much as $500,000. This kind of system is used to develop high-level production plans and master production schedules that can be monitored at any business or product category level. The high-end systems also support make-to-stock, make-to-order, or assemble-to-order operations.

Production scheduling software packages are mostly customized products, with no single vendor controlling more than 19 percent of the market. As customers have become more educated and demanding of sophisticated capabilities in the software, the market has become highly competitive.

5.3.2.3 FUTURE CHALLENGES TO PRODUCTION SCHEDULING SOFTWARE TECHNOLOGY

Specific technological barriers exist in three primary areas:

- There are no standard process plan formats that allow production scheduling software to take input from CAPP systems.
- There are no standard resource information bases, or even standard specifications for resource requirements derived from CAD and CAPP systems.
- There are no mechanisms for linking production scheduling systems to the actual events on the shop floor. In general, rescheduling based on unanticipated changes in the shop-floor situation is not provided by any scheduling product.

5.4 PRODUCTION SIMULATION SYSTEMS

Simulation systems are useful for studying the dynamics of a real-world system to learn about its behavior and, more importantly, its performance. These simulations are used to:

- Plan new manufacturing projects and major renovation
- Optimize operations and resources
- Help managers evaluate how decisions affect overall production
- Improve information flow and sharing among different facets of the operation.

A 1992 survey by *Industrial Engineering Magazine* [5] showed that 88 % of the industrial engineers who responded to a questionnaire recognized the importance of production simulation software. The software, which displays factory operations graphically, is used in a variety of ways for plant layout, fine-tuning existing systems, and justifying the procurement of new equipment.

5.4.1 GENERAL PRODUCTION SIMULATION SYSTEM FUNCTIONS

There are two types of inputs to simulation software: 1) simulation languages, which allow users to develop their own simulation programs, and 2) user interfaces that allow users to develop a simulation by choosing from a menu. A simulation software tool usually provides the following capabilities: data fitting, model building, animation, and statistical analysis. Outputs usually take the form of graphical displays and printed reports of the results from each simulation run.

Production simulation, process planning, and production scheduling are interrelated. Not only do simulations use process plans and production schedules as inputs, but the results from a simulation usually lead to changes in process plans and production schedules.

Simulation software has the following general features [6,7]:

- Material flow simulation
  - Factory-floor event scheduling
  - Process animation
- Equipment simulation (cranes, robots, machine tools, conveyors, transporters, and automated guided vehicle systems)
- Operator simulation
- Graphical capabilities
  - Visual interactive simulation models of manufac turing operations, with 2-D or 3-D displays
  - Multiple window displays
- Analysis capabilities
  - Modification and optimization of simulation models
  - Equipment utilization analysis
  - Repair and rework analysis
  - Raw material consumption and throughput analysis
- Report generation
- CAD interface for layout drawings

Some software packages also include special functions such as:

- Short-term adjustments to production plan
- Dynamic handling of changing shop-floor conditions, including the ability to handle unpredictable events such as machine breakdown and disruptions to work order schedules
- Optimizing facility capability while creating a master production schedule

5.4.2 PRODUCTION SIMULATION SOFTWARE MARKET

In a 1993 survey of the simulation software market [8], there were about 45 different software systems offered by about 20 different vendors. Their prices ranged from $1,000 to $90,000 per copy.

5.4.3 FUTURE CHALLENGES TO PRODUCTION SIMULATION SOFTWARE TECHNOLOGY

To meet future challenges, simulation software needs to provide better model-building tools, more automated output analysis tools, and libraries of reusable models. Applying virtual reality technology to production simulation will make more powerful simulators for training factory operators, supervisors, and managers.

Specific technological barriers exist in the following areas [9]:

- A general lack of understanding by manufacturers of simulation software technology and how to apply it
- The software's limited ability to model materials handling features
- Low quality and slow speed of animation and graphical displays
- A lack of standard definitions for resources (e.g., machine tools, tools, fixtures, robots, materials, process parameters) that govern the physical laws of manufacturing

Some of these issues are discussed in more detail in Chapter 7.

5.5 RECOMMENDATIONS

Based on the analysis of commercial software systems in this chapter, we recommend the following actions for the MSE project:

1) Consider the following characteristics when selecting commercial software systems to install:

Market penetration: Wide industrial use of a system indicates that it is capable of practical use in a manufacturing environment, However, it may not represent the latest state of the art.

Functionality: Systems providing state-of-the-art functionality may make wider use of computer-interpretable information and require less human intervention, thus providing greater potential for integration. If such a system is newly introduced it may not have made much market penetration.

Openness: Systems with open semantics and open architectures, or systems that allow direct interfacing of user applications, are compatible with integration.

Standards conformance: It is desirable that systems acquired for the MSE project should support data exchange standards such as IGES or STEP.

Price/performance/quality: All other things being equal, the MSE project should acquire commercial systems that have high customer satisfaction and competitive pricing.

2) Encourage participation in MSE activities by software vendors.

Software vendors should be invited to participate in developing and testing integration mechanisms, and in particular to modify or extend their existing systems to meet the integration guidelines developed by the MSE project.

CHAPTER 6: INTEGRATED SYSTEMS DEVELOPMENT BY U.S. MANUFACTURING INDUSTRY

The principal objective of this part of the background study was to determine those areas of relevance to the MSE project in which U.S. manufacturing firms have recently undertaken in-house software development. This was seen as a metric for the potential significance of MSE integration efforts in those areas, and also as a means for identifying barriers to system integration as perceived by industry.

More than 80 industrial projects that fell within the MSE project scope were examined.[6] The background study team studied, in particular, the motivations for industrial software development and the lessons learned regarding successful and unsuccessful methodologies, use of standards, and areas in which new standards may be useful.

The study was effectively limited to larger companies, since smaller organizations generally do not have the resources to develop manufacturing applications software. Small and medium-sized companies generally use off-the-shelf products to provide links between software modules. This sometimes restricts them to the use of compatible software products supplied by a single vendor and communicating via proprietary interfaces. Alternatively, they may be able to link system modules from different suppliers by using sequential file transfer formats such as IGES (Initial Graphics Exchange Specification; see Section), using commercially available translators, without having to write any integration software themselves.

6.1 MANUFACTURING SOFTWARE DEVELOPED BY U.S. INDUSTRY

The industrial software development projects examined have been grouped into eleven categories, each discussed in one of the following subsections. The ordering of topics roughly follows that used in the two previous chapters. The parallel cannot be exact, since this chapter is concerned not with individual engineering activates and the systems supporting them, but with composite systems supporting more than one activity. The most frequently found areas of industrial integration were process planning, control programs and engineering analysis.

6.1.1 DESIGN NORMALIZATION

The quality of a CAD model depends not only on the capabilities of the system used to construct it, but also to some extent on the way the system is used. Geometric approximations used within CAD systems, or in data transfer of complex curve and surface geometry between CAD systems, may lead to stored product descriptions having small geometric discontinuities and other anomalies in the representation of physical parts. Similar discrepancies sometimes result from poor user methodology by CAD system users, perhaps in external client or subcontractor companies not under the direct control of the organization wishing to use the data.

In order to use such flawed models in automated manufacturability analysis, process planning, engineering analysis, control program generation, or direct manufacturing systems (e.g., stereolithography), it is often necessary to "clean up" or "normalize" the design geometry by removing the anomalies, which may affect surface junctions, curve junctions, normals to curves, location of features, etc. Several companies have developed software to perform these normalizations as needed for their particular manufacturing processes.

Some normalization systems clean up the design models stored in the CAD system itself. Alternatively, normalization processes may be built into the translators used to convert design data into other forms used by analysis, planning, NC generation, or production control systems.

6.1.2 ENGINEERING ANALYSIS

With the aim of generating optimal designs, many companies increase automation of the design process by linking their CAD systems to various types of design analysis software. As mentioned in Chapter 4, off-the-shelf finite element (FE) packages are routinely used for structural, vibration, thermal and fluid flow analyses. Many such packages (and also numerous CAD systems) provide tools to assist the user in the necessary conversion of CAD models to FE models for these purposes. But for other applications, and even some specialized applications in the areas listed above, companies often develop their own product-specific analysis software. This embodies specialized mathematical models, and may obtain necessary parameter values and boundary condi-

---

[6] See Chapter 2 for a discussion of the MSE project scope.

tions automatically from the design data through implementation of a direct interface to the CAD system.

6.1.3 Cost Estimation

Companies differ widely in how they perform the cost estimation function, and the survey revealed several cases where in-house software had been developed for this purpose. Costs may be estimated at various stages in the product realization process, though in general the earlier the stage the less accurate the result. Crude estimates are nevertheless useful during the design phase. One approach is based on the use of off-the-shelf spreadsheet packages, which may be programmed to reflect a particular organization's cost estimation formulae and interfaced as appropriate to a CAD system. Accurate estimations require detailed process planning information, however, and some companies have developed specialized process planning systems (see below) primarily as tools for bid preparation.

6.1.4 Process Planning

In large manufacturing firms, computer-aided process planning (CAPP) is an important thrust. Several of the projects examined were essentially company-built process- or inspection-planning systems.

Even when off-the-shelf CAPP systems are used, companies often find it necessary to develop supporting software for the following purposes:

- identification of the features of the design which will affect process selection and the corresponding company-specific GT codes
- organization of, and provision for access to, product and plan databases
- storage and maintenance of information on the company's manufacturing resources: machines, tooling, etc.
- interfacing the company's CAD system to the process planning system
- conversion of the output of the CAPP system for presentation to company production engineers and cost estimators (human, automated, or both)

6.1.5 Control Programs

Most control program generation is adequately supported by off-the-shelf software, because standard control program representations have been available for 20 years and are now commonly implemented in most CAD/CAM systems and machine tools. Certain factors, however, encourage companies to develop their own specialized software for control program generation:

- Some manufacturing equipment has no standard control language. This includes almost all robotic manipulators and most "direct manufacturing" systems
- "Generic" control programs may need to be modified to use specific features of a given machine-type. Common examples of modifications are adding new operations and parameters, or dynamic derivation of control parameters from in-process measurements and computations
- In some commercial systems, algorithms for the generation of tool paths on complex curvilinear surfaces (e.g. for molds and dies) still have some limitations

6.1.6 Control Program Databases

The problem of getting the control programs from the generating systems to the equipment controllers almost requires company-specific software solutions.

Several companies have developed software systems to create a database of control programs. This database is stored in a standard, neutral format, or in machine-specific form with appropriate additional attributes for identification, version control, etc. Controllers that have "third-party access" capability (either built into commercial software or added as a company-specific modification) can then be directed to obtain control programs dynamically from the repository, using standardized communications protocols.

Controllers that are capable of converting the neutral format to their internal format "on the fly"can be directed to do this as part of the access. For controllers that lack this capability, some repository systems can invoke their own conversion routine and deliver the converted code on demand. Other systems simply require that the machine-specific control program be sent from the repository where it had been stored at some previous time.

The objective of such systems is to minimize management and tracking of control programs and their various encodings, and to avoid unnecessary reconstructions. This saves planning time and, in some cases, production time. Such systems also allow control codes to be reused, thus improving reliability and reducing validation requirements.

6.1.7 Process Simulation

Some companies have built their own process plan simulation and validation software, using general-purpose simulation and visualization systems.

Companies who use commercial manufacturing simulation systems still find it necessary in many cases to convert the outputs of the CAPP system to the input forms required for the simulation. The more advanced process simulation systems often require access to some stored form of design geometry and tolerance information as well. IGES files output from the CAD system are commonly used, but they are not always adequate for the purpose.

6.1.8 MANUFACTURING RESOURCE PLANNING (MRP II)

Companies generally find it necessary to build software to create inputs for MRP packages from other information sources. First there is the need to obtain data on existing and projected inventory levels from the inventory and procurement systems, and to provide them to the MRP system in the desired form. Then there is a need to obtain summary information from process plans for products in the job list, including resource requirements, stock materials, fixtures, and tooling lists.

There may also be problems with the outputs of MRP software regarding acquisition requirements and schedules, which need to be communicated to the company's procurement and inventory management systems. Similarly, tool and fixture building requirements and schedules have to be communicated to the corresponding tool cribs and tool and fixture management systems. In addition, the company has to devise a way to get the "released job" list out to the shop floor scheduler.

When procurement and inventory systems are acquired from an MRP software vendor, some mechanism — typically requiring human assistance — is provided for transfers to and from the MRP system. In these cases the file formats usually are compatible. But in a large corporation, this is rarely the case. Almost every company using MRP has some internally developed, custom software for converting MRP information to forms required by the supporting services and systems. This software alerts those services/systems to schedule specified acquisitions, and feeds back inventory updates to the MRP information base.

Many companies use internally developed software to extract summary information from process plans, in order to create the MRP information base for product requirements. Similarly, the task of getting tool and fixture building orders to the shop is commonly performed by internally developed software, as is the task of releasing job orders.

In many companies, however, little emphasis is placed on these interchanges (apart from procurement/inventory) because MRP software is not directly linked to production systems in the manufacturing facility.

6.1.9 RESOURCE MANAGEMENT

Most of the companies contacted for the background study have some kind of tool management system, either developed internally or purchased off-the-shelf. The internally developed software provides a link to the procurement system for ordering tooling components and other consumable materials, and provides a link to the MRP system to obtain tool and fixture building orders. If the tool management system is an off-the-shelf product, the company still has to build these interfaces.

Some companies have developed systems for allocating human resources as well. These systems track specific skills and skill levels in the company's work force, treating them as "capabilities," in much the same way that a process planner deals with equipment capabilities. The human resource allocation system demands that the planned operations are associated with an indication of the skill level needed for their effective performance. Then, as the operations are scheduled, the allocator schedules the staffing to maximize product quality.

Another resource management concern is equipment maintenance. Some companies have developed software to "feed" preventive maintenance schedules — and in some cases requirements and statistical estimates of downtime associated with remedial maintenance — into the shop floor scheduling system.

6.1.10 PRODUCTION SCHEDULING AND CONTROL

The MRP-produced production plan identifies "jobs" to be performed — tools and fixtures to be built, quantities of products to be manufactured, etc. — over some period of time. Specific jobs are said to be "released" to the shop when the requisite materials, tools, or fixtures are available. The mechanism of this release is the transfer of information between the production planning systems and the scheduler. Several companies have developed software to provide this link for the particular combination of MRP and scheduling systems they use.

The interactions between predictive scheduling systems and control systems are usually implemented via a shop-floor monitoring system. Several companies have developed software to make the job schedule (from the scheduler) available to the shop-floor monitoring system, and a few are developing feedback software to provide the scheduler with live job status and equipment status data from the factory floor.

Identifying which task or operation is being performed by a controller usually requires either a manually updated information base or an internally developed enhancement to the controller software and its status reports. Bar coding systems for in-process materials are sometimes used to automate such updating procedures.

Companies with reactive scheduling systems typically link them directly to the controllers, manual key stations, and bar code readers. These links may use any of several standard protocols.

Because facilities typically use a number of standard protocols for communications involving machine controllers, several companies have developed a "controller communication center" —a hardware/software system that speaks a single standard protocol to all "non-controller" systems such as schedulers and monitors, and performs the required translations between protocols. Some off-the-shelf systems for this purpose already exist, and companies which use such systems only have to supply the software to link the scheduler and monitor systems to them.

6.1.11 PRODUCTION SIMULATION

The two primary functions of production simulation systems were outlined in Section 4.3.5. Commercially available systems of this kind are usually stand-alone packages. Although many companies use general-purpose commercial simulation packages, the use of internally developed, special-purpose simulation software is also common. A number of the companies surveyed had embedded a simulation system into an overall production planning system; this has required the creation of interface software providing links to other modules performing MRP and/or scheduling functions. There are no standard means for communicating between systems of these types, and it has therefore always been necessary to develop interfaces tailored to the particular software packages involved.

6.2 MOTIVATIONS FOR COMPANY EFFORTS TO DEVELOP IN-HOUSE SYSTEMS

Developing manufacturing software in-house is extremely expensive. Not only must the applications be developed and installed, they must be continually supported and upgraded. This represents a significant commitment of staff and resources. A detailed cost-benefit analysis is required to justify the decision to develop new applications rather than purchase them off-the-shelf.

The primary reasons identified by companies for developing and enhancing systems in-house include:

- *Specialized applications:* No off-the-shelf system is available to perform a certain function. Some software is too specialized to warrant commercial development. In that case, the user has no choice but to develop the application in-house or contract for custom development. This is especially true of exotic engineering analysis systems.
- *Integration with legacy systems:* New applications have to be integrated with existing systems that were developed in-house. When companies develop their own interfaces and schemas for information sharing, it becomes virtually impossible to integrate third-party applications into the system. Once a company starts down this path, it must continue developing its own applications or replace existing systems.
- *Legacy hardware:* Many companies are using obsolete hardware that is no longer supported by the original equipment manufacturer or by software applications developers. Software applications for outdated hardware platforms must be developed by the user.
- *Competitiveness:* Many companies view certain applications as crucial to maintaining their competitive advantage and responsiveness to the market. Companies may prefer to develop such applications in-house to safeguard proprietary data and processes.
- *Expense:* A company may determine that it is less expensive to develop certain applications in-house than to modify off-the-shelf commercial systems. Actual costs or savings to the company can be difficult to determine, depending on the accounting methods used.
- *Look and feel:* Companies sometimes prefer to have a particular look and feel for their operator interfaces that is not available in off-the-shelf products. This can promote user acceptance of new tools and capabilities.
- *Maintainability:* For critical applications that must work around the clock and be extremely reliable, many companies prefer to develop the software in-house and keep the software developers on staff.

6.3 COMMON APPROACHES TO INTEGRATION

Companies meet their needs for integrated manufacturing software applications in many different ways. The following approaches are among the most common:

- *Build application from scratch, in-house.* For a variety of reasons, many companies develop some of their own manufacturing software applications, which are designed to work with existing systems. This is usually done by staff programmers, based on an internal analysis of requirements.
- *Modify or add to a commercial application.* A company may need to modify a commercial application to fit the company's own requirements. Depending on the extent of the modifications, this can be more cost-effective than building a system from scratch. Some software vendors offer source code as an option with their system. In most cases this is the only way the system can be modified directly by the company. Another common alternative is to get the vendor to make the changes, usually for a fee.
- *Develop "wrappers" using "open interfaces."* Using interface libraries supplied with commercial applications, it is possible to develop software that integrates the application with the rest of the company-specific environment. The commercial application essentially is "wrapped" in a software interface that gives it an external appearance compatible with other applications by converting formats, information files, etc.
- *Use macro languages.* Many applications allow users to develop their own additions and extensions to the base application. This is generally done to provide user-specific functions, or to alter the user interface.
- *Create a central information repository.* Many companies have created centralized data repositories for information on design, process plans, features, control programs, etc., in an effort to facilitate data exchange between incompatible applications. The data may be in some neutral format for automatic retrieval, or in text form for the user to enter manually.

6.4 STANDARDS USED IN COMPANY SOFTWARE DEVELOPMENT EFFORTS

When developing integrated systems or additions to off-the-shelf systems, many companies take advantage of available standards whenever possible. In most cases, a particular standard is used because one or more of the systems involved supports it. The following sections describe standards (both formal and de facto) that are used commonly in the manufacturing industry.

6.4.1 CAD DATA EXCHANGE STANDARDS

Initial Graphics Exchange Specification (IGES) defines a neutral data format for representing 2-D and 3-D wireframe drawings, as well as some solid geometry specifications, tolerances, and other annotations. The most recent version is IGES 5.2, but many systems use older versions. This standard is widely supported by CAD systems for both input and output, and by many other design and planning systems for the input of design drawing and geometry. The encoding of annotations is not very standardized, and there are problems with interpreting information other than the appearance of a drawing.

Design eXchange Format (DXF) is a proprietary de facto standard, providing an alternative to IGES. It offers some improvement in annotation and geometry representation capabilities. DXF is supported by in particular by many PC-based CAD applications.

STEP AP 202 (ISO 10303-202) is an emerging standard intended to be an alternative to IGES for representations of the drawing itself, including annotations, tolerance specifications, etc. While it is believed to resolve many of the IGES problems, it is not yet widely supported by CAD systems, and consequently is not yet used commonly in company systems.

STEP AP 203 (ISO 10303-203) offers a significant improvement over IGES for exchanging design geometry information. Several newer systems use draft versions of AP 203 for exchanging geometry information among solid modelling packages, CAD systems, and finite-element analysis tools.

6.4.2 DATABASE STANDARDS

Standard Query Language (SQL), a general database query language, is used widely for interrogating manufacturing data repositories based on off-the-shelf relational database systems. Many other such repositories are actually company-developed file system indexing schemes.

6.4.3 MACHINE CONTROL PROGRAM STANDARDS

APT (Automatically Programmed Tools) is a high-level text language used to write machine control programs. The APT program is processed into an ordered set of instructions for NC (Numerical Control) machines. The output is a cutter location (CL) file, a part-oriented neutral file that cannot be input directly to an NC machine. The CL file must be post-processed for the target machine controller.

EIA RS-274D (M and G codes) is a standard language used to program most CNC machine tools. RS-274D control programs, however, are not portable between controllers. Most CAD/CAM systems generate APT CL data files. These CL files must be post-processed into machine-specific command representations, which normally use the RS274D command code standard.

EIA RS-494 BCL (Binary Cutter Location) is a standard machine tool programming language developed to address the deficiencies of APT and RS-274D.

6.4.4 COMMUNICATIONS STANDARDS

Manufacturing Message Specification (MMS) is a collection of machine control and information exchange message formats. MMS is used by a number of machine controllers for communicating with higher-level control and scheduling systems. It also is used for robots and many programmable logic controllers. But the subsets of the messages used (and the way they are used) have little in common.

SECS II, an MMS-like control and information exchange language, originally was designed for PC board and wire-laying equipment. Like MMS, it is sufficiently general-purpose to be used for many other fabrication machines in the electronics industry.

TCP/IP/Ethernet is the most widely used network services and transmission standard for direct communication among design and planning systems. In many cases, the only direct communications are file transfers. This protocol is widely supported by CAD system vendors and by most software development systems that produce company-specific software.

Many different (and incompatible) serial protocols are used for other direct communications.

6.5 PROBLEMS ENCOUNTERED IN INTEGRATION EFFORTS

Since only one person was contacted for each system examined, the problems reported reflect a collection of subjective viewpoints. Nevertheless, it is believed that enough systems were analyzed to provide a representative sample of the kinds of difficulties faced by companies in integrating their manufacturing systems. The most frequently mentioned problems were:

- *Lack of access to the internal data and functionality of commercial systems:* Vendors differ widely in their attitude toward the "openness" of their products. In the worst cases, it may be necessary to output files, usually in IGES or some proprietary format, to select what information can be used. Other systems provide proprietary macro languages. These can easily be used to enhance the system capability, but are of little assistance in interfacing between systems. The easiest packages to work with were those that provided dynamic procedural interfaces—often referred to as Applications Programming Interfaces (APIs)—that permit access software to be written in standard programming languages.
- *Unavailability of standards:* Several companies, particularly those working in the process planning area, commented that they try to be STEP-compliant, but are obliged to work with STEP standards (e.g., tolerance and form features) that are still in draft state. Another problem area was the interface between knowledge-based systems and conventional CAD or other application systems written in sequential languages such as FORTRAN or C. This was suggested as one area where the availability of standards would be particularly valuable.
- *Legacy systems:* Many companies need interfaces that link newer systems with legacy systems dating from a period when the closed system was the norm, and hardware and operating systems showed greater diversity than they do today. There is also the associated problem of reconfiguring the overall integrated system when an older module is replaced, since the original interface was probably highly customized and may require extensive reconfiguration.
- *Development time:* In many cases, integrated system development efforts exceed their time and cost estimates. It is currently difficult to predict what difficulties will be met with in linking two or more systems, so this is not surprising. One result of the MSE project should be to reduce the number of unexpected hurdles .
- *User/operator acceptance:* The problem of user acceptance is likely to arise when any new system is introduced, whether it is developed in-house or by a commercial vendor. With an in-house system, however, there is a particular danger that resources may be devoted primarily to achieving functionality, while user interface development is neglected. A commercially developed system cannot be marketed unless it has a polished user interface, but that problem does not arise for systems developed in-house, and the user interface is usually the first area to suffer cuts if a project overruns time and cost estimates. This leads to a system that users may find difficult to work with.

6.6 MOST SIGNIFICANT INTERFACE NEEDS

Most of the systems examined during the background study linked some internally developed custom software to a major off-the-shelf package. The most important requirement for the industrial customer was to get a "distributed" or "extended" system that performed a critical task reliably. Thus, the most significant need was to have a system that worked, and that continued working when the off-the-shelf package was upgraded, regardless of what it took to achieve that. Other perceived needs arose largely from the limited support of standards by off-the-shelf systems—particularly CAD systems, but also cost estimation and MRP systems.

A number of general concerns were seen as critical in integrating off-the-shelf systems. One was gaining access to system data via neutral data formats and/or some application programming interface (API). Because of the shortcomings of IGES, there was a commonly identified need for standard representations of geometry, tolerances, and features. For many planning systems, the need for a standard process plan representation also was cited.

Finally, many companies cited a need to capture information that the off-the-shelf systems did not. This included the intent and rationale for design and planning choices, so that downstream interactive systems could be used to make or modify decisions more intelligently. There also was an identified need to determine exactly what information was important to the company in making its decisions. In many cases the company was making do with information maintained by the software, rather than imposing its own information requirements on the software systems.

6.7 RECOMMENDATIONS

In this section we consider what could be done to help reduce the cost of manufacturing software development. Several approaches are examined below.

6.7.1 REDUCING THE NEED FOR COMPANY-SPECIFIC SOFTWARE

The development of specialized systems, information bases, user interfaces, etc., will in many cases continue to be a company responsibility, because the demand will be too low to induce vendors to develop off-the-shelf software. Some academic development in this area might prove useful, but the small market makes it unlikely that commercial software will be developed, and the small impact area makes it unsuitable for NIST (or other government) investment.

The integration of legacy systems with newer systems may be improved in the future, however. Fundamentally, all systems become legacies when newer systems are acquired. But standardization of interfaces would allow either of the systems to be accessed in a known way, without the need for specialized interface code. Standardization of interfaces to common manufacturing engineering and production systems, whether formal or de facto, will have wide applicability and impact when incorporated into commercial product systems. This should be a SIMA activity.

6.7.2 SOLVING DEVELOPMENT PROBLEMS

The time and expense required to develop and maintain software generally can be reduced by the use of common libraries and common information bases, and by the use of software engineering methods and tools.

SIMA can and should support the development of common libraries and information bases, particularly commercial products. Software engineering methods and tools, on the other hand, are general-purpose information technologies that are not specific to manufacturing. Development of these technologies, while desirable and possible, is therefore outside the scope of SIMA.

Access to the "internal" information bases of commercial systems requires the support of the vendor. The development and documentation of *some* external interfaces to allow access to this information should be strongly encouraged. SIMA can support the development of standards for these interfaces where possible, and can work with vendors to implement standards such as STEP Data Access Interface (SDAI) and Application Interface Specification (AIS) where appropriate.

Solving the problem of communications between systems requires changes in certain mindsets that are common in the commercial systems market. First is the "island of automation" view, where all control is seen as coming from the keyboard, with all input coming either from the keyboard or from files whose formats are specified by the vendor. Second is the "family of systems" view, where companies believe that all the software they need will be provided by one vendor, that it will all work smoothly together, and that it will work with nothing else. Third is the "master database" view, which holds that all the information a company needs can be stored in a common database with an external access protocol. Each vendor, however, tends to believe that its system provides *the* master database to be used by everyone else. The vendor's system may therefore be configured not to *use* the standard external access protocol, but only to provide it to other systems.

The "island of automation" view can be overcome by working with vendors to implement standards for data formats and communications interfaces. To some extent it also can be overcome by providing "wrappers" around a commercial system. This technique also can be used to overcome the "family of systems" view, if the interfaces are well documented. Some conversion routines and wrappers could themselves become third party commercial products, and this should be encouraged, particularly when the base product vendor is unwilling.

6.7.3 ADDRESSING INDUSTRY NEEDS

In August 1993, NIST held a workshop for industry leaders to address the question of their needs for systems integration. A report summarizing the results of the workshop can be found in [2]. Most of the systems integration needs identified by industry representatives during the workshop can be addressed by the development of standard interfaces. However, the deficiencies mentioned in the last paragraph of Section 6.5 (the need for capture of engineering rationale and company decision support data), because they are not supported by commercial systems andbecause they have wide applicability, represent potential areas of SIMA activity.

Determining what information is necessary to a company's decision-making processes is apparently a company-specific problem. But there is a common thread: how to make standard models and formats for information exchange extensible so that company-specific fields can be added, company-specific management rules and controls can be implemented, and standard but "useless" information can be ignored or converted to useful information. This general facility is an appropriate SIMA activity.

As to the capture of information on engineering rationales, the general mechanisms for accomplishing this are not understood, nor is the way in which those mechanisms would be applied to the manufacturing environment.

6.7.4 SPECIFIC RECOMMENDATIONS

1) With an view to the improvement of integration mechanisms, the MSE project should encourage, or if appropriate participate in, the development of:

- Standard neutral data formats for exchanging engineering and production information
- Standard application programming interfaces (APIs) providing access to the internal functionality and data repositories of commercial systems
- Third-party, off-the-shelf, standards-conforming conversion and wrapper software

This will involve active support of standards development, as well as the development of public domain "reference implementations". It also will involve cooperation with vendors to incorporate the standards into commercial products. Specific areas where standards are needed are product geometry, features, tolerances, process and production plans, and production resource and inventory information.

2) The MSE project should encourage and participate in development of standard libraries and information bases for manufacturing applications.

In addition, NIST should support development of commercial subroutine or "object" software libraries for specific manufacturing applications such as engineering analysis and design normalization. Most important is the commercial provision of standard information bases for materials, tooling and equipment, standard parts, and process plans.

3) The MSE project should explore methods for the capture and inclusion of engineering rationale information in product data, and protocols for its exchange between systems.

This involves research into the application of "intent capturing" techniques to manufacturing engineering situations (design and planning). It also involves the development and standardization of exchange formats for such information.

Some level of research must be completed before standardization is initiated. In the meantime, it is necessary to ensure that an avenue for such exchange is created, either through STEP files or by means of companion files linked to the STEP objects. It may be necessary to participate in some emerging standards activity in this area, if it is initiated elsewhere.

4) The MSE project should explore methods for allowing commercial software products to capture and make use of company-specific information requirements and rules.

This involves ensuring that the relevant standards permit and support such mechanisms. Vendors should also be encouraged to incorporate these mechanisms into commercial products. In addition, the MSE project should support the development of integration frameworks and tools that permit companies to capture and exchange company-specific information and rules in their integrated systems.

The MSE project should develop a reference architecture which defines the scope of activities which represent design engineering, manufacturing engineering, and production. This reference architecture would be an information model which represents the typical set of activities performed by industry.

CHAPTER 7: RESEARCH TRENDS IN PRODUCT REALIZATION

It is important for the MSE project to be aware of current research trends in product realization, which aim to fill certain technology gaps in currently available product realization systems. Improved systems incorporating new technology may become available during the lifetime of the MSE project, and may then displace older, less highly automated capabilities. This could lead to changes in the way the integrated system is best modeled in terms of functional subsystems. Corresponding changes will occur in the information flows between subsystems, affecting both the nature and the routing of the information.

This chapter discusses research directions that are likely to have the greatest impact on the MSE project if they succeed. It is not a complete summary of research in the area of product realization. Most of the material presented concerns research in universities—for obvious reasons, technical developments within software vendor companies are much harder to track. There is some overlap between the topics covered, and they do not all fit neatly under the three major headings of design engineering, manufacturing engineering, and production research.

7.1 DESIGN ENGINEERING RESEARCH

The amount of research in design engineering has increased markedly in recent years, to the point where it now represents a major proportion of the total product realization research effort. A major fraction (up to 70 %) of the total life cycle cost for a product is committed in the early stages of design [10]. Thus, improving the design process and relating it more intimately to manufacturing planning activities through the use of concurrent engineering is likely to result not only in shorter development time but also lower production cost and better product quality.

The major implications of design engineering research for the MSE project are that new types of design-related systems are likely to emerge during the lifetime of the project, and that existing CAD systems will acquire additional functionality. New requirements may therefore arise as regards not only the nature of the information transmitted between modules of an integrated system, but also the location of interfaces within the overall system structure.

Design engineering can be broken down into the four stages of product planning, functional design, configuration design, and detail design (see Chapter 4); these four stages correspond to those defined in Reference [11], though different terminology is used here. Although early design research concentrated almost exclusively on providing computer aids for detail design, more recent work has focused on extending the use of CAD systems back to the configuration and functional phases.

7.1.1 PRODUCT MODELING

The output of the design process is a specification of the product to be manufactured, which usually includes some non-textual description of it in the form of drawings or product models. These descriptions are referred to as *representations* or *models* of the design. The model acts as a substitute for the real thing, and provides answers to queries about the real product.

Different types of geometric models are generated by the various classes of CAD systems discussed in chapters 4 and 5. They include the 2-D drawing, the 3-D wireframe model, the solid model, and solid model enhancements containing parametric, constraint, and form feature information with their associated engineering semantics.

No single computer model can provide complete answers to all the queries that may arise during the product realization cycle. The complexity of this cycle for mechanical and electromechanical products often makes it appropriate to generate different models of the product for different stages of the design process. These models may be crude in the early design stages. However, the output of detail design should include a fully detailed geometric description of the product; it may also contain a great deal of nongeometric information of various types discussed in the following sections.

7.1.1.1 FEATURE MODELING

This topic is dealt with more extensively under the manufacturing engineering heading (Section 7.2). As mentioned in Chapter 4, many modern CAD systems provide facilities for feature-based design, allowing the designer to operate in terms of what may be thought of as volumetric elements having some significance in terms of the intended functionality of the product or in terms of manufacturing operations. For example, a company manufacturing gearboxes might configure its feature-based system to allow design of gear-wheels in terms of cylindrical blanks, shaft holes, and gear teeth, instances of each of these three feature types being definable through the specification of a small number of key parameter values. Then the designer does not have to be concerned with the precise details of the geometry of gear teeth—all that is taken care of by the system, and hence the design can proceed much more quickly.

The examples given above are, strictly speaking, *form features* [16,17]; they are usually represented within the system in terms of the geometry and connectivity of faces of the part model. Other types of features that have been suggested are *material* and *precision* features, the last relating to tolerance data associated with the model. However, material type can be represented by a simple attribute or label, and most current approaches to tolerance representation also use an attribute approach, so that there is no particular virtue in including these types of information as special types of feature. Nevertheless, it is sometimes convenient to provide an alternative representation of a volumetric or form feature by means of an attribute. For example, a beveled edge on a block corresponds to a prismatic volume of material to be removed, but its presence could be indicated sufficiently well for some purposes by a label "beveled" attached to the representation of a sharp edge in the model of an unmodified block. Although feature technology has now existed for several years, there still is no clear consensus on some of these representational issues, which remain an active subject of research.

A related research problem is that of creating feature-based extensions to the STEP standard fortransfer of CAD data between systems (see Chapter 8). This effort has been under way for several years, but is still far from conclusion, partly because the technology is still in a developmental phase. Initially the focus was on defining a range of standard feature types, but opinion now seems to be moving toward the idea of a standard means of *defining* features, in terms of admissible elements, the way they are connected together in the part model, and constraints on the geometric relationships between them. This will provide a ready means for the configuration of feature-based systems to operate in terms of company-dependent feature classes both for design and for other applications.

7.1.1.2 PARAMETRIC, VARIATIONAL, AND CONSTRAINT-BASED MODELING

Parametric, variational, and constraint-based modeling [18] were defined in Section 4.1.5. Parametric modeling is reasonably well understood and easy to implement in what is known as a *procedural* or *history-based* manner. The model is described as a sequence of operations on geometric elements, whose eventual outcome is the desired shape. For example, a block may be created with dimensions X, Y, and Z. Then a through hole may be created in the center of one face, with radius R. Specific parts may now be defined by assigning values to X, Y, Z, and R. The system will generate any one of these by running through the sequence of specified operations using the specified numerical values. It is easy to edit the procedural description to create variants of the basic part family.

Variational or constraint-based modeling presents more problems. The constraints are specified as equations associated with the relevant elements in the model. When the model is changed, these equations must be solved simultaneously (not sequentially, as in parametric modeling) to determine the details of the new configuration satisfying the old constraints and also the new changes. This can lead to severe computational problems since the constraint equations are in general nonlinear and there may be many of them. While existing commercial systems can currently handle constraints in the modeling of 2-D cross-sectional views, the extension of this capability into 3-D is still a subject of intensive research, regarding both the nature of the constraints that can be implemented and the method of solving the equations.

These modeling methodologies currently pose problems with regard to standards. The STEP standard (see Chapter 8) can be used for transferring CAD data between different CAD systems, or between CAD systems and manufacturing application systems, but at present it makes no provision for the transfer of parametric or constraint-based models, despite their generation by existing CAD systems in wide use by industry. A research effort has been started recently to rectify this situation, but it is not yet clear how much work needs to be done in making the necessary extensions to STEP.

There is a strong relation between the methodologies of this section and feature-based modeling. Features are naturally defined in a parametric manner, and inter-feature constraints need to be specified to ensure part functionality. Many current research programs are investigating the interaction between these areas of development and also the area of tolerance modeling as described in the next section.

7.1.1.3 TOLERANCE MODELING

Engineering tolerances are not well handled by existing CAD systems. It is possible to include them in a product model in the form of annotations, but these usually require human interpretation and cannot be used by automated application systems handling functions downstream of design. The crux of the technological problem is that the semantics of tolerances are not well understood [19,20]. Tolerance technology has been developed over the last few decades in an *ad hoc* manner, having its origins in shop-floor practice. Efforts are now under way to develop a more formal mathematical theory of tolerances to facilitate computer processing. Until this is achieved, however, the lack of effective tolerance modeling capabilities will remain a major void in CAD technology, preventing full automation and integration of design with a range of manufacturing and quality-related activities.

7.1.1.4 VIRTUAL PROTOTYPING

Virtual or computational prototyping is generally understood to be the construction of computer models of products for the purpose of realistic graphical simulation, often in a "virtual reality" environment [21]. This provides the ability to test part behavior in a simulated functional context without the need to manufacture the part first. The idea of "virtual" prototypes originated in the computer graphics community, whereas most of the models discussed above were developed by the engineering community. There is no clear-cut distinction, however; all such models can be used to provide answers to engineering queries.

Virtual prototyping lends itself to realistic process modeling. The availability of a graphical model of a part or product allows simulation of the effects of manufacturing processes. For example, it is possible to generate animated simulations of material removal during machining processes.

The advantages of using virtual prototypes in an immersive virtual reality environment are currently being studied by a few large manufacturing companies. Boeing uses them for "fly-throughs" of complex structures to provide visual checks for interference of parts. Caterpillar and the German company AEG use virtual reality environments to aid in the design of cabs for earth-moving equipment and trucks, respectively.

One significant problem with virtual prototyping is that no standard interfaces exist between CAD systems and virtual reality (VR) systems. Currently, it is common practice to generate models for VR purposes in a proprietary format designed for quite another purpose, to serve as input to stereolithography and other solid free-form fabrication (or rapid prototyping) systems. Automated feedback of information in the reverse direction (i.e., interpretation of the results of VR simulations in the original CAD model) is effectively non-existent at present, however, and this situation needs to be remedied before VR can be fully integrated into the design process.

7.1.1.5 MODELING FOR ENGINEERING ANALYSIS

As mentioned in Chapters 4 and 5, analysis and simulation tools provide support for the design process, aiding designers by computing information about functional behavior, cost, and other concerns pertinent to the design process. Many analysis and simulation tools are currently available, but the need for highly trained specialists to operate some of them is a strong barrier to their use by small companies in particular. One major problem is that the generation of computer models suitable for analysis from given initial CAD models is a lengthy procedure requiring specific skills. Another is that the analysis results cannot in general be fed back automatically into the design process. This section gives details of research aimed at overcoming both these problems.

One of the most common types of engineering analysis model is the finite element (FE) model (see Section 4.1.6), a specialized approximate representation of a part in terms of a mesh of simple geometric elements, used as the basis of structural and other types of analysis. The elements are usually either triangles or quadrilaterals in 2-D (e.g., cross-sectional) analysis, and tetrahedra or hexahedra (distorted cubes) in 3-D analysis. In the case of structural analysis, loads are specified at the nodes of the mesh (usually at the corners of elements where they connect to each other), and the resulting displacements of the mesh are calculated, again in terms of the nodes.

Although the analysis is automatic once the mesh is set up and the loading conditions imposed, a "good" FE model cannot in general be created automatically from a detailed geometric product model. This is especially difficult in 3-D for a number of technical reasons. As mentioned above, the creation of good FE models generally requires the knowledge and experience of a highly trained human operator. Encapsulating that knowledge in a rule-based system has so far proved difficult [12]. Consequently, the interface between CAD and FE analysis is at present far from fully automated, and setting up analysis models can be a lengthy and painstaking task that sometimes creates bottlenecks in the design cycle.

There are two promising alternatives to FE analysis [13]. One is the *boundary element* method, in which the computation uses a mesh created to approximate the exterior of the object but not its interior. This simplifies the mesh generation problem, but the method is less fully developed than the conventional FE method, and cannot be used under all circumstances. The other alternative approach is the *boundary integral* method, which shows much promise since it allows the analysis to be performed directly on the CAD model rather than on a mesh approximation of it. This appears to circumvent the problem of mesh generation entirely. The first commercial system based on this approach recently has become available, but it is too early to assess how well it overcomes the FE interface problem.

The second major problem with FE methods is that the computed results relate to the mesh model, but they must be interpreted with respect to the original CAD model. At present, for example, if the computed stress concentration in a part exceeds acceptable limits, then a human operator will note that fact from the FE output and make the necessary changes to the CADmodel. What is ultimately needed in design optimization is the automatic feedback of FE results into the CAD system, with necessary design changes generated by an expert advisory system and directly implemented in the CAD model. This is a long-term prospect at present. The nearest approach to it is in certain optimization systems which modify component strength by adding or deleting layers of elements on the exterior of the FE model. This corresponds to increasing or decreasing material thickness in the part, but at present there is generally no direct feedback to the CAD model. New standard interfaces between FE and CAD will need to be defined if this type of automated feedback becomes a practical possibility for general use.

Analysis and simulation tools are most frequently used in the detail design phase, after the part is fully described. However, as emphasis shifts towards the use of concurrent engineering, where decisions must be made earlier in the design cycle [14,15], FE and other analysis tools will need to be developed to support the design in its earlier phases as well, for example by providing approximate results on the basis of incomplete design information. While this is a recognized problem to which solutions are needed, not much research in this area is known to the authors.

A final point is that not all analysis models are geometric. A model used for estimating production cost is much more likely to take the form of an algorithm or set of formulae, taking into account the time taken by manufacturing operations, the operational and depreciation costs of the equipment used, costs related to tool wear, and so on.

7.1.2 CONCURRENT ENGINEERING TOOLS: "DESIGN FOR X" AND LIFE-CYCLE DESIGN

In the past, design engineering and manufacturing engineering have been regarded as sequential activities, with the results of one process "thrown over a wall" to initiate the second. This traditional, compartmentalized approach to product realization is inflexible and incapable of rapid response to changes in market requirements [15].

Concurrent engineering is viewed as a way of breaking down this rigid mode of working. It enables reduction of product development times by cutting down design iteration, production costs, and post-manufacture design corrections, by addressing production issues throughout the design process, as explained in Chapter 4. To a large extent, concurrent engineering is an organizational issue of getting people to work together in new and flexible ways, but there are certain areas of research aimed at providing computer aids to streamline the process [15,22].

Four fundamental characteristics of concurrent engineering have been identified. First, it involves the collaboration of people working in different engineering disciplines. Second, the product models and associated analyses and process plans evolve continuously while remaining consistent with each other. Thirdly, concurrent engineering systems must facilitate the reuse of previous designs. Last, and most importantly for SIMA, concurrent engineering will be most effective through the use of integrated software systems handling many different aspects of the product realization process.

One type of computational aid under development for concurrent engineering is the "Design forX" (DFX) technique, where the "X" stands for almost any downstream product realization activity (e.g., manufacturing, assembly, testing). The intention is to provide continual feedback to the designer to help in improving or simplifying the corresponding downstream activity. For example, a particular aspect of a design may cause it to be very expensive to manufacture. In the past, such implications were discovered only after the design process was complete, giving rise either to unnecessarily high production costs or the necessity for redesign. Using DFX, where X in this case would be "manufacture," the designer will be alerted to the problem early in the process, so that design improvements can be made quickly and inexpensively.

DFX attempts to determine all the factors in a particular design activity that influence a specific downstream activity and to use this knowledge in optimizing the design from the point of view of that activity. A significant problem arises when two or more DFX systems associated with different activities offer conflicting views. Research into methods for resolving such conflicts is currently in a very early stage.

The two DFX techniques that have received the most attention are Design for Assembly (DFA) and Design for Manufacturing (DFM). At present, both have been applied mainly in the detail design phase, although there has been some work relating to the configuration phase, particularly in the case of DFA.

Concurrent engineering methodologies such as DFX can in principle be extended to any type of post-design activity and can be applied throughout the product realization cycle. Taking into account such issues as product reliability, testability, maintainability, disposability, and recyclability results in what is known as "life-cycle design" [23]. Not all these issues are within the SIMA scope, however. Life-cycle engineering is much more developed in the electronics and software fields than it is for mechanical products.

Not only is DFX applicable to concurrent engineering and life-cycle engineering, it is also an enabling technology for *continuous improvement*. In this practice, a product's design and production process is reviewed constantly during its production lifetime to improve its performance, lower its cost, or otherwise increase its appeal to consumers. In the past, modification of a design or established production process has been viewed by management as disruptive and harmful [24]. In the context of continuous improvement, however, design change is both necessary and desirable. The objective of continuous design improvement is to determine all factors that can play a role in the success of a product—including customer requirements, product cost, quality, consistency, reliability, and maintainability—and to find ways of improving the design with respect to those factors. Companies using continuous improvement often have innovative and effective approaches to design [25].

7.1.3 DESIGN REUSE, VARIANT DESIGN, AND DESIGN INTENT

An important engineering design research issue concerns the reuse of previous design information. This obviously has strong relevance to continuous design and concurrent engineering. The term "variant design" refers to the retrieval and modification of previouslyexisting design specifications to satisfy new design goals and constraints []. The retrieval process can range in sophistication from a simple manual search to automatic identification of similar designs based on some criterion such as part functionality. Similarly, design modification techniques range in complexity from manual changes to the design specification to automatic modification based on new design objectives and constraints.

The capabilities of current CAD systems do not cover the requirements of variant design. Most design retrieval procedures are only performed by part or assembly name, no provision being made for the representation of part functionality information in a manner suitable for automated use. "Reasoning" capabilities in variant design systems draw on research in artificial intelligence—particularly analogy- and case-based reasoning research. Much of the success of variant design systems, therefore, will depend on advances in these fields [26].

Another fairly new but related area of research concerns the capture of "design intent" during the design process. The intention here is to store—in addition to the drawings, models, and information currently representing archived designs—an account of the decision process that led to their creation. The purpose of including design intent information is "to organize information needed to answer questions about the evolution of the designed artifact and the process through which it matured." [10] This capability is central to variant design, which aims to keep and reuse as much prior design information as possible. Despite claims to the contrary, current CAD systems provide little, support in this respect. The capture of design intent is particularly valuable for products having a long life, when it may not be possible to ask the original designer why particular design decisions were made. Many vendors of parametric and variational CAD systems claim that their systems do indeed represent design intent, but this is only true in the sense that a system stores constraints imposed by the designer, to be adhered to in any subsequent design modification. What is at issue in the present context is not the constraints themselves, whose importance is well understood —design intent in the sense used here is concerned not with the —mere *presence* of a constraint, but rather with the *reason why* that particular constraint was applied.

Several issues need to be addressed before the capture of design intent can become a practical reality. These include how specific the design intent information should be, whether it should include the same amount of detail for all stages of design, and how it should be represented in a computer-understandable format to allow automatic query facilities. A major problem is that of deciding *how* the design intent information is captured. Any requirement for its direct manual input by the designer is likely to slow down the design process significantly, and could distract attention from other important considerations. Another issue is how to record informal (e.g., "back-of-the-envelope") and abstract design information. Clearly, important standardization issues will arise if CAD systems acquire the capability for capturing and representing design intent information.

7.1.4 DESIGN OF ASSEMBLIES

Existing CAD systems are oriented almost exclusively toward detail design of individual parts. The construction of assembly models is therefore a follow-up activity. However, configuration design most naturally proceeds in the reverse direction, starting with a general layout. The interactions between parts in the assembly are then specified, and the detailed geometry of the components in non-interacting regions are filled in later.

For example, in a connecting rod of a car engine, the primary functional surfaces are the bearing surfaces, which are the most important elements of the part in the initial stage of configuration design. The next stage is likely to concern methods for assembly and disassembly of the connecting rod to allow for its installation and maintenance. The central part of the component may be regarded simply as a piece of material to hold the two ends together; its precise shape may be determined at a later stage, at which time an analysis may be performed to find the minimum-weight configuration that provides the required strength. Thus, for much of the process, the shape definition of the connecting rod is incomplete. Similarly, tolerance data (see Section 7.1.1.4) are essentially concerned with interactions between components, and so are associated with the product model during assembly definition rather than at the stage of individual part design.

Efforts are under way to develop CAD systems that permit the top-down design of assemblies [27]. There is a related deficiency in the standards area, since the STEP standard (see Chapter 8) only allows the representation of assemblies as collections of positioned and oriented parts. It makes no provision for capturing details of the functional interactions between parts, a crucial aspect of configuration design.

7.1.5 MODELING THE DESIGN PROCESS

One major aim of research into modeling the design *process,* rather than the designed products themselves, is to extend the use of engineering design systems back from the configuration and detail phases of design to earlier phases. There are two types of design process models, *descriptive* and *prescriptive*. Descriptive models result from studying the processes, strategies, and problem-solving methods used by designers [28]. Prescriptive models are divided into two categories: those that prescribe the manner in which the design process should proceed, and those that prescribe certain attributes of the product model of the designed artifact.

Descriptive models have been developed primarily through the study of protocols, cognitive processes, and design examples. Design protocol studies attempt to record the actions of a designer during the evolution of a design. For instance, the designer is encouraged to think aloud, and is queried to clarify or expand design decisions. Protocol studies generally have been performed at the functional and configuration design stages.

Cognitive models describe, simulate, or emulate the skills that humans use to solve problems. Research to develop cognitive models of the design process is relatively new, but it is seen by some researchers as essential for the future development of CAD systems.

Research also has been conducted on defining a standard or "canonical" design process. Thereare considerable difficulties in doing this, in view of the diversity of approaches to design. However, there is general agreement on the nature of some basic design principles. The most popular prescriptive artifact model at present is based on Taguchi's approach [29]. This uses statistical techniques for the allocation of dimensions and tolerances in detail design. The objective is to decrease the sensitivity of the design to variations resulting from the manufacturing process. This permits the desired functionality to be attained with lower manufacturing costs and fewer parts rejected during the inspection process.

7.1.6 LEGACY DESIGN DATA

Industry currently faces major problems in handling design data generated by technology that has now been superseded. For example, the Boeing 737 airliner is currently being redesigned, but much of the design data for the original version exists only in the form of drawings, since this aircraft predates the use of general-purpose CAD systems. There is a strong incentive, therefore, to find methods of capturing information from drawings, or from the more primitive types of CAD representations, and to use it to create product models that can be used in today's more powerful systems. This is an important, albeit specialized, emerging aspect of systems integration.

There are many facets to this problem. One approach is to use a document scanner with a drawing and to try to output the results in some standard or proprietary CAD format. This has been tried for 10 years or more with no marked success. Even if it works well, the result would be only a computer representation of the original drawing, whereas a solid model or other more sophisticated type of representation would be more useful.

A related problem, therefore, is generating solid models from 2-D drawing data. Some experimental systems exist for this purpose, but they succeed only in a limited domain. There also has been some partial success in generating feature data directly from 2-D drawing files. Yet another approach is to bypass the original design representation completely, and to generate a CAD representation directly from measured information from a laser scan of the actual part. This is another active area of research.

An interesting consequence of the legacy data problem is the emergence of agencies that ship manually generated drawings to developing countries, where cheap labor is used to construct CAD models of the same parts or products.

7.2 MANUFACTURING ENGINEERING RESEARCH

Research in manufacturing engineering has been intensive in recent years, particularly in the area of process planning for machined parts. The current emphasis is on generative methods, although the variant approach was more popular previously. There seems to be a general consensus that form features provide the key to automated generative process planning, and there are some related efforts aimed at developing a feature-based approach to part classification and coding forvariant planning. There is a small but significant amount of work on process planning for non-machined parts, in particular for sheet metal, die-cast, and injection molded components.

7.2.1 PROCESS PLANNING

The function of the process planning activity [30,31] was reviewed in Chapter 4. The following paragraphs briefly describe some representative current research issues in this area.

7.2.1.1 DESIGN BY MANUFACTURING FEATURES

One approach to concurrent engineering is to require the designer to work "in manufacturing mode," in effect designing from the outset in terms of specific production operations [32]. For example, the designer of a part might be constrained to start with an initial block of material and to create the shape of the final part by subtracting from it volumes corresponding to machining features such as pockets or slots. If "standard" machining strategies are available for every available type of feature, then a process plan is immediately available on completion of the design process.

Recent opinion has moved away from this idea for several reasons. First, it does not provide a natural way for designers to operate; their concern is primarily with functionality rather than the processes used in making a product. Second, this approach presupposes that the manufacturing process is known before design commences, which is by no means always true. Third, design in terms of manufacturing operations does not result in a product model containing information that is immediately useful for other activities in the product realization cycle, such as FE analysis or assembly planning. Thus, it is still necessary to provide some means for recognizing features relating to these further activities.

7.2.1.2 FEATURE RECOGNITION

The initial motivation for working with features came from a growing realization that part models of purely geometric types do not readily provide the kind of information most immediately useful to a process planning system. At one time, it was thought that the solid model would be able to do this, but experience proved otherwise. There are two main approaches to solid modeling [33]:

- A *boundary representation (b-rep)* system represents a part as a connected collection of faces with specified geometry.
- A *set-theoretic* or *constructive solid geometry (CSG)* system represents it as a set of points in 3-D space, expressed in terms of combinations of simple volumetric primitives such as blocks and cylinders expressed in the same way.

It was found that b-rep and CSG modelers provided information at too high and too low a level, respectively, for easy interpretation by a process planning system. The appropriate "median"level proved to be the form feature, expressed as a (usually connected) set of faces in a b-rep model, or as interactions between two or more primitive volumes in a CSG model. Despite the popularity of the CSG approach some years ago, all existing commercial CAD modeling systems are now based primarily on the b-rep methodology.

Much attention has been given to the problem of automatically recognizing form features for manufacturing processes (machining in particular) from a model of a part, usually in the form of a solid model of one of the types discussed above [16]. In a b-rep context, this involves identifying a set of part faces that match some predefined sets of rules characteristic of each recognizable feature type. For example, a rectangular pocket consists of five faces: a rectangular floor perpendicular to four walls connected at right angles to each other at the corners (and therefore forming two mutually perpendicular parallel pairs). This has proved to be an easy problem to solve in simple cases, but is much more difficult in cases where features overlap and their characteristic face patterns are modified as a result.

The first commercial generative process planning systems for machined parts based on the automatic recognition of manufacturing features from a solid model are now available. However, they are only successful for a limited part domain, and their capability needs to be extended to cover other types of manufacturing processes.

7.2.1.3 FEATURE MODEL TRANSMUTATION

Many modern CAD systems allow the designer to design in terms of features. These systems provide a range of frequently occurring functional features and also offer the facility for extending this range with user-defined features to meet the specialized requirements of any particular product range. The design process with such a system results in a product model containing design feature information. The problem for process planning, however, is that design features and manufacturing features are generally not the same.

One illustration of this problem is a rib of material created by the designer as a strengthening element. If the rib exists on a machined part, then it defines two machining features, one to remove material on either side of it. Whereas feature recognition takes as its input a pure geometric model, the corresponding process when the input is a design feature model is known as *feature model transmutation* (also *feature mapping, feature conversion, feature transformation*—there is no agreement yet on the terminology). Here the problem is to input a design feature model and output the corresponding feature-based model for some other activity such as process planning or inspection [34,35].

Although not much has yet been demonstrated in this area, feature model transmutation will probably prove to be easier than feature recognition, since the input model contains more information. An essential preliminary task will be to check each design feature to see whether it is also a manufacturing feature; if it is, the scale of the remaining problem is reduced. No commercial systems yet provide a capability of this kind. For those having the capacity for automatic feature recognition, any design feature information in the input model is simply ignored-during the recognition process.

7.2.1.4 TOLERANCE ALLOCATION

This is the process of assigning manufacturing tolerances to features of individual components. Tolerances, like features, have a design view and a manufacturing view, and tolerances imposed by the designer must at the process planning stage be reinterpreted in the manufacturing context. If a certain feature is to be generated by a combination of operations—for example, by a roughing and a finishing operation—the tolerance may be distributed between them, so that each must be performed to some specified accuracy, while the overall accuracy is within the tolerance originally specified by the designer. Determining a solution to this problem requires a database of available manufacturing resources that details the accuracy of which each is capable, as well as its operational costs. The object of tolerance allocation is to assign tolerances to operations in such a way that the original specification is met but manufacturing cost is minimized. In general, the tighter the tolerance allocated to a feature, the more expensive it is to manufacture. There are usually many feasible solutions to this problem, and it is difficult to determine one that is optimal or near-optimal. One current approach to tolerance allocation research makes use of genetic algorithms [36].

7.2.1.5 OPERATIONS SEQUENCING, FIXTURING

These are two further aspects of process planning. Suppose that the manufacturing features of a part are known. A particular type of feature has only a limited number of ways in which it may be manufactured; for a cylindrical hole in a machined part the possibilities include drilling, boring, reaming, or some combination of these operations. The particular choice of operation, or combination of operations, may be made on the basis of (1) tolerances of location, size, and form associated with the hole definition in the part model, and (2) available manufacturing resources—in this case machine tools and cutting tools. Different operations have different inherent accuracies, and the same operation performed on different machine tools also has a range of accuracies depending, for example, on the rigidity of the machine tool and hence the amount it deflects when cutting forces are applied.

Thus, given the manufacturing features and the tolerance data, a set of operations may be determined for manufacturing the part. But the problem then arises as to how those operations should be sequenced. This is a difficult choice to make automatically. There are some easy aspects; for example, if a machined part exhibits a pattern of identical holes, all of them will normally be grouped together in the sequence, since they use the same setup and the same tools. Also, there are some natural precedences, since the roughing operation for a feature must precede its finishing operations.

Tolerances also play a part in operations sequencing [37,38]. Location tolerances are specified (in modern practice) with respect to *datum reference frames* (DRFs) which act as local coordinate systems. They are usually expressed in terms of elements of the part model; for example, three orthogonal planar surfaces may be specified as datums, and their combination as a DRF. Toachieve the necessary tolerances, it is necessary for the datum elements to be generated prior to the features referencing the DRF. This consideration, based on the necessity to achieve specified accuracy, gives rise to a partial ordering of operations.

Another aspect of sequencing relates to minimizing some objective function based on manufacturing cost, manufacturing time, or some combination of the two. One expensive operation in the manufacture of machined parts is setting up the part on the bed of the machine tool. It is therefore desirable to minimize the number of setups required during the overall process, and operations tend to be sequenced in groups that can be performed in the same setup. Finally, the part must be held while being machined, and fixtures must be chosen that do not obscure any of the features to be machined in that setup. This might require the definition of two or more setups for a set of features which, in the absence of fixtures, could be machined in one.

These examples show that many different factors must be taken into account in operations sequencing, which is therefore a difficult problem. Solutions are most advanced for rotational parts turned on a lathe, but for other types of machined parts and parts produced using other manufacturing methods, much remains to be done before automatic sequencing of operations becomes available in commercial process planning systems.

7.2.1.6 Planning of Non-Machining Production Methods

The vast majority of process planning research has concerned machined parts. Planning for the manufacture of sheet metal parts is the second most well-developed area, but there is only a small amount of research into the automated planning of other important manufacturing methods such as die-casting and injection molding [39].

7.2.1.7 Process Representation

A process plan model is a description, in some formal terms, of the desired behavior of a discrete-process manufacturing environment. It provides a means for communicating process plan information to other product realization processes and may also be edited to produce alternative plans in a variant planning environment. There has been significant research and development in this area, including work within the international standards community on defining a standard model for process planning.

A process plan model should support the following: task decomposition into subtasks, task concurrence, task synchronization, task sequence alternatives, task prioritization, task constraints, and resource allocation [40].

7.2.1.8 Process Capabilities

Manufacturing process capability is “the physical ability of a manufacturing process to perform one or more form-generating operations to some level of accuracy and precision” [41]. Physicalability is generally defined with respect to those attributes of manufacturing equipment used to produce part features. Examples of part attributes affecting process capability requirements include size, type and orientation of manufacturing features, part geometry, tolerances, surface finish requirements, and material. These must be matched with manufacturing resources that provide the required process capability in terms of resource type, maximum allowable dimensions, and achievable accuracy.

Given a known process and resource, the desired part attributes in terms of tolerances and surface finish are achieved through the appropriate choice of control parameters, such as (for machining) cutter speeds and feeds. These depend in turn upon knowledge of the part material. Knowledge of process capability is therefore fundamental in process planning.

Much information on manufacturing process capability is easily available from catalogs and handbooks of manufacturing machinery. Some manufacturers and software vendors are now beginning to make these data available on electronic media such as CD-ROM or as ready-populated databases in CAPP systems. These are essential preliminaries to the use of process capability information in integrated product realization systems.

Research in this crucial area has mainly been concerned with determining the precise nature of the information required, as well as its representation and standardization.

7.2.1.9 RESOURCE DATABASES IN GENERAL

This topic is closely related to the last. The integration of any planning activity requires access to databases of available resources for that activity. Taking machining as an example, databases are needed that contain details of all machine tools available (including their process capabilities as described above), all cutting tools, and tool holders. Materials databases are also essential—for machining purposes, the required information concerns hardness and "machinability," which affect the choice of feeds and speeds required to achieve particular tolerances and surface finishes.

One of the factors influencing the cost of machining is tool wear. This provides a further example of information that can usefully be stored. Various mathematical formulae have been proposed for the representation of tool life in terms of various factors such as tool type and material. The availability of this type of information, when associated with a tool database and linked to some means of recording individual tool usage, makes it possible to predict when a tool will need regrinding or replacement. Coupled with information on tool cost, this allows calculation of an important component of the cost of machining operations. The creation of accurate tool life models is an ongoing area of research.

7.2.1.10 PROCESS PLANNING METRICS

When using CAPP, and generative CAPP in particular, it is important to have measures available for the evaluation of both process plans and the systems used to generate them. In general, thereare many feasible process plans for the manufacture of any particular part. The selection of an optimal or near-optimal plan from among those possible is usually made on the basis of manufacturing cost, manufacturing time, or some combination of the two since they are related.

Current process planning research indicates the desirability of generating not just a single master plan but a collection of closely related plans, all being near-optimal when judged by the chosen objective. This allows for flexibility in the event of machine breakdowns, for example, when the plan in use can be diverted to an alternate branch making use of an alternate resource.

The evaluation of process planning *systems* is desirable for both diagnostic and quality reasons [42]. Some quantitative measure of the "goodness" of a plan, as discussed above, is a prerequisite. Two systems with corresponding technical capabilities can be compared in terms of precision, speed, robustness, resource utilization, and flexibility.

Precision measures whether stated planning objectives have been met and specific product attributes achieved. Speed measures the time required to generate the plan, and resource utilization measures how much of the available search space was explored in generating it. Robustness is concerned with the frequency and cost of system failures, and the ability to recover from them. Ideally, evaluation of both plans and planning systems should be decomposable into atomic components so that different parts of a process plan, or performance in different phases of the planning process, can be assessed individually.

The comparative evaluation of planning systems is more difficult when different capabilities are involved. In this case, it is harder to be quantitative; the assessment then depends on the particular combination of capabilities required by a particular manufacturing organization.

7.2.2 INSPECTION PLANNING

Inspection planning is the determination of a strategy for checking that a part has been manufactured according to specification [43]. Measurements may be taken using various kinds of devices, including laser range-finders and coordinate measuring machines with mechanical probes. The former are usually used to scan the part and measure the positions of a large array of closely-spaced points on its surface. These points may then be used to construct a surface that can be compared with the nominal surface specified in the original design. The most accurate results are obtained when the laser beam is roughly perpendicular to the part surface; as a result, this process is not well suited for measuring the surfaces of holes.

Coordinate measuring machines are programmed to contact the part surface with a probe at certain selected points. Surfaces are then calculated from the resulting measurements and compared with the nominal design surfaces to find whether the two are within the specified tolerance.

Inspection planning can be regarded as a feature-based activity. Tolerances apply to features, and this provides a natural decomposition of the overall inspection problem into sub-problems. However, not all features can be inspected in the same part orientation, and more than one setup is generally required. Thus, as with process planning, it is desirable to find a strategy that minimizes the number of setups needed.

Much of the current research in inspection planning is mathematical in nature. For laser range-finding, the problem is developing economical methods for finding best-fit surfaces to very large numbers of points, which are subject to small errors of measurement. For coordinate measuring machines, it is desirable to have a strategy for inspecting each feature type that requires the smallest number of point measurements consistent with a given accuracy in the computed surfaces. This minimizes inspection time, and hence also reduces inspection cost.

7.2.3 Assembly Planning

Assembly planning determines how a product will be built from its individual components [44]. Like process planning for manufacture, it is driven by a combination of different requirements and constraints. There are many significant research projects aimed at achieving eventual full automation of assembly planning, but at present this prospect still lies in the future. From the purely geometric point of view, it is necessary to consider the trajectory of a part or subassembly as it is moved into position in the final product. In many cases this will consist of two parts, one of which is well-defined (for example, by the linear motion required to fit a cylindrical peg into a cylindrical hole) and one of which is less well-defined (the path bringing the peg from its initial position to the starting point of its insertion trajectory). The latter part of the overall path must be planned so that the new component does not collide with any other part of the assembly environment while it is in motion. The automation of path planning for assembly therefore requires extensive use of solid modeling techniques, including collision checking and avoidance. The problem is complicated by the possible necessity to rotate the component as it traverses the path in order to avoid some obstacle.

Apart from the geometric problems associated with assembly planning, there are also combinatorial problems. For a product with many parts, there may be an immense number of different ways to combine individual parts into subassemblies and ultimately into the full assembly. For some parts there will be obvious sequences of events—a simple example is that a bolt must be installed before a washer and then a nut can be fitted to it. However, in a typical assembly, a significant proportion of parts do not have such obvious sequences, and some means of pruning down the possibilities is needed as the plan is generated.

Automated assembly may make use of equipment ranging from simple but inflexible pick-and-place devices to multi-degree-of-freedom robots. Automated assembly planning must of course regard the assembly device as part of the environment, and its geometry must be taken into account in collision avoidance calculations and path planning. Robotics in itself is an enormous and fruitful field of research, but much of this is outside the scope of the MSE project, which will use established hardware technology and software systems in this area.

7.2.4 Quality Planning

Quality planning determines how organizational quality goals will be met []. Much of the research in quality planning is aimed at developing management strategies and organizational structures that raise quality levels []. New management techniques such as Total Quality Management (TQM) and Concurrent Engineering (CE) fall into this category, as does research on the effects of certified quality procedures (such as ISO 9000) on business operations. Quality planning research is addressing all phases of the manufacturing life cycle. Considerable attention is being paid to developing metrics for evaluating the quality of designs, the quality of process plans, and the quality of production schedules.

An important aspect of quality planning is sharing quality data throughout the product life cycle. Quality data arises either internally (e.g., from shop-floor inspection) or externally (e.g., customer complaints). Ongoing research on quality database architectures is aimed at providing an integration mechanism to unify such data and make them accessible to all phases of the product life cycle. Other work is aimed at developing methods for interpreting quality data captured at one phase (e.g., process control) so as to be usable during "upstream" processes (e.g., tolerance allocation in design). This work includes the design and population of process capability data bases [47].

Other research is aimed at developing better analytical tools for quality planning. Issues include the application of statistical design of experiments technology to support planning decisions at all life-cycle phases. Within the scope of the MSE project, such applications include make-or-buy decisions, product configuration and tolerance synthesis, development of process capability data to support process planning, and robust scheduling. Other statistical tools being researched include statistical process control (SPC) for correlated product characteristics, statistical tolerancing and its relationship to yield and manufacturing costs, and Taguchi design methods.

As product tolerances are tightened, it no longer becomes feasible to maintain quality through high-precision measurements [48]. Often, measurement procedures are no longer at the traditional accuracy of 10 times better than the tolerance. They are frequently no better than four-to-one or, in a few cases, one-to-one. That is, the uncertainties of measurements used to provide closed-loop control of manufacturing processes are as large as the uncertainties in the manufacturing processes themselves. Under these conditions, much greater attention must be paid in quality planning to the cost-effective use of available resources. The control loops can be at widely varying levels in an enterprise. They may involve machine-level process control or may extend all the way to new product planning. One of the most intense areas of research is in the use of process capability data in the detailed design phase. Statistical tolerancing, which has the potential to greatly reduce production costs, is still poorly understood, particularly in its effects on the performance of assemblies.

7.3 PRODUCTION RESEARCH

Both of the topics discussed under this heading are very broad, requiring highly inter-disciplinary research for their future development. The first of the following sections is concerned withproduction scheduling and control, the second with simulation of manufacturing systems.

7.3.1 PRODUCTION SCHEDULING AND CONTROL

During recent years, it has become possible to gather data about events on the shop floor almost as soon as they happen. This has given rise to the possibility of integrated production scheduling and control systems running in real time [49].

There are many production scheduling and control software tools on the market today, but they are inadequate for inclusion in a real-time system. Having been designed to run in an off-line, completely stand-alone mode, they can neither be run in real time nor integrated with shop-floor data collection systems or other product realization software. In addition, most of these tools use narrowly focused and obsolescent techniques, making them incapable of handling a broad spectrum of emerging user requirements. These include handling multiple performance objectives, dynamic reassignment of job priority, and integrability with other software systems. The provision of this added functionality together with real-time capabilities requires solutions to a wide range of problems in operations research, computer science, and information management.

The major focus of research today is production scheduling. Many approaches to the modeling and solution of scheduling problems are currently being investigated. Most current systems use dispatching rules, which in general can optimize only one performance measure of the production system, using only one type of information. This may concern the jobs, the machines, or the work-in-process inventory. Currently, simulation techniques are used to establish which are the best rules to use in optimizing a particular performance measure.

More speculative scheduling research deals with the use of artificial intelligence techniques, including expert systems, neural networks, genetic algorithms, inductive learning, and fuzzy logic. These allow the use of both quantitative and qualitative knowledge in the decision-making process, can make use of much more complex rules, and can function in terms of a range of information about the entire job shop, including current jobs, expected new jobs, status of machines and material transporters, and status of inventory and personnel.

Development of a practical real-time system will require decomposition of the overall very complex scheduling problem into both a temporal and spatial hierarchy of subproblems, each to be solved by an appropriate combination of the techniques listed above.

7.3.2 SIMULATION OF MANUFACTURING SYSTEMS

Simulation models are computer-based tools used to characterize and analyze the physical, logical, and operational aspects of (in the present context) engineering and factory floor functions [50,51]. One of their primary uses is to assess the impact of proposed changes in those functions, to help in making beneficial decisions. Often, simulation provides the only viable technique for performing such analyses. However, for SIMA purposes, simulation will be usedfor such purposes as checking the validity of plans generated by activities in the manufacturing engineering and production stages of product realization.

Two types of simulation are of interest to the MSE project: *continuous* and *discrete-event*. Continuous simulation refers to real-time (or near-real-time) animation of individual processes. Examples are the simulation of material removal in machining, of the flow of molten material in filling an injection mold, or of the vibration of some component subjected to oscillatory loading. This type of simulation is used mainly in the design and manufacturing engineering stages of product realization.

However, the discrete-event type of simulation generally is used in the production stage. Examples of the kinds of events of interest are the arrival of workpieces at a machining station and their subsequent departure on completion of the machining process performed there. Typically, such a simulation will be based on a model of the job shop and the resources it contains, and may deal with the flow of many different types of components through the shop. Simulated inputs are usually generated randomly over time, according to specified statistical distributions. Simulations of this kind are effectively computational experiments, and significant conclusions can therefore be drawn only after the performance of series of experiments using the same model.

Commercial simulation tools are commonly used in industry today. However, they have serious limitations. Building a simulation model for use in solving a particular problem is time-consuming and difficult. Each simulation tool has its own model-building language, which takes days of training and months of practice to master. Research currently in progress is aimed at developing new graphics-based model-building tools, which would provide ready-built representations for a wide range of manufacturing resources, entities, and activities, and would allow the user to construct models by menu selection and filling in templates. An associated possibility is the idea of constructing complex models from simpler submodels stored in a library.

Once a model has been built, users often need to translate and/or re-enter existing system definition data, much of it already in computer-readable form, into the proprietary format used by the simulation tool. Such data may originate in a variety of sources such as shop-floor data collection systems, MRP systems, parts databases, and process planning systems. To overcome this problem, some simulation system vendors are working to provide automated import of data directly from other systems or databases. This will improve the integration potential of simulation programs and will reduce the current high cost of developing, validating, using, and maintaining models.

Currently, effective use of a simulation system requires expertise in statistics. A simulation is essentially a series of carefully designed statistical experiments, whose outputs must be carefully analyzed. These activities are both time-consuming and difficult, and a significant academic research effort is being devoted to their automation.

Another problem is inflexibility. Usually, models are constructed with a single narrow purposein mind. But a requirement frequently arises for reuse of the model for some other analysis. This generally necessitates changes to the model, which have to be consistent with the nature of the new purpose. Such modification and reuse of models currently is difficult, because there is a close coupling between the nature of the model built and the purpose for which it is intended. Today, experts often build new models for different analyses of the same product realization system. Research is in progress to develop general-purpose modeling techniques that will facilitate analysis from multiple viewpoints, using the same simulation model.

Vendors also are working toward providing animated graphics to show visually how the results of a simulation develop over time.

7.4 CONCLUSIONS

Research into engineering design is very broad in scope, involving many scientific and technological disciplines. Design research is of vital importance, especially in the developing climate of concurrent engineering, since decisions made at the design stage commit a large proportion of the overall manufacturing cost of any product and also have a fundamental impact on its quality.

Manufacturing engineering research has a narrower scope, but even so requires input from a variety of disciplines. Many functional requirements are involved, which interact with each other in complex ways. Issues related to integrating manufacturing engineering systems into larger product realization systems are of crucial importance in this area.

Both production scheduling and simulation suffer from severe integration problems at present, and research leading to the development of system interfaces and standardized data formats is potentially very significant for the MSE project. Production scheduling, in particular, also requires the development of better scheduling algorithms before it can play a fully effective role in the project.

Significant developments are likely to occur in all three areas during the lifetime of the MSE project. Important new capabilities will appear in commercially available product realization packages during the same period. The project staff must therefore keep abreast of product realization research and must continually reassess the impact of new developments on the project.

7.5 RECOMMENDATIONS

The following specific recommendations are grouped under the main headings of the chapter.

7.5.1 DESIGN ENGINEERING

1) Engage in standardization activities relating to the transfer of design information not currently covered by the STEP standard.

This includes feature-based, parametric and constraint-based design data, and design rationale information.

7.5.2 MANUFACTURING ENGINEERING

1) Develop process plan representation models and manufacturing resource models, with a view to their eventual standardization.

2) Capture the semantics of the dialogue which takes place between the manufacturing engineering function and other engineering functions in design and production.

This is a necessary step towards automating the dialogue

3) Encourage and monitor study of the critical role of form features in representing different application views of product data.

Both feature recognition and feature mapping are needed, as are also application-oriented feature taxonomies and a feature definition language allowing users to configure systems to their precise requirements.

7.5.3 PRODUCTION

1) Analyze functionality and interfaces of state-of-the-art production scheduling tools and discrete-event simulation systems within the MSE project scope. Develop functional and information models for them. Working with system vendors, develop a proposed neutral file and/or database format for information shareable with communicating systems.

2) Use results from the above effort as the basis for developing functional, interface, and information management and exchange standards for production scheduling and control systems, as well as for discrete event simulation systems.

3) Develop a standard graphical user interface for model building and graphical postprocessors to aid in understanding simulation results.

CHAPTER 8: STANDARDS RELATED TO MANUFACTURING APPLICATIONS

> *"Standards govern the design, operation, manufacture, and use of nearly everything that mankind produces . . . . How standards come about is a mystery to most people, should they even ponder the question."*
>
> John H. Gibbons, Director
> Office of Technology Assessment, 1992
> "Global Standards: Building Blocks for the Future"

This chapter is principally concerned with standards related to manufacturing applications, and particularly with the MSE project's strategy for adopting certain standards or supporting specific standards development activities. Standards related to general information technologies are discussed in Chapter 11.

Standards, particularly international ones, are increasingly used to create common markets and to influence marketing patterns in global trade. Many products have a worldwide market (cars, aircraft and consumer electronics products are obvious examples) and in the absence of international standards manufacturers are faced with the difficult problem of providing products meeting many different local codes for acceptability, safety and so on.

Another aspect of international standards, of particular relevance in the MSE context, is that they facilitate the operation of distributed enterprises. Manufacturing is increasingly occurring on a national or global scale. Within the U.S.A., many products are built from components designed and/or manufactured in widely separated locations. From a broader viewpoint, multinational corporations and international strategic partnerships proliferate. These developments lead to strong requirements for reliable data interchange over worldwide networks.

It was described earlier how the integration of product realization systems can benefit manufacturing industry, and how standards can play a crucial role in the achievement of integration. Both from this point of view and from their ability to enable distributed operations, standards strongly impact the competitiveness of U.S. industry. It follows that an effective standards strategy is needed within the MSE project to ensure that it provides the most appropriate standards support for the relevant industrial sectors.

The analysis in this chapter, and that in Chapter 11 on information technology standards, is based on a study of about 100 standards and standards-setting activities. The strategy outlined here addresses the overall MSE project—a strategic framework is proposed within which individual MSE project groups will make their own decisions regarding the use and development of standards in particular technical areas. Two appendices are included. Appendix A tabulates existing standards by technical area. Appendix B presents summary information about each standard surveyed.

8.1 THE EVOLUTION OF STANDARDS AND THE ROLE OF SIMA

By engaging in standards-related activities, the MSE project can further a number of goals, including:

- Enhancing the effectiveness and competitiveness of U.S. manufacturing industry by helping to develop and establish needed standards in the MSE area
- Ensuring the value of MSE project outputs to industry by making them compatible with existing standards
- Reducing uncertainties or development costs for the MSE project by adopting accepted standards for products and processes wherever possible

Before determining what kinds of investments in standards are appropriate, and what are the associated risks, it is useful to consider the "life cycle" of standards—how they come into being, how they are used and how they evolve. The process of setting industry standards has changed in recent years. In general, it follows a cycle that includes:

- Recognizing the need for a standard
- Gaining corporate and/or national support for standards investment
- Applying appropriate technology (either a technology "push" to develop new technology, or a technology "pull," where established practices are used)
- Codifying and documenting a consensus solution
- Testing and evaluating the standard
- Implementing the standard in industrial environments
- Periodically revising or reaffirming the standard to meet changing needs
- Retiring or replacing the standard when it reaches obsolescence

Four methods can be identified by which consensus standards are created in the MSE area:

- Dominance in the market of a particular mechanism, resulting in the "de facto" standardization of its characteristics
- Agreement among a group of vendors to supply a common interface so that their products interoperate, thus creating a "standard" within their combined customer base
- Agreement among a group of influential users to require particular features and interfaces in the products of their software suppliers
- Consensus by a committee of technical experts formed under the auspices of a formal standards-making body

Only the last of these results in a *formal* standard; the results of the other processes are variously termed *de facto*, *industry* or *informal* standards. However, informal standards are often subsequently ratified, usually with some adjustments, by formal standards-making bodies.

Due to the increasing prominence of standards in international trade and the economic strategies of industrial nations, the dominant method of standards development currently appears to be the second in the above list, followed by eventual formal ratification. While this saves time and bypasses some of the political complications of formal standardization from a clean sheet, the creation of such informal standards inevitably involves less than fully open participation by all interested parties. This may complicate the process of gaining the broad public consensus and support ultimately needed for a formal standard.

Whatever the process adopted, a solution proposed as a new standard has no practical value until it is effectively deployed in an industrial environment. In the case of software integration standards, this value is realized when the solution is implemented in off-the-shelf products from multiple vendors, so that out-of-the-box interoperation is made possible. A major goal of the MSE project is to encourage the implementation, by various means, of common integration solutions by the vendors of product realization software products. The development and adoption of formal standards embodying those solutions is a related goal that may be attained before, during or after their actual deployment.

8.2 TECHNICAL AREAS OF STANDARDIZATION

This section identifies the technical scope for standards of relevance to the MSE project. This scope is limited to consensual technical standards that facilitate information sharing and software integration. The scope does not address, for instance, regulatory or safety standards established by governmental bodies.

Some areas of standardization are of central importance to the MSE project. These relate to such global areas of product realization applications as system architecture, information modeling, and information exchange. This chapter will identify the areas of highest potential for MSE interaction with standards-making efforts, and will make recommendations as to how and when such interaction should occur.

As regards MSE requirements for standards in the general information technology (IT) area, the project intends to be a user of available standards. Little if any priority will be given to the development of any new capability in this area. However, where deficiencies are identified in IT standards coverage, appropriate organizations will be notified of our project requirements so that future standards will become more responsive to product realization requirements in the IT area.

There also exist standards whose interest to the MSE project is only peripheral. These may be used only for some particular specialized manufacturing application, or may relate only to a particular product family. Such peripheral standards will only be employed by the project if they are necessary for the creation of demonstration systems to meet the integration needs of specific industrial partners. It is not envisaged that the project will engage in new standards development work in peripheral areas.

Several taxonomies exist for categorizing technical areas of standardization. Committees of standards-setting organizations are usually organized according to these technical areas. Examples include the International Organization for Standardization (also known as ISO), the American Society of Mechanical Engineers (ASME), and the American Society for Testing and Materials (ASTM).

Taxonomies of manufacturing-related standards have also been developed in the United States by the Association for Manufacturing Technology, and in Europe by a joint project of the European Committee for Standardization (CEN), the European Committee for Electrotechnical Standardization (CENELEC), and the European Telecommunications Standards Institute (ETSI). This last taxonomy is of particular interest in that it is being closely studied by the Computer-Integrated Manufacturing Standards Board of the American National Standards Institute (ANSI). It is possible that ANSI will adopt the taxonomy with only minor amendments.

This European-developed taxonomy classifies advanced manufacturing standards into seven broad categories, which are shown below, together with their subcategories:

M1: *Internetworking:* Manufacturing Environment Architecture (M1.1); OSI-Standards for Industrial Applications (M1.2); Standards for Industrial Communications (M1.3); and Functional Standards for Industrial Communications (M1.4)

M2: *Data:* General Method for Definition of Application Data (M2.1); Applications Data (M2.2); Standard Parts Libraries (M2.3); and Group Technology (M2.4)

M3: *Processing:* Software Portability (M3.1); Software Modularity (M3.2); General Programming Languages (M3.3); Operating Systems (M3.4); Database Systems (M3.5); Knowledge Based Techniques (M3.6); Data Security (M3.7); Application Languages (M3.8); and Software Tools and Methodologies (M3.9)

M4: *Control equipment:* NC Equipment for Machines (M4.1); Coordinate Measuring Machine Controllers (M4.2); Robot Controllers (M4.3); Programmable Controllers (M4.4); Process Control Subsystems (M4.5); Transport System Controllers (M4.6); Automatic Testing Equipment (M4.7); Data Entry Terminals (M4.8); and Sensors (M4.9)

M5: *Human aspects:* Man-Machine Interface (M5.1) and Ergonomics (M5.2)

M6: *Mechanical aspects:* Machines (M6.1); Industrial Robots (M6.2); Auxiliary Equipment (M6.3); and Machine Data (M6.4)

M7: *General aspects:* Methodology (M7.1); Operational Safety (M7.2); Documentation (M7.3); Performance Testing (M7.4); Implementation Guidelines (M7.5); Operating Environment (M7.6); Terminology (M7.7); and Maintenance and Systems Integrity (M7.8)

None of the existing taxonomies is completely satisfactory for the purposes of this chapter, being either incomplete (due to, for example, the limited scope of a standards-setting organization) or overly complex. For instance, the CEN/CENELEC/ETSI taxonomy has many categories in which there are no standards at present. Consequently, a simplified taxonomy is used in this chapter. Cross-references to the CEN/CENELEC/ETSI taxonomy are given where appropriate.

The standards of interest to the MSE project fall into two major classes; those specifying how computer systems and software applications in general can communicate with each other, and those concerned with the vocabulary and semantics of communication regarding specific product realization topic areas. The first grouping covers standards concerning such matters as programming languages, networking protocols and database access methods (CEN category M3). Some of these are of major importance to the MSE project, and their further discussion is deferred until Chapter 11. The remainder of the present chapter is devoted to discussion of standards relating to industrial practices (CEN categories M1, M2, and some of M7) and manufacturing equipment (CEN categories M4, M6, and some of M7). These are the standards relating to specialized product realization topics. Roughly speaking, industrial practices standards are related to product realization activities above the shop floor, particularly in relation to manufacturing systems integration. Conversely, manufacturing equipment standards relate to the integration of shop floor equipment, covering communications, performance models, etc.

For MSE purposes, standards for industrial practices will be broken down into the following subcategories:

- *Frameworks:* this heading includes standards related to open architectures and methodologies for systems integration (matching CEN category M1.1)
- *Information exchange:* including standards specific to product realization applications (matching CEN category M2.2). There are three sub-groupings:
  - Electronic Data Interchange (EDI)
  - Manufacturing Management Data
  - Product Data (including STEP resources and application protocols)
- *Product realization:* including standards for product life cycle functions, and further subdivided into three sub-groupings:
  - Calibration and Performance Testing (CEN categories M7.4; M7.8)
  - Design (no CEN category identified)
  - Product Standards (CEN category M2.3)

Standards for manufacturing equipment will be classified under the two headings:

- *Communications* (CEN categories M1.2—M1.4)
- *Performance models* (CEN category M7.4)

Appendix A lists the titles of the standards surveyed for possible application in the MSE project, partitioned according to the MSE classification defined above.

8.3 MANUFACTURING APPLICATIONS STANDARDS RELEVANT TO THE MSE PROJECT

In Chapters 4 through 7 the main body of material has been consistently ordered to cover topics relating to design, manufacturing engineering and production, in that sequence. An attempt will be made to do the same in this chapter, though the application areas of some standards (notably STEP, as will be seen) spread across a wide range of activities, and therefore it is not always possible to draw clear-cut boundaries.

8.3.1 DESIGN-RELATED STANDARDS

It is possible to distinguish four different classes of standards of importance in engineering design. Firstly there are safety standards. These generally relate to the design of particular products, and their intention is to ensure that those products are safe to use. Examples include safety codes for pressure vessels, or for clearances in high voltage electric power systems. Currently, standards of this kind are often paper documents, consulted by the designer and applied as constraints on the design process. In some cases such standards have been implemented in the form of knowledge bases, in which case there is potentially a greater level of automation and integration in their use. Standards of this kind will not be listed in the present document, since as mentioned earlier they are generally very product-specific. From the MSE point of view the major problem of dealing with such standards in the environment of an integrated system is the general one of interfacing to a knowledge-based system. At present there is no standard way of achieving this, but formats such as KIF and KQML are under development to fill this technology void.

The second class of standards relates to product functionality, and in particular the interchangeability of commonly used parts in assemblies. Examples include standards for roller chains, screw threads and gear teeth. They allow components to be bought in from outside suppliers in full confidence that they will interoperate with components made in-house or obtained from other sources. Standards of this kind have much in common with safety standards; they are often embodied in paper documents, but also increasingly in the form of information bases accessible to designers.

The third class of design-related standards may be classed as *presentational.* Typically, these govern the appearance of design information on a drawing or on the screen of a CAD system. A drafting standard, for example, specifies a wide range of conventions including the significance of different line styles, the manner in which dimensional information is shown on a drawing, and the interpretation of the symbols used to represent tolerances. The intention of this type of standard is to ensure consistent interpretation of design information by all of the people who need to make use of it. Presentational standards, being concerned purely with the human interpretation of design data, have no direct relevance to the MSE project; for purposes of building integrated systems it is necessary for the information to be interpretable by the computer. However, the presentational standards have over the past several years played an important role in the development of the modern standards dealing with computer-interpretable information discussed in what follows. In effect, they have laid the groundwork for the new standards by specifying the nature of the design information to be captured. An example is the ANSI Y14.5 standard on the representation geometric tolerances on drawings. This has been used as the basis for Part 47 of the STEP standard, which defines formats for the computer-interpretable representation of the same information.

The fourth class of design-related standards is the one of primary interest to MSE. It contains the standards, referred to in the previous paragraph, concerned with computer-interpretable information storage and exchange in an electronic environment. Examples of this category include IGES for CAD data exchange and STEP for product model exchange. Such standards generally retain the capability for dealing with human-interpretable information (e.g., for the display of ANSI Y14.5 tolerancing on a CAD-generated drawing), but also make provision for the fully automated handling of such information. We expect that as the automation-oriented standards are extended they will first incorporate and eventually render unnecessary the capabilities of the human-interpretable standards.

The most important design-related standard for MSE project purposes is STEP, though in fact STEP is ultimately intended to cover the whole of the product life-cycle and is already being developed to cover a range of product realization activities downstream of design. The overall standard is designed to provide a comprehensive suite of capabilities for the exchange, sharing and archiving of product information in computer-sensible form, without loss of completeness or data integrity. The standard is currently under continual development, and the first parts actually to be formally ratified were issued as ISO documents in late 1994. Numerous other parts are in various stages of preparation at the time of writing. As regards the nature of the information transferable by STEP, there has been strong emphasis in the early stages on the capture of shape information (geometry and topology), though in the new facilities currently under development there is an increasing concentration on non-geometric information. The dozen STEP documents forming the initial release of the standard ISO 10301 are as follows:

- Part 1: Overview and Fundamental Principles
- Part 11: The EXPRESS Language Reference Manual
- Part 21: Clear Text Encoding of the Exchange Structure

- Part 31: Conformance Testing Methodology and Framework: General Concepts
- Part 41: Integrated Generic Resources: Fundamentals of Product Description & Support
- Part 42: Integrated Generic Resources: Geometric and Topological Representation
- Part 43: Integrated Generic Resources: Representation Structures
- Part 44: Integrated Generic Resources: Product Structure Configuration
- Part 46: Integrated Generic Resources: Visual Presentation
- Part 101: Integrated Application Resources: Draughting
- Part 201: Application Protocol: Explicit Draughting
- Part 203: Application Protocol: Configuration Controlled Design

Some of these parts provide infrastructure for the standard as a whole; for example, Part 11 defines a formal information modeling language that is widely used throughout STEP, and Part 21 specifies how the information in any STEP exchange file is physically formatted. Part 31 is the first of the 30-series parts which define means for validating the conformance of STEP translators and other software to the standard. The remaining parts listed are all contributions to the definition of the STEP standard means for representing and transmitting product data. The 40-series parts provide generic resources useful for many purposes within STEP; Part 101 defines and configures a subset of these generic resources for a particular application area, drafting (though the standard actually uses the British spelling, draughting). The practical exchange, sharing or archiving of data will be achieved using the Application Protocols, and currently just two of these have been defined. The first is AP201, which handles data for explicit drafting (in which there is no logical associativity between numerical dimensions and the lengths etc. of geometric entities on the drawing. The second, AP203, deals with configuration-controlled design; it allows the association of positioned and oriented part models into assembly models, and also handles such administrative matters as release status and sign-offs on part and assembly models.

Many additional parts of STEP are in various stages of development. On the infrastructure side, one of the most important for the MSE project is Part 22: STEP Data Access Interface (SDAI). This will provide a dynamic means of accessing models stored in STEP format in a database. Several language bindings for the SDAI (initially FORTRAN, C and C++) are also being worked on.

Other emerging parts of the standard of immediate relevance to the MSE work are mainly concerned with specific product realization activities. They include the following:

- Part 45: Integrated Generic Resources: Materials
- Part 47: Integrated Generic Resources: Shape Tolerances
- Part 49: Integrated Generic Resources: Process Structure and Properties
- Part 104: Integrated Application Resources: Finite Element Analysis
- Part 105: Integrated Application Resources: Kinematics
- Part 202: Application Protocol: Associative Draughting
- Part 204: Application Protocol: Mechanical Design using Boundary Representation
- Part 205: Application Protocol: Mechanical Design using Surface Representation
- Part 207: Application Protocol: Sheet Metal Die Planning &Design
- Part 208: Application Protocol: Life Cycle Product Change Process
- Part 213: Application Protocol: Numerical Control Process Plans for Machined Parts
- Part 214: Application Protocol: Core Data for Automotive Mechanical Design Processes
- Part 224: Application Protocol: Mechanical Product Definition for Process Planning

This list excludes certain parts dealing with topics out of the MSE project scope, such as electronics design and manufacture, plant engineering, shipbuilding and the design and manufacture of composite structures. It should be noted that the topic areas are moving beyond the product design stage to include tooling design and the process planning of machined parts. Further extensions into the manufacturing engineering area are certain to occur as the standard develops further. The entries in the list are currently at various different stages in the standardization process, ranging from working drafts to Draft International Standards. MSE staff will need to evaluate the state of maturity of any particular part before deciding whether to adopt it for use in the project. The use of parts still under development will provide useful feedback to the developers, based on practical experience in the project.

Other standards for the computer-interpretable representation of product data also exist. These include the older standard IGES (ANSI Y14.26: Digital Representation of Product Data, informally known as the Initial Graphics

Exchange Specification), and EDIF, which captures the design data for electronic products. It should also be noted that some of these are national standards, and some are international. In addition, some standards have been adopted for mandatory use in Federal procurements. Table 8-1 presents the relationships.

Acronym	NATIONAL Standard	GOVERNMENT Standard	INTERNATIONAL Standard
DXF	defacto	none	none
EDIF	EIA EDIF Version 3 0 0	MILSTD 1840A	in development by IEC TC93
IGES	US PRO/IPO-100	FIPS 177	none
STEP	US PRO/IPO-200-nnn 12 documents	MILSTD 1840A FIPS in development	ISO 10303-nnn

Table 8-1 Cross reference list of PD standards

Most of these standards for product data representation are less concerned than STEP with product life-cycle issues, and primarily provide computer-interpretable representations of the shape or geometry of a product design. In common with STEP, they do not in general capture design criteria, design rationale, intended functionality, safety codes or other aspects. In fact, as pointed out in Chapter 7, most CAD systems have no facilities for capturing any of this information in computer-sensible form in any case. There are a few exceptions; some electronic CAD (ECTAD) systems can handle certain design rules such as the minimum spacing between tracks on a printed wiring board, which can act as constraints for routing algorithms. Corresponding facilities do not yet exist in the mechanical CAD area, although modern feature-based parametric and variational design systems described in earlier chapters will probably be able to support them in the future. It was noted in Chapter 7 that although standards such as STEP support data exchange between traditional geometry-based CAD and solid modeling systems, they are not currently capable of handling the features, tolerances and parametric and variational relationships defined by this new generation of CAD systems. An activity to support inclusion of these capabilities within STEP has just started at the time of writing.

An aspect of inter-system communication not yet addressed by STEP (except through the low-level facilities being proposed for the SDAI) is that of application programming interfaces (APIs).

There is one standard in this area, the Application Interface Specification (AIS) developed by CAM-I, which recently completed a period as an ANSI Draft Standard for Trial Use. This is STEP-compatible, and is intended to provide a standard means of access to the functionality of geometric modeling systems. The AIS allows an application program to call upon both high and low-level capabilities within the modeler, including Boolean and other modeling operations. At the data access level there is some overlap of functionality with the SDAI mentioned above, though the latter is primarily concerned with the manipulation and querying of individual elements in a database, and the validation of data, in a data-sharing environment. Besides the AIS there also exists a *de facto* standard API, in the sense that several vendors of CAD and CAD-related systems have based their products upon a single commercially available solid modeler (ACIS from Spatial Technology Inc.). The ACIS proprietary API provides a ready means for communication amongst these systems, which together account for a significant proportion of the total CAD market.

8.3.2 STANDARDS RELATED TO MANUFACTURING ENGINEERING

Manufacturing engineering activities give rise to specifications of processes to be used in the production of an artifact. Accordingly, the majority of standards in this area are concerned with the control of automated manufacturing and inspection equipment. The longest-established of these is the APT language for machine tool control, designed for the specification of cutter paths in

NC machining. APT is a high-level geometric language, which is processed to generate low-level instructions comprehensible to machine tool control units. There are mature standards at this level as well, notably EIA RS-494 (Binary Cutter Location language) and EIA RS-274D (M and G codes). More recently, DMIS (a US national standard) has come into use for the high-level programming of automatic inspection machines. Whereas most versions of APT only deal with the control side, DMIS also handles the passing of measurements back to the control system for further processing. An effort is under way within ISO TC184/SC1/WG4 to define an international standard based on DMIS. The Next Generation Control (NGC) project, coordinated by NCMS, is working in the general area of standard interfaces to intelligent controllers, and has recently issued a draft document under the designation RS-274/NGC.

The NIST Rapid Response Manufacturing Intramural Project is developing information models which specify a common subset of manufacturing resource data. This data may include characteristics of machine tools, cutting tools, tool holders, tool adaptors, inserts, collets, etc. This type of information is required to perform a variety of manufacturing engineering functions, including process planning, cost estimation, and NC code generation. Current CAE applications typically use and maintain this information in subtly different ways, which prevents sharing of the information between applications and results in multiple redundant stores of manufacturing resource data. It is expected that the project's results will provide a catalyst for a standardized and publicly available manufacturing resource data structure by providing proven results and a working strawman to appropriate standards organizations.

It was shown in the last section that the STEP standard is being developed to handle data relating to process planning, with an initial emphasis (in APs 213 and 224) on machined parts. Part 213 is intended to provide a means for the representation of a process plan, in terms of sequenced operations. This information is at a higher level than NC control data, since it is intended for the control of a collection of manufacturing resources rather than a single machine tool. Part 224 captures a specialized part description expressed in terms of machining features, suitable as input to an automated process planning system. Both of these have relevance for the MSE project.

Part 49 of STEP is intended to provide high-level generic resources for the representation of plans for any manufacturing engineering purpose. Indeed, it may have even wider application. Part 49 defines the resource constructs for the elements of a plan, and specifies the information necessary to represent the execution of a general process, including relationships between the steps in the process.

8.3.3 STANDARDS RELATED TO PRODUCTION ACTIVITIES

The only relevant standards that currently exist in the production area are concerned with communications, and these are covered in Part III of this report. There is an activity in ISO TC184/SC4/WG8, under the name of MANDATE, which may ultimately create standards for the representation of production-related data.

8.3.4 RECOMMENDATIONS FOR SIMA INVOLVEMENT

This section starts with some general recommendations regarding standards activities in the MSE project. Specific standards development activities are then identified in which the project should invest effort, and some recommendations made concerning particular areas in which existing standards should be adopted.

8.3.5 GENERAL RECOMMENDATIONS

The following should be considered for incorporation into future MSE projects to strenghten the relationship between SIMA and the industrial automation standards community.

1) Establish links to standards-making bodies relevant to MSE objectives.

MSE projects will be encouraged to identify all relevant standardization efforts that can provide support technology for their work, and all those that are expected to derive benefit from project results. The interaction should be approached as a two-way opportunity — MSE projects can gather needed technology elements and can tap into national and international resources, while standards efforts will benefit from better definitions of user requirements, pilot implementations of their work and drafts of new standards descriptions.

Links to these standards efforts can take many forms, and will vary with the degree of importance that the standards effort has to the project objectives. The most active form is that of participation in technical standards projects. Other possibilities include attendance at key overview meetings, review and comment on standards development ballots, and review of committee documents. It should be noted that actually holding office in a standards body requires a multi-year institutional commitment, but that technical participation is otherwise possible on a very flexible basis. Guidelines should be developed for the MSE project for reducing or withdrawing from active participation at the end of the project. The benefits derived by MSE staff from participation in standards development will include:

- increased understanding of the rationale and intended use of new standards
- access to the latest standards technology
- knowledge of relevant national and international R and D projects
- opportunity for professional development of MSE staff

2) Wherever appropriate, utilize MSE projects to demonstrate research feasibility of relevant evolving standards.

The effectiveness of a standard in the MSE field is not established just on the fact that consensus has been achieved and a document published. A standard is successful only if it solves the original industrial problem that prompted its development, and if quality implementations of it are available for industrial use. The MSE project can further this process by implementing drafts of critical standards during their developmental period and feeding back to the standards-making team their benefits and limitations. This will enable draft documentation to be appropriately amended before it is frozen into a formal standard.

3) Establish an ongoing effort to maintain awareness of emerging standards activities (both formal and otherwise), and to reevaluate from time to time the priorities for MSE involvement in them.

This is expected to be a continuous low-level activity. Its aim will be to identify new standards development thrusts, both those going through formal standardization procedures and those being promoted by vendors, users, industry or government consortia and projects. This will enable periodic reevaluation and reassignment of MSE standards efforts as judged appropriate for the needs of the project. The field of view should not be restricted to the USA. Information sources include the technical literature, personal contacts, Internet interest groups, etc. The risks and value to SIMA of associating with identified activities should be evaluated.

4) Maintain and enhance the compendium of standards activities given in the appendices of this report as a service to US industry.

Appendices A and B of this report present basic information on published standards and standards organizations that are relevant to the MSE project. If expanded by the addition of standards development efforts, the resulting compendium would become a valuable resource to the community involved with product realization systems. MSE may wish to contribute this initial work to the ANSI National Standards Network so as to make the listing accessible via the Internet. Such a compendium could be appropriately indexed and cross-referenced and could ultimately form a component of a framework for integrated systems development.

5) Assemble and maintain a library of standards and training materials related to standards.

In those areas of standardization judged to be high priority for MSE projects, a library of explanatory material should be created, including presentations, summary articles and background papers. These are available through informal contacts in standards efforts and give valuable insight about how the subject technology can be understood and best applied.

6) Establish criteria for individual groups within the MSE project to use in deciding whether to adopt specific standards, or to become involved in specific standards development activities.

The criteria for involvement in standards activities should take into account both the costs incurred and the potential benefits derived.

8.3.6 RECOMMENDED PARTICIPATION IN STANDARDS DEVELOPMENT ACTIVITIES

As a general policy, the MSE project should look to U.S. industry to drive the process of standards creation. Individual groups within the MSE project should take the lead in initiating a standards activity only if industry clearly expresses a need for the standard and provides support for its development. Otherwise, MSE projects should limit their involvement to established standards activities. Some of the more important of these from the MSE point of view are listed below:

- ISO TC184/SC4—This subcommittee is the forum for the development of ISO 10303, the standard informally known as STEP, which has the broadest detailed technical scope of any relevant standards activity. The MSE project should limit its attention to the Integrated Generic Resources and Application Protocol portions of STEP as these define the information exchange for a wide range of engineering activities. Projects to be followed should be determined by the nature of specific MSE requirements. At a minimum, selected AP documents should be reviewed, while technical participation in STEP development work should be considered in particularly relevant areas. A significant proportion of the effort that has gone into developing

STEP has been expended on the creation of technologies to support the formal and computer-sensible specification of standards in general (e.g., Parts 11 and 12, the 20 and 30 series Parts of ISO 10303). When developing extensions to an existing standard or developing a new standard, the MSE project should use STEP formal methods and technology for standards development and specification, in order to ensure compatibility with established STEP standards. Within the same ISO subcommittee another group is developing the ISO 13584 (Parts Library) standard, which may have a strong impact on the structure of databases in the MSE project. At least one meeting a year should be attended, and all committee documents reviewed.

- ISO/TC3-10-57/JHG (Joint Harmonization Group) — The purpose of the JHG is to harmonize all standards related to the geometric properties of products throughout the product realization cycle. This is important to the MSE project in that it relates manufacturing engineering functions to product characteristics. The project should at least track the activities of this group, and should preferably attend its meetings.
- ASME Management Control Systems — This ASME standards committee deals with requirements for the "establishment and execution of a controlled management system". However, the background study was unable to determine the committee's status, schedule or deliverables. These should be determined to in order to judge the relevance of its work to the MSE project.
- The MSE project should consider the stimulation of a U.S.-led effort to develop a standard framework for manufacturing information. Inputs to this activity should include the European CIMOSA framework (see Section 9.6.3), the Sematech framework (see Section 9.6.4) and the CFI results (although the two last are principally targeted on the electronics applications area). The framework should encompass models of manufacturing processes, resources, products, and controls. The proposed effort should be initiated outside the formal standards-setting arena, but with a view to the eventual introduction of a standard via an appropriate ISO committee, possibly ISO TC184/SC5/WG1, the group working on standards for enterprise integration.

8.3.7 Recommended Standards for the SIMA Project to Adopt

The MSE project should make use of existing STEP (ISO 10303) standards and modeling methodology whenever this is possible. This is because STEP provides the widest coverage by any standard of the MSE technical area, and because use of various parts of the overall standard in different areas of the MSE work will provide consistency of approach. When individual MSE groups develop new models or exchange formats, strong consideration should be given toward their eventual incorporation into STEP activities. As a fall-back position, at least for CAD data exchange, IGES (an ANSI standard) may be used where the corresponding STEP standard, or its implementation, is not yet available. Only as a last resort, and as a means for implementing short-term integration solutions, should proprietary data exchange formats such as DXF be adopted for MSE use.

There are two reasons why the background study team has only a limited number of recommendations for adopting specific standards within the MSE project. First, there are surprisingly few standards relating to manufacturing systems integration. This is the primary reason why an MSE standards strategy is important; the lack of relevant standards is one of the most significant barriers to progress in systems integration. Secondly, most MSE decisions on the use of standards will be domain specific, and hence best left to individual specialized groups within the overall project. However, these decisions should conform to the criteria for standards involvement recommended earlier in this section.

References

In addition to the standards listed in the appendices, the following documents were used in preparing this chapter:

AMT, *Standards Handbook*, Association for Manufacturing Technology, McLean, VA, March 1994.

ASME, *Personnel of Codes, Standards, and Related Accreditation and Certification Committees*, Pub. **AS-11**, American Society of Mechanical Engineers, New York, NY 1993.

Bloom, H. M., Formulation of Position on U.S. Standards Role in Enterprise Integration, NIST

Intergovernmental Report NISTIR 5484, August 1994, National Institute of Standards and Technology, Gaithersburg, MD.

IHS, *Index and Directory of Industry Standards* (in 4 volumes), Information Handling Services, Englewood, CO, 1990.

Information Technology Steering Committee (ITSTC), *Directory of European Standardization Requirements for Advanced Manufacturing Technology and Programme for the Development of Standards*, Memorandum M-IT-04 (Issue 3), CEN/CENELEC, Brussels, December, 5-6, 1991.

Lamprecht, J. L., *ISO 9000: Preparing for Registration*, ASQC Quality Press, Milwaukee, WI, 1992.

Roundtree, J. E., *Directory of DOC Staff Memberships on Outside Standards Committees*, National Institute of Standards and Technology, Gaithersburg, MD, 1993.

U.S. Congress, Office of Technology Assessment, *Global Standards: Building Blocks for the Future*, **TCT-512**, U.S. Government Printing Office, Washington, DC, 1992.

PART III: SUPPORTING TECHNOLOGIES

Part III is a study of information technology methods, tools and standards supporting the integration of product realization systems. Part III is organized as follows:

Chapter 9:	Systems Integration Process
Chapter 10:	Systems Integration Technologies
Chapter 11:	Standards Related to Information Technology

CHAPTER 9: SYSTEMS INTEGRATION PROCESS

This chapter defines basic terminology related to the integration of manufacturing software systems and describes a process for the realization of such systems. Some of the tools and techniques providing support for elements of the integration process are described, and recommendations are made regarding their employment in the SIMA MSE project.

9.1 INTEGRATED SYSTEMS

9.1.1 TERMINOLOGY

A *system* is a free-standing entity that performs some identifiable function or set of functions and has identifiable inputs and outputs. *Integration* is the process of getting *separate* systems to work effectively together. Integration produces a new system, called a *distributed system*, whose components are systems in their own right. The term *integrated system* is used, more generally, to mean both distributed systems and *extended* systems — systems which are created by adding modules to an existing system to provide additional functionality.

In a distributed system, each component system is considered a *subsystem*. A given system can be a subsystem in more than one distributed system. Every function of the distributed system can be decomposed into a set of subfunctions, each of which can be performed by one of the subsystems.

An *interface* is the common boundary between two or more subsystems or modules, at which those components must interact in performing a function of the integrated system. An *interface specification* describes the behavior of the participating subsystems at a particular interface, that is, the information and functional services each can expect from the other. An interface specification can prescribe messages and events, with rules for their occurrence and rules for the behavior of the systems when they occur, and shared information, with constraints on what each system can understand, access and modify.

9.1.2 CHARACTERISTICS OF INTEGRATED SYSTEMS

It must be recognized that an integrated system is, in all cases, a *new* system, and there can be a significant expenditure of resources in constructing one. It is therefore desirable to be able to measure the return on the investment of these resources. This would require that integration be characterized in quantifiable terms. The following notions are all expected to lead to quantifiable measures of success in integration:[7]

- *Interoperability:* the degree to which each component of the integrated system tolerates upgrade or replacement of the components to which it has interfaces
- *Modularity:* the degree to which the separate functions of the integrated system are identifiable with distinct subsystems and modules
- *Practicability:* use of established products for the systems involved and standard interface mechanisms and specifications
- *Adaptability:* the marginal cost of modifying the distributed system to extend or modify the functions it performs, assuming systems or modules which perform those functions are available
- *Integration effectiveness:* the degree of coupling between subsystems in performing a given function—what percentage of the information created by each producer system flows to the consumer system as rapidly as it can be used, as against flows which require human intervention or delays in operation of the consumer systems
- *Performance:* time and resources used by the integrated system in performing its functions, typically measured against the time and resources required for human labor and the un-integrated component systems to perform the same functions
- *Reliability:* the degree to which the distributed system is functional when needed and produces correct/acceptable results when performing its functions
- *Maintainability:* the ease of getting a system operational again in the event of a failure or a component upgrade
- *Quality, as perceived by the user:* performance, capacity, reliability, and maintainability, which to some extent is a function of the relative complexity of the distributed system

In any given integration activity, there will be tradeoffs among these, and some will be emphasized at the cost of others.

---

[7] These are preliminary ideas on quantifiable characterizations of integrated systems, and they have not yet been subjected to independent review.

9.2 SYSTEMS INTEGRATION PROCESS

Integration of manufacturing software systems is primarily a software engineering process. Multiple physical platforms may be involved, often giving rise to physical plant and network concerns, but these are secondary considerations. The major activities in the process of realizing integrated manufacturing systems, therefore, are the four major software engineering processes: requirements definition, system specification, system implementation, and validation.

A conventional software engineering project is either creation of a system to perform a particular set of functions, or modification of an existing system to perform different or additional functions. A systems integration project, however, involves both the creation of a new system (the distributed system) and the modification of many of the existing component systems. Moreover, the component systems are not, in general, modified as to the functions they perform, but rather as to the interfaces they provide. These differences affect all four phases of the software engineering process, and dictate the use of an approach differing from those described in textbooks on the subject.

9.2.1 REQUIREMENTS DEFINITION

The goals of the general software engineering requirements definition activity are:

- to understand the process to be automated
- to identify the aspects of that process which are to be supported by the new system
- to identify the additional functions which the new system is to perform
- to identify the objects and information those functions operate on, and
- to identify the rules for, and constraints on, the performance of those functions.

This activity produces a model of the partially or fully automated process, identifying the functions to be performed, their inputs and outputs, sequencing and timing of the functions, and other constraints. In a partially automated system the requirements for human interaction must also be defined. Requirements definition leads to the specification of system performance criteria and technical constraints. Methodologies and tools for representing processes and functions, and the constraints on them, are a standard part of the software engineer's toolkit. They are discussed further in Section 9.4.

In understanding the process, it is necessary to identify the real-world and conceptual objects that the process deals with and the information which must be associated with those objects in order to automate the functions. Each function has a collection of inputs (objects and information items that it uses) and a collection of outputs (objects and information items it produces). It is often necessary to understand the interrelationships of these objects and information items in order to understand the detailed requirements for processes and functions. Methodologies and tools for representing objects and information are another part of the software engineer's toolkit. They are discussed in Section 9.5.

Requirements definition is essentially the same for all development, modification, and integration projects. However, in integration projects, it is usually the case that most of the process elements, functions, objects and information items have already been identified and documented during the design and implementation of the component systems. What remains is to identify those aspects of the overall process that were not formerly supported and therefore become the targets of the integration process. It is also necessary to identify commonalities and conflicts in the models of process, function, object and information used by the component systems, and to reconcile them with a single comprehensive model. This reconciliation may lead to changes in the details of functional and performance specifications for the component systems.

Since it is expected that the process, functions, objects and information will be common to many manufacturing domains, it is a goal of the MSE project to perform the requirements analysis task for several manufacturing systems, to document the resulting models and identify the commonalities and differences in them.

9.2.2 SYSTEM SPECIFICATION

Two different perspectives are commonly used in developing a specification of system structure. The "black box" perspective describes only the requirements for the system and its effects on its environment. This perspective deliberately excludes information about how the system performs its functions—hence the phrase, black box. A black box perspective of a system can be described by the functions it performs, their inputs and their outputs. Time and other resources required to perform the function are important characteristics of the system. The multiple functions performed by a single system are not independent. This creates additional constraints, which must be included in the system description.

The "white box" perspective provides a transparent view of how the system functions, which is often called decomposition. The white box is ultimately composed of a number of black box descriptions of the component modules of the system. A white box description of a system as a linked set of subsystems can be called a *structural decomposition* of the system; it may be hierarchical, with intermediate levels of white box descriptions. This decomposition is an important part of the specification of a distributed system, because it reveals the relationship of the distributed system to its components, and allocates to individual subsystems the functions and constraints seen in the black box model of the distributed system.

The white-box specification activity begins by decomposing each of the required functions into a set or series of simpler functions. Several levels of decomposition may occur, and the process stops at a level when it is clear to the engineers how to map the functions onto software modules or systems. Systems and modules are then identified and functions are assigned to them. The specifications and constraints for the functions then become specifications and constraints for the systems/modules to which they are assigned.

In addition, when a major function involves subfunctions assigned to *different* subsystems, the performance of the major function requires coordination of the actions of those subsystems, and an *interface* between them arises. A major aspect of the systems specification activity is the specification of those interfaces.

In the particular case of systems integration, the assignment of functions to systems at some fairly high level has already occurred, and the component systems already exist. Thus this activity simply identifies the systems which are available to perform the component functions of the process. The critical aspect is therefore the specification of the interfaces; these determine how the components work together to perform the desired overall function.

The result of the systems specification process, then, has two components:

- the assignment of functions to systems, and
- the detailed specification of the interfaces between systems.

These two specifications are referred to as the system *architecture*. The process of developingan architecture may involve specifications at several levels of detail. It may also involve the application of standard patterns of interconnection and the use of standard interface mechanisms. Such choices improve the adaptability of the distributed system and its components to small changes in the functional specification, and thus make some levels of the architecture applicable to similar distributed systems. These notions are further discussed in Section 9.6.

The SIMA project will specify higher-level architectures for common activities of manufacturing enterprises, in the expectation that they will apply to many specific manufacturing domains.

9.2.3 System Implementation

Implementation is the process of producing a system which conforms to the architecture and meets the identified requirements. The result of this phase is a system which supports the modelled process in the planned ways.

In the case of systems integration, the major components already exist, but significant effort may yet be expended in linking them together to create the integrated system. Chapter 10 discusses the mechanisms of integration and the corresponding implementation concerns.

Methodologies and tools for software implementation and implementation management are the subject of much literature and will not be discussed here. They apply just as much to the integration of existing systems, and to the modification of the components, as to the development of systems from scratch.

The SIMA MSE program emphasizes the creation of specifications and standards that enable the integration of manufacturing systems. This emphasis does not eliminate the need to create pilot-system implementations based on those specifications. System implementations are the testable artifacts of this process, and will be undertaken to prove the utility of the specifications.

9.2.4 System Validation

Software development methodologies emphasize testing as a means of determining that the implementation meets the requirements. More particularly, *integration testing* is the process of determining whether the overall system supports the process as planned, i.e. that the system performs the specified major functions, abides by the specified rules and constraints, and meets the specified performance criteria.

The process of determining whether each subsystem or module conforms to its functional and interface requirements is known as *unit testing*. In the integration process, as in any software development process, both forms of testing are required. Where off-the-shelf systems are used, it may be presumed that a component system conforms to its functional requirements, but it is important to test whether it also meets the interface requirements specific to the distributed system. An advantage can be gained here when standard interfaces are specified. In that case, conformance to the interface may be the same as conformance to a national or international standard for which test suites, testbeds or other certification facilities may be provided by external organizations. *Conformance testing* is defined as "the testing of a candidate product for the existence of specific characteristics required by a standard in order to determine the extent to which that product is a conforming implementation [52]." Note, however, that conformance to the standard only guarantees interoperability at some level corresponding to the standardized mechanism, and may not guarantee total conformance to the *intent* of the interface specification.

SIMA interface specifications, therefore, should utilize and contribute to existing and developing standards. It is important that these specifications are clear as to their intent and contain statements of what constitutes conformance to them. Such statements help to promote uniform interpretation and implementation of the specifications, and aid in achieving the maximum benefit from their use.

9.3 MODELING

A model is an approximate description of some actual process or object. Engineers use models as tools in designing and building physical objects. Similarly, systems and software engineers use models to represent the characteristics of a system from several clearly defined points of view.

Creating a new system or improving an existing one requires that a great deal of information about that system be specified. Modeling is an important aid for the design of complex systems because it allows systems, processes, and objects to be partitioned into understandable elements with identified relationships. Understandable elements can be mapped into known implementation mechanisms, and the relationships can be used to define interfaces facilitating the distribution of work among different groups of implementers.

The following sections deal, respectively, with models of the process to be performed (or supported) by the system being designed, and models of the objects manipulated by the system being designed. In that sense, they are models of engineering requirements on the system.

The modeling process usually begins with a real-world prototype of the objects and processes to be modeled, often resulting from the analysis of some manually performed activity. Several definitions are important to understanding modeling:

- The *modeling perspective* is the orientation used when examining the prototype for the purpose of constructing the model. That is, the modelling perspective determines those aspects of the process and objects that are to be modelled.
- A single prototype is often modeled using several different perspectives. A *unified model* is one in which multiple models have been linked together to create a coherent whole.
- A *model view* is the orientation used in determining the requirements of a particularsubsystem for information and functionality from the unified model. Employing a modeling perspective to create a model is similar to employing a model view in the context of a unified model. Thus we call the focus of either function a *subject*.
- The *modeling methodology* refers to the process, guidelines, representation rules and tools used to create models.

9.4 REQUIREMENTS MODELING

Requirements modeling is really an application/linking of other modeling techniques and as such is "art" not "science-formalized modeling".

Function modeling, activity modeling, information modeling, simulation modeling, and process modeling can all be viewed as requirements modeling methods. The difference is that each provides a particular view (subset) of the complete set of requirements that are typically needed to completely define important characteristics of a system.

Missing from the current set of requirements tools are structured methods for collecting and managing user requirements. User requirements represent statements by the users of "what is needed" of a system. Other classes of requirements address "what is" and "how to provide what is needed".

Within the context, a requirements management system serves three purposes. First, it is used to capture and manage the fundamental building block of all remaining requirements activities — user requirements. Secondly, it can be used to collect requirements in support of other requirements modeling activities. Thirdly, it can be used to integrate, manage, and track all classes of requirements.

9.5 MODELING A PROCESS

The first step in understanding a new or modified system is to model the process the system is to support. Modeling the process covers such system aspects as functions, activities, information flow, events, interactions, time, and resources. There are several different approaches to modelling a process.

Functional modelling identifies all the functions a system is to perform, the inputs they require and the outputs they generate. It also includes decomposition of the primary functions into a *composition* (or sequence) of subfunctions, with the additional purpose of identifying the *elementary functions* of the process. This technique is usually used in defining specifications for algorithms to accomplish the *purpose* of a process.

Activity modelling represents a process as a set or sequence of interrelated tasks, each having inputs and outputs. Each major activity decomposes into its component interrelated subtasks, and so on, until tasks which can be directly implemented (or directly supported) by software have been identified. A function of the system is a *scenario* describing the set or sequence of actions (in the cases of parallel or sequential actions respectively), and the information flows resulting in the performance of that function. Relationships among subtasks can be modeled according to time dependencies, or according to input/output dependencies, or both. This technique may be used in modelling an existing process which is currently executed primarily by human labor or by independent systems. One models what they *do,* not what they are trying to accomplish. This is, however, a solid foundation for evolution of independent systems into an integrated complex, on the assumption that the individual systems reliably perform their intended tasks.

Process modeling depicts the response of a system to specific inputs or events, identifying the set or sequence of actions it takes and the results those actions produce. Process models in this sense are much more specialized than function or activity models and typically rely on the existence of one of the other models to determine the context for the detailed behavior specifications. Such models can be used to produce very clear engineering specifications for the behavior of interacting systems at interfaces. In order to avoid confusion, we will use the term behavior modelling for such modelling methods.

The remainder of this section describes common methods for modelling a process, including examples of all of the above approaches.

9.5.1 PROJECT EVALUATION AND REPORTING TECHNIQUE (PERT)

PERT was initially developed as a means of breaking down large and complex projects into manageable component activities. PERT models characterize subtasks as having required predecessors and successors. PERT orders subtasks into a directed graph in which the *root* node generates the final output of the major task and each arc between nodes is implicitly labelled by an object which is the output of the source and the input of the destination. PERT models assume that an output must be complete before a task to which it is input can be begun. This defines a partial ordering of all subtasks, indicating which ones can be performed in parallel and which must be sequential.

Extended PERT models, such as the Work Breakdown Structure (WBS), add information about resources and duration to the tasks. If resources are not considered, the duration information allows the PERT chart to become an optimal task schedule, which minimizes and predicts total time by maximizing parallelism. When resources are assigned to tasks, the model becomes a true schedule, in which parallel tasks cannot use the same resources at the same time. Extended PERT modeling systems may also support formal hierarchical decomposition, where a subtask in one model becomes the major task of another model.

These modelling methods have established graphical representations and commercially available tools to support model development. Some of these tools can additionally generate textual representations of the models.

9.5.2 IDEF0

The acronym IDEF stands for Integrated DEFinition (for Function Modeling). IDEF0, formerly the Structured Analysis and Design Technique (SADT), is actually an *activity* modelling approach that models subtask relationships solely in terms of inputs and outputs, with no concern for timing. It does, however, distinguish three types of input:

- source objects and information, which the subtask processes or consumes to produce its output
- control objects and information, which condition the behavior of the subtask in performing the process
- resource objects and information—resources used or required by the process, whose availability may affect the timing of the process

IDEF0 supports hierarchical decomposition. Unlike PERT, IDEF0 makes no assumptions about the completeness or finality of the source objects flowing into a subtask (there is disagreement about whether it should do so for controls and resources). As a consequence, it can be used to model continuous, interacting and cyclic processes.

IDEF0 defines a graphical representation, and is supported by commercially available tools for model development. These can commonly output alternative textual representations of the models.

9.5.3 PETRI NETS

The Petri net provides a functional modelling method that sees a process as performing a single function with a definable result. The process function is decomposed into a directed, but not necessarily acyclic, graph of component functions in which an arc connects the function which produces a given partial result to each function which operates on that result. Petri nets thus model the functional dependencies inherent in the component functions of a process, and as a consequence, the timing dependencies inherent in their implementation.

Petri nets are commonly used to define the functional dependencies of a process, and thus to identify flows and critical paths. They are also often used to simulate processes in order to obtain rough time estimates, identify bottlenecks, and analyze responses to perturbations in inputs or resources.

Petri nets have established graphical representations, and commercially available tools are available to support model development, some of which can also generate textual representations of the models.

9.5.4 FINITE STATE AUTOMATA

Finite state automata (FSA) are behavior modelling methods. The class includes a large number of published methodologies, variously called *state tables, decision tables, state charts,* etc. In all of these methodologies, each system or subsystem is modelled as having a set of *states,* a setof rules for the transition from one state to another, and usually a set of actions to be taken, or results to be produced, during each transition. The rules for transition are based on *events,* i.e. detected changes of state in external and internal objects.

Most extended FSA fall into the general category of Augmented Transition Networks (ATN). These allow the overall state of the process to be decomposed into multiple variables and multiple sub-machines, so that the true state of the whole system is a combination of the modeled states. Extended FSA also model actions that create events and change values (states plus) of internal objects and external objects. In addition, they can model *derived* combinations of states and events, which may be used to control transitions.

FSA and ATN models are particularly useful for formally describing the response of a system to unpredictable situations and events, i.e. situations in which the proper *sequence* of actions cannot be readily described.

There are many decision support tools that implement simple FSAs, and several academic languages and systems for ATNs. Like Petri nets, these tools are often used to simulate systems (and specifications) in order to determine their effective behavior.

9.5.5 RULE-BASED MODELS

Rule-based models describe the functions and behavior of systems as a set of *deductive logic sentences,* i.e. constructs of the form: IF <conditional expression> THEN <result statement>. The <conditional expression> is some AND/OR combination of "facts", information available externally and internally to the system. The <result statement> can be any combination of new facts, actions the system must take, and results the system must produce. Such methods often include both functional and behavioral notions.

There are many available languages for capturing and exchanging such models, none of which is in wide use. The Knowledge Integration Framework (KIF), however, has recently been proposed as a standard.

Commercially available tools, often known as *knowledge engineering systems,* support various rule-based modelling languages. While such tools can be used for various levels of functional or behavior specification, they are commonly used as implementation tools, and many are strongly biased in that direction.

9.5.6 SERVICE MODELS

Service models are behavior models which describe a system in terms of the distinct functions it can perform, the information required to perform each function, and the information produced on completion of each function. They assume that a function is performed in response to an explicit request, either from another system or from a human user. Service models use a "onerequest/one response" paradigm. Such a system performs only one function at a time, and only responds once to any given request, on completion.

System-to-system service models are modeled by an *interface definition language* (IDL). Similar IDLs are used and defined by the OMG Common Object Request Broker Architecture (CORBA), the X/Open Distributed Communications Environment (DCE), the ISO 13886 Language-Independent Procedure Call (LIPC), and the ECMA Portable Common Tools Environment (PCTE).

All of these IDLs assume the following rules:

- A particular system offers a specific set of services (functions) to other applications.
- The set of services is described in a single *interface specification schema,* along with all information and objects necessary to specify the services.
- Each service is modeled as a procedure call (see Section 10.1.1), which includes the name of the service, a list of inputs and their data types, a list of output results and their data types, and possible error results.

There are no standard languages for modelling human-to-system interfaces.

For the CORBA and DCE IDLs, there exist software tools that can read IDL and produce code templates directly supporting implementations. No tools exist to relate such IDL specifications to other models.

9.5.7 PROTOCOL MODELS

Protocol models specify the functionality of systems in terms of the *messages* they can process, and possibly the behavior of a system in response to each message. Protocol models are effectively limited to specifying interfaces between two systems which involve some form of direct exchange (see Section 10.1). In general, they specify the form and content of the messages, how they interrelate, and under what circumstances they can be sent.

Protocol models take a more general view of direct interfaces than do IDLs, in that they can deal with multiple requests. They can handle requests that affect other outstanding requests, multiple notifications for a given request, and notifications with no associated request. On the other hand, they model the elements of the interface, and may or may not identify the correspondence to the functions and activities performed by either of the communicating systems.

The ISO Open Systems Interconnection projects have adopted several standard languages for protocol modeling, including

- ISO 8824:1993 (ASN.1) describes message format and content. Additions to ASN.1 in1993 allowed requests and responses to be modeled and some requirements for message interactions to be stated.
- ISO 9074 (Estelle) can be used to model the behavior of a communicating party *protocol machine* in sending and receiving messages of various types. Estelle is like the Ada language in that it includes the concepts of event and priority, with actions represented by procedure calls.

Both ASN.1 and Estelle are used as specification languages at a very low level. Support tools are available for ASN.1 and Estelle that will read protocol descriptions, check them for correctness of form, and possibly produce code fragments that can become part of implementations.

9.6 MODELING OBJECTS

In describing the process performed by a system, it is necessary to identify the objects and information on which the system acts, and the objects and information it produces. When the process is decomposed into separate functions (possibly implemented by separate systems), further details of the objects and information, including new intermediate objects, become a part of the specification. Thus models of the *shared* objects and information of a system, called the *universe of discourse* of the system, become an important part of the engineering specifications, and are especially critical to integration.

The universe of discourse has four elements: objects, relationships, properties, and operations. Various modelling methodologies address some or all of these elements in somewhat different ways.

An *information model,* or conceptual schema, is a formal description of the possible states of the objects, properties, and relationships within a system. *Information analysis* is the process by which these objects, properties, and relationships are discovered. A *data model* is a formal description of an organization of information units conforming to an explicit or implicit information model.

Object-oriented analysis is the process of identifying the objects and operations in a universe of discourse, and then classifying objects by the set of operations they support. The information units which are attached to an object, and the relationships among objects, are then determined from the need to support these operations. An *object model* is a formal description of the object classes and the operations they support, and usually includes the required information units and relationships.

The remainder of this section describes popular methods for information modeling and object-oriented modeling.

9.6.1 THE ENTITY-ATTRIBUTE-RELATIONSHIP (EAR) METHOD

The EAR method is the oldest accepted information analysis method. It places each element of a universe of discourse into one of the following categories:

- *Entity:* any interesting object
- *Value:* an information unit having a simple representation
- *Relationship:* an association between two or more entities
- *Attribute:* an association between an entity and a value

Entity *types* are distinguished by the set of attributes and relationships the member entities possess. More advanced EAR models attach particular significance to the relationships "is a part of" and "is a kind of". The latter relationship is also referred to as a *subtype* relationship.

EAR methods were closely associated with the development of relational databases, and as a consequence, often have problems with multivalued attributes and relationships (i.e., situations in which an entity may have the same type of conceptual association to more than one entity or value). This gives rise to *value structures* (sets or lists of values), and to *reified relationships* (entity types which represent associations as objects).

EAR methods lead to a number of model representation languages, both graphical and textual. Two of these have been standardized:

- IDEF1-X (FIPS 183) has both a standard graphical representation and a less frequently used textual representation. The graphical format is most frequently used for the publication of models.
- EXPRESS (ISO 10303-11) has both a standard graphical representation (EXPRESS-G) and a standard textual representation (EXPRESS). The latter is far more commonly used, and in fact the graphical form cannot capture all the relationships representable by the language form.

Both of these (and several others) are supported by commercial software tools, at least for the development of the models.

9.6.2 THE BINARY (OR *n*-ARY) RELATIONSHIP METHOD

The binary method is an information analysis technique which classifies each element of a universe of discourse into one of the following categories:

- Non-lexical object (an entity, in non-NIAM terms)
- Lexical object (an entity name)
- Binary relationship (always viewed as bi-directional)

Binary methods allow and require the modeler to specify whether a relationship between entity classes is 1-to-1, 1-to-many, many-to-1, or many-to-many, and whether a relationship is possessed by all members of a class or only some. In many such methods, however, *n*-ary relationships and relationships that themselves have properties must be *reified* into entities. Like EAR methods, most binary methods attach particular significance to *subtype* relationships.

The only popular binary method is NIAM (Natural-language Information Analysis Method), although the binary method is a common component of several object-oriented analysis methods (see below).

The NIAM method has no standardized representations, either textual or graphical, although the several graphical representations in use are quite similar. The de facto standard textual language, RIDL, is not widely used. However, the EXPRESS textual language mentioned in Section 9.5.1 can be used to capture NIAM models without information loss, provided certain syntactic conventions are followed.

9.6.3 THE INTERPRETED PREDICATE LOGIC (IPL) METHOD

The IPL models the universe of discourse as consisting entirely of objects for which certain *propositions* hold. The conceptual schema consists of a set of *sentences* made up of:

- Terms and variables (entity instances, classes, and names)
- Predicates (verbs, relationships)
- Logical connectives (IMPLIES, AND, OR, etc.)
- Quantifiers (all, at least one, at most one, etc.)

Mathematically, the IPL model is a *first order functional calculus,* and has proved in practice to be capable of modelling both static or dynamic behavior of objects and information.

Languages embodying such methods have been used to capture information for a number of integration activities. The results, however, are not necessarily very readable. For those familiar with it, the DAPLEX model used for the Multibase project is a good example, as is the Semantic Unification Meta-Model currently proposed as a standard. The rule-based language KIF has also been used in the IPL context.

9.6.4 OBJECT-ORIENTED METHODS

The term *object-oriented* is applied to many different methods whose objective is to develop some form of object model. True object-oriented analysis methods concentrate on the identification of the conceptual objects in a universe of discourse and classify those objects by the set of basic operations they support. The type of a class is established by the ability of its members to support a certain set of operations.

Operations themselves are considered to have two components:

- the *message,* sometimes called the *operation signature,* which identifies the nature of the operation, the types of its parameters, and the types of its results, and
- the *method,* or the actual implementation of that operation on a particular object

By definition, a class supports a common set of messages. One class may be described as a *subclass* of another, and it is said to *inherit* all operations from the base class, that is, every object in the subclass supports all operations of the base class, and possibly others as well. But object modeling methodologies disagree on whether a class and its subclasses must also support a common set of methods.

Object modeling methods also disagree as to whether a class can be modeled as a subclass of more than one base class. This concept is referred to as *multiple inheritance.* The underlying issue is whether a model implies that classes not described as having a superclass/subclass relationship must be disjoint. Such a rule is convenient for implementations, but requires unnatural models of some real-world situations.

The properties and relationships of a class can be deduced from the set of operations the class must support. In many cases, the requirement to have a particular property is made evident by an operation that returns the value of that property. The representations of properties and relationships in an implementation are chosen to optimize the methods that use them. They are said to be *encapsulated* in the object implementation, and are of little interest to the model.

Many object modeling methods distinguish between *primitive types* (the types of machine-representable information units), and *abstract types* (the types for which operations are modelled). This distinction gives rise to rules about what is encapsulated and what is not, and what types can appear where in messages.

The principal object-oriented modeling methodologies are named after the authors of the books in which they were introduced (e.g., Rumbaugh, Booch, Schlaer-Mellor).

9.7 ARCHITECTURES AND FRAMEWORKS

As described at the beginning of this chapter, the distributed system performs complex functions by breaking them down into component actions that can be performed by subsystems. The systems specification process for the distributed system assigns specific actions to specific subsystems, and specifies the interfaces between those subsystems that are needed to perform the complex functions of the integrated system. Such a specification is called an *architecture.*

A *framework,* on the other hand, supplies general architectural principles and high-level unified models of the activities and objects in the domain, together with general-purpose interface mechanisms, in support of the development of architectures for specific distributed systems.

This section further explains these concepts and their significance in developing integratedsystems.

9.7.1 ARCHITECTURES

An architecture for a distributed system is a description of the system at several levels of specificity viewed from multiple perspectives. SIMA MSE architectures will be described at three levels of specificity: the *reference architecture,* the *engineering architecture,* and the *implementation architecture.*

Overall, the architecture is viewed in terms of the system's function, behavior, information content, interfaces, hardware, software, and communications networks. However, each level of architecture is not viewed from all perspectives. Instead, views are added as the architecture becomes more specific.

The *reference architecture* for a distributed system is created by mapping the overall system function onto logical groupings of activities expected to be performed by the system. A reference architecture has the following principal elements:

- *Functional decomposition:* the breakdown of each function of the distributed system into component functions, each having specified inputs and outputs
- *System model:* assignment of functions to subsystems, which identifies and characterizes the subsystem types

- *Information model* or *universe of discourse:* the entire collection of objects, relationships, and information units involved in interactions between subsystems, considered as a coherent whole

The *engineering architecture* adds detail to the principal elements of the reference architecture. It maps the logical grouping of activities into available or instantiable subsystems, and it identifies the types of exchange mechanisms and subsystem interfaces. An engineering architecture has the following principal elements:

- *Subsystem model:* identification of subsystems to be treated as black boxes, and the functions assigned to them;
- *Interface model:* identification of the interactions between subsystems—the set of output-to-input connections implied by the functional decomposition—in terms of both information and material
- *Integrating mechanisms:* identification of the exchange mechanism to be used for each identified interface and the applicable standards for mechanism or content

The final level of architecture is the *implementation architecture.* The implementation architecture adds final detail to the elements described in the reference and engineering architecture levels. This level is the most specific in that it identifies versions of products, languages, and protocols. Vendor software and hardware products that perform the functions and interactions defined by the engineering architecture are identified here, and detail is provided tocreate system-specific code. In addition to the elements described in the previous two levels, the implementation architecture has the following elements:

- *Hardware, operating systems, and communication protocols:* Identification of the types of hardware platforms, operating systems, and communication protocols to be used in the system implementation
- *System model:* identification of specific products that implement subsystems defined in the engineering architecture
- *Interface model:* specification of particular implementations of the specified exchange mechanisms, including selection of database management systems and other repositories
- *Information model:* specification of data models, file formats and messages corresponding to the objects in the conceptual schemas of the engineering architecture

The specification of integrated manufacturing systems in terms of reference and engineering architectures will be one of the core activities of the SIMA MSE program. It is in this activity where the matches will be made between integration requirements and the system specifications. This activity should indicate useful combinations of existing standards and should result in recommendations for new and emerging integration standards. Specific integration mechanisms, protocols and standards are covered in the two following chapters.

9.7.2 FRAMEWORKS

As stated above, development of a system architecture requires as input technical constraints (legacy applications and existing infrastructure), system performance criteria, business constraints, and organizational characteristics (organization structure and culture must be compatible with the intended system instantiation). In addition, it will be constrained by a development methodology, architecture structure, style guides, and pre-existing architectural artifacts. An *architecture framework* (from now on simply called *framework*) organizes many of the constraints and resources used in architecture development.

For the sake of this discussion, a framework is defined as an environment for developing architectures or implementations of distributed systems (this is not a universally agreed upon definition). Frameworks can include a generic reference architecture, conceptual guidelines for developing a system architecture, tools and methodologies for completing the architecture, and may even provide some integration mechanisms. As defined here, frameworks often include biases toward particular approaches to decomposition and modeling, and they may include partial information models or engineering models. While these biases may lead to less than optimal architectures and integrated systems, the time they save can far outweigh the difference in value between an optimal system and an adequate one.

If a framework is to support the use of modeling, then a decision has to be made regarding the forms of model representation to be made available. Most frameworks support only a limitednumber of modeling technologies.

Frameworks enable the creation and reuse of models associated with a distributed system. These models are not limited to describing actual systems. Some may be partially specified as inputs to further system development activities. There is a spectrum of models, with the most generally applicable constructs at one end (base models) and specifications of real systems at the other. In between are the partial models, in the present context referred to as *industry models*, which are generally applicable to enterprises of a given industry.

The potential reduction in the time and cost of developing system specifications through the use of frameworks should be an important consideration for the SIMA program. The identification of a framework for system specification, its use, and subsequent lessons learned may be a valuable contribution to the manufacturing community by the SIMA program.

9.7.3 CIMOSA (COMPUTER INTEGRATED MANUFACTURING OPEN SYSTEMS ARCHITECTURE)

To conclude this chapter, two examples of frameworks are briefly described. The first, CIMOSA, was developed by an international consortium of twenty-two companies within the European Community research program ESPRIT (European Specific Program for Research and Development in Information Technology).

The CIMOSA modeling framework [53] is that portion of the CIMOSA specification that contains and supports the creation of system models. It is divided into two parts: a *reference* architecture and a *particular* (engineering) architecture. CIMOSA's reference architecture contains the base model and partial models appropriate to particular industries, and so provides a reference from which to begin defining the models of a particular enterprise.

The developers of CIMOSA wanted its users not to be encumbered by the limitations of existing information technology tools when defining their requirements. They therefore organized the CIMOSA framework in the form of a conceptual cube. Because humans can only consider a finite quantity of information at a given time, models are divided into four mutually exclusive views along one of the axis directions: function, information, resource and organization. Along a second axis direction the cube is subdivided according to the types of models used (by the CIMOSA project) in system development: requirement, design, and implementation models. The third, model development, axis direction has the following subdivisions: base (generic), partial, and particular models. Now the cube makes sense as a framework or *reference model*, consisting of cells that can be "populated" by using partially populated cubes as inputs to the next population stage.

9.7.4 SEMATECH CIM APPLICATION FRAMEWORK SPECIFICATION

This specification [54] has been created by Sematech Inc., a U.S. industrial consortium. It is intended as an aid to the development of computer-integrated manufacturing (CIM) systems for use specifically in the semiconductor industry. It reflects the need of that industry to move awayfrom inflexible mainframe-based systems towards more flexible distributed systems using client-server architectures. The framework has been developed with a strongly object-oriented approach. It is partitioned into a number of *functional groups* dealing with different application areas:

- Factory services and common facilities
- Factory management
- Management of manufacturing specifications
- Labor management
- Plan management
- Schedule management
- Material management
- Machine control
- Process control

Four levels of information abstraction have been identified:

- Company-specific
- Semiconductor manufacturing
- Manufacturing
- Generic

The first and fourth are implementation-dependent, and the scope of the framework is therefore restricted to the second and third levels.

9.8 RECOMMENDATIONS

The techniques and tools identified in this chapter may not optimally support all the modeling needs of the MSE project. Therefore, as a first step towards meeting the project's modeling requirements, sufficient resources should be allocated to identify the most appropriate methods to use, and to acquire the associated tools. Further specific recommendations are given below:

1) The MSE project should adopt a software engineering methodology for the development of an optimal integration strategy. The strategy should be validated through system implementations.

2) Specifications created by the MSE project should state requirements for conformance to those specifications.

3) The MSE project should determine and acquire the most appropriate methods and tools for its purposes.

MSE staff will need to be trained in the use of the software tools, which should be refined if necessary to suit MSE project needs. Guidelines must be developed to ensure the uniform useof the tools within the project.

4) The MSE Project should promote a single reference architecture.

This architecture should include a scheme for identifying its component systems and their functionalities, a common universe of discourse, and a model for information exchange between pairs of subsystems and information publication by component subsystems.

5) The MSE project should be prepared to work with multiple engineering architectures for both short- and medium-term developments, provided they conform to the reference architecture.

The boundaries and groupings of software packages should be flexible, to reflect the different ranges of functionality of available packages. The project should accept several different integration mechanisms, provided the groupings and mechanisms are consistent with the reference architecture. It is also appropriate to support multiple implementations even of the same engineering architecture, as a way to test the robustness of that architecture with different component products.

6) The MSE project should identify a framework for system specification.

CHAPTER 10: SYSTEMS INTEGRATION TECHNOLOGIES

The previous chapter surveyed those aspects of software engineering that are relevant in building an integrated system from previously existing software components having diverse provisions for inter-component communication. The survey highlighted the important role of system modeling in the design of the internal interfaces of the integrated system.

The transition from a collection of system models to an actual system requires the specification of the mechanisms that instantiate the system interfaces, including the timing of the exchanges, the means of exchange, and the form of the exchange. These *integration mechanisms* are specified in the engineering architecture. When the implementation architecture calls out specific application packages to implement specific architectural elements, those packages will be required to exhibit exactly the interfaces specified. In this chapter, we discuss the exchange mechanisms that may be specified in the engineering architecture, the availability of software and standards supporting those mechanisms, and the implementation considerations that may be necessary to make application packages conform to a given set of interface specifications.

10.1 EXCHANGE MODELS

An exchange model characterizes the behavior of a particular class of information exchange between two (or more) application subsystems. These exchanges fall into two large categories:

- *Direct exchanges* involve the transmission of a message from a sending application to a specific receiving application, and usually the return of a response to that message.
- *Indirect exchanges* involve one application's provision of information in some accessible location, with a defined organization and format, and the subsequent retrieval of that information by another application.

Although these categories appear clear-cut as viewed by the applications, their actual implementation may involve (possibly multiple) physical mechanisms of different types. For example, logically direct exchanges can be performed using an *indirect* technique based on shared memory or even files, while almost all indirect exchanges require the applications to communicate *directly* with a repository service (for example via procedure calls, as described below).

10.1.1 THE PROCEDURE CALL

The *procedure call* is the established mechanism for direct communication between subsystems that act as components of a single executable system. A subsystem is here defined as a library or package, consisting of one or more subprograms (or procedures, or object classes), each of which performs a particular service. Another component—the *caller* or *client*—invokes the subsystem and provides the parameters of the requested service. The subsystem that receivedthe call then performs the service and returns parameters describing the results.

Some CAD systems provide a similar mechanism, based on the use of what is often called a *macro language*. These languages are invariably proprietary. The user writes the procedure for a particular subsystem (referred to in this case as a *macro*) in such a language, and it is then incorporated into the operational CAD system to provide additional specialized functionality. The CAD system invokes the macro (i.e. calls the procedure), according to rules defined by the user, at certain points in the CAD system processing. Although this mechanism allows the linking of a subsystem to the CAD system, macro languages usually do not provide for linking the CAD system to any external software package.

Some CAD and other application systems provide an *application programming interface* (API). This is a set of procedures, usually written in some standard programming language, callable from an external user-generated program. Such procedures may provide access to some of the functions of the application system, or provide access to the data it maintains, or both. Those APIs which provide access to the functions of the system permit direct exchange between the applications themselves via procedure call; those which only provide access to the data are implementing an indirect exchange via procedure call.

10.1.2 THE REQUEST/RESPONSE (OBJECT SERVER) MECHANISM

The *request/response* (or *object server*) mechanism is the standard paradigm for most direct exchanges. It generalizes the procedure call mechanism for independent programs, which may reside on different platforms. Each subsystem, when acting as server, offers a set of integrated services to other client subsystems.

An important characteristic of this mechanism is that every interchange begins with a client request and ends with a server response. It is also commonly required that the client program wait for the server to respond before engaging in further activity.

The Common Object Request Broker Architecture (CORBA) object server model and the OSI Open Distributed Processing (ODP) *remote call* model are both examples of this mechanism. Both are defined in 1994 standards, each specifying the client application programming interface (API) and a corresponding exchange protocol.

10.1.3 THE REQUEST BROKER/TRADER

The *broker/trader* model extends the object server model of direct exchange. In this model, the broker is a single point of contact that intercepts most of the client's service requests and transfers them to the appropriate server. The broker maintains a directory of servers and the services they offer, as well as two forms of the service request and response: the client-interface form and the server-specific form.

On receipt of the client's request, the broker determines which server will be used, converts therequest to the server-specific form, and passes it on to the server. It later converts the server's response to the client-interface form and returns it to the client.

CORBA implicitly assumes the existence of a broker, because at present it specifies no standard protocol for communicating with a server. Thus the broker, which performs the server-specific protocol conversion mapping, is required to achieve the common interface at the client. The ODP Trader, which performs the same functions, is an optional value-added service providing a standard "as-server" protocol for communicating with the client, and a standard "as-client" protocol for communicating with the server. Thus, the trader behaves like a server to the client, and vice versa.

10.1.4 THE QUEUED EXCHANGE

The *queued exchange* model strongly resembles the object server model, except in one important way: the client does not expect a prompt reply. Instead it expects a notification, possibly much later, that the requested service has been completed. The request is posted to a server queue (with information on release, priority, authorization, resource requirements, etc.), then processed at some later time according to algorithms defined by the server. Communication by electronic mail may be considered an example of this type of exchange mode; considerable time may elapse before a given message is responded to, and response time may depend on some priority system of the recipient.

10.1.5 THE BLACKBOARD

The *blackboard* is an indirect exchange mechanism providing a location (the blackboard itself) in which separate subsystems post announcements, requests, and replies, and are notified of the postings of other subsystems. The information on a blackboard can be persistent, but is subject to being overwritten by new postings.

A blackboard server differs from a request/response server in that it automatically notifies each client about other subsystem postings, whereas a conventional server must wait for the client to post a request.

Blackboard implementations often devolve into a set of client/server relationships where requests and replies are exchanged through the blackboard. In effect, they become direct exchanges implemented through a blackboard.

10.1.6 THE COMMON DATABASE

The *common database* is an indirect exchange model in which separate subsystems share information through a common repository of persistent information objects. The common repository has a formally described organization and a small set of selective retrieval and update operations, which are used by all parties for all accesses. Depending on the organization, these operations may apply to a single object or information unit, or to a large group of similarinformation units. If multiple subsystems use and modify many shared information units in separate operations, the common database is essential.

10.1.7 THE CONTAINER

A *container* is a collection of information objects—usually a large collection—that is always is exchanged in its entirety. The objects in a container may together be regarded as constituting a single object in its own right, at some higher level. A container is almost invariably generated initially by a single subsystem, and the performance of any service using it may use some or all of the information it contains. A stored process plan, an IGES drawing file, a bill of materials, and an NC program are all examples of containers.

Conceptually, a container is a *file* object. It may be maintained locally as a complex data structure on disk or in memory, designed to suit the internal workings of a particular subsystem, but it is actually exchanged as an object stream.

The Container is the most common indirect exchange model, as well as the most common exchange mechanism used by current engineering and production support software. The container model simplifies the exchange of information and represents a small evolutionary step in interchange capabilities for most existing systems.

10.1.8 THE CONTAINER BASE

The *container base* is an indirect exchange model which is a combination of the database and container mechanisms. It is a common repository providing shared simultaneous access to multiple systems. Most of the information is kept in containers and is accessible only as a unit, i.e. in the container form. The database component provides an index to the containers by technical content and typically also includes control information, such as data formats, versions, dates, etc., and possibly references to related containers.

The container base is primarily a means of managing, locating and relating containers, treated as objects in their own right. Most product data management (PDM) systems are really container bases, as are, for example, specialized databases containing stored NC programs or product model files.

10.2 APPLICABILITY OF EXCHANGE MODELS

Deciding which exchange mechanism to use depends on the nature of the interaction between subsystems, the quantity of data involved, and how the recipient subsystem uses the data.

If one subsystem specifically requests a service of another, some direct exchange model is required. In most cases, however, a large volume of data must be shared, and this requires that an indirect exchange mechanism be used in addition to the direct exchanges.

As mentioned in Section 10.1.6, the use and modification by multiple subsystems of large quantities of low-level data dictates the use of a common database. However, if a subsystem fetches *collections* of information objects and converts them to its own internal form at the beginning of processing, the container model is preferable.

10.2.1 ENGINEERING (DESIGN, ANALYSIS AND PROCESS PLANNING)

Many current design and manufacturing engineering systems use the container mechanism for integration. In the next phase of engineering systems integration, many of these systems should become servers to a central human-interface process (and in some cases to each other) using request/response interfaces supported by containers or databases. Subsystems such as design advisory systems, which actually contribute to the creation and modification of a common design will in most cases require access to a common design database. A similar approach is needed where multiple subsystems are interacting to develop a single process plan. On the other hand, interchanges between the design and engineering analysis systems, or between design and manufacturing engineering systems, involve communication and processing of the entire design. These interchanges can use containers. Since containers are routinely used to store archival information, maintenance of the individual containers in a container base is an important contribution to engineering information management.

10.2.2 PRODUCTION

Interfaces among scheduling and control systems on the shop floor require either a direct exchange model or a blackboard. Maintenance of production information (on jobs, tools, workpiece tracking, resource information, and schedules) requires common databases. Plans and control programs may be exchanged as containers, and might profitably utilize a container base, although they are managed internally by both the producer and consumer systems as complex data structures.

10.2.3 ENTERPRISE (HORIZONTAL) INTEGRATION

The term *horizontal integration* is used here to describe linkages between engineering, production, and other sectors of a single enterprise (sales and marketing, procurement and inventory, delivery, financial systems, etc.). Here the queued exchange model is most likely to be successful. Most of the exchanged information will probably be in containers, processed asynchronously according to organization-dependent rules for version release, authorization, priority, etc.

10.2.4 MULTI-ENTERPRISE (VERTICAL) INTEGRATION

The term *vertical integration* is used here to describe linkages between a manufacturer's systems and those of its suppliers. Here again the queued exchange model is most likely to be successful, although direct exchanges may occur in the future. Most of the information exchanged will probably be in containers, because direct access to the databases of other companies is fraughtwith legal and competitive concerns.

10.3 SOFTWARE SUPPORT FOR THE EXCHANGE MECHANISMS

The maturity of the exchange mechanisms specified above, in terms of standardization and software support, is by no means uniform. In selecting a mechanism for a particular interface it is important to consider the relative availability of compatible implementations.

10.3.1 THE PROCEDURE CALL

The procedure call is supported by nearly all standard programming languages. It is therefore a readily available mechanism when new software packages are being developed. On the other hand, only a limited number of product realization software packages provide a procedure call interface allowing access to the functionality of the package. One exception is in engineering analysis packages, which are often implemented as libraries of callable procedures rather than as stand-alone packages. Increasing many CAD packages now also provide an API, though in some cases this only provides access to the system's information base rather than to the system functions.

As mentioned earlier, many CAD packages have a macro facility allowing software written in a *non-standard* language—the *macro language* of that CAD system—to be integrated via a procedure call mechanism, but this mechanism provides no means of accessing the functionality of any external package, or of making CAD system functionality available to an external package..

10.3.2 THE REQUEST/RESPONSE (OBJECT SERVER) MECHANISM

There are a number of standard request/response protocols. The X/Open Distributed Communications Environment (DCE) Remote Procedure Call and the Common Object Request Broker Architecture (CORBA) are the most commonly implemented general-purpose protocols, but they are not currently used by any off-the-shelf manufacturing software. These implementations provide libraries which may be used by new software or new generations of existing packages for direct communication.

The ISO Manufacturing Message Specification (MMS) is also a standard request/response protocol, specifically targeted to production control systems, although it is also used for somewhat more general purposes. This protocol, and the derivative SECS II, is commonly supported by existing machine controllers and other production software systems.

10.3.3 THE REQUEST BROKER/TRADER

Many *directory servers* can be acquired off-the-shelf; they employ various standard directory service protocols, such as the CORBA Object Request Broker interface, the ISO Trader interface,CCITT X.500 directory services, etc. The use of these in the integration of manufacturing software systems is important in three areas:

- Access to suppliers' order entry services and related vertical integration exchanges,
- Access to external advisory, analysis and evaluation services, and
- Access to external information bases, such as materials data and tooling catalogs.

In each of these cases, the access is to software services out of the control of the overall manufacturing enterprise.

10.3.4 THE QUEUED EXCHANGE

Standards for electronic mail may be relevant here; these include Internet SMTP, MIME and CCITT X.400. Some operating systems also provide queued messaging services.

10.3.5 THE BLACKBOARD

There are no existing standards, though some current application products use blackboard functionality internally.

10.3.6 THE COMMON DATABASE

General-purpose database management systems designed for simultaneous access by multiple software applications residing on physically remote platforms are commonly available. There are two basic database technologies in common use: *relational* and *object-oriented*.

Relational database technology is mature and supported by international standards for both organization and access. Relational systems are optimized for finding the relevant set of objects for a particular purpose among many objects of the same structure, and for handling multiple views of the interrelationships among objects. Many existing production planning packages, some scheduling systems and some process planning packages use relational databases for a large portion of their input and output information. Unfortunately, there are no standards for how such information should be organized, so each package defines its own organization, and most packages use vendor-specific interfaces to the database system (rather than the standard interfaces), in order to improve performance. As a consequence, when two systems which need to communicate both use off-the-shelf relational database systems, they still differ in which systems they are able to use and how they expect the information to be organized. The *external view* mechanism provided by relational database systems can overcome the organization problem, but only when the representation of the information is consistent. Thus the use of a relational database inside an application package as the common database for a system is complicated by the frequently-encountered vendor philosophy that _other packages can use *my* information." The essential problem is that there are no standards for the information content and organization of product realization data in a relational database.

Object-oriented database technology is not mature, in that there are no existing organization or access standards, and most of the general-purpose commercial products have limited field experience and few supporting tools. Object-oriented and navigational systems are optimized for interrelating objects with different or variable structures and for finding all objects closely related to a given object, but all of this assumes that all the applications share a common view of the objects and their interrelationships. Object-oriented systems are needed for maintaining the complex information structures of a product model, and may also be conveniently used for representing process plans and some resource information. Several newer CAD and CAPP systems are internally linked to a particular commercial OODBMS and most older ones have some equivalent home-grown navigational system. Since there are no standards for either organization or access, sharing of information contained in these systems is restricted to new software that uses the API provided by the system vendor.

10.3.7 THE CONTAINER

The principal container implementation is the file, as supported by the operating system. File technology is mature and supported by de facto standards for representation and access on many media. There are several commonly supported standards for remote file access, allowing for easy exchange via container in a distributed environment.

In addition, existing formal and de facto standards for the exchange of manufacturing information define standard file formats such as IGES, DXF and STEP. These formats are commonly supported by many off-the-shelf systems for design, engineering analysis, process planning, and production control. Thus, of all the exchange methods, the container mechanism is the only mature one.

10.3.8 THE CONTAINER BASE

Container base technology is not mature in the sense of being supported by standards, although the mechanisms are well known. It combines the advantages of the highly mature database and container technologies. Many home-grown industrial systems, and a rapidly growing number of commercial tools, implement container bases as the means of information exchange.

Most Product Data Management systems and CAM database systems are container base implementations. These systems, however, typically define their own information content and organization and their own interfaces, so that integration via such a system is product-dependent rather than based on standard approaches. In most cases, the database component of the container base is a commercial DBMS, which may be either relational or object-oriented. In the relational case, standardized interfaces to the underlying DBMS are possible, but the problem mentioned in Section 10.3.6, that there are no standards for information content and organization, still applies.

10.4 COMMUNICATIONS TECHNOLOGIES

In any truly distributed system, the separate application packages may communicate using any of several techniques: interprocess communication within a platform, point-to-point serial communications links, local area networks (LANs), or even wide area networks (WANs).

Although new physical communications technologies emerge frequently, the software technology seen by the application is highly mature and readily supported by off-the-shelf products conforming to formal or *de facto* standards. The TCP/IP standards provide a common appearance and reliable network services for application-to-application and application-to-service links on most platforms. They are commonly implemented on all standard LAN technologies, and supported by gateways for both serial links and WANs. These protocols are commonly supported by file servers, database systems and implementations of request/response protocols.

TCP/IP is also commonly supported on many platforms as a local interprocess communication mechanism, although Windows OLE is emerging as a competing standard.

Factory floor communications use many standard serial link protocols and many standard messaging protocols. Integrating factory floor systems with anything else, therefore, is one of the few areas in which the communication technologies might be a barrier.

10.5 INTEGRATING EEXISTING PACKAGES

In order to integrate existing software packages into a target distributed system, it is necessary for the package to provide the interfaces expected in the distributed system. The following methods are available for to solve such problems. The survey of industry-developed integrated systems given in Chapter 6 provides examples of several of them.

- *"Plug and play"*. If the package already provides exactly the interfaces expected, only configuration should be needed in order to integrate it. This entails the provision of means, in both the package and elsewhere in the distributed system, for controlling the interaction of the package with the rest of the integrated system.
- *Modify the application*. By modifying the application programs themselves, any necessary interface can be constructed. But this is only possible if one has access to the source code for the system. In most cases, this translates to having the system vendor perform the modifications. In the long run, this is the most desirable mechanism, but it is only desirable from the point of view of the vendor if it improves the overall market for the software product, which usually implies that the desired interfaces should be standardized.
- *Extend the application information base*. Some application packages allow users to develop their own additions to the information base, i.e., to define new objects or newattributes of existing objects, or both. This allows the application to be extended to include additional information in some interface it already supports, but not to provide new interfaces or change the form of an interface.
- *Extend the application functions*. Some application packages allow users to develop their own additions to the functionality of the package, by adding new routines to be invoked before/after some existing function or on command from the user. These additions are usually phrased in some package-defined macro language, which greatly limits what functions can be defined. This allows some component functions to be embedded directly in the application, and there is actually a small market for software macros to perform certain common analyses and derivations for a particular CAD system. It also allows information to be derived or converted, but it rarely provides a means of creating a new interface or changing the form of an existing one.
- *Modify or extend the user interface*. Some application packages allow users to develop their own additions and modifications to the user interface. This allows the user to give commands to perform some added function or provide some added information, but not to influence the interface to other packages.
- *Pre- or post-process files and databases*. If an application produces an output file containing the necessary information, but in the wrong format for input to another module via a standard interface, a post-processor can be developed to convert the output file to the format desired for a standard interface. Similarly, a pre-processor can be written to convert an input file from the standard format to the format the application package expects. In general, the application input or output could be a database instead of a data file if the application provides an applications programming interface (API) to its information base, or at least documents the format used. Similarly, the standard interface could be to a database, rather than a file.
- *Develop a wrapper for a human-interactive application*. Using pre- and post-processors, and possibly other mechanisms above, embed the off-the-shelf application in a software envelope that exhibits the standard interface. When the application is essentially human-interactive, the invocation of the pre- or post-processor may be added to the user interface, as indicated above.
- *Develop a wrapper for an automatic application*. This is an extension of the last approach. In order to wrap an essentially automatic application, it is additionally necessary for the wrapper software to handle the request/response messages, or provide a blackboard interface, whichever is expected. In some cases, this may also require dynamically invoking/executing the application package itself. Such software tends to be as much platform-dependent as its is application-dependent.
- *Convert a human-interactive package to an automatic application*. This is analogous to the development of a wrapper for an automatic application, except that the wrapper mustreroute the human-interface elements of the package to some medium in which the wrapper software can intervene. The wrapper must simulate the human commands to match the request messages and deliver the package responses intended for the human agent as response messages to the remote requestor. For many systems, such a simulation is simply not possible, but for simulation systems and engineering analysis systems with crude human interfaces, this is sometimes a viable technique.

10.6 TIME-FRAME FOR INTEGRATION STRATEGIES

The extent to which manufacturing software can be integrated depends on when the integration is expected to be complete. It is possible to modify existing software in 18 to 24 months. Developing entirely new software packages could require 3 to 5 years, and the acceptance of a radically new approach in the integrated system requires longer still. The choice of integration strategies therefore depends in part on the expected date of availability. The SIMA program has to make a similar choice in that it must target the appropriate complexity of integration approaches given the duration of the program. System capabilities and performance requirements must be determined with careful consideration of the time available for the evolution of technology and standards. The following sections discuss additional constraints on systems integration in terms of development time.

10.6.1 SHORT-TERM INTEGRATION ACTIVITIES (0 TO 2 YEARS)

In the short term, existing subsystems can be modified to improve their interfaces with others. Such modifications cannot significantly alter either the way the package works internally or the way in which it is used. In most cases, short-term integration operations must be manually controlled to some extent, because the subsystems are often built to interact with an operator. Most integration activities consist of converting input information sources and formats to the files and formats expected by the particular subsystem, and converting output files to the form used in the shared repository. The shared form may be a container or a database, as appropriate. This may involve pre- and post-processing the information, and extending the application or the human-interface by macros, as described above.

10.6.2 MEDIUM-TERM INTEGRATION ACTIVITIES (3 TO 5 YEARS)

For integration activities in the 3 to 5 year time-frame, it is reasonable to expect that software packages can be developed or modified significantly to use any available sharing technology, so long as it does not significantly alter the way in which the package is *used,* that is, automatically or interactively. In the medium term, however, it is possible to change the way the package *works*.

In the medium term, totally automated packages, and those that interact with a user only to set up initial parameters, could become servers, using some de facto standard protocols and interface specifications. In other words, the service provided by the package could be automaticallyinvoked and executed, and the results automatically returned. Many of the inputs and outputs will continue to be files, and only some of their formats will be standardized. Thus the integrated form of the server will be a achieved by using a wrapper that performs the file conversions developed in the short term. On receipt of a request, the wrapper would perform the input file conversions, invoke the package service, perform the output file conversions, and then return the response.

Highly interactive packages ideally could become queued servers in the medium term. For them, falling back to manually controlled integration would not be a significant difference. The requirement to schedule a human operator to work with the system prevents any major improvement in automated integration, but continuing improvement in indirect information sharing is important.

On the other hand, software may be developed in the medium-term future that is capable of performing some functions totally automatically that now require interaction. It will take time for them to be accepted, however.

10.6.3 LONG-TERM ACTIVITIES (MORE THAN 5 YEARS)

Integration activities that are only minimally constrained by time have the opportunity to explore many more approaches to manufacturing systems integration. For the long term, it is possible to explore:

- The use of integration and manufacturing technologies not yet completely developed
- The use of artificial intelligence techniques to automate processes currently believed to require interactive human judgment
- Optimal solutions to integrated manufacturing problems that are currently decomposed to produce feasible approximate solutions
- Standardization of frameworks: manufacturing architectures, software components, exchanges and information models on a large scale
- Ways to convince vendors of manufacturing software packages that provision of facilities to enable integration is in their own interest
- Provision of services currently made difficult by the manufacturing culture, market competition, legal concerns, etc.

10.7 RECOMMENDATIONS

1) The MSE project should target short-term and medium-term development activities.

For short-term integration, emphasis should be on developing de facto standard forms and constructing the necessary wrappers and interchange mechanisms. For medium-term integration, the project should concentrate on developing servers that provide totally automated functions to support client systems concerned with design, engineering, planning, and control.

2) Towards the end of the project, an activity with a low level of effort should be initiated to monitor emerging possibilities for more efficient and advanced integration techniques in the longer term.

The output of the latter activity should be recommendations for post-SIMA projects in selected high-risk/high-payoff technology areas.

3) The primary integration mechanisms should be Request/Response, Common Databases, Containers, and Container bases.

The project needs to give careful consideration to choosing indirect communications mechanisms (databases, containers and indices), because most information will be exchanged via these mechanisms.

4) The MSE project should consider establishing a clearinghouse for the manufacturing community to post system integration requirements, concerns and issues.

This could enable the project to set up a matrix relating integration problems for manufacturing software systems with specifications and standards supporting solutions to those problems.

CHAPTER 11: STANDARDS RELATED TO INFORMATION TECHNOLOGY

This chapter is concerned with standards related to information technologies applied to manufacturing. Standards related to manufacturing applications are discussed in Chapter 8. Also, see Chapter 8 for a discussion of the role of standards in the MSE project.

Information technology provides tools for achieving systems integration. However, MSE resources should primarily be devoted to developing standards related to manufacturing, not the underlying information technology. This general policy might be suspended if development of a particular information technology standard is important to achieving a particular MSE goal. This chapter identifies relevant information technology standards and presents recommendations on the adoption of specific standards.

11.1 TECHNICAL AREAS OF STANDARDIZATION

A widely accepted taxonomy of information technology standards is provided by the subject classification contained in IEEE Standard 1175, "Reference Model for Computing System Tool Interconnections." Nine areas of standards are identified:

- *Communication systems:* communication protocols (e.g., MMS, RPC, OSI and TCP/IP), network services (e.g., Internet), and application services for intercomputer communication. Some communication system standards are of central interest to the MSE project; others (for example in the area of network services) are only of peripheral interest.
- *Data exchange formats and standards:* file formats, character sets, and encoding schemes, and product data exchange standards such as STEP, are key enablers for sharing and exchanging information between applications.
- *Description exchange formats:* formats for communicating data models (including EXPRESS, IDEF1X and ASN.1).
- *Database systems*: database languages, programming interfaces (e.g., SDAI), and query languages. SQL is the international standard language for formulation of relational database organization and for access to information so organized. OQL is a trial-use standard of the OMG/ODMG consortium for access to object-oriented databases. RDA is a standard for remote database access.
- *Document exchange formats*: document markup languages, interchange formats specific to documents (such as SGML).
- *Human interface systems*: windowing systems and user interfaces.
- *Programming language systems*: general-purpose computer languages.

Software platforms relevant to tool interconnections (operating systems): environments for software operations, such as tools, architectures, and operating system interfaces. (POSIX is an international standard for operating system service interfaces and facilities. Among these facilities are standard means for invoking automated processes and binding them to specific input and output files and streams.)

- *Support view of software tools*: methodologies and tools that support the development, design, quality control, and management of software (that is, software viewed in its support role, rather than as software *per se*).

The European classification of standards for advanced manufacturing technology (see Section 8.2) categorizes IT standards in a different way. As in the IEEE classification, nine subcategories are defined (under class M3), but these differ substantially from those listed above. Since the IEEE standard is widely accepted both in the United States and elsewhere, it is a more appropriate reference for the MSE project.

- *Object technology:* is a collection of services (e.g., events, naming, externalization, persistence, transications, and queries) with object interfaces that provide basic functions needed by most or all applications that support an object request broker architecture. Object technology will be a key enabler for integration of new and legacy capability across applications.

11.2 RECOMMENDATIONS FOR SIMA INVOLVEMENT IN IT STANDARDS ACTIVITIES

Most of the decisions on adopting specific information processing standards should be deferred until the conclusion of a requirements analysis for the work of the MSE project. Two specific recommendations are however made at this stage:

1) MSE staff should attend one or two meetings of the ISO/IEC 10746 (Open Distributed Processing) group per year, with further participation determined by a subsequent evaluation.

This is potentially one of the highest-leverage communications standards activities for SIMA. Distributed processing can be highly application-specific, and open systems standards are important to gaining broad acceptance of SIMA results. It is much more likely that the specific needs of manufacturing applications will be addressed by this activity if the MSE project is involved.

2) The MSE project (and the SIMA projects as a whole) should adopt IEEE 1175, particularly the suite of software development standards (Support View of Software Tools).

This should be preceded by a survey of supporting commercial tools, and acquisition of those most appropriate for MSE purposes. The adoption of a consistent set of standards in the area of software development will greatly improve the productivity of MSE project efforts, and the IEEE suite of standards fills this need very well.

By their nature, information technology standards are not specific to one manufacturing domain or another, so adoption of other standards should be done primarily on an MSE-wide level. Decisions on involvement with standards-setting activities in the area of information technologies should conform to the criteria for standards involvement recommended in Chapter 8.

REFERENCES

[1] Bloom, H.M., Technical Program Description Systems Integration for Manufacturing Applications (SIMA), NISTIR 5476, NIST, Gaithersburg, MD, 1994.

[2] Stewart, S.L. and Pinholster, G., "Systems Integration Needs of U.S. Manufacturers, Conference Report," Journal of Research of the NIST, **99**(5), September-October 1994.

[3] 1992 CAD/CAM, CAE Requirements: Survey of Industry Experts and Leading Users, D. H. Brown Associates, Inc., Tarrytown, NY, February 1990.

[4] World Discrete Manufacturing Software Markets, Frost and Sullivan, Inc., Mountain View, CA, 1993.

[5] "Majority of IEs Think Graphics/Animation Very Important," a survey result in Industrial Engineering, July 1992, p. 43.

[6] Law, A. and McComas, M. "How to Select Simulation Software for Manufacturing Applications," Industrial Engineering, July 1992, pp. 29-35.

[7] Law, A. "Selecting Simulation Software for Manufacturing Applications: Practical Guidelines and Software Survey," Industrial Engineering, May 1989, pp. 34-46.

[8] "Simulation Software Buyer's Guide," Industrial Engineering, July 1993, pp. 46-57.

[9] Norman, V. "Future Directions in Manufacturing Simulation," Industrial Engineering, July 1992, pp. 36-37. may not represent the latest state of the art.

[10] Ullman, D.G., "Issues Critical to the Development of Design History, Design Rationale and Design Intent Systems," in Proc. 1994 ASME Design Theory & Methodology Conf., Minneapolis, MN, September 1994 (ASME Publication DE-Vol 68, pp. 249-258, 1994).

[11] Pahl, G. and Beitz, W., Engineering Design, (ed. K. Wallace), The Design Council (London) and Springer-Verlag, 1984.

[12] Tworzydlo, W.W. and Oden, J.T., "Towards an Automated Environment in Computational Mechanics," in Computer Methods in Applied Mechanics & Engineering **104**(1):87-143, 1993.

[13] El-Zafrany, A., Techniques of the Boundary Element Method, Ellis Horwood, New York, 1993.

[14] Cutkosky, M.R. and Tenenbaum, J.M., "CAD/CAM integration through concurrent process and product design," in Intelligent and Integrated Manufacturing Analysis andSynthesis, American Society of Mechanical Engineers, New York, 1987.

[15] Cutkosky, M.R. and Tenenbaum, J.M., "Providing computational support for concurrent engineering," Int. J. Systems Automation: Research and Applications **1**, 239-261, 1991.

[16] Shah, J.J., "Assessment of features technology," in Computer Aided Design **23**(5):331-343, 1991.

[17] Pratt, M.J., "Aspects of form feature modelling," in Geometric Modelling: Methods and Applications, H. Hagen and D. Roller (eds.), Springer-Verlag, 1991.

[18] Hoschek, J. and Dankwort, W. (eds.), Parametric and Variational Design, Teubner-Verlag, Stuttgart, 1994.

[19] Hopp, T.H., "The Language of Tolerances," in Quality through Engineering Design, W. Kuo (ed.), Elsevier, 1993, 317-332.

[20] Voelcker, H.B., "A current perspective on tolerancing and metrology," Manufacturing Review **6**(4):258-268, 1993.

[21] Larijani, L. C., The Virtual Reality Primer, McGraw-Hill 1993.

[22] Boothroyd, G., "Product Design for Manufacture and Assembly," Computer Aided Design **26**(7):505-520, 1994.

[23] Finger, S. and Dixon, J.R., "A review of research in mechanical engineering design, Part 2: Representations, analysis, and design for the life cycle," Research in Engineering Design 1, 121-137, 1989.

[24] Ashton, J.E., "Managing design for continuous improvement in a system job shop," Manufacturing Review **5**(3), September 1992.

[25] Hoover, C.W. and Jones, J.B., Improving engineering design: Designing for competitive advantage, National Academy Press, Washington, D.C., 1991.

[26] Fowler, J.E., Variant Design for Mechanical Artifacts — A State of the Art Survey, NISTIR 5356, National Institute of Standards and Technology, Gaithersburg, MD, February 1994.

[27] Sodhi, R. and Turner, J.U., "Towards Modelling of Assemblies for Product Design," Computer Aided Design **26**(2):85-97, 1994.

[28] Finger, S. and Dixon, J.R., "A review of research in mechanical engineering design, Part 1: Descriptive, prescriptive, and computer-based models of design processes," Researchin Engineering Design 1, 51-67, 1989.

[29] Singpurwalla, N., A Bayesian Perspective on "Taguchi's Approach" to Quality Engineering and Tolerance Design, Technical Report 91-2, George Washington University, Washington, DC 1991.

[30] Chang, T.C. and Wysk, R.A., An Introduction to Automated Process Planning Systems, Prentice-Hall, 1985.

[31] Eversheim, W. and Schneewind, J., "Computer-aided process planning: State of the art and future development," Robotics and Computer-Integrated Manufacturing, **10**(1/2):65-70, 1993.

[32] Chang, T.C., Expert Process Planning for Manufacturing, Addison Wesley, 1990.

[33] Hoffmann, C.R., Geometric and Solid Modeling, Morgan Kaufmann, San Mateo, CA, 1989.

[34] Bronsvoort, W.F. and Jansen, F.W., "Feature modelling and conversion—key concepts for concurrent engineering," Computers in Industry **21**, 61-86, 1993.

[35] Shah, J.J., Hsiao, D. and Leonard, J., "A systematic approach for design-manufacturing feature mapping," in Geometric Modeling for Product Realization, P.R. Wilson, M.J. Wozny and M.J. Pratt (eds.), North-Holland, 1992.

[36] Iannuzzi, M. and Sandgren, E., "Optimal tolerancing: The link between design and manufacturing productivity," in Proc. 1994 ASME Design Theory & Methodology Conf., Minneapolis, MN, September 1994 (ASME Publication DE-Vol 68, 1994).

[37] Gilman, C.R., Dong, X. and Wozny, M.J., "Feature-based fixture design and set-up planning," in Artificial Intelligence in Optimal Design and Manufacturing, Z. Dong (ed.), Prentice-Hall, 1993.

[38] Boerma, J.A. and Kals, H.J.J., "Fixture design with FIXES: The automatic selection of positioning, clamping and support features for prismatic parts," Annals of the CIRP **37**(1):443-446, 1989.

[39] Wozny, M.J., Pratt and C. Poli, "Topics in feature-based design and manufacturing," in Advances in Feature Based Manufacturing, J.J. Shah, M. Mantyla and D.S. Nau (eds.), Elsevier Science B.V. 1994.

[40] Ray, S.R., "Using the ALPS process plan model," in Proc. 1992 ASME Manufacturing International Conf., D. Durham (ed.), (ASME publication G00657-1992).

[41] Algeo, M.E.A., A state-of-the-art survey of methodologies for representing manufacturing process capabilities, NISTIR 5391, National Institute of Standards and Technology, Gaithersburg, MD, March 1994.

[42] Cook, D.J., "Metrics for the evaluation of planning systems," Proc. Workshop on Automated Factory of the Future, at 1994 IEEE International Conf. on Robotics & Automation, San Diego, CA, May 1994.

[43] Spyridi, A.J. and Requicha, A.A.G., "Automatic planning for dimensional inspection," Manufacturing Review **6**(4):314-319, 1993.

[44] Homem de Mello, L.S. and Lee, S. (eds.), Computer-Aided Mechanical Assembly Planning, Kluwer, Boston, 1991.

[45] Juran, J.M., ed., Quality Control Handbook, 3rd edition, McGraw Hill, New York, 1979.

[46] Vasseur, H., Cagan, J., and Kurfess, T., "Economic Analysis of Quality Innovation," Manufacturing Review, 6(4):343-352, December, 1993.

[47] Harry, M. J., and Stewart, R., Six Sigma Mechanical Design Tolerancing, Motorola University Press, Schaumburg, IL, 1988.

[48] Swyt, D., Challenges to NIST in Dimensional Metrology: The Impact of Tightening Tolerances in the U.S. Discrete-Part Manufacturing Industry, NISTIR 4757, National Institute of Standards and Technology, Gaithersburg, MD, 1992.

[49] Jones A., Rabelo L., and Yih Y., "Job Shop Scheduling," in Encyclopedia of Operations Research, 1995 Edition, to appear.

[50] Swain. J., "Flexible Tools for Modeling," OR/MS Today, December 1993, 62-78.

[51] Grant, H., Jones, A., and Polito, J., "Technology for Enterprise Integration: The Role of Computer Simulation," in Proc. Conf. on New Directions in Simulation for Manufacturing and Communications, Tokyo, Japan, Aug. 1-3, 1994, 269-278.

[52] ISO, Product Data Representation and Exchange: Part 31: Conformance Testing Methodology and Framework: General Concepts, ISO Standard 10303-31, 1994.

[53] ESPRIT AMICE Consortium (eds.), CIMOSA: Open System Architecture for CIM, 2nd edition, Springer-Verlag, 1991.

[54] Sematech Inc., Collaborative Manufacturing System Computer Integrated Manufacturing (CIM) Application Framework Specification 1.0, Technology Transfer Report 93061697C-ENG, 31 March 1994, Sematech Inc.

Glossary of Acronyms

In cases where an acronym refers to a formal or *de facto* standard, or a proposed new standard under development, the name of the standards body or developing organization is given in brackets. This will aid in locating the corresponding entry in Appendix B.

AIS	Application Interface Specification (CAM-I)
ALPS	A Language for Process Specification (NIST -no id-)
ANSI	American National Standards Institute
AP	Application Protocol
API	Application Program Interface
APT	Automatically Programmed Tools (ANSI X3.37, ISO 4342)
ASME	American Society of Mechanical Engineers
ASN.1	Abstract Syntax Notation 1 (ISO/IEC 8825)
ASTM	American Society for Testing and Materials
B-rep	Boundary representation
BCL	Binary Cutter Language (EIA RS 494-B)
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CAM-I	Consortium for Advanced Manufacturing International, Inc.
CAPP	Computer-Aided Process Planning
CBEMA	Computer Business Equipment Manufacturers Association
CEN	European Committee for Standardization
CENELEC	European Committee for Electrotechnical Standardization
CFI	CAD Framework Initiative
CIM	Computer Integrated Manufacturing
CIMOSA	Computer Integrated Manufacturing Open System Architecture (EC)
CL	Cutter Location (EIA RS 274-D)
CMM	Coordinate Measuring Machine
CNC	Computer Numerical Control
CORBA	Common Object Request Broker Architecture (OMG)
CRADA	Cooperative Research and Development Agreement
CSG	Constructive Solid Geometry
DARPA	Defense Advanced Projects Research Agency
DBMS	Database Management System
DCE	Distributed Communications Environment (X/Open)
DFA	Design for Assembly
DFM	Design for Manufacture (or Manufacturability)
DFX	Design for X
DMIS	Dimensional Measurement Interface Specification (ANSI/CAM-I 101)
DoD	Department of Defense
DXF	Design eXchange Format (Autodesk Inc.)
EAR	Entity-Attribute Relationship
EC	Commission of the European Communities
ECMA	European Computer Manufacturers Association
EDI	Electronic Data Interchange
EDIF	Electronic Data Interchange Format (EIA EDIF)
EDM	Electro-Discharge Machining
EIA	Electronic Industries Association
ESPRIT	European Specific Program for Research & Development in Information Technology
ETSI	European Telecommunications Standards Institute
FDDI	Fiber Distributed Data Interfaces (ISO -no id-)
FE	Finite Element
FIPS	Federal Information Processing Standard
GT	Group Technology
HPCC	High Performance Computing and Communications
IDEF0	Integration Definition for Function Modeling (NIST FIPS 183)
IDEF1X	Integration Definition for Information Modeling (NIST FIPS 184)
IDL	Interface Definition Language (ECMA PCTE, ISO 13886 LIPC, OMG CORBA, X/Open DCE)
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers

IETF	Internet Engineering Task Force
IGES	Initial Graphics Exchange Specification (ASME Y14.26)
IITA	Information Infrastructure Technology Applications
IPL	Interpreted Predicate Logic
ISO	International Organization for Standardization
ITU	International Telecommunications Union
JHG	Joint Harmonization Group
KIF	Knowledge Interchange Format (DARPA)
KQML	Knowledge Query and Manipulation Language (DARPA)
LIPC	Language-Independent Procedure Call (ISO/IEC DIS 13886)
MEL	Manufacturing Engineering Laboratory
MIME	Multipurpose Internet Mail Extensions (IETF RFC 1341)
MMS	Manufacturing Message Specification (ISO/IEC 9506)
MRP I	Materials Requirement Planning
MRP II	Manufacturing Resource Planning
MSE	Manufacturing Systems Environment
NC	Numerically Controlled
NCMS	National Center for Manufacturing Sciences
NGC	Next Generation Controller
NIAM	Natural-language Information Analysis Method (formerly Nijssen's Information Analysis Method)
NII	National Information Infrastructure
NIST	National Institute of Standards and Technology
NURBS	Non-Uniform Rational B-Splines
ODMG	Object Data Management Group
ODP	Open Distributed Processing (ISO/IEC 10746)
OMG	Object Management Group
OODBMS	Object-Oriented Database Management System
OQL	Object Query Language (OMG ODML/OQL)
OSF	Open Systems Foundation
PCTE	Portable Common Tools Environment (ECMA 149)
PDM	Product Data Management
POSIX	Portable Operating System Interface (ISO/IEC 9945)
RDA	Remote Database Access (ISO/IEC 9579)
RDBMS	Relational Database Management System
RIDL	Reference and IDeas Language
RPC	Remote Procedure Call (OMG CORBA, OSF DCE RPC, Sun ONC RPC)
SDAI	Standard Data Access Interface (ISO 10303 - 22)
SECS II	SEMI Equipment Communications Standard (SEMI E5)
SEMI	Semiconductor Equipment and Materials International
SGML	Standard Generalized Markup Language (ISO 8879)
SIMA	Systems Integration for Manufacturing Applications
SMTP	Simple Mail Transport Protocol (IETF RFC 822)
SQL	Structured Query Language (ISO/IEC 9075)
STEP	STandard for the Exchange of Product model data (informal name of ISO 10303)
STL	File format for stereolithography applications
TCP/IP	Transport Control Protocol/Internetwork Protocol (DoD MIL-SPEC-1777/1778)
WBS	Work Breakdown Structure

APPENDIX A: STANDARDS BY SUBJECT

This Appendix lists the titles of standards by subject category. The category of Computing Technology (and its subcategories) are discussed in Chapter 11. All other categories are discussed in Chapter 8. Details about individual standards can be found in Appendix B, where standards are listed by organization and identification.

SUBJECT: COMPUTING TECHNOLOGY

IEEE 1175 (Title: *Reference Model for Computing System Tool Interconnections)*

SUBJECT: COMPUTING TECHNOLOGY/COMMUNICATION SYSTEMS

ISO 7498 (Title: *Information Processing Systems — Open Systems Interconnection)*

ISO 8072 (Title: *Information Processing Systems — Open Systems Interconnection — Transport Service Definition)*

ISO 8073 (Title: *Information Processing Systems — Open Systems Interconnection — Connection Oriented Transport Protocol Specification)*

ISO 8649 (Title: *Information Processing Systems — Open Systems Interconnection — Protocol Specification for Association Control Service Element — Service Definition for Association Control Service Element)*

ISO 8650 (Title: *Information Processing Systems — Open Systems Interconnection — Protocol Specification for Association Control Service Element)*

ISO/IEC 8802 - 3 (Title: *Information Technology — Local and Metropolitan Area Networks — Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications)*

ISO/IEC 8802 - 4 (Title: *Information Technology — Local Area Networks — Part 4: Token-passing bus access method and physical layer specifications)*

ISO/IEC 8802 - 5 (Title: *Information Technology — Local and Metropolitan Area Networks — Part 5: Token ring access method and physical layer specifications)*

SUBJECT: COMPUTING TECHNOLOGY/COMMUNICATION SYSTEMS/APPLICATION SERVICES

IETF -no id- (Title: *Telnet [Internet])*

IETF RFC xx (Title: *Network File System (NFS))*

57ISO 8751 (Title: *Information Processing Systems — Open Systems Interconnection — File Transfer, Access and Management (Parts 1 - 5))*

ISO 9074 (Title: *Information Processing Systems — Open Systems Interconnection — Estelle: a Formal Description Technique based on an Extended State Transition Model)*

ISO 9040/9041 (Title: *Information Technology — Open Systems Interconnection — Virtual Terminal Basic Class Service/Basic Class Protocol)*

ISO 11578 (Title: *Remote Procedure Call)*

ISO/IEC 9579 (Title: *Remote Database Access)*

ISO/IEC 10746 (Title: *Reference Model for Open Distributed Processing)*

ISO/IEC DIS 13886 (Title: *Information Technology — Language Independent Procedure Call)*

ISO/ITU X.500 (Title: *Directory Access)*

ITU X.901 (Title: *Reference Model for Open Distributed Processing)*

OMG CORBA (Title: *Remote Procedure Call)*

OSF DCE rpc (Title: *Remote Procedure Call)*

Sun ONC rpc (Title: *Remote Procedure Call)*

SUBJECT: COMPUTING TECHNOLOGY/COMMUNICATION SYSTEMS/NETWORK SERVICES

DoD MIL-SPEC-1777/MIL-SPEC 1778 (Title: TCP/IP)

IEEE 488 (Title: *Digital Interface for Programmable Instrumentation)*

IEEE 802.3 (Title: *Ethernet physical link)*

IEEE 802.4 (Title: *Token bus)*

IEEE 802.5 (Title: *Token Ring)*

IETF RFC 822 (Title: *Standard for the Format of ARPANET Internet Text Messages)*

IETF RFC 1341 (Title: *Multipurpose Internet Mail Extensions)*

ISO -no id- (Title: *FDDI)*

ISO -no id- (Title: *LDDI)*

ISO/IEC 10021 (Title: *Information Technology — Text Communication— Message-oriented Text Interchange Systems (MOTIS))*

ITU V.35 (Title: *T1 line)*

N/A Ethernet V2 (Title:)

SUBJECT: COMPUTING TECHNOLOGY/DATA FILE EXCHANGE FORMATS

ANSI/CBEMA X3.122 (Title: *Computer Graphics Metafile for the Storage and Transfer of Picture Description Information)*

ANSI/CBEMA X3.124 (Title: *Graphical Kernel System (GKS) Functional Description)*

IETF -no id- (Title: *External Data Representation [XDR])*

ISO 646 (Title: *Information Processing — 7 bit coded character set for information interchange)*

ISO 2022 (Title: *Information Processing — 8 bit single-byte coded graphic character sets)*

ISO 6937 (Title: *Information Processing — Coded character sets for text communication (parts 1 and 2))*

ISO 8859 (Title: *Information Processing — 8 bit single-byte coded character sets (parts 1-5))*

ISO 10303 - 21 (Title: *Implementation methods: Clear text encoding of the exchange structure)*

ISO/IEC 8825 (Title: *Information Technology — Open Systems Interconnection — Specification of Encoding Rules for Abstract Syntax Notation One (ASN.1))*

ISO/IEC 9592 (Title: *Information Processing Systems — Computer Graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS))*

SUBJECT: COMPUTING TECHNOLOGY/DATABASE SYSTEMS

ANSI/CBEMA X3.135 (Title: *Information Systems — Database Language — SQL with Integrity Enhancement)*

ANSI/CBEMA X3.138 (Title: *Information Systems — Information Resource Dictionary System (IRDS))*

DARPA -no id- (Title: *Knowledge Query and Manipulation Language (KQML))*

ISO 10303 - 22 (Title: *Standard Data Access Interface (SDAI))*

ISO/IEC 9075 (Title: *Information Technology — Database Language — SQL)*

OMG -no id- (Title: *ODML/OQL)*

SUBJECT: COMPUTING TECHNOLOGY/DESCRIPTION EXCHANGE FORMATS

DARPA -no id- (Title: *Knowledge Interchange Format (KIF))*

ISO 5806 (Title: *Information Processing — Specification of Single-Hit Decision Tables)*

ISO 5807 (Title: *Information Processing — Documentation Symbols and Conventions for Data, Program and System Flowcharts, Program Network Charts and System Resources Charts)*

ISO 8790 (Title: *Information Processing Systems — Computer System Configuration Diagram Symbols and Conventions)*

ISO 10303 - 11 (Title: *Descriptive methods: The EXPRESS language reference manual)*

ISO/IEC 8824 (Title: *Information Technology — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1))*

ISO/IEC 11179 (Title: *Data Element Attributes)*

NIST FIPS 183 (Title: *IDEF0)*

NIST FIPS 184 (Title: *IDEF1X)*

OMG -no id- (Title: *CORBA Object Model)*

SUBJECT: COMPUTING TECHNOLOGY/DOCUMENT EXCHANGE FORMATS

ISO 8613 (Title: *Information Processing — Text and Office Systems — Office Document Architecture (ODA) and Interchange Format (Parts 1-8))*

ISO 8879 (Title: *Information Processing — Text and Office Systems — Standard Generalized Markup Language (SGML))*

ISO/IEC 10744 (Title: *HyTime)*

SUBJECT: COMPUTING TECHNOLOGY/HUMAN INTERFACE SYSTEMS

X/Open Group -no id- (Title: *X-Window System)*

SUBJECT: COMPUTING TECHNOLOGY/PROGRAMMING LANGUAGE SYSTEMS

ANSI/CBEMA X3.9 (Title: *Programming Language FORTRAN)*

ANSI/CBEMA X3.23 (Title: *Programming Language COBOL)*

ANSI/CBEMA X3.159 (Title: *Programming Language — C)*

DoD MIL-STD-1815A (Title: *Reference Manual for the ADA Programming Language)*

IEEE X3.97 (Title: *Pascal Computer Programming Language)*

ISO 7185 (Title: *Pascal Computer Programming Language)*

ISO 9989 (Title: *Programming Language — C)*

SUBJECT: COMPUTING TECHNOLOGY/SOFTWARE PLATFORMS RELEVANT TO TOOL INTERCONNECTIONS (OPERATING SYSTEMS)

ECMA (European Computer Manufacturers Association) 149 (Title: *Portable Common Tool Environment)*

ISO/IEC 9945 (Title: *Information Technology — Portable Operating System Interface (POSIX))*

Microsoft -no id- (Title: *Windows)*

X/Open Group -no id- (Title: *X/Open Portability Guide)*

SUBJECT: COMPUTING TECHNOLOGY/SUPPORT VIEW OF SOFTWARE TOOLS

IEEE 730 (Title: *Standard for Software Quality Assurance Plans)*

IEEE 828 (Title: *Standard for Software Configuration Management Plans)*

IEEE 829 (Title: *Standard for Software Test Documentation)*

IEEE 830 (Title: *Guide for Software Requirements Specifications)*

IEEE 1012 (Title: *Standard for Software Verification and Validation Plans)*

IEEE 1016 (Title: *Recommended Practice for Software Design Descriptions)*

IEEE 1058.1 (Title: *Standard for Software Project Management Plans)*

IEEE 1074 (Title: Standard for Developing Software Life Cycle Processes)

SUBJECT: INDUSTRIAL PRACTICES

ISO 9000 (Title: *Quality Management and Quality Assurance Standards)*

ISO/TR 10314-1:1990 (Title: *Industrial Automation — Shop Floor Production — Reference Model for Standardization and a Methodology for Identification of Requirements)*

ISO/TR 10314-2:1990 (Title: *Industrial Automation — Shop Floor Production — Application of the Reference Model for Standardization and Methodology)*

SUBJECT: INDUSTRIAL PRACTICES/FRAMEWORKS

EC -no id- (Title: *CIM-OSA)*

ISO/TR 12186:1993 (Title: *Manufacturing Automation Programming Language Environment Overview (MAPLE))*

ISO/TR 13345:1994 (Title: *Industrial Automation Systems — Specification of Subsets of the Protocol for ISO/IEC 9506)*

NIST -no id- (Title: *Manufacturing Systems Integration)*

SUBJECT: INDUSTRIAL PRACTICES/INFORMATION EXCHANGE/MANUFACTURING MANAGEMENT DATA

ASME Board on Safety Codes and Standards -no id- (Title: *Management Control Systems)*

ISO 10303 - 49 (Title: *Integrated generic resources: Process structure, property and representation)*

ISO/IEC 9506-1:1990 (Title: *Industrial Automation Systems — Manufacturing Message Specification — Part 1: Service Definition)*

ISO/IEC 9506-2:1990 (Title: *Industrial Automation Systems — Manufacturing Message Specification — Part 2: Protocol Specification)*

NIST -no id- (Title: *ALPS - A Language for Process Specification)*

SEMI E5 (Title: *SEMI Equipment Communications Standard (SECS II))*

SUBJECT: INDUSTRIAL PRACTICES/INFORMATION EXCHANGE/PRODUCT DATA

ASME Y1 (Title: *Abbreviations)*

ASME Y10 (Title: *Letter Symbols)*

ASME Y14 (Title: *Engineering Drawing and Related Documentation Practices)*

ASME Y14.26 (Title: *Computer Aided Processing of Engineering Drawings and Related Documentation (IGES))*

ASME Y14.5-1994 (Title: *Dimensioning and Tolerancing)*

ASME Y14.5.1-1994 (Title: *Mathematical Definition of Dimensioning and Tolerancing Principles)*

ASME Y32 (Title: *Graphic Symbols and Designations)*

Autodesk DXF (Title: *Data Exchange Format)*

CAM-I AIS (Title: *Application Interface Specification)*

EIA EDIF (Electronic Industries Association) (Title: *Electronic Data Interchange Format)*

ISO 10303 - 1 (Title: *Overview and fundamental principles)*

ISO 10303 - 40 Series (Title: *Integrated generic resources)*

ISO 10303 - 100 Series (Title: *Integrated application resources)*

ISO 10303 - 200 Series (Title: *Application protocols)*

ISO 10303 - 1200 Series (Title: *Abstract test suites)*

ISO TC 3-10-57 (Title: *Joint Harmonization Group)*

SUBJECT: INDUSTRIAL PRACTICES/MANUFACTURING

ANSI/CAM-I 101 (Title: *Dimensional Measurement Interface Standard (DMIS))*

ANSI X3.37 - 1987 (R1993) (Title: *Programming Language APT)*

ASME B89 (Title: *Dimensional Metrology)*

ASME B89.3.2 (Title: *Dimensional Measurement Methods)*

EIA RS 274-D (Title: *Interchangeable Variable Block Data Format for Positioning, Contouring and Contouring/Positioning Numerically Controlled Machines)*

EIA RS 494-B (Title: *32 bit Binary CL Exchange (BCL) Input Format for Numerically Controlled Machines)*

ISO 4342 (Title: *Numerical Control of Machines — NC Processor Input — Basic Part Program Reference Language)*

ISO 6983-1 (Title: *Data Format for Positioning, Line Motion and Contouring Control Systems)*

ISO Handbook 3 (Title: *Statistical Methods)*

ISO Handbook 33 (Title: *Applied Metrology—Limits, fits, and surface properties)*

SUBJECT: INDUSTRIAL PRACTICES/MANUFACTURING/CALIBRATION AND PERFORMANCE TESTING

ASME PTC 19.1 (Title: *Measurement Uncertainty)*

ISO/IEC -no id- (Title: *Guide to the Expression of Measurement Uncertainty)*

SUBJECT: INDUSTRIAL PRACTICES/MANUFACTURING/DESIGN

ASME B4 (Title: *Limits and Fits)*

ASME B46 (Title: *Classification and Designation of Surface Qualities)*

SUBJECT: INDUSTRIAL PRACTICES/MANUFACTURING/PRODUCT STANDARDS

ASME B47 (Title: *Gage Blanks)*

ISO 13584 (Title: *Part Libraries)*

SUBJECT: MANUFACTURING EQUIPMENT

ASME B5 (Title: *Machine Tools — Components, Elements, Performance, and Equipment)*

ASME B89.4 (Title: *Coordinate Measuring Technology)*

ASME B94 (Title: C*utting Tools, Holders, Drivers, and Bushings)*

IEEE 416 (Title: *ATLAS Test Language)*

APPENDIX B: STANDARDS INFORMATION SUMMARY

This Appendix lists summary information about each standard identified in the MSE background study. The information listed for each standard or standards-setting activity is:

- *Designation:* the name of the organization and the code and title of the standard or standards activity
- *Subject Area:* the subject classification, as described in Chapters 8 and 11 and in Appendix A
- *Scope:* the technical scope of the standard or activity
- *Credentials:* whether the entry is company-specific, national, or international in nature
- *Status:* the status of issue of the standard
- *Comments:* comments by MSE project staff are listed as submitted during the study

Entries are listed alphabetically by designation. To find the listing for a named standard, please refer to the listing of acronyms and standards names for the proper designation.

Designation:	ANSI/CAM-I 101 (Title: *Dimensional Measurement Interface Standard (DMIS))*
Subject Area:	Industrial Practices/Manufacturing
Scope:	Machine control interface to coordinate inspection equipment.
Credentials:	ANSI accredited standards-making organization
Status:	Revision 2.1 1992
Comments:	Revision 3.0 ready but on hold pending patent litigation.

Designation:	ANSI/CBEMA X3.9 (Title: *Programming Language FORTRAN)*
Subject Area:	Computing Technology/Programming Language Systems
Scope:	Programming language
Credentials:	U.S. national; international
Status:	FORTRAN 9x; 1978; Reaff 1989
Comments:	Referenced from IEEE Std 1175. Also an ISO standard.

Organization: *Programming*	ANSI/CBEMA X3.23 (Title: *Language COBOL)*
Subject Area:	Computing Technology/Programming Language Systems
Scope:	Programming language
Credentials:	U.S. National (ANSI)
Status:	1985
Comments:	Referenced from IEEE Std 1175. Also an ISO standard. Some commercial PDM systems are reportedly wriiten in COBOL.

Organization:	ANSI/CBEMA X3.122 (Title: *Computer Graphics Metafile for the Storage and Transfer of Picture Description Information)*
Subject Area:	Computing Technology/Data File Exchange Formats
Scope:	Graphics exchange
Credentials:	U.S. National (ANSI)
Status:	1986
Comments:	Referenced from IEEE Std 1175. Not very good for raster images.

Organization:	ANSI/CBEMA X3.124 (Title: *Graphical Kernel System (GKS) Functional Description)*
Subject Area:	Computing Technology/Data File Exchange Formats
Scope:	
Credentials:	U.S. National (ANSI)
Status:	1985
Comments:	Referenced from IEEE Std 1175. Also an ISO standard: ISO 7942 for the 2D version, and ISO 8805 for GKS3D. Generally being superseded by PHIGS (see ISO/IEC 9592)

Organization: ANSI/CBEMA X3.135 (Title: *Information Systems — Database Language — SQL with Integrity Enhancement)*
See: ISO/IEC 9075

Organization: ANSI/CBEMA X3.138 (Title: *Information Systems — Information Resource Dictionary System (IRDS))*
Subject Area: Computing Technology/Database Systems
Credentials: U.S. National (ANSI)
Status: 1988
Comments: Referenced from IEEE Std 1175. There exists a competitive ISO standard. (Contact Dave Gradwell, U.K., for more info.]

Organization: ANSI/CBEMA X3.159 (Title: *Programming Language — C)*
See: See ISO/IEC 9989 (1990)

Designation: ANSI X3.37 - 1987 (R1993) (Title: *Programming Language APT)*
Subject Area: Industrial Practices/Manufacturing
Scope: High-level language for control of numerically controlled machine tools
Credentials: US national standard
Status: Revised 1993
Comments: See also ISO 4342:1985

Organization: ASME B4 (Title: *Limits and Fits)*
Subject Area: Industrial Practices/Manufacturing/Design
Scope: "Standardization of tolerances and corresponding symbols for rough and finished (except threaded) parts, mainly applicable to the establishment of preferred sizes, tolerances, and fits in limits and fits, together with the principles that should govern the inspection of these parts."
Credentials: U.S. National (ANSI)
Status: B4.1-Preferred Limits and Fits for Cylindrical Parts; B4.3-General Tolerances for Metric Dimensioned Products; B4.4-Inspection of Workpieces; Documents published in late 70's and early 80's
Comments: Important topic for harmonization across manufacturing applications; probably superceded by ISO work. See ISO/TC 3.

Organization: ASME B5 (Title: *Machine Tools— Components, Elements, Performance, and Equipment)*
Subject Area: Manufacturing Equipment
Scope: "The standardization of machine tools and of the elements of machine tool construction and operation relating primarily to their use on manufacturing operations, including work and tool holding elements, driving mechanisms that constitute an inherent part of the machine tool, components and associated appurtenances; nomenclature, designations, sizes, capacities, and tests for accuracy of machine tools and of work and tool holding parts or elements; movements and adjustments of machine tool elements; and parts and elements for adjusting, guiding, and aligning work or tools, including slots and tapes, but excluding perishable tools."
Credentials: U.S. National (ANSI)
Status: B5.51-Manufacturing Systems and Components; B5.52-Machining Centers; Very active; recently issued standards
Comments: These standards are key for process capability work.

Organization: ASME B46 (Title: *Classification and Designation of Surface Qualities)*
Subject Area: Industrial Practices/Manufacturing/Design
Scope: "Classification and designation of surfaces according to quality of surface."
Credentials: U.S. National (ANSI)

Status: Standards on measurement methods and designations; Active

Comments: B46 provides standards used in design, production, and inspection. Closely tied to ISO/TC 57. These standards are applicable to several SIMA projects.

Organization: ASME B47 (Title: *Gage Blanks)*

Subject Area: Industrial Practices/Manufacturing/ Product Standards

Scope: "To simplify gage design through the adoption of standard blanks and components for various common types of dimensional control gages, and to append with further related data."

Credentials: U.S. National (ANSI)

Status: B47 standard; Active

Comments: B47 provides a standard used in design, manufacture and test of gages. This standard may be applicable to several SIMA projects, mainly because of the "related data" part of the scope.

Organization: ASME B89 (Title: *Dimensional Metrology)*

Subject Area: Industrial Practices/Manufacturing

Scope: "Calibration and the specific conditions relating thereto. It shall encompass the inspection and the means of measuring the characteristics of the various geometrical configuratios such as lengths, plane surfaces, angles, circles, cylinders, cones, and spheres."

Credentials: U.S. National (ANSI)

Status: B89.3.2; In progress

Comments: Includes the following subcommittees: B89.1—Length; B89.2—Angles; B89.3—Geometry; B89.4—Coordinate Measuring Technology; B89.5—General Principles and Definitions; B89.6—Environment

Organization: ASME B89.3.2 (Title: *Dimensional Measurement Methods)*

Subject Area: Industrial Practices/Manufacturing

Scope: "This Standard formalizes the requirements for developing dimensional measurement plans which ensure compliance of workpieces with drawings conforming to ANSI Y14.5. Attributes and variables gaging are included. Compliance may be ensured either by process control or by final inspection."

Credentials: U.S. National (ANSI)

Status: In progress

Comments: This standard will lay out data element requirements, as well as technical considerations, for mechanical part inspection plans.

Organization: ASME B89.4 (Title: *Coordinate Measuring Technology)*

Subject Area: Manufacturing Equipment

Scope: All aspects of coordinate metrology. Technologies include CMMs, vision systems, theodolites, photogrammetry, etc. Issues include relationship to tolerancing standards and inspection methods, international harmonization, calibration requirements, performance measurement.

Credentials: U.S. National (ANSI)

Status: Various; Varied

Comments: This group provides the liaison to ISO/TC 3 and is heavily involved in ISO Joint Harmonization work.

Organization: ASME B94 (Title: *Cutting Tools, Holders, Drivers, and Bushings)*

Subject Area: Manufacturing Equipment

Scope: "The standardization of cutting tools, holders, drivers, bushings, punches, dies, and metal stamping tool accessories for use on or in conjunction with machine tools and/or allied equipment, including nomenclature, classification, sizes, marking, performance, testing, dimensions, and tolerances."

Credentials: U.S. National (ANSI)

Status: Various; Varied

Organization: ASME Board on Safety Codes and Standards -no id- (Title: *Management Control Systems)*

Subject Area: Industrial Practices

Scope: "Development of standard(s) which set forth requirements for the establishment and execution of a controlled management system. The standard(s) will describe a system which is flexible in application and suitable without the limits imposed by the size of an organization or the nature of the product or service."

Credentials: U.S. National (ANSI)

Organization: ASME PTC 19.1 (Title: *Measurement Uncertainty)*

Subject Area: Industrial Practices/Manufacturing/ Calibration and Performance Testing

Scope: "Specifies procedures for the evaluation of uncertainties in individual test measurements, arising from both random errors and fixed errors, and for the propagation of these errors into the uncertainty of a test result. Methods for treating spurious data points (outliers) and for determining the uncertainty of least squares regressions are also given."

Credentials: U.S. National (ANSI)

Comments: Important topic for all measurement applications. Probably superceded by ISO work. See ISO/IEC Guide to the Expression of Measurement Uncertainty.

Organization: ASME Y1 (Title: *Abbreviations)*

Subject Area: Industrial Practices

Scope: "Standardization of abbreviations on drawings and in text for the physical sciences and engineering."

Credentials: U.S. National (ANSI)

Status: Various; Varied

Organization: ASME Y10 (Title: *Letter Symbols)*

Subject Area: Industrial Practices

Scope: "Standardization of letter symbols and signs for equations and formulas."

Credentials: U.S. National (ANSI)

Status: Various; Varied

Organization: ASME Y14 (Title: *Engineering Drawing and Related Documentation Practices)*

Subject Area: Industrial Practices/Information Exchange/Product Data

Scope: "The development and maintenance of national standards for defining and documenting a product through out its life cycle and related certification activities. . . ."

Credentials: U.S. National (ANSI)

Status: Various; Varied

Comments: Key subcommittees include Y14.26 (Drawings and Related Documentation); Y14.5 (Dimensioning and Tolerancing); Y14.34 (Parts Lists, Data Lists, and Index Lists); Y14.35 (Drawing Revisions); and Y14.100 (Government/Industry Drawing Practices). The last is a liaison to MIL STD 100, the DoD standard for engineering drawings.

Organization: ASME Y14.26 (Title: *Computer Aided Processing of Engineering Drawings and Related Documentation (IGES))*

Subject Area: Industrial Practices/Information Exchange/Product Data

Scope: Electronic exchange of drawing data

Credentials: U.S. National (ANSI)

Status: Version 5.2

Comments: IGES functionality is also available in the first release of STEP (ISO 10303).

Organization: ASME Y14.5M-1994 (Title: *Dimensioning and Tolerancing)*

Subject Area: Industrial Practices/Information Exchange/Product Data

Scope: Representation of tolerance information on drawings

Credentials: U.S. National (ASME)

Status: Recent revision of ANSI Y14.5M-182 (R 1988)

Comments: Provides informational basis for ISO 10303-47. Closely related to ISO 1101.

Organization: ASME Y14.5.1M-1994 (Title: *Mathematical Definition of Dimensioning and Tolerancing Principles)*
Subject Area: Industrial Practices/Information Exchange/Product Data
Scope: Provides interpretation of Y14.5M tolerancing in mathematical terms
Credentials: U.S. National (ASME)
Status: Supporting document for Y14.5M, providing means for the consistent interpretation of tolerance data by automated applications
Comments: This is a difficult topic, and some important questions remain to be answered in future versions of this document.

Organization: ASME Y32 (Title: *Graphic Symbols and Designations)*
Subject Area: Industrial Practices
Scope: "To standardize graphic symbols and related designations used for communications in engineering disciplines and with the public. To coordinate and establish standards for graphic symbols, reference designations, device function designations except for switch-gear device function designations covered by ANSI C37, and terminal markings unless they are covered by special product standards."
Credentials: U.S. National (ANSI)
Status: Various; Varied
Comments: Includes subcommittees for railroad use, fluid power diagrams, and mechanical and acoustical elements as used in schematic diagrams.

Organization: Autodesk, Inc. (Title: *Data Exchange Format)*
Subject Area: Industrial Practices/Information Exchange/Product Data
Scope: A competitor, mainly of IGES (ASME Y14.26) for the electronic transfer of drawing and other product modeling data.
Status: De facto standard
Comments: Widely used, especially for transfer of data between PC-based CAD systems.

Organization: CAM-I AIS (Title: *Application Interface Specification)*
Subject Area: Industrial Practices/Information Exchange/Product Data
Scope: An application programming interface (API) for CAD systems with solid modeling capabilities. It provides not only query facilities but also full access to the geometric construction operations implemented by the modeler.
Credentials: U.S. National (ANSI)
Status: Draft standard for trial use

Organization: DARPA Knowledge Sharing Initiative InterlinguaWorking Group (Title: *Knowledge Interchange Format, Version 3.0)*
Subject Area: Computing Technology/Description Exchange Formats
Scope: Proposed standard knowledge interchange format
Credentials: Under development; may form the basis of a future standard
Status: 1992

Organization: DARPA Knowledge Sharing Initiative External Interfaces Working Group (Title: *Specification of the KQML Agent Communication Language)*
Subject Area: Computing Technology/Database Systems
Scope: Language for communication with knowledge-based agents
Credentials: Under development; may form the basis of a future standard
Status: Draft, 1994

Comments: The specified subject area stretches the concept of a database system to include a knowledge-based agent

Organization: DoD MIL-SPEC-1777/MIL-SPEC 1778 (Title: *TCP/IP)*
Subject Area: Computing Technology/Communication Systems/Network Services
Scope: Network service protocols
Comments: TCP is transport layer (4); IP is addressing layer (3) of OSI model.

Organization: DoD MIL-STD-1815A (Title: *Reference Manual for the ADA Programming Language)*
Subject Area: Computing Technology/Programming Language Systems
Scope: Programming language
Credentials: U.S. National (ANSI)
Status: 1983
Comments: Referenced from IEEE Std 1175

Organization: EC -no id- (Title: *CIM-OSA)*
Subject Area: Industrial Practices/Frameworks
Scope: Open systems architecture for computer-integrated manufacturing
Credentials: Developed by European ESPRIT projects

Organization: ECMA (European Computer Manufacturers Association) 149 (Title: *Portable Common Tool Environment)*
Subject Area: Computing Technology/Software Platforms Relevant to Tool Interconnections (Operating Systems)
Scope: Software tools architecture and interface specification
Credentials: Regional European (E.C.?)
Status: 1990
Comments: Referenced from IEEE Std 1175

Organization: EIA EDIF (Electronic Industries Association) (Title: *Electronic Data Interchange Format)*
Subject Area: Industrial Practices/Information Exchange/Product Data
Scope: Representation and exchange of digital product data for electronic products such as integrated circuits.
Credentials: ANSI accredited standards-making organization
Status: 1992
Comments: Widely used. Also underconsideration by ISO TC93

Organization: EIA RS 274-D (Electronic Industries Association) (Title: *Interchangeable Variable Block Data Format for Positioning, Contouring and Contouring/Positioning Numerically Controlled Machines)*
Subject Area: Industrial Practices/Manufacturing
Scope: Applies to the variable block data format used as input to numerically controlled machine tools.
Credentials: ANSI accredited standards-making organization
Status: 1988
Comments: Widely used.

Organization: EIA RS 494-B (Electronic Industries Association) (Title: *32 bit Binary CL Exchange (BCL) Input Format for Numerically Controlled Machines)*
Subject Area: Industrial Practices/Manufacturing
Scope: Numerically controlled machining input data in a part-oriented, machine independent format represented as a series of 32 bit binary integers.
Credentials: ANSI accredited standards-making organization
Status: 1988

Organization: IEEE 416 (Title: *ATLAS Test Language)*
Subject Area: Manufacturing Equipment

Scope: It is a test-oriented language independent of test equipment, and provides a standard abbreviated English language used in the preparation and documentation of test procedures for manual, automatic, or semi-automatic implementations.

Credentials: U.S. National (ANSI)

Status: 1981

Comments: There is a companion IEEE Std 771, Guide to the Use of ATLAS. There is also a subset language, C/ATLAS, defined in IEEE Stds 716 and 717.

This language, initially developed for the avionics industry, may have some interesting features for the definition of interface performance tests.

Organization: IEEE 488 (Title: *Digital Interface for Programmable Instrumentation)*

Subject Area: Computing Technology/Communication Systems/Network Services

Scope: A byte-serial, bit parallel means to transfer data among a group of instruments and systems.

Credentials: U.S. National (ANSI)

Status: 1978

Comments: In the U.S., this is the most widely used interface to test equipment. It is also known as HPIB (Hewlett-Packard Interface Bus).

Organization: IEEE 730 (Title: *Standard for Software Quality Assurance Plans)*

Subject Area: Computing Technology/Support View of Software Tools

Credentials: U.S. National (ANSI)

Status: 1989

Comments: Referenced from IEEE Std 1175

Organization: IEEE 802.3 (Title: *Ethernet physical link)*

See: ISO/IEC 8802-3

Organization: IEEE 802.4 (Title: *Token bus)*

See: ISO/IEC 8802-4

Organization: IEEE 802.5 (Title: *Token Ring)*

See: ISO/IEC 8802-5

Organization: IEEE 828 (Title: *Standard for Software Configuration Management Plans)*

Subject Area: Computing Technology/Support View of Software Tools

Credentials: U.S. National (ANSI)

Status: 1990

Comments: Referenced from IEEE Std 1175

Organization: IEEE 829 (Title: *Standard for Software Test Documentation)*

Subject Area: Computing Technology/Support View of Software Tools

Credentials: U.S. National (ANSI)

Status: 1983; Reaff. 1991

Comments: Referenced from IEEE Std 1175

Organization: IEEE 830 (Title: *Guide for Software Requirements Specifications)*

Subject Area: Computing Technology/Support View of Software Tools

Credentials: U.S. National (ANSI)

Status: 1984

Comments: Referenced from IEEE Std 1175

Organization: IEEE 1012 (Title: *Standard for Software Verification and Validation Plans)*

Subject Area: Computing Technology/Support View of Software Tools

Credentials: U.S. National (ANSI)

Status: 1986

Comments: Referenced from IEEE Std 1175

Organization: IEEE 1016 (Title: *Recommended Practice for Software Design Descriptions)*

Subject Area: Computing Technology/Support View of Software Tools

Credentials: U.S. National (ANSI)

Status: 1987

Comments: Referenced from IEEE Std 1175

Organization: IEEE 1058.1 (Title: *Standard for Software Project Management Plans)*

Subject Area: Computing Technology/Support View of Software Tools

Credentials: U.S. National (ANSI)

Status: 1987

Comments: Referenced from IEEE Std 1175

Organization: IEEE 1074 (Title: *Standard for Developing Software Life Cycle Processes)*

Subject Area: Computing Technology/Support View of Software Tools

Credentials: U.S. National (ANSI)

Status: 1991

Comments: Referenced from IEEE Std 1175

Organization: IEEE 1175 (Title: *Reference Model for Computing System Tool Interconnections)*

Subject Area: Computing Technology

Scope: Describes the interactions which must be considered when buying, building, testing, or using computing systems tools.

Credentials: U.S. National (ANSI)

Comments: This trial use standard references 42 associated standards in the following areas:

- Support view of software tools
- Software platforms relevant to tool interconnections (operating systems)
- Database systems
- Human interface systems
- Programming language systems
- Communications systems
- Data file exchange formats
- Document exchange formats
- Description exchange formats

Organization: IEEE X3.97 (Title: *Pascal Computer Programming Language)*

See: See ISO 7185

Organization: IETF RFC 822 (Title: *Standard for the Format of ARPANET Internet Text Messages)*

Subject Area: Computing Technology/Communication Systems/Network Services

Scope: Defines message format for Simple Mail Transfer Protocol (SMTP)

Credentials: International

Comments: See also ISO/IEC 10021 - 6

Organization: IETF RFC 1341 (Title: *Multipurpose Internet Mail Extensions)*

Subject Area: Computing Technology/Communication Systems/Network Services

Scope: Defines standard means for identification of message content-types

Credentials: International

Comments: Has important applications in Internet architectures, including the World Wide Web, where it is used in the automatic invocation of appropriate local tools for processing incoming files of various types.

Organization: IETF -no id- (Title: *External Data Representation [XDR])*

Subject Area: Computing Technology/Data File Exchange Formats

Scope: Data encoding rules for many Internet exchanges

Credentials: International

Comments: Uses C as data description language. Limits data structures which can be passed: integers, floats and character strings in fixed structures. Libraries available for most C implementations (only). Used for OSF and Sun rpc and thus for most CORBA implementations, via mapping to C.

Organization: IETF -no id- (Title: *Telnet [Internet])*

Subject Area: Computing Technology/Communication Systems/Application Services

Scope: Remote interactive processing protocol

Credentials: International

Comments: Commonly supported by TCP/IP. No API.

Organization: IETF RFC xx (Title: *Network File System (NFS)*)

Subject Area: Computing Technology/Communication Systems/Application Services

Scope: File transfer, access and management protocol

Credentials: International

Organization: ISO 646 (Title: *Information Processing — 7 bit coded character set for information interchange)*

Subject Area: Computing Technology/Data File Exchange Formats

Scope: Character set

Credentials: International

Status: 1983

Comments: Referenced from IEEE Std 1175

Organization: ISO 2022 (Title: *Information Processing — 8 bit single-byte coded graphic character sets)*

Subject Area: Computing Technology/Data File Exchange Formats

Scope: Character set

Credentials: International

Status: 1986

Comments: Referenced from IEEE Std 1175

Organization: ISO 4342 (Title: *Numerical Control of Machines — NC Processor Input — Basic Part Program Reference Language)*

Subject Area: Industrial Practices/Manufacturing

Scope: High-level language for control of numerically controlled machine tools

Credentials: International

Status: 1985

Comments: See also ANSI X3.37

Organization: ISO 5806 (Title: *Information Processing — Specification of Single-Hit Decision Tables)*

Subject Area: Computing Technology/Description Exchange Formats

Credentials: International

Status: 1986

Comments: Referenced from IEEE Std 1175

Oganization: ISO 5807 (Title: *Information Processing — Documentation Symbols and Conventions for Data, Program and System Flowcharts, Program Network Charts and System Resources Charts)*

Subject Area: Computing Technology/Description Exchange Formats

Credentials: International

Status: 1985

Comments: Referenced from IEEE Std 1175

Organization: ISO 6937 (Title: *Information Processing — Coded character sets for text communication (parts 1 and 2))*

Subject Area: Computing Technology/Data File Exchange Formats

Scope: Character set

Credentials: International

Status: 1983

Comments: Referenced from IEEE Std 1175

Organization: ISO 6983-1 (Title: *Data Format for Positioning, Line Motion and Contouring Control Systems)*

Subject Area: Industrial Practices/Manufacturing

Scope: Numerical control of machines - program format and definition of address words.

Credentials: International

Status: 1982

Comments: Referenced from EIA RS 494.

Organization: ISO 7185 (Title: *Pascal Computer Programming Language)*
Subject Area: Computing Technology/Programming Language Systems
Scope: Programming language
Credentials: International
Comments: Referenced from IEEE Std 1175 as IEEE 770 X3.97.

Organization: ISO 7498 (Title: *Information Processing Systems — Open Systems Interconnection)*
Subject Area: Computing Technology/Communication Systems
Scope: Open Systems Interconnection (OSI)
Credentials: International

Organization: ISO 8072 (Title: *Information Processing Systems — Open Systems Interconnection — Transport Service Definition)*
Subject Area: Computing Technology/Communication Systems
Scope: OSI transport layer protocol
Credentials: International
Status: 1986
Comments: Referenced from IEEE Std 1175. See also ISO 8073

Organization: ISO 8073 (Title: *Information Processing Systems — Open Systems Interconnection — Connection Oriented Transport Protocol Specification)*
Subject Area: Computing Technology/Communication Systems
Credentials: International
Status: 1988
Comments: Referenced from IEEE Std 1175. See also ISO 8072

Organization: ISO 8571 (Title: *Information Processing Systems — Open Systems Interconnection — File Transfer, Access and Management (Parts 1 - 5))*
Subject Area: Computing Technology/Communication Systems/Application Services
Credentials: International
Scope: OSI file transfer
Status: 1988, except Part 5 (1990)
Comments: Referenced from IEEE 1175. Part 5, concerning protocol implementation, is an ISO/IEC standard

Organization: ISO 8613 (Title: *Information Processing — Text and Office Systems — Office Document Architecture (ODA) and Interchange Format (Parts 1-8))*
Subject Area: Computing Technology/Document Exchange Formats
Credentials: International
Status: Varied
Comments: Referenced from IEEE Std 1175

Organization: ISO 8649 (Title: *Information Processing Systems — Open Systems Interconnection — Protocol Specification for Association Control Service Element — Service Definition for Association Control Service Element)*
Subject Area: Computing Technology/Communication Systems
Scope: OSI application connection protocol
Credentials: International
Status: 1988
Comments: Referenced from IEEE Std 1175. See also ISO 8650.

Organization: ISO 8650 (Title: *Information Processing Systems — Open Systems Interconnection — Protocol Specification for Association Control Service Element)*
Subject Area: Computing Technology/Communication Systems
Credentials: International

Status: 1988

Comments: Referenced from IEEE Std 1175. See also ISO 8649

Organization: ISO 8790 (Title: *Information Processing Systems — Computer System Configuration Diagram Symbols and Conventions)*

Subject Area: Computing Technology/Description Exchange Formats

Credentials: International

Status: 1987

Comments: Referenced from IEEE Std 1175

Organization: ISO 8859 (Title: *Information Processing — 8 bit single-byte coded character sets (parts 1-5))*

Subject Area: Computing Technology/Data File Exchange Formats

Scope: Character set

Credentials: International

Status: 1983

Comments: Referenced from IEEE Std 1175.

Organization: ISO 8879 (Title: *Information Processing — Text and Office Systems — Standard Generalized Markup Language (SGML))*

Subject Area: Computing Technology/Document Exchange Formats

Credentials: International

Status: 1986

Comments: Referenced from IEEE Std 1175.

Organization: ISO 9000 (Title: *Quality Management and Quality Assurance Standards)*

Subject Area: Industrial Practices

Scope: The ISO 9000 Series consists of five standards, ISO 9000-9004. ISO 9001-9003 provide models of quality assurance at decreasing levels of scope. ISO 9000 provides guidelines to the selection of the appropriate model, and ISO 9004 provides guidelines for implementing the model.

Credentials: International

Status: Standards Series; 1992

Comments: This is developed by ISO/TC 176. The various parts of the ISO 9000 series are:

ISO 9000: Quality Management and Quality Assurance Stadards: Guidelines for Selection and Use

ISO 9001: Quality Systems-Model for Quality Assurance in Design/Development, Production, Installation, and Servicing

ISO 9002: Quality Systems-Model for Quality Assurance in Production and Installation

ISO 9003: Quality Systems-Model for Quality Assurance in Final Inspection and Test

ISO 9004: Quality Management and Quality System Elements Guidelines

The ISO 9000 series is the same as ANSI/ASQC Q90-94 series. They are also closely related to MIL-Q-9858A and MIL-I-45028A, both of which will eventually be replaced by ISO 9000 within DoD.

Organization: ISO 9040/9041 (Title: *Information Technology — Open Systems Interconnection — Virtual Terminal Basic Class Service/Basic Class Protocol)*

Subject Area: Computing Technology/Communication Systems/Application Services

Scope: Remote login

Credentials: International

Organization: ISO 9074 (Title: *Information Processing Systems — Open Systems Interconnection — Estelle: a Formal Description Technique based on an Extended State Transition Model)*

Subject Area: Computing Technology/Communication Systems/Application Services

Scope: Models behaviour of a `protocol machine' in sending and receiving messages of various types
Credentials: International
Status: 1989, amended 1993

Organization: ISO 9989 (Title: *Programming Language — C)*
Subject Area: Computing Technology/Programming Language Systems
Scope: Programming language
Credentials: International
Status: 1990
Comments: Referenced from IEEE Std 1175 as ANSI/CBEMA X3.159

Organization: ISO 10303 - 1 (Title: *Overview and fundamental principles)*
Subject Area: Industrial Practices/Information Exchange/Product Data
Scope: Provides an overview of the STEP standards.
Credentials: International

Organization: ISO 10303 - 11 (Title: *Descriptive methods: The EXPRESS language reference manual)*
Subject Area: Computing Technology/Description Exchange Formats
Scope: Conceptual schema language
Credentials: International
Status: IS

Organization: ISO 10303 - 21 (Title: *Implementation methods: Clear text encoding of the exchange structure)*
Subject Area: Computing Technology/Data File Exchange Formats
Scope: Used for file exchanges conforming to Express models.
Credentials: International
Status: IS
Comments: Can pass arbitrarily complex objects. Uses only display characters, good for debugging and experimental development. Tools and libraries available. Use for file transfers.

Organization: ISO 10303 - 22 (Title: *Standard Data Access Interface (SDAI))*
Subject Area: Computing Technology/Database Systems
Scope: "A specification for an Application Program Interface (API) to ISO 10303 data representations is described. The functional specification itself is given in an implementation language form. EXPRESS is used to describe the data types which may be accessed through the interface. The behavior of SDAI implementations is described in English. Bindings of the functional specification to the C, FORTRAN, and C++ programming languages are provided."
Credentials: International
Status: CD
Comments: Emerging standard for STEP databases. Currently a moving target; bindings for other languages are under development. Requires DB schema to be described in EXPRESS. Use it for direct access to databases built from EXPRESS models.

Organization: ISO 10303 - 40 Series (Title: *Integrated generic resources)*
Subject Area: Industrial Practices/Information Exchange/Product Data
Scope: Together with 100 Series resources, provides a collection of information models used as resources for building application protocols (200 Series). The 40 Series resources are defined independent of application.
Credentials: International
Comments: Not actually the basis for exchange, used in building exchange models. Link SIMA models to these where appropriate.

Organization: ISO 10303 - 49 (Title: *Integrated generic resources: Process structure, property and representation)*

Subject Area: Industrial Practices/Information Exchange/Manufacturing Management Data

Scope: "Specifies the elements of a process plan. A process plan is the specification of instructions for a task. This Part does not specify any particular process, but defines the elements to exchange process information. This Part specifies the information necessary to represent the execution of a process including the relationships between the steps in the process. The process plan can be used to define or enhance a product definition. The process plan can also be a set of instructions to complete a task without regard to a product definition."

Credentials: International

Organization: ISO 10303 - 100 Series (Title: *Integrated application resources)*

Subject Area: Industrial Practices/Information Exchange/Product Data

Scope: Together with 40 Series resources, provides a collection of information models used as resources for building application protocols (200 Series). The 100 Series resources are defined to support a single application or range of applications.

Credentials: International

Comment: Used in building exchange models.

Organization: ISO 10303 - 200 Series (Title: *Application protocols)*

Subject Area: Industrial Practices/Information Exchange/Product Data

Scope: Application protocols are developed for a particular application context using the integrated resources and descriptive methods.

Credentials: International

Comments: Being implemented on many CAD systems and some related design systems, such as solid geometry packages. Part 202 may supplant IGES. Parts 203 and 207 may be important to SIMA, when available in implementations. Use for exchanges where appropriate.

Organization: ISO 10303 - 1200 Series (Title: *Abstract test suites)*

Subject Area: Industrial Practices/Information Exchange/Product Data

Scope: Define test requirements for implementations of 10303 Series 200 exchange formats.

Credentials: International

Organization: ISO 11578 (Title: *Remote Procedure Call)*

Subject Area: Computing Technology/Communication Systems/Application Services

Scope: Remote command/response

Credentials: International

Status: DIS 1994

Comments: All remote services are calls. Protocol is more complete than OSF rpc (q.v.) and Sun rpc (q.v.), may replace them. IDL will be mapped to many more languages, but may have little effect on primary (C) community. API: language-based "bindings" not yet defined.

Organization: ISO 13584 (Title: *Parts Library)*

Subject Area: Industrial Practices/Manufacturing/Product Standards

Scope: Seven documents describing the representation of component part library information in digital form.

Credentials: International

Status: Circulated for first ballot review April 1995.

Comments: Under development by ISO/TC 184/SC 4 (the developer of STEP).

Organization:	ISO Handbook 3 (Title: *Statistical Methods)*
Subject Area:	Industrial Practices/Manufacturing
Scope:	"The International Standards contained in this Handbook set out the practical methodology which a user requires in order to be able to process and interpret, statistically, testing and inspection results whenever goods are assessed from a sample."
Credentials:	International
Status:	1989

Organization:	ISO Handbook 33 (Title: *Applied Metrology—Limits, fits, and surface properties)*
Subject Area:	Industrial Practices/Manufacturing
Scope:	This Handbook includes 58 International Standards, subdivided into five groups: terminology; indication of tolerances and surface conditions on technical drawings; limits and fits; properties of surfaces; and common measuring instruments.
Credentials:	International
Status:	1988
Comments:	This is related to the efforts of the ISO/TC 3-10-57/Joint Harmonization Group.

Organization:	ISO -no id- (Title: *FDDI)*
Subject Area:	Computing Technology/Communication Systems/Transmission services
Scope:	High-speed physical
Credentials:	International
Comments:	Strongly recommended for large data volume.

Organization:	ISO -no id- (Title: *LDDI)*
Subject Area:	Computing Technology/Communication Systems/Transmission services
Scope:	High-speed physical
Credentials:	International
Comments:	Acceptable, but less commonly available than FDDI

Organization:	ISO TC 3-10-57 (Title: *Joint Harmonization Group)*
Subject Area:	Industrial Practices/Information Exchange/Product Data
Scope:	To develop a roadmap of all standards related to the specification, manufacture, and inspection of geometric characteristics of products.
Credentials:	International
Status:	Standards Framework; In progress
Comments:	This has the potential to be an extremely important activity with respect to SIMA. We should find a way to keep up to speed on what is happening in this group.

Organization:	ISO/IEC 8802 - 3 (Title: *Information Technology — Local and Metropolitan Area Networks —Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications)*
Subject Area:	Computing Technology/Communication Systems
Credentials:	International
Status:	1992
Comments:	Also ANSI/IEEE Std 802.3, 1992 Edition. Referenced from IEEE Std 1175

Organization:	ISO/IEC 8802 - 4 (Title: *Information Technology — Local Area Networks — Part 4: Token-passing bus access method and physical layer specifications)*
Subject Area:	Computing Technology/Communication Systems
Credentials:	International
Status:	1990
Comments:	Also ANSI/IEEE Std 802.4, 1990 Edition. Referenced from IEEE Std 1175. This is Map and MiniMap. Use MiniMap only if absolutely necessary.

Organization: ISO/IEC 8802 - 5 (Title: *Information Technology — Local and Metropolitan Area Networks —Part 5: Token ring access method and physical layer specifications)*
Subject Area: Computing Technology/Communication Systems
Credentials: International
Status: 1990
Comments: Also ANSI/IEEE Std 802.5, 1992 Edition. Referenced from IEEE Std 1175

Organization: ISO/IEC 8824 (Title: *Information Technology — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1))*
Subject Area: Computing Technology/Description Exchange Formats
Credentials: International
Status: 1990
Comments: Referenced from IEEE Std 1175. Used for network messages conforming to ISO standards (RPC, MMS, RDA/SQL) and for some file formats.

Organization: ISO/IEC 8825 (Title: *Information Technology — Open Systems Interconnection — Specification of Encoding Rules for Abstract Syntax Notation One (ASN.1))*
Subject Area: Computing Technology/Data File Exchange Formats
Scope: Encoding rules for ASN.1 (ISO 8824).
Credentials: International
Status: 1990
Comments: Referenced from IEEE Std 1175. Multiple parts. Part 1 is Basic Encoding Rules. Part 2 is Packed Encoding Rules.

Organization: ISO/IEC 9075 (Title: *Information Technology — Database Language — SQL)*
Subject Area: Computing Technology/Database Systems
Credentials: International
Status: 1989
Comments: Referenced from IEEE Std 1175 (as ANSI/CBEMA X3.135).

Supported by all major vendors. Use it for direct access to all relational data bases.

Organization: ISO/IEC 9506 - 1 Annex C (Title: *MMS File Services)*
Subject Area: Manufacturing Equipment
Scope: File transfer, access and management protocol
Credentials: International
Comments: Strictly binary; used to move NC code. Unofficial part of MAP. API: defined by MAP (not common), others are different and usually better, but not common. See ISO 9506-1

Organization: ISO/IEC 9506 (Title: *Manufacturing Message Specification)*
Subject Area: Manufacturing Equipment
Scope: Remote command/response
Credentials: International
Comments: Includes the service specification (Part 1), protocol (Part 2) and other parts.

Variable services adequate for conveying any data, command or response, but then real protocol is user-defined. Task protocol adequate for speaking to controllers about simple tasks; too many protocol rules for more general use. Commonly supported by controllers of many kinds.

Organization: ISO/IEC 9579 (Title: *Remote Database Access)*
Subject Area: Computing Technology/Communication Systems/Application Services
Credentials: International
Status: 1990

Comments: Provides generic (Part 1), SQL specialization (Part 2), and IRDS specialization (Part 3). Commonly supported for remote access to SQL data bases, over many different network and transmission services. Use for relational databases ONLY. API: embedded SQL (ISO 9075-2), commonly supported. Part 1 could conceivably be used with a new "specialization" for OODBs, etc., but other approaches should also be considered. No API for Part 1.

Organization: ISO/IEC 9592 (Title: *Information Processing Systems — Computer Graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS), Parts 1 - 4)*)
Subject Area: Computing Technology/Data File Exchange Formats
Credentials: International
Status: 1989 except Part 4 (1992)
Comments: Language bindings for PHIGS are apecified in the several parts of ISO/IEC 9593

Organization: ISO/IEC 9945 (Title: *Information Technology — Portable Operating System Interface (POSIX)*)
Subject Area: Computing Technology/Software Platforms Relevant to Tool Interconnections (Operating Systems)
Scope: Operating system interface
Credentials: International
Status: 1990
Comments: Also ANSI/IEEE Std 1003. Referenced from IEEE Std 1175. Strongly aligned with Unix. Use wherever possible, encapsulate departures.

Organization: ISO/IEC 10021 (Title: *Information Technology — Text Communication— Message-oriented Text Interchange Systems (MOTIS) [in 7 parts]*)
Subject Area: Computing Technology/Document Exchange Formats
Scope: Internet messaging standard
Credentials: International
Status: 1990, with revisions to 1994
Comments: See also IETF RFC 822

Organization: ISO/IEC 10744 (Title: *HyTime)*
Subject Area: Computing Technology/Document Exchange Formats
Scope: Multimedia exchange standard
Credentials: International
Status: DIS

Organization: ISO/IEC 10746 (Title: *Reference Model for Open Distributed Processing)*
Subject Area: Computing Technology/Communication Systems/Application Services
Scope: Provides a framework for standardization of open distributed processing. The standard consists of four parts: an overview and guide; a descriptive model (definitions and framework); a prescriptive model (requirements for conformance); and architectural semantics (formalization of concepts)
Credentials: International
Status: WD
Comments: ISO/IEC JTC1/SC 21/WG 7.
Also ITU-T X.901

Organization: ISO/IEC 11179 (Title: *Data Element Attributes)*
Subject Area: Computing Technology/Description Exchange Formats
Scope: Attribute naming conventions for data elements
Credentials: International
Status: IS
Comments: Conformance to parts of 11179 is required by DoD.

Organization: ISO/IEC DIS 13886 (Title: *Information Technology — Language Independent Procedure Call)*

Subject Area: Computing Technology/Communication Systems/Application Services
Scope: Language-independent interface definition specification
Credentials: International
Status: Draft standard, 1995
Comments: Under development by ISO/IEC JTC1 SC22

Organization: ISO/IEC -no id- (Title: *Guide to the Expression of Measurement Uncertainty)*
Subject Area: Industrial Practices/Manufacturing/ Calibration and Performance Testing
Scope: A comprehensive guide to evaluation and reporting of measurement uncertainty.
Credentials: International
Status: 1992
Comments: Important topic for all measurement applications.

Organization: ISO/ITU X.500 (Title: *Directory Access)*
Subject Area: Computing Technology/Communication Systems/Application Services
Scope: Remote directory access
Credentials: International

Organization: ISO/TR 10314-1 (Title: *Industrial Automation — Shop Floor Production — Part 1: Reference Model for Standardization and a Methodology for Identification of Requirements)*
Subject Area: Industrial Practices
Credentials: International
Status: 1990

Organization: ISO/TR 10314-2 (Title: *Industrial Automation — Shop Floor Production — Part 2: Application of the Reference Model for Standardization and methodology)*
Subject Area: Industrial Practices
Credentials: International
Status: 1991

Organization: ISO/TR 12186 (Title: *Manufacturing Automation Programming Language Environment Overview (MAPLE))*
Subject Area: Industrial Practices/Frameworks
Credentials: International
Status: 1993

Organization: ISO/TR 13345 (Title: *Industrial Automation Systems — Specification of Subsets of the Protocol for ISO/IEC 9506))*
Subject Area: Industrial Practices/Frameworks
Credentials: International
Status: 1994

Organization: ITU V.35 (Title: *T1 line)*
Subject Area: Computing Technology/Communication Systems/Transmission services
Scope: Long run serial
Credentials: International
Comments: V.35 at 1.44 Mbps with HDLC but linking local-net to data highway is preferable.

Organization: ITU X.901 (Title: *Reference Model for Open Distributed Processing)*
See: ISO/IEC 10746

Organization: Microsoft -no id- (Title: *Windows)*
Subject Area: Computing Technology/Software Platforms Relevant to Tool Interconnections (Operating Systems)
Comments: De facto standard. Use in lieu of X-windows on PC-specific HCIs.

Organization: N/A Ethernet V2 (Title:)
Subject Area: Computing Technology/Communication Systems/Transmission services
Scope: Ethernet physical link
Comments: Strongly recommended. See IEEE 802.3 and ISO/IEC 8802-3.

Organization: NIST FIPS 183 (Title: *IDEF0)*
Subject Area: Computing Technology/Description Exchange Formats
Scope: Federal standard function modeling language
Comments: Primarily a graphical language

Organization: NIST FIPS 184 (Title: *IDEF1X)*
Subject Area: Computing Technology/Description Exchange Formats
Scope: Federal standard information modeling language
Comments: Primarily a graphical language, with a parseable language added on. See also its alternative, EXPRESS (ISO 10303-11)

Organization: NIST -no id- (Title: *ALPS - A Language for Process Specification)*
Subject Area: Industrial Practices/Information Exchange/Manufacturing Management Data
Scope: A means for the representation of process plans, intended to bridge the gap between manufacturing engineering snd production control
Credentials: None
Status: Laboratory study

Organization: NIST -no id- (Title: *Manufacturing Systems Integration)*
Subject Area: Industrial Practices/Frameworks
Scope: Open systems architecture for computer-integrated manufacturing
Credentials: None
Status: Laboratory study

Organization: OMG CORBA (Title: *Remote Procedure Call)*
Subject Area: Computing Technology/Communication Systems/Application Services
Scope: Remote command/response
Comments: All remote services deal with objects and messages. This is a good and general model. Most implementations map to OSF rpc, with problems and subterfuges. API is standard, but defined only for C at the moment. IDL is general, has ISO features, but resembles C.

Organization: OMG -no id- (Title: *CORBA Object Model)*
Subject Area: Computing Technology/Description Exchange Formats
Scope: Conceptual schema language
Comments: Use it for specifying object interfaces. Mapping to/from Express needed.

Organization: OMG -no id- (Title: *ODML/OQL)*
Subject Area: Computing Technology/Database Systems
Comments: Just apearing, will be commonly supported by major vendors. Use it for direct access to OODBs, where possible. Project carried out by ODBTG, which is a separate organization from OMG, but related.

Organization: OSF DCE rpc (Title: *Remote Procedure Call)*
Subject Area: Computing Technology/Communication Systems/Application Services
Scope: Remote command/response
Comments: All remote services look like subroutine calls. One of several UNIX-based competitors. Supports C-language calls to C-language procedures, linkage to other languages is NOT easy, although possible. Supports asynchronous calls, with UNIX environment problems, has real problems with pointer objects and callbacks. API: C, with annotations.

Organization: SEMI E5 (Title: SEMI Equipment Communications Standard (SECS II))
Subject Area: Industrial Practice/Information Exchange/Manufacturing Management Data

Scope: Control of semiconductor manufacturing equipment

Comments: SEMI is an international consortium concerned with the manufacture of semiconductors

Organization: Sun ONC rpc (Title: *Remote Procedure Call)*

Subject Area: Computing Technology/Communication Systems/Application Services

Scope: Remote command/response

Comments: All remote services look like subroutine calls. Another competitor, less commonly emulated. Simpler, efficient, but more limited in capability. API: C, with annotations.

Organization: X/Open Group -no id- (Title: *X/Open Portability Guide)*

Subject Area: Computing Technology/Software Platforms Relevant to Tool Interconnections (Operating Systems)

Credentials: U.S. Industry

Comments: Referenced from IEEE Std 1175.

Organization: X/Open Group -no id- (Title: *X-Window System)*

Subject Area: Computing Technology/Human Interface Systems

Credentials: Industry

Status: Version X.11

Comments: Referenced from IEEE Std 1175. See FIPS Pub. 158. Use wherever possible. Problems with color.