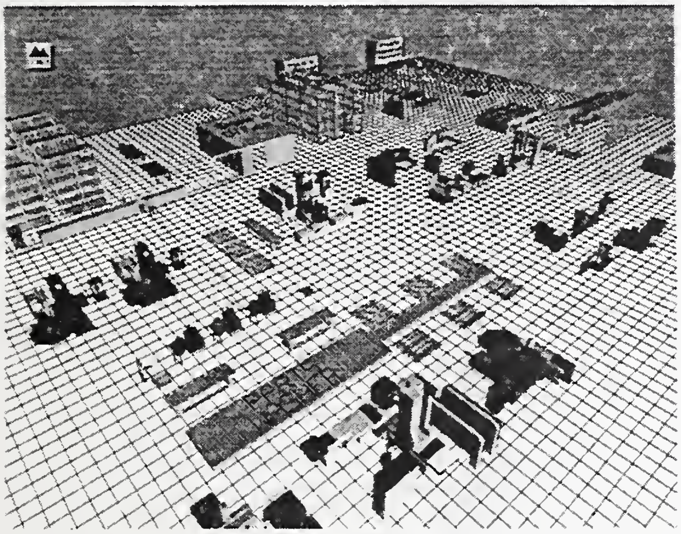
NISTIR 5846

Computer-Aided Manufacturing Engineering Forum

Second Technical Meeting Proceedings
Hilton Hotel, Gaithersburg, MD
August 22 - 23, 1995



Mike Smith
Swee Leong

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

and

U.S. Navy Manufacturing Technology Program

NIST

NISTIR 5846

Computer-Aided Manufacturing Engineering Forum

Second Technical Meeting Proceedings
Hilton Hotel, Gaithersburg, MD
August 22 - 23, 1995

Mike Smith
Swee Leong

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

and

U.S. Navy Manufacturing Technology Program

May 1996

U.S. DEPARTMENT OF COMMERCE
Michael Kantor, Secretary

TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arati Prabhakar, Director

Table of Contents

CAME Forum Overview and Objectives

The second technical meeting of the Computer-Aided Manufacturing Engineering (CAME) Forum convened on August 22-23, 1995 in Gaithersburg, Maryland. This meeting built on work completed at and since the first technical meeting held in March, 1995 and brought together software developers, manufacturing engineers, manufacturing managers, and university-based researchers that could address important issues related to developing a Manufacturing Engineering Tool Kit (METK). The tool kit project is jointly sponsored by the U.S. Navy Manufacturing Technology Program and the National Institute of Standards and Technology.

This second meeting was attended by 26 representatives from industry, vendors, government, and academia, many of whom participated in the earlier technical meeting where manufacturing engineering data validation needs were identified and a context for developing the METK was established. A list of participants in the second technical meeting is provided in Appendix A as is the meeting agenda. The meeting offered an opportunity for NIST scientists to update participants on the status of ongoing METK development work; demonstrate capabilities of available manufacturing engineering application software tools; and solicit advice, suggestions, concerns, and consensus about the emerging manufacturing data validation tools and their development and application.

Specifically, the objectives of this second technical meeting of the CAME Forum were:

1. *Provide an update on METK project status.*
2. *Demonstrate the baseline manufacturing engineering tool kit.*
3. *Review and enhance the manufacturing data validation methodology.*
4. *Develop and discuss manufacturing data generation and validation implementation issues.*
5. *Identify and address system integration issues.*
6. *Get an overview of NIST's related production system engineering activities.*

This one and one-half day meeting was organized around three major segments, each segment lasting about one-half day. The first segment provided the update information and demonstrations needed to make informed and intelligent assessments of METK development objectives and progress. The second segment sought ideas from meeting participants through three breakout groups assigned specific questions and tasks drawn from meeting objectives 3, 4, and 5 above. In the third segment, breakout groups prepared outbriefs and presented results to all meeting participants. More specifically, the three segments were as follows:

1. *Information Sharing*
   - Project Status Update
   - Manufacturing Validation Issues
   - Implementation Planning Issues
   - Baseline System Demonstration
   - System Integration and Interface Issues

2. *Developing the Issues -- Breakout Group Discussions*

- Business Case and Implementation Strategy
- Validation Methodology and Application
- METK Architecture and System Integration Issues

3. *Reporting Results -- Breakout Group Outbriefs*

- Related NIST Production System Engineering Activities
- Outbrief Presentation Preparation
- Breakout Group Reports
- Open Forum, Next Steps, and Meeting Wrap-up

METK Overview and Project Status

A brief review of the project (see Appendix B) provided participants not familiar with the previous METK project with a context for further presentations, demonstrations, and discussions. Importantly, the Navy-sponsored CAME Project (of which the METK development effort is a part), is designed to lower manufacturing costs, reduce delivery times, and improve product quality through the coordinated development and use of commercial-off-the-shelf(COTS) computerized tools that apply scientific and engineering methods to design and implementation of manufacturing systems, processes, and equipment. The project calls for establishing an alliance of users, researchers, developers, and vendors who work together to develop architectures, databases, and techniques for tool integration; develop common data bases for manufacturing engineering data; construct prototype tool kit environments using COTS tools; test and validate tool kits using real world data; develop solutions applicable to large and small shops; and recommend potential standards based on results.

The METK project will:

- Construct an integrated manufacturing engineering tool kit and common databases that provide
  1) Product data and work flow management
  2) Process Planning
  3) Engineering data validation through simulation, and
  4) Other capabilities for a family of machined parts.
- Base functionality upon extensions to the capabilities of commercial off-the-shelf (COTS) software and hardware.
- Conduct the project as a collaborative effort by users, vendors, academic researchers, and representatives of other government agencies in project planning, requirements definition, system design, development, testing, and evaluation.
- Perform "alpha" testing and evaluation of the tool kit environment at industry and government manufacturing sites.
- Recommend new industry standards based upon project results.

The status of the METK development effort is shown in the project Gantt chart provided in Appendix C. Work to date centers on initiating and solidifying industrial, academic, and government collaboration; defining, acquiring, and installing the baseline development system; and developing requirements and specifications for the METK environment. Industrial, academic, and government partners are engaged in the effort through various arrangements (CRADAs, MOUs, and direct contracts). A baseline development environment is in place in NIST's Advance Manufacturing Systems And Networking Testbed. Test parts, manufacturing processes, and a model shop are identified for developing the prototype METK environment.

Manufacturing Engineering Data Generation and Validation Issues

To make breakout group discussions more productive, three presentations set the stage by establishing objectives, raising issues, and posing questions. These three presentations dealt with (1) Manufacturing Data Validation Issues (Appendix D); (2) Implementation Plan and Issues (Appendix E); and (3) CAME METK Architectural Review. An additional presentation on the Production System Engineering Environment under development by NIST was also made to provide a broader context for discussing data generation and validation issues (Appendix F). Each presentation is briefly summarized below.

Manufacturing Validation Issues

Manufacturing validation issues are presented in the form of a briefing and a report, both contained in Appendix D and summarized in this section. The objective of the effort is to define a methodology for the validation of manufacturing data that may be applied using either manual methods or automatic systems to help ensure that machined parts are produced correctly the first time. Manufacturing data include, at a minimum:

1. Manufacturing order
2. Routing sheet
3. Stock material specification
4. Intermediate workpiece geometry and shape information
5. Operation sheet
6. Machine setup sheet
7. Workpiece setup sheet
8. Tool list
9. Fixture list
10. NC program

Engineering data which support modeling and validation of manufacturing data include:

1. Product data
2. Manufacturing resources
3. Production data
4. Manufacturing system and resource functional models
5. Engineering process management data
6. Manufacturing engineering knowledge

The manufacturing data validation methodology will be developed by:

- Identifying data elements to be validated
- Knowing the types of errors and where they occur
- Establishing techniques for identifying errors

The key elements of the proposed methodology are

- Check data integrity
- Verify the manufacturing process
- Evaluate cost and performance

Remaining issues and questions:

- Are the proposed data package elements correct?
- Can the data formats be standardized?
- Should the proposed methodology be modified?
- How can the methodology be applied incrementally during the data generation process?

Implementation Plan and Issues

The major implementation issues explored in this briefing (see Appendix E) are:

- Manufacturing data package
- Data model for baseline system demonstration
- Reference SIMA architecture
- Manufacturing data management and control
- Manufacturing data state space diagram
- Organization chart

The briefing in Appendix E shows graphically the elements, flows, and relationships within each of these areas and also provides an example using a Naval Surface Warfare Center (NSWC) scenario.

An implementation and rollout plan and related issues are also described. Plan elements and issues include:

- Prototype system at NIST
- Prototype system "roadshow" to demonstrate capabilities
- Software licenses issues

CAME METK Architecture Review

The CAME METK architecture review briefing, provided in Appendix F, offers a starting point for developing and integrating CAME tools. The briefing describes a preliminary CAME METK architecture and highlights the significant manufacturing integration architectural issues that must be addressed.

The briefing begins by describing the current situation in most manufacturing facility where computer-aided design and manufacturing software tools are generally stand alone applications with minimal integration. The briefing shows an evolving architecture beginning with an integrating architecture that provides product data and workflow management. Common data storage, query, and reporting capabilities are added to this basic architecture. Finally, generalized application interfaces are provided that permit integration of manufacturing engineering and data validation tools to form the METK.

The briefing describes in greater detail the components and capabilities of the proposed architecture including application interfaces, federated databases, reference data sharing, reference data extension, generated data sharing, modification back-propagation, redundant data storage, result variability between similar tools, and engineering data package validation support.

Remaining questions and issues:

- What needed functionality is not provided by the proposed METK architecture?
- Which issues are not fully addressed by the proposed METK architecture?
- How does your organization wish to be involved in the development of the final METK architecture specification?

SIMA Production System Engineering Environment

The SIMA Production System Engineering Environment briefing, provided in Appendix G, describes NIST's program for integrating a number of production planning and control applications. The *goal* of the SIMA production project is "to develop solutions for integrating production engineering, planning, scheduling, and simulation systems with other manufacturing life cycle applications." NIST's *strategy* for accomplishing this is:

- Identify industrial collaborators with diverse production systems
- Leverage on-going research on OA-sponsored projects
- Develop process and information models for production applications
- Coordinate efforts and integrate results with other SIMA projects
- Use a product data management system to integrate production engineering and production planning systems
- Test solutions using simulation capabilities established for the AMSANT facilities

The *technical approach* is:

- Develop production system engineering process model
- Select computing platform(s) and candidate tools
- Identify baseline production engineering problem and test case data
- Load test case data into selected tools
- Define information models and links between tools
- Identify integration opportunities and interface specifications
- Integrate and test commercial tools
- Apply environment to new engineering problems

Elements of the SIMA production engineering environment include:

- Process specification
- Production system design
- Production line design
- Plant layout
- Manufacturing simulation
- Project management
- Production cost estimating

Baseline System Demonstration

A demonstration of the toolkit applications has been prepared to illustrate the functionality of a prototype METK. The demonstration is comprised of two scenarios in which information in an manufacturing data package is generated and validated. An manufacturing data package contains the information needed to perform the manufacturing operations required to produce a part. The package contains various elements including NC programs, operations sheets, routing data, CAD models, tool lists, fixture lists, and machine lists. The manufacturing data package used in the demonstration is for a small prismatic machined part. The first scenario involves tasks performed to generate and validate the NC program, operation sheet, tool lists, and fixture lists elements of the manufacturing data package. The second scenario involves the validation of the process routing data.

The first scenario of the demonstration consists of creating a solid model geometry in the Pro-Engineer CAD application. The CAD solid model geometry output is used as input into the generative process planning application ICEM Part. ICEM Part then creates a process plan and stores the information in the Oracle database. A CNC program is also produced by the ICEM Part application. Interface software is then executed to extract the machine tool, cutting tools, raw stock and fixture information from the database. This interface software was developed by Robert Judd, Ohio University under the Intelligent Machining Workstation project. This interface is currently implemented as UNIX shell scripts. The scripts query the Oracle database for the appropriate information, create the directory structure needed by Deneb VNC, and construct a simulated workcell in VNC. The workcell consists of: a pre-developed kinematic VNC model of the EMCO 100 milling workstation; a blank fixtured to the machine table; geometric models of the tooling mounted on the machine; and the appropriate CNC program. The demonstration then executes Deneb VNC to simulate the machining process. VNC is used to help identify any errors in the CNC program as well as any tool crashes or part gouges. This is a major part of the NC program validation process.

The second scenario of the demonstration simulates the workflows between factory workstations used to create the prismatic product. A virtual factory is being modeled in Deneb Quest. Each workstation in the virtual factory will perform processes that represent manufacturing processes. The processes were selected by industrial participants at the first technical meeting of the Computer Aided Manufacturing Engineering Forum on March 21-22 1995. The inclusion of additional workstations in the virtual factory which perform other types of processes will be considered as the needs of the forum participants change over the life of the project. The Quest model development has focused on the workflow/routing required to produce the prismatic product used in the first scenario of the demonstration. The Quest simulation environment is intended to validate the process routing data in the manufacturing engineering data package.

Manufacturing Data Validation Tool Issues and Reports

The manufacturing data validation tool development and implementation issues raised during the first segment of the meeting provided the basis for breakout group discussions and subsequent reports during the latter two meeting segments. Three breakout groups formed to address issue in three different areas. One participant in each group served as facilitator and other group members self-selected into the group. Some participants with specific interests or expertise served in more than one group, moving between groups from time to time. The three breakout group and their specific objectives were:

1. ***Implementation*** -- Develop the business case and implementation approach for manufacturing data generation and validation as part of the METK.
2. ***Validation*** -- Review the manufacturing data validation needs and develop scenarios for applying the validation methodology.
3. ***Architecture*** -- Review and test the proposed METK architecture by illustrating how the data generation and validation applications and data would operate.

The end users and manufacturing managers were encouraged to join the Implementation Group, led by Steve Ray of NIST; end users and developers were encouraged to join the Validation Group, led by Charles Parks of Ohio University; and developers and vendors were encouraged to join the Architecture Group, led by Alan Brown of the Software Engineering Institute (SEI). Each group appointed a recorder and spokesperson; the facilitator managed the group, keeping it focused on its assigned task and ensuring balanced participation among group members. Each group began with a set of questions or tasks that helped direct group discussion. These questions served as points of departure for meaningful discussion of issues and provided a framework for developing the groups' reports to the forum during plenary session. These questions and the groups' findings are reported below.

<u>Implementation</u>

The implementation breakout group used the objectives given above and the following guidance as a framework for discussions and reporting:

1. Develop the business case for integrated manufacturing engineering data generation and validation.
    - How are manufacturing data typically generated and validated today?
    - What are the major problems and opportunities for improvement in the data generation/validation process?
    - What tangible benefits should result from improvements in the data generation/validation process (e.g., cost reduction, cycle time reduction)?
    - How could the value of an improved data generation/validation process be assessed?
    - Develop a "test plan" for assessing the value of an integrated manufacturing data generation/validation methodology. [Note: there may be more than one plan depending on the type of facility].
2. Develop the implementation approach for introducing an integrated manufacturing engineering data generation and validation methodology in a typical manufacturing facility.
    - Who is involved and how?
    - What is the best time frame for implementation?
    - What should be the scope of the implementation (part types, processes, organizations, etc.)

- What are the minimum and preferred hardware/software/process/data infrastructure requirements to support implementation?
- Who should be on the primary implementation team (i.e., responsible for implementation actions)?
- What training, if any, should be provided and for whom?
- What are the major steps in the implementation process, what is their approximate timing, and who is responsible for each?
- How should the implementation be evaluated?

This group considered implementation in terms of both the business case that justifies the METK and the approach for implementing the METK in an organization. In addressing these issues, the group developed both strategic and tactical business case for manufacturing data validation, it considered how data validation is accomplished today and the problems and opportunities associated with these approaches, the benefits of improved data validation methods and ways for measuring improvements, and the cultural considerations for implementing improved methods.

Business Case. The group concluded that the "bottom line" for selling any system to management decision makers is how the system will help the organization produce products faster, better, and/or cheaper -- which translates into speed (or cycle time), product quality, and development and production cost. The group saw the need for both strategic and tactical business cases. The strategic business case looks at the competitive environment to determine if and how well an organization can gain and/or maintain its competitive position in a given market. Managers must decide if they plan to enter or remain in markets that are under significant global competitive pressures. If they choose to compete, they must find ways to bring better and greater varieties of products to markets more quickly and at reduced cost.

The primary drivers for the *strategic* business case include environmental concerns and regulations, changing markets (products, etc.), changing customer expectations, new technologies, and increased competition (especially international). Metrics for determining how well organizations are anticipating and responding to these strategic drivers include flexibility, agility, and efficiency (which also translate into speed, quality, and cost).

The *tactical* business case is generally associated with a significant crisis or threat to an organization and demands serious management attention, commitment of sufficient resources, and application of a full-time dedicated staff of competent, experienced personnel. Evidence of improvement (e.g., metrics) includes elimination of re-entry of data ("enter it once, use it many times"), reduction in scrap and rework, reduced cost of prototyping, fewer Engineering Change Orders, and lower warranty costs.

Currently, most manufacturing data are generated manually and data validation is accomplished through physical prototyping. Opportunities for improvement can be identified through developing a baseline that measures cycle times, error rates, scrap, rework, engineering changes, duplication of effort, etc. The potential benefit that could accrue to improve methods can be estimated through benchmarking against organizations judged to be superior in terms of cost, quality, and speed. However, because the METK has not been implemented anywhere, benchmarking cannot provide a complete picture of the potential benefits to be gained through the data validation methodology proposed. The alternative is to use a "best of class" approach that exposes some of the inefficiency in current methods.

Implementation. This group suggested that implementation be accomplished through a relatively small but highly visible pilot project where benefits can be clearly demonstrated. The pilot should permit comparisons against the "as is" approach as well as that of competitors and other benchmarks. The group prescribed the following elements for the implementation approach:

- *Vision* -- senior management must see the strategic need for improved data validation methods and understand its benefits.
- *Business Case (strategic)* -- a legitimate business case must be made that demonstrates beyond reasonable doubt that the improved methods **will** result in reduced cycle time, better quality, and reduced cost that leads to a stronger competitive position.
- *Champion* -- a single individual who if fully committed to the effort must accept primary responsibility for ensuring the success of the project.
- *Catalyst (project)* -- improved data validation methods must be implemented in conjunction with an important, highly visible pilot project that **must** be successful for the organization to gain or maintain the competitive position it seeks; without a compelling reason for improved data validation methods, change is less likely because of the security of the status quo -- typical "catalyst" projects are characterized by high visibility, new products, tight schedules with real deadlines and consideration of entire product life cycle.
- *Tactical Business Case* -- clear and measurable outcomes that are directly linked to improved data validation methods (e.g., reduced data entry, reduced errors, lower prototyping costs).
- *Management Buy-In* -- senior and middle managers must fully support the change to ensure commitment of resources, adequate management attention, and the rewards and incentives to motivate success.
- *Project Planning* -- the project team (led by the "Champion") must develop and follow a well conceived implementation plan that addresses skills, resources, training, infrastructure requirements, tools and technology, and vendors.

A major issue raised during discussion of the implementation approach is the assumption that the METK must be technically mature prior to implementation. Software developers present expressed concern over putting a critical project at risk by using relatively untested software products (e.g., data and application interfaces) that could impede project completion if they fail to perform as expected. The implementation group expressed their conviction that no software products will be taken seriously by top managers until the products are technically mature and proven to be effective. The consensus of the forum is that software development and testing must occur in a relatively low-risk environment where adequate control can be maintained; implementation will be most successful where well-tested software proves to be the critical success factor in a high visibility, time constrained effort.

The implementation process will likely involve top managers (sponsorship), a champion who sees the project through to completion, end users, functional managers, vendors, customers, and suppliers. The implementation team should include representatives from marketing, manufacturing engineering, design, systems planning, quality, reliability, test, and production.

Some discussion revolved on how vendors will work together to solve the integration and interface problems faced in developing and implementing the data validation approach. Will a commercial entity emerge that will develop the "glue" that allows different applications to access and exchange information? Will selected vendors choose to work together to develop a neutral interface that will allow application to work together? What role will NIST or other government entities (e.g., NSWC, USTACOM) play in moving the data validation methodology forward? These questions remain as do a number of options for developing the validation methodology.

Validation

The validation group reviewed manufacturing data validation needs and developed scenarios for applying the validation methodology presented. The group used the following additional guidance to assist their discussion:

1. *Review the list* of manufacturing data to be validated and determine what, if anything, is missing.
2. Describe and diagram the *typical sequence* in which manufacturing data are generated during manufacturing engineering planning.
3. Develop a *preferred mapping* of manufacturing data validation activities to the sequence developed above.
4. As appropriate, cluster data validation actions into *logical groupings* designed to identify and resolve manufacturing data problems as early as possible in the planning process.

The breakout group examined the list of manufacturing data types presented in the data validation methodology. This initial list contained the following items:

1. Manufacturing order
2. Routing sheet
3. Stock material specification
4. Intermediate workpiece geometry and shape information
5. Operation sheet
6. Machine setup sheet
7. Workpiece setup sheet
8. Tool list
9. Fixture list
10. NC program

The validation group added several additional items to the list of manufacturing data to be validated. These items are list below:

1. Lot Size
2. Availability of material and time frame
3. Manufacturing History
4. Cost Estimate, Customer Quote, Final Cost
5. Tool Preset Information, Fixturing Data
6. Consolidated tool list
7. QA Validation
8. Process Instructions
9. Environmental Issues
10. Bill of Materials

The group acknowledged that lot size is important in manufacturing planning because the choice of processes and tooling may vary depending on the length of the production run. Low volume and/or prototype products may be manufactured using methods optimized for quality and time to initial production; higher volume products may be optimized for quality and unit production costs and unit production time -- and may require additional capital investment.

To better understand when and how these manufacturing data might be validated, the group developed an illustrative sequence that shows where various manufacturing engineering data are generated during manufacturing engineering planning. This sequence, shown in Figure 1, was modeled after processes used by members of the validation group during their product development cycle. The group noted that substantial time may elapse between any of the major milestones (noted as "Status x") due to customer delays in authorizing manufacturing planning, delays in obtaining materials, or delays in scheduling production after manufacturing planning is completed.

The group used this manufacturing planning sequence as the basis for identifying areas where manufacturing data validation is particular critical or difficult. Participants responded to the question:

> *What are the most likely sources of error/problems in manufacturing engineering data? (problems/error types: feasibility, high cost, quality, time to release)*

A total of 30 responses were offered and then grouped into similarity categories as shown in Table 1. Two of the responses were general in nature, noting that the validation process and resulting problems and errors are different depending on whether manufacturing data are generated manually or using automated methods. The group was not able to take definitive positions on which errors/problems are more likely with automated versus manual data generation system. The group did conclude that while automated systems are clearly able to develop manufacturing data more rapidly and to consider many factors simultaneously, detection of some critical manufacturing data errors (e.g., errors in understanding and in some assumptions) may be difficult to automate; manual systems, while more time consuming, may be more likely to detect these types of errors before they become manufacturing or production problems.

The group noted that errors that occur in categories A through D are largely associated with input to the manufacturing planning process. These errors concern the availability, accuracy, timeliness, completeness, and consistency of data provided to the manufacturing engineer about the product and the manufacturing environment (e.g., processes available and their capabilities). Categories E through G are areas were the manufacturing engineer can introduce errors by selecting the wrong processes, operation sequences, tooling, or fixtures or by generating inaccurate NC programs. These errors typically occur between "Status 2" and "Status 0" shown in Figure 1.

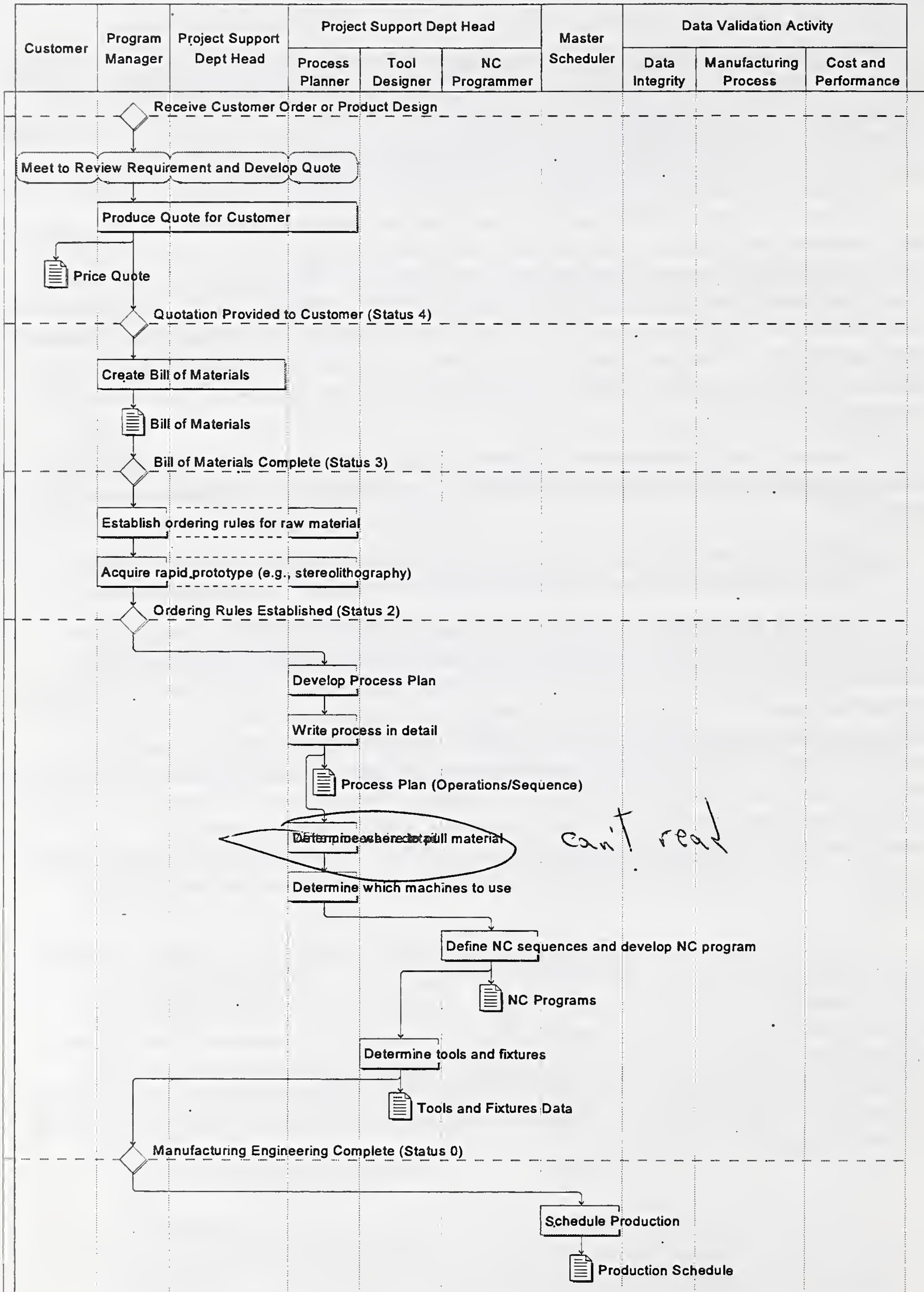


Figure 1. Example Manufacturing Engineering Data Generation Sequence

Table 1. Likely Sources of Errors/Problems in Manufacturing Engineering Data

Item	Description
	Category A: Understanding
A1	Improper understanding or expectation of process capabilities
	Category B: Assumptions
B1	Incorrect assumptions
B2	Wrong assumptions
B3	Change in state of production facility
	Category C: Data
C1	Data sources -- late engineering data
C2	Data sources -- quality of data -- poor GD&T; improper tolerances
C3	Wrong inputs, e.g., selection of a tool which is not available
C4	How to get the right data from the right place at the right time
C5	Incomplete data
C6	Wrong input data
C7	Data -- lack of or wrong
	Category D: Producibility
D1	Design does not lend itself to efficient manufacturing
D2	How to validate process plan for producibility
D3	How to validate design for manufacturability
	Category E: Process Plan (steps/sequence)
E1	Variety of configurations
E2	Process/operation sequence
E3	Omission of operation steps
E4	Process instructions
	Category F: Process Implementation (tools, NC code, fixtures)
F1	NC Program errors
F2	Tool list errors
F3	Fixtures incomplete or wrong
F4	Required accuracy not met
F5	Machine tool selection
F6	Machine is not capable of performing process at all or to the desired accuracy
F7	Wrong parts ordered
	Category G: QA, Cost, Performance
G1	Impact on cost, time, and quality
G2	Quality/inspection requirements
G3	Quality assurance validation

The validation breakout group next examined the recommended data validation methodology by responding to the following question:

What are the two most important concerns/issues with the validation methodology presented?

The responses to this question were grouped into three major categories corresponding to the level at which the concern or issue should be addressed. These responses are shown in Table 2.

Table 2. Most Important Concerns/Issues with the Validation Methodology

Item	Description
	Category A: Overarching Issue and Concerns
A1	It should be the position of METK program to do as much as possible using conformance testing [1] of the application tools
A2	A single validation methodology may not be appropriate; probably needs different methods
A3	How efficient is the validation method -- how quickly are errors detected?
A4	The ultimate validation is the part; the methodology does not include this -- the part should change assumptions
	Category B: Validation Methodology Application Issues and Concerns
B1	Focus is on small machined parts/small batch manufacturing -- is this acceptable?
B2	Difficulty in applying a methodology if it requires a major change in the way people work
B3	There are different types of data that require different validation methods -- the proposed methodology [2] offers only one approach
B4	When to apply the methodology -- conformance testing vs. point of generation vs. time of use
	Category C: Validation Methodology Operational Issues and Concerns
C1	Process planning data is based on assumptions about lot size and schedule
C2	Metal removal rates may be modified per setup (the speed/feedrate may be adjusted depending on the tool conditions -- rigidity & precision of the tool)
C3	There may be different routings depending on the variables -- must have the options of when and how to perform validation of routing data

[1] Conformance testing refers to validating application software that generates the manufacturing engineering data rather than validating the data that are produced by the application software.

[2] Further explanation of this item: Different data can be validated by different methods. Pick the best method for the data. Methods may vary from conformance testing of the generator (e.g., no validation) to validate every time the process plan is created.

Item	Description
C4	Documentation of process plan, NC program, and tooling must be consistent. For example, if tooling changes, the NC program may have to be re-generated and validated
C5	Validation procedures need to be established including test cases (machine features, machine tool, tooling, speed/feeds, etc.) and testing methods. Reference and use benchmark test cases and test methods if available
C6	Need for feedback between major functions such as design systems, process planning and simulation systems

The validation group summarized their most important concerns as:

- Consider conformance testing explicitly in the methodology
- Determine when best to apply the methodology (conformance testing, data generation, use of data)
- Establish test cases
- Ultimate validation is the part -- need to incorporate feedback from production
- Routing vs. NC generation -- need to address specific documents
- May need multiple methodologies
- Need to show feedback model to show how errors detected affect design and manufacturing planning
- Process may be different based on lot size

Additional issues and concerns identified during the validation group outbrief to the plenary session are:

- Manufacturing engineering planners must understand both the design and manufacturing problem (customer requirements)

- The requirements of the engineering data validation methodology need to be solidified and documented to guide developers and users in understanding what to develop and what the benefit will be

- Few models are available to adequately test the efficacy of the validation methodology -- may require a creative design of experiments to capture effects of the validation methodology on cost, quality, and cycle time

- May want to use quality functional deployment (QFD) to link validation methodology requirements to customer requirements

Architecture

The architecture group was assigned the task of reviewing the proposed METK and testing it by illustrating how the data generation and validation applications and data would function within the architecture. Additional guidance provided to this group was:

1. Identify needed *functionality that is not provided* by the proposed METK architecture.
2. Identify *issues not fully addressed* by the proposed METK architecture.
3. List the *"hot" issues* that influence development of the METK architecture and should be given special consideration.
4. *Test the METK architecture* by illustrating how manufacturing data preparation and validation might work within the proposed architecture. Surface and describe problems and opportunities for improvement.

The architecture breakout group was generally favorable toward the METK architecture presented (see Appendix F). Specific issues discussed and concerns expressed include:

- More detailed description of the architecture to better understand how applications and data will be accessed and linked
- Clarification of "application interfaces" and their role; distinction made between integration which occurs at a high level and interfaces which require detailed description
- "Push vs. pull" of application interfaces -- bi-directional flow needed
- Competing needs of short and long-term goals
  > Extend existing prototype
  > General architecture for METK
  > Standard industry agreements and schemas for manufacturing engineering

The group concluded that the METK architecture needs to be described at different levels: process, services, and mechanisms. Figure 2 illustrates these three levels and the issues to be addressed at each level. In addition, the group recognized the importance of knowing what is in scope and out of scope for the METK project and architecture and the need to be aware of the needs of other projects and systems. Figure 3 illustrates how METK fits within a broader view of product design and manufacturing.

Several vendors participated in the architecture and offered their suggestions regarding how vendors could best support METK development. Specific concerns and suggestions that are particularly relevant to vendors are:

- Specific changes to tools for METK are not considered viable <u>unless</u> they provide broad appeal to customers.
- External translators between formats are more feasible in the short term.
- Vendors are currently working with industry consortia and standard groups who require common formats or data translators
  > METK should leverage this work
  > Vendors would be willing to provide this data.
- To reduce the burden on vendors, any common data formats and schemas defined by the METK project require vendor input to ensure they are "close" to current tool data.
- Customer expectations about translators and input/output formats are very high
  > Bi-directional data conversion
  > No loss of information

  > Performance, performance, performance, . . .

- External control of some data sources is needed to avoid re-entry of data (e.g., current shop floor layout, tool catalogs, others, . . .)
- Specific actions that vendors could carry out for the METK project include
  > Deneb to come up with an ACIS translator to aid conversion and integration of data
  > Work with Matrix to look at moving translation between formats as part of use of Matrix (i.e., remove manual input to start the translation)

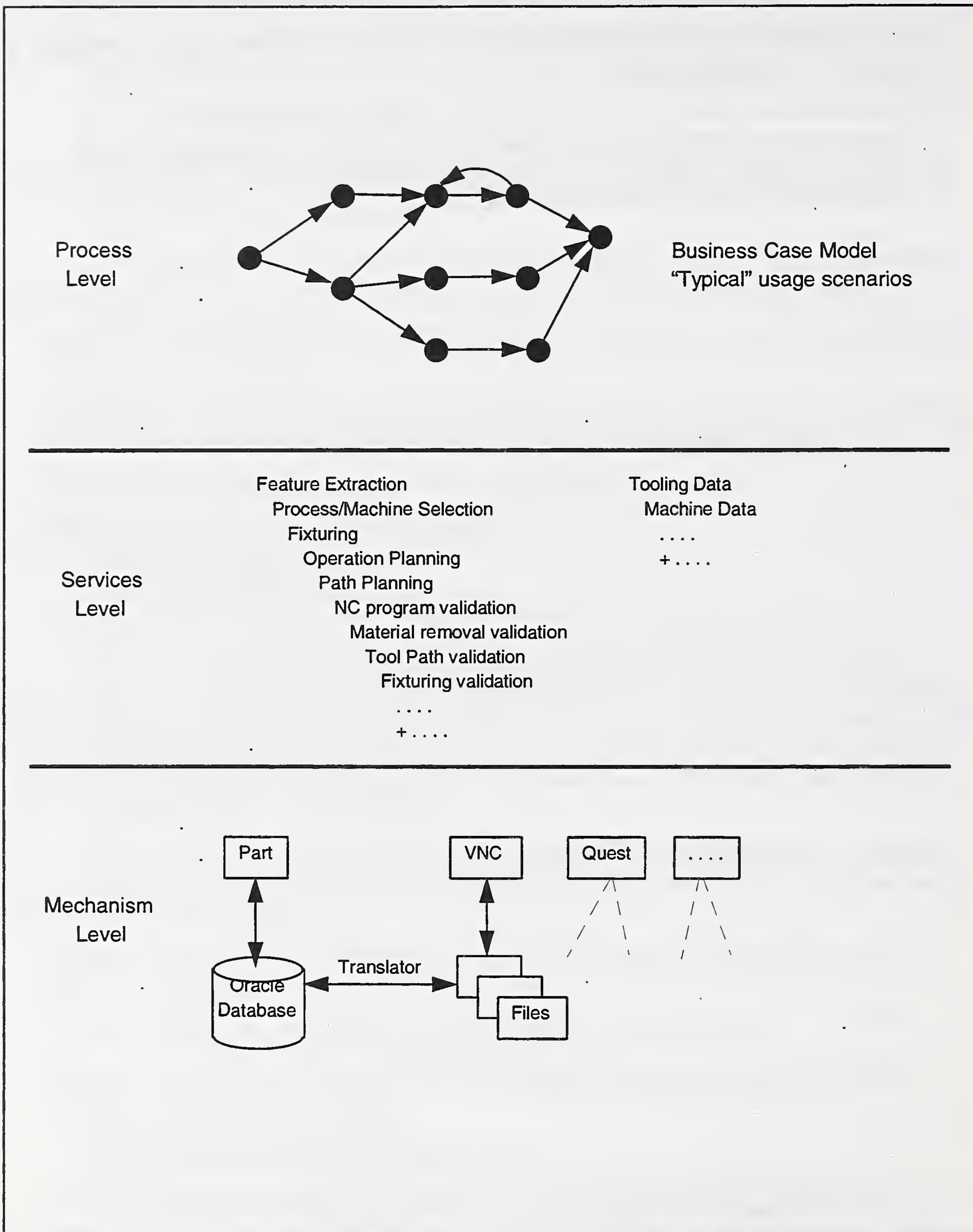


Figure 2. METK Architecture Levels

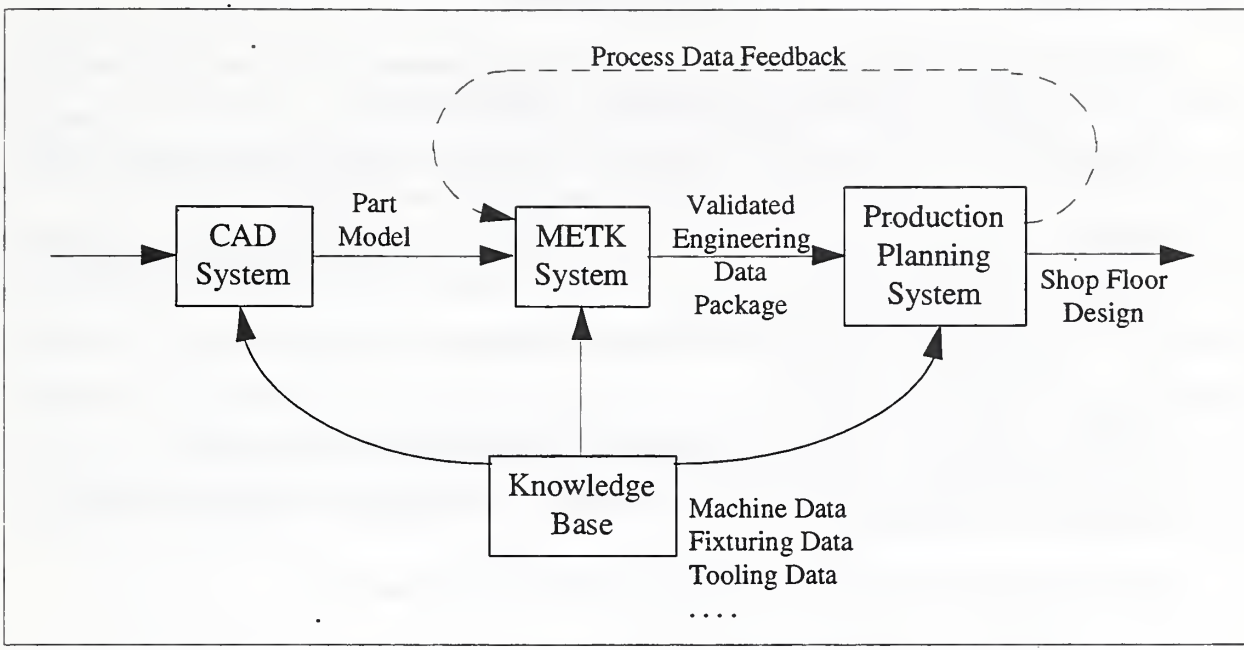


Figure 3. Boundary of METK Project

The architecture group concluded that several intermediate products are needed to ensure successful development of the METK architecture. These are

- Expertise in each of the individual tools as a precursor to the integration effort
- A more detailed list of services of METK
- Mid-level view of architecture at the services level (more detailed than current design)
- Data schemas and structures for any shared data
- Typical usage scenarios
  > Clearly illustrate where integration benefits lie
  > Provide actual job shop example
  > Develop enhanced scenario with the specific tools chosen (Matrix, Part, Deneb)
- A prioritized list of improvements that can be made to the current integration prototype.

An excellent summary of the major architecture issues is provided in Appendix H, "Integration Aspects of the METK Project," prepared by Alan Brown, facilitator for the Architecture breakout group. This briefing provides a context for discussing integration and architecture design issues by describing the range of integration approaches and the roles various organizations might play. Much of the thinking in this paper was captured in the report of the architecture group.

Next Steps

The second technical meeting of the CAME Forum concluded by discussing outstanding issues that need further development and discussion. These issues are:

- Further discuss and develop the real business issues that will drive implementation of the manufacturing engineering data generation and validation approach.
- Discuss the roll-out of METK system versus the needs of the participating organization.
- Discuss how to get vendors together to
  - > Identify and resolve interface/integration issues
  - > Provide input to NIST
- Vendors need better (and more concrete) guidance on what needs to be done.
- Help vendors find the areas of greatest payoff for collaboration -- what is the business case for their investment?

The next steps in the METK development that need to be taken are

> Develop a business process model.
> Identify additional test parts for validating the methodology.
> Refine the state transition diagram for manufacturing engineering data generation and validation.

Appendices

Appendix A:

CAME Forum Second Technical Meeting Agenda and Participant List

NIST Computer-Aided Manufacturing Engineering Forum

Technical Meeting Agenda

Tuesday, August 22, 1995

7:45 a.m.	Continental Breakfast	
8:30 a.m.	Welcome and Introductions	C. McLean
8:45 a.m.	Meeting Objectives and Overview	M. Smith
9:00 a.m.	Project Status Update	S. Leong
9:15a.m.	Manufacturing Validation Issues	C. McLean C. Chen
10:00 a.m.	Implementation Plan & Issues	S. Leong
10:30 a.m.	Break Transportation provided between hotel and NIST	
10:45 a.m.	Baseline System Demonstration (at NIST) 1. Matrix / ICEM_Part / Deneb VNC / Deneb Quest 2. Model shop and layout 3. Shop scenario	MSE[1] Group
12:00 p.m.	Lunch (at hotel)	
1:00 p.m.	System Integration and Interface Issues Overview	Frank Riddick Alan Brown
2:00 p.m.	Breakout Groups Discussion 1. Systems Integration Issues Group 2. Manufacturing Validation Group 3. System Implementation Issues Group	All
4:30 p.m.	Session Wrap-up	M. Smith
5:00 p.m.	Adjourn	

[1]Manufacturing Systems Engineering

NIST Computer-Aided Manufacturing Engineering Forum

Technical Meeting
Preliminary Agenda

Wednesday, August 23, 1995

7:45 a.m.	Continental Breakfast	
8:30 a.m.	SIMA[2] Production System Engineering Environment	C. Mclean S. Leong
9:15 a.m.	Breakout Groups Reconvene (to prepare outbrief)	All
10:00 a.m.	Break	
10:15 a.m.	Systems Integration Issues Report and discussion	Spokesperson
10:45 a.m.	Manufacturing Validation Issues Report and Discussion	Spokesperson
11:15 a.m.	Business Case and Implementation Report and Discussion	Spokesperson
11:45 a.m.	Meeting Wrap-up	C. McLean S. Leong M. Smith
12:00 p.m.	Adjourn	

[2]Systems Integration for Manufacturing Applications

NIST CAME Forum Participant List

Name	Organization	Phone	Fax	E-Mail
John Bachinsky	Watervliet Arsenal	518/266-5719	518/266-4555	bachinsky@wva-emhi.army.mil
Donald Bailey	Ohio Aerospace Institute	216/962-3050	216/962-3120	DonBailey@oai.org
Alan Brown	Software Engineering Institute	412/268-6194	412/268-5758	awb@sei.cmu.edu
Chin-Sheng Chen	Florida International University	305/348-3753	305/348-3721	chen@eng.fiu.edu
Mitchell De Jong	ICEM Technologies	612/482-4330	612/482-4899	mitchell.dejong@cdc.com
Don Elson	Black & Decker	410/716-3410	410/716-3653	
John Eskuri	Deneb Robotics	810/377-6900	810-377-8125	eskuri@deneb.com
Shaw Feng	NIST	301/975-3551	301/258-9749	feng@cme.nist.gov
Simon Frechette	NIST	301/975-3335	301/975-9749	frechette@cme.nist.gov
Jeremiah Halley	McDonnell Douglas	314/234-9874	314/232-5443	
John Handelman	Ford Company	313/592-2807	313/592-2381	jhandelman@rrm.org
David Hyde	Matrix	203/944-2020 x160	203/944-2022	hyde@matrixnet.com
Mike Iuliano	NIST	301/975-3891	301/258-9749	iuliano@cme.nist.gov
Robert Judd	Ohio University	614/593-0106	614/593-0007	juddrp@bobcat.ent.ohiou.edu
Swee Leong	NIST	301/975-5426	301/258-9749	leong@cme.nist.gov
Chuck McLean	NIST	301/975-3511	301/258-9749	mclean@cme.nist.gov
Erroll Palmer	Naval Ordnance Station/Louisville	502/364-6592	502/364-6415	erroll_p@louis.nosl.sea06.navy.mil
Charles Parks	Ohio University	614/593-1540	614/593-0778	cparks@bobcat.ent.ohiou.edu
Steve Ray	NIST	301/975-3524	301/258-9749	ray@cme.nist.gov
Frank Riddick	NIST			riddick@cme.nist.gov
Dave Schmitz	DSMC	703/805-3765		schmitzd@dsmc.dsm.mil
David Sly	Cimtechnologies Corp.	515/296-9914	515/296-9909	info@cimtech.iastate.edu
Allison Smith	Framework Technologies	414/276-5120		AJSatFTC@aol.com
Mike Smith	SAIC	703/827-4772	703/893-2187	mike_smith@cpqm.saic.com
Jack Spalding	Litton Amecon	301/454-9526		
David Stieren	NIST	301/975-3197	301/869-3750	dstieren@enh.nist.gov

Appendix B:

"A Brief Review"

NIST

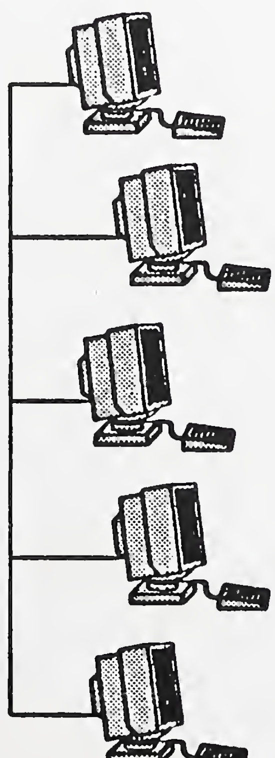
A Brief Review

Chuck McLean
Leader
Manufacturing Systems Engineering Group

CAME Industry Meeting
August 1995

MEL

Computing Environments in the Manufacturing Enterprise



- Business Management
- Design and Product Data Management
- Manufacturing Engineering
- Production Planning and Control
- Distribution and Logistic Support

NIST

Navy CAME Project

Goal:

To lower manufacturing costs, reduce delivery times, and improve product quality through the coordinated development and use of advanced software tools

Strategy:

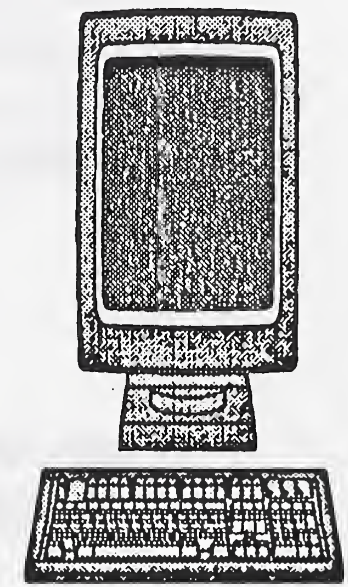
- Establish consortium of users, researchers, developers, and vendors
- Develop architectures, databases, and techniques for tool integration
- Develop common data bases for manufacturing engineering data
- Construct prototype tool kit environments using COTS tools
- Test and validate tool kits using real world data
- Develop solutions which are applicable to large and small shops alike
- Recommend potential standards based upon results

MEL

NIST

Manufacturing Engineering Tools

- Limited number of tools are available
- Some problems with existing tools include:
  - *Not widely used*
  - *May be difficult to learn*
  - *Per site costs are high*
  - *Many capabilities have not been computerized*
  - *Many techniques are not well-defined*
- The most significant problem is that existing tools are not integratable, i.e., they do not work together!



MEL

Industry Needs

- Integrated engineering tool kits and databases
- Capabilities to reliably predict manufacturing system behavior
- Standard manufacturing engineering databases
- Improved tools for performing a number of engineering functions, for example: producibility analysis, tolerance analysis, process planning, system simulation, tool and fixture design
- Plug-compatibility between commercial engineering tools

NIST

Typical MSE Applications

System Level	Product Concept	Product Development	Production Planning	Production
Equipment	- Equipment concepts	- Tolerance studies	- Tool design - Equipment selection	- Capability analyses - SPC - Shop Floor Data collection
Process	- Process concepts	- Producibility analyses - Design for X	- Process planning - Inspection procedures	- Process mapping - Time and motion
Cell / Department	- Cell concepts	- Cell design	- Assembly line design - Equipment selection - Cell layout - Scheduling	- Performance studies - Quality standards
Factory	- Capacity analysis - Factory concepts	- Site location	- Plant layout - Make-or-buy - Material handling	- Performance studies - Incentive programs
Enterprise	- Production strategy	- Cost estimating - Teaming	- Outsourcing - Quality management	- Benchmarking - Supplier tracking

MEL

The CAME Solution

Computer-Aided Manufacturing Engineering (CAME) is broadly defined as:

the use of computerized tools in the application of scientific and engineering methods to the design and implementation of manufacturing systems, subsystems, processes, and equipment.

The overall goal of CAME is to lower manufacturing costs, reduce delivery times, and improve product quality through the coordinated development and use of advanced tools.

CAME Program Focus

The CAME Program will place an emphasis upon providing:

- an integrated framework,
- operating environment,
- common databases, and
- interface standards

for a wide variety of emerging tools and techniques for designing manufacturing processes, equipment, and enterprises, as well as tools for evaluating the producibility of product designs.

The "Vision"

An integrated computing environment for manufacturing engineering which includes:

- engineering tools
- other application software
- databases and reference libraries
- operating systems
- computer and communications hardware

Who will use CAME?

Current users:
- Manufacturing Engineers
- Design Engineers
- Quality Engineers
- Industrial Engineers
- System Engineers
- Process Planners
- Tool Designers
- Etc.

Additional future users:
- Manufacturing Strategists
- Producibility Engineers
- Enterprise Designers
- Benchmarkers
- Etc.

METK Project Abstract

- Construct an integrated manufacturing engineering tool kit and common databases that provide: 1) product data and work flow management, 2) process planning, 3) engineering data validation through simulation, and 3) possibly other capabilities for a family of machined parts.

- Functionality will be based upon extensions to the capabilities of commercial off-the-shelf software and hardware.

- Conduct the project as a collaborative effort by users, vendors, academic researchers, and representatives other government agencies in project planning, requirements definition, system design, development, testing, and evaluation.

- Perform "alpha" testing and evaluation of the tool kit environment at industry and government manufacturing sites.

- Recommend new industry standards based upon project results.

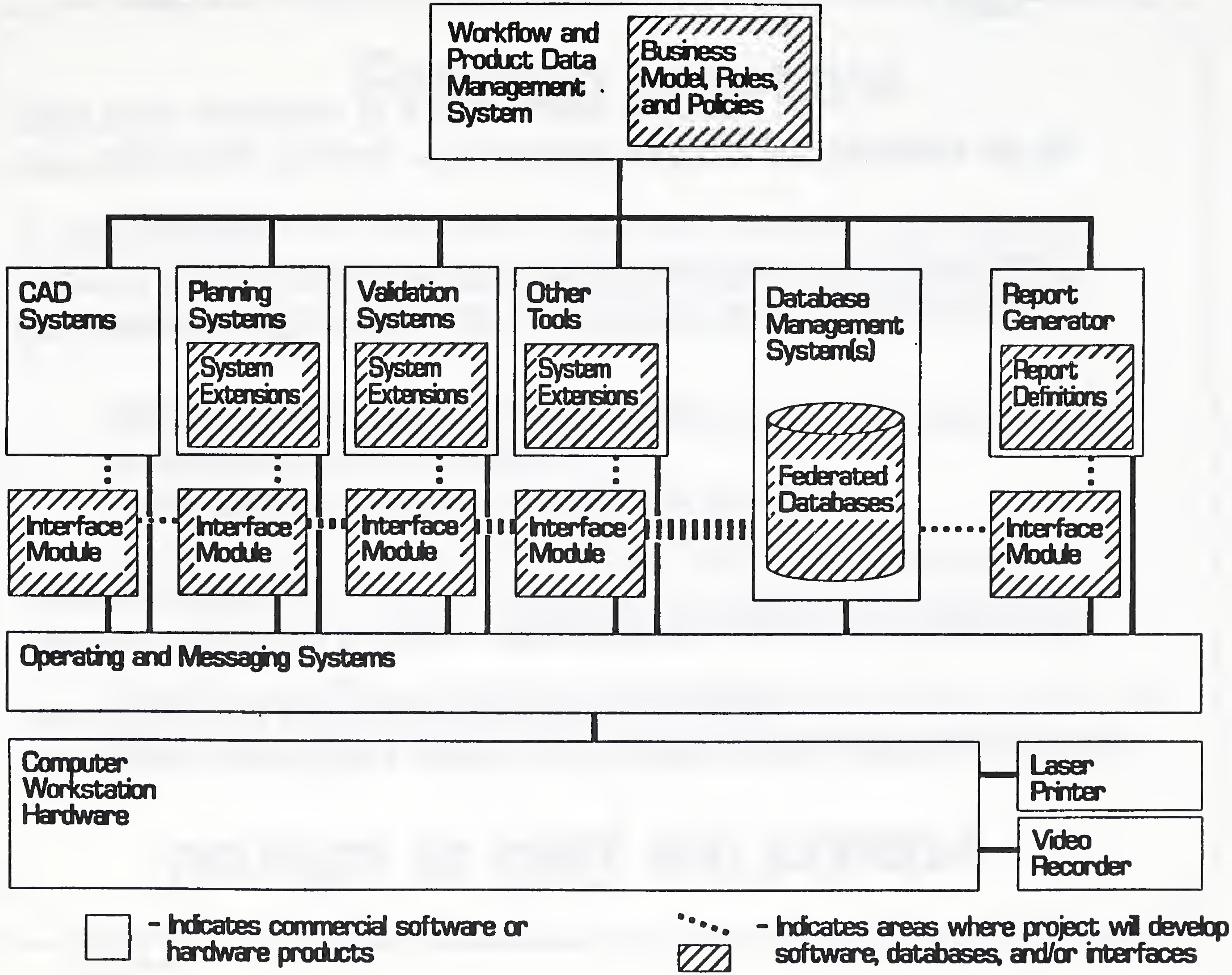
NIST

Requirements Synopsis

- Loading of product design data from several CAD formats
- Management of product, process, and engineering change order data
- Workflow management for planning and validation processes
- Planning of routings and operations for machined parts
- Selection of resources, e.g. material stock, tools, fixtures
- Validation of engineering data through simulation, checklists, etc.
- Management of resource data required for planning and plan validation
- Preparation of hard copy reports, e.g., plans, tool lists, inventory status
- Recording of simulation runs on video tape

MEL

System Concept for the Manufacturing Engineering Tool Kit



Benefits to DoD, and Industry

Widespread Applicability - Applies to DoD sites, primes and subcontractors, and non-defense commercial manufacturing facilities.

Improved Engineering Function - Manufacturing engineers will make better decisions resulting in:

- more products produced correctly the first time,
- overall reduction in engineering time,
- fewer scrapped parts and less rework.

Productivity Benefits - Better utilization of shop floor equipment, quicker response for rush orders, shorter development times and higher quality for new products.

New Engineering Products - Commercial software and hardware will be made more accessible to a larger group of users.

NIST

Research Directions

- Identification and evaluation of new tools, methods, and algorithms for rapidly translating product design data into manufacturing plans, process designs, and system specifications

- Modeling and evaluation of manufacturing engineering solutions using simulation and analysis software

- Development of information/functional models, databases, and architectures to support engineering and planning tool integration

- Establish repository of engineering information and process models and testbed for evaluating integration solutions, conformance testing, etc.

MEL

Appendix C:

"Project Status"

Project Status

Swee Leong
Manufacturing Engineer

CAME Industry Meeting
August 1995

METK Project Status

1. Welcome New Visitors/Participants

- McDonnell Douglas
- Ford Motor Company
- Ohio Aerospace Institute
- Ohio University
- CimTechnologies
- Framework Technologies
- New NIST Staff

2. Test part/manufacturing process data status

- Product model
- Drawing
- Machine/tooling/fixture
- NC program
- Routing sheet

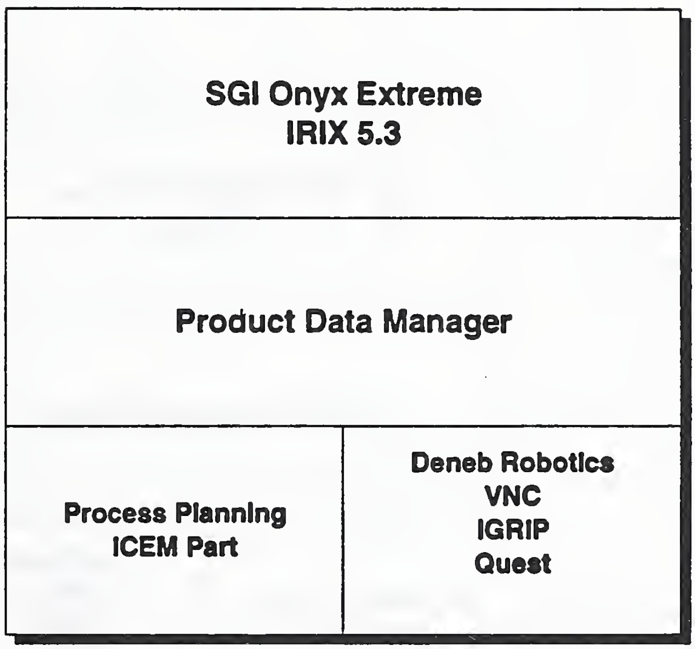
3. CRADA / MOU / Contract

4. System Baseline Overview

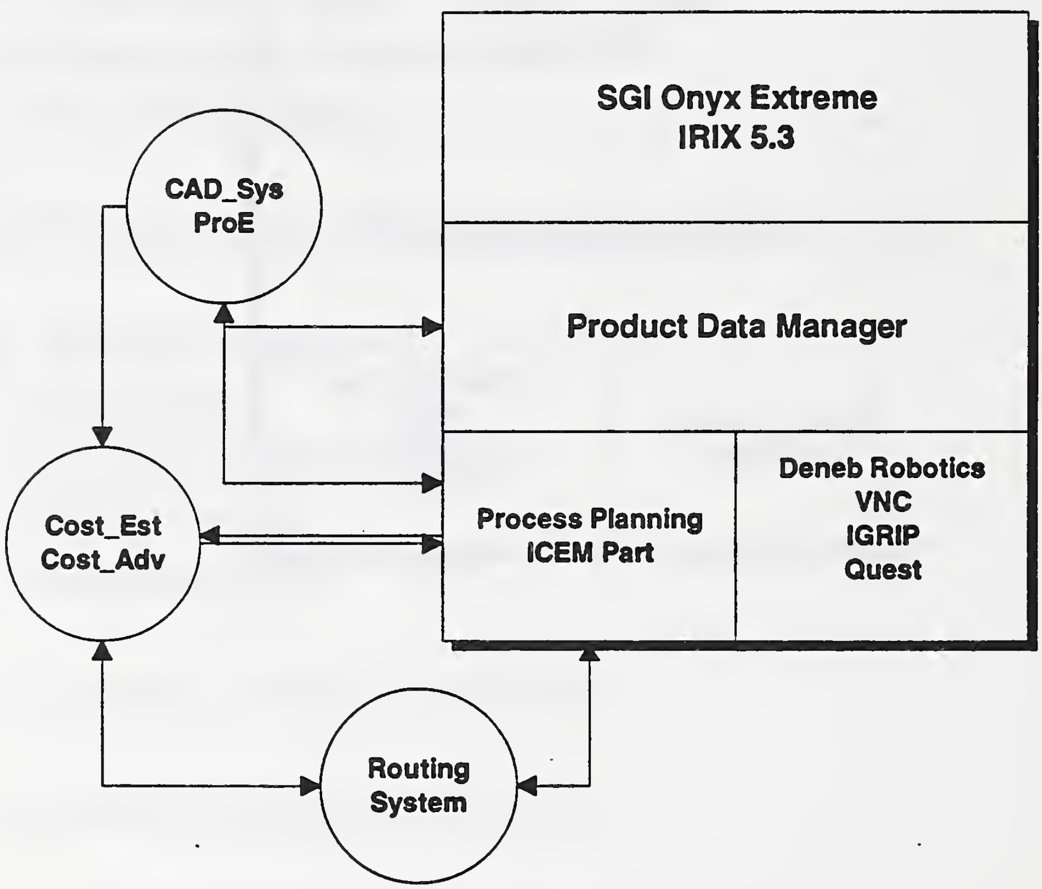
5. System Baseline Demo

6. Project Schedule

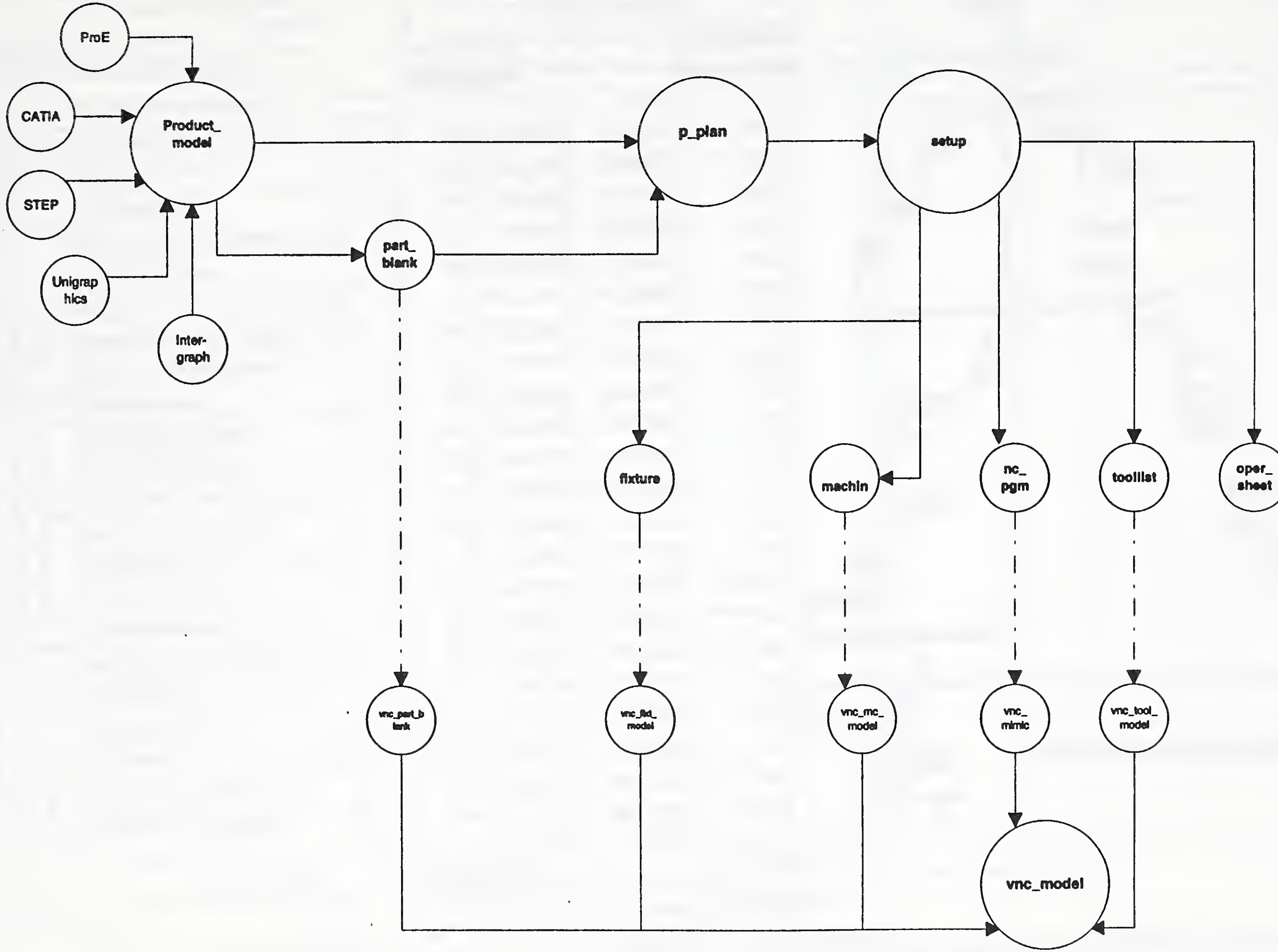
Baseline System



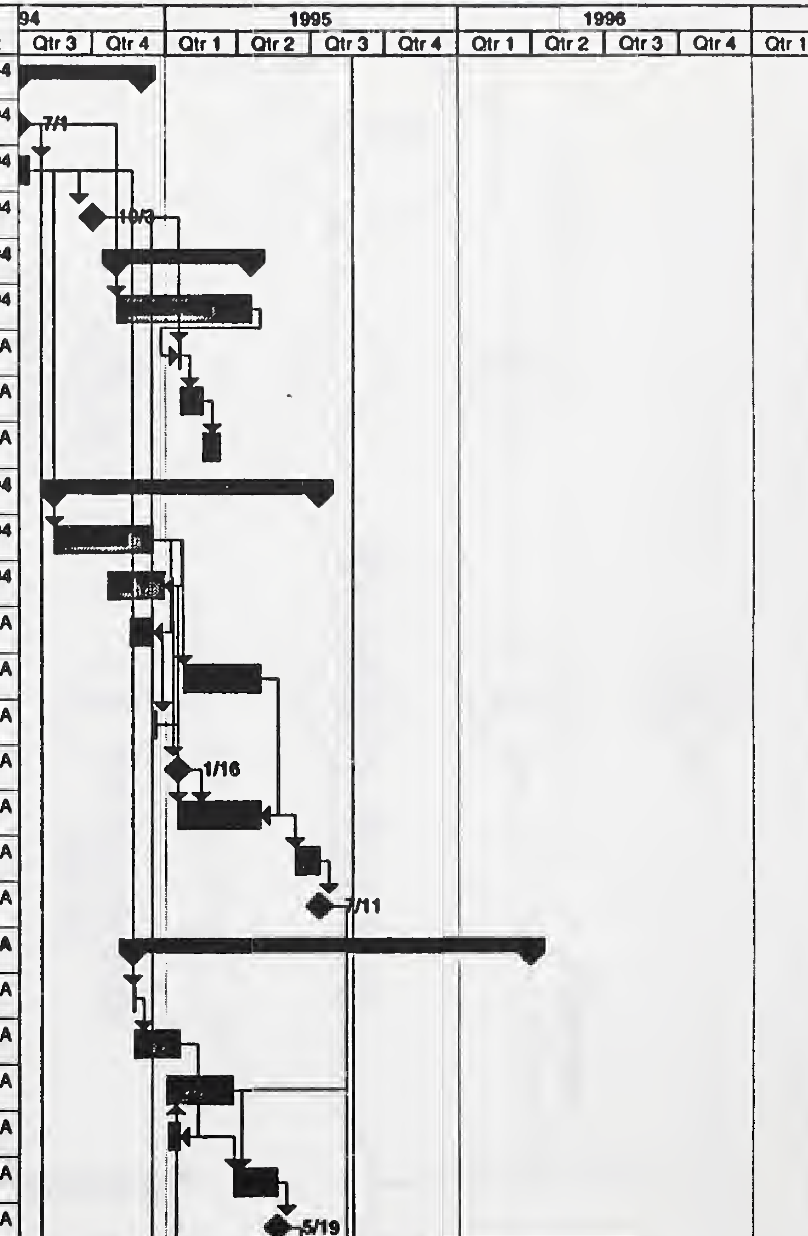
Baseline System



Data Schema for Baseline System Demo



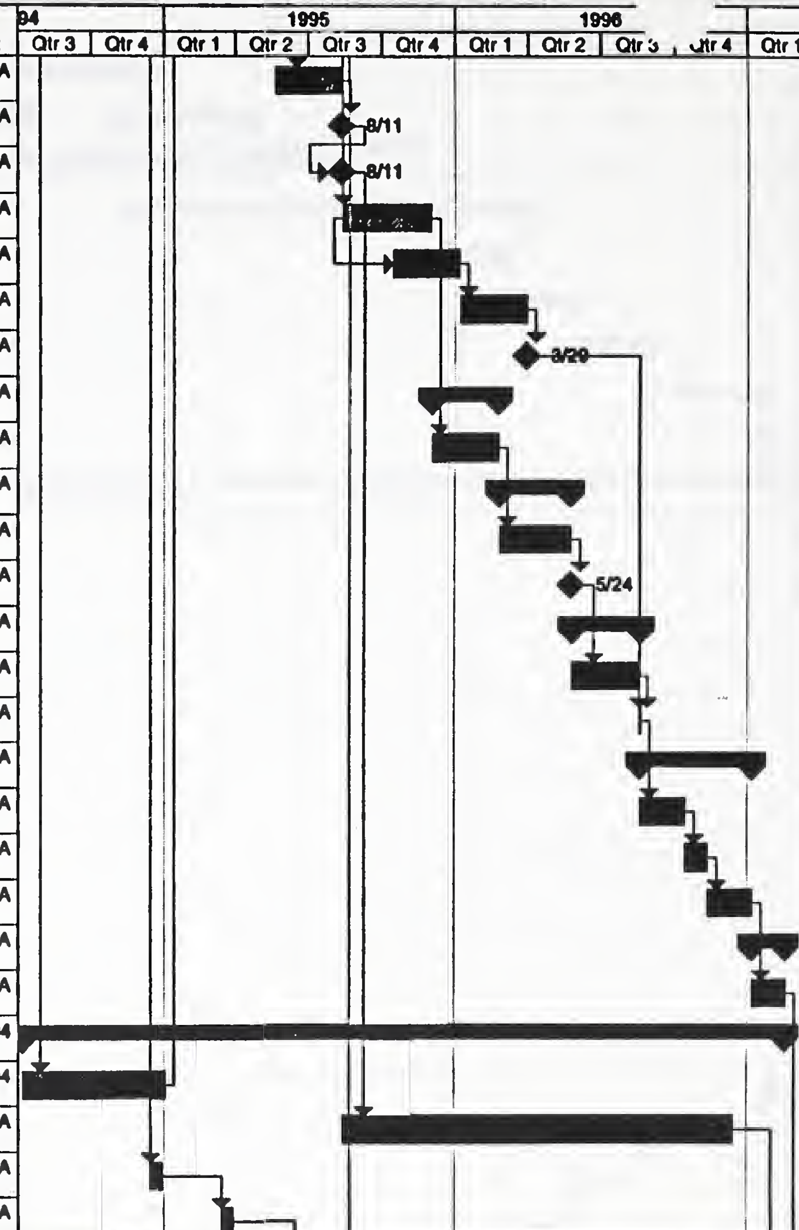
ID	Task Name	Group	Duration	Start	Finish	Actual Start
1	**T1. Prepare & maintain technical development**		111d	Fri 7/1/94	Fri 12/2/94	Fri 7/1/94
2	Project started	T4	0d	Fri 7/1/94	Fri 7/1/94	Fri 7/1/94
3	Define initial toolkit objective/scope/plan	T3	10d	Fri 7/1/94	Thu 7/14/94	Fri 7/1/94
4	Develop detailed project plan	T1	9w	Mon 10/3/94	Fri 12/2/94	Mon 10/3/94
5	**T2. Initiate Industrial collaborations**		120d	Tue 11/1/94	Mon 4/17/95	Tue 11/1/94
6	Organize Industrial consortium	T2	24w	Tue 11/1/94	Mon 4/17/95	Tue 11/1/94
7	Kick off meeting	T2	2d	Wed 1/18/95	Thu 1/19/95	NA
8	Prepare MOU & CRDAs	T2	4w	Fri 1/20/95	Thu 2/16/95	NA
9	Sign agreements	T2	15d	Fri 2/17/95	Thu 3/9/95	NA
10	**T3. Select, acquire, and install baseline systen**		237d	Mon 8/15/94	Tue 7/11/95	Mon 8/15/94
11	Select software systems	T3	12w	Mon 8/15/94	Thu 12/15/94	Mon 8/15/94
12	Procurement of software	T3	10w	Fri 10/21/94	Thu 12/29/94	Fri 10/21/94
13	Purchase hardware	T3	4w	Fri 11/18/94	Thu 12/15/94	NA
14	Attend software training classes	T3	14w	Mon 1/23/95	Fri 4/28/95	NA
15	Install hardware	T3	2d	Fri 12/16/94	Mon 12/19/94	NA
16	AMSANT facility completed	T3	0d	Mon 1/16/95	Mon 1/16/95	NA
17	Install & configure software	T3	15w	Mon 1/16/95	Fri 4/28/95	NA
18	Prepare baseline demos	T3	22d	Mon 6/12/95	Tue 7/11/95	NA
19	Perform initial baseline demo	T3	0d	Tue 7/11/95	Tue 7/11/95	NA
20	**T4. Prepare specifications and contracts**		355d	Mon 11/21/94	Fri 3/29/96	NA
21	Attend COTR	T4	2d	Mon 11/21/94	Tue 11/22/94	NA
22	Develop contracting plan	T4	8w	Wed 11/23/94	Tue 1/17/95	NA
23	Develop preliminary specifications	T4	12w	Mon 1/2/95	Fri 3/24/95	NA
24	Develop bidders list	T4	2w	Wed 1/4/95	Tue 1/17/95	NA
25	Prepare SOW	T4	8w	Mon 3/27/95	Fri 5/19/95	NA
26	RFPs released	T4	0d	Fri 5/19/95	Fri 5/19/95	NA



Project: Manufacturing Engineering Tool Ki
Date: Mon 8/21/95

Task — Summary — Rolled Up Progress
Progress — Rolled Up Task
Milestone — Rolled Up Milestone

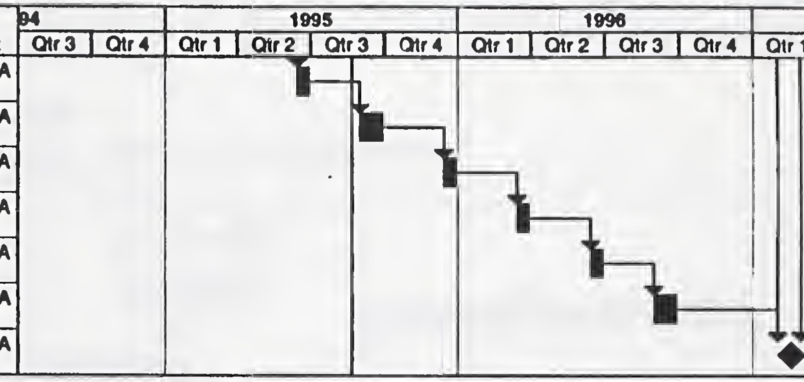
ID	Task Name	Group	Duration	Start	Finish	Actual Start
27	Evaluate & negotiate proposals	T4	12w	Mon 5/22/95	Fri 8/11/95	NA
28	Contracts awarded	T4	0d	Fri 8/11/95	Fri 8/11/95	NA
29	Begin coordination of contractors	T4	0d	Fri 8/11/95	Fri 8/11/95	NA
30	Develop system design document	T4	16w	Mon 8/14/95	Fri 12/1/95	NA
31	Develop unit / integration test plan	T4	12w	Mon 10/16/95	Fri 1/5/96	NA
32	Develop user document	T4	12w	Mon 1/8/96	Fri 3/29/96	NA
33	Complete specification	T4	0d	Fri 3/29/96	Fri 3/29/96	NA
34	**T5. Develop individual tool modules & databas**		60d	Mon 12/4/95	Fri 2/23/96	NA
35	Develop & test software	T5	12w	Mon 12/4/95	Fri 2/23/96	NA
36	**T6. Integrate and test tool modules and databa**		65d	Mon 2/26/96	Fri 5/24/96	NA
37	Perform integration test & revision	T6	13w	Mon 2/26/96	Fri 5/24/96	NA
38	Systems operational and documented	T6	0d	Fri 5/24/96	Fri 5/24/96	NA
39	**T7. Demonstrate system**		61d	Mon 5/27/96	Mon 8/19/96	NA
40	Prepare for pilot system demo	T7	12w	Mon 5/27/96	Fri 8/16/96	NA
41	Demo pilot system @ NIST	T7	1d	Mon 8/19/96	Mon 8/19/96	NA
42	**T8. Conduct "alpha" testing at other sites**		100d	Tue 8/20/96	Mon 1/6/97	NA
43	Conduct outsiders training	T8	8w	Tue 8/20/96	Mon 10/14/96	NA
44	Install selected system @ navy alpha site	T8	20d	Tue 10/15/96	Mon 11/11/96	NA
45	Test and provide feedback	T8	8w	Tue 11/12/96	Mon 1/6/97	NA
46	**T9. Prepare final documentation package**		30d	Tue 1/7/97	Mon 2/17/97	NA
47	Final report	T9	6w	Tue 1/7/97	Mon 2/17/97	NA
48	**T10. Manage tool kit environment project**		682d	Fri 7/8/94	Mon 2/17/97	Fri 7/8/94
49	Hire technical staff	T10	13w	Fri 7/8/94	Sun 1/1/95	Fri 7/8/94
50	Coordinate contractor efforts & review projec	T10	350d	Mon 8/14/95	Fri 12/13/96	NA
51	Prepare January, 1995, progress report	T10	10d	Thu 12/15/94	Wed 12/28/94	NA
52	Prepare April, 1995, progress report	T10	10d	Wed 3/15/95	Tue 3/28/95	NA



Project: Manufacturing Engineering Tool Ki
Date: Mon 8/21/95

Task		Summary		Rolled Up Progress	
Progress		Rolled Up Task			
Milestone		Rolled Up Milestone			

ID	Task Name	Group	Duration	Start	Finish	Actual Start
53	Prepare July, 1995, progress report	T10	10d	Thu 6/15/95	Wed 6/28/95	NA
54	Prepare October, 1995, annual report	T10	20d	Fri 9/1/95	Thu 9/28/95	NA
55	Prepare January, 1996, progress report	T10	10d	Fri 12/15/95	Thu 12/28/95	NA
56	Prepare April, 1996, progress report	T10	10d	Fri 3/15/96	Thu 3/28/96	NA
57	Prepare July, 1996, progress report	T10	10d	Sat 6/15/96	Fri 6/28/96	NA
58	Prepare October, 1996, annual report	T10	20d	Sun 9/1/96	Fri 9/27/96	NA
59	End of project	T10	0d	Mon 2/17/97	Mon 2/17/97	NA



Project: Manufacturing Engineering Tool Ki
Date: Mon 8/21/95

Task

Progress

Milestone

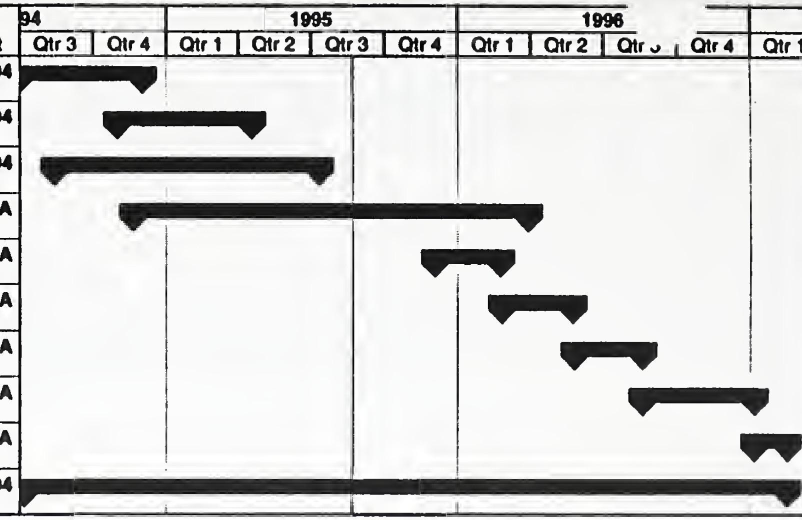
Summary

Rolled Up Task

Rolled Up Milestone

Rolled Up Progress

ID	Task Name	Group	Duration	Start	Finish	Actual Start
5	T1. Prepare & maintain technical development		111d	Fri 7/1/94	Fri 12/2/94	Fri 7/1/94
5	T2. Initiate industrial collaborations		120d	Tue 11/1/94	Mon 4/17/95	Tue 11/1/94
10	T3. Select, acquire, and install baseline system		237d	Mon 8/15/94	Tue 7/11/95	Mon 8/15/94
20	T4. Prepare specifications and contracts		355d	Mon 11/21/94	Fri 3/29/96	NA
46	T5. Develop individual tool modules & databas		60d	Mon 12/4/95	Fri 2/23/96	NA
46	T6. Integrate and test tool modules and databs		65d	Mon 2/26/96	Fri 5/24/96	NA
19	T7. Demonstrate system		61d	Mon 5/27/96	Mon 8/19/96	NA
42	T8. Conduct "alpha" testing at other sites		100d	Tue 8/20/96	Mon 1/6/97	NA
46	T9. Prepare final documentation package		30d	Tue 1/7/97	Mon 2/17/97	NA
46	T10. Manage tool kit environment project		682d	Fri 7/8/94	Mon 2/17/97	Fri 7/8/94



Project: Manufacturing Engineering Tool Ki
Date: Mon 8/21/95

Task | Progress | Milestone | Summary | Rolled Up Task | Rolled Up Milestone | Rolled Up Progress

Appendix D:

"Manufacturing Validation Issues"

Manufacturing Validation Issues

C. McLean
Leader, Manufacturing Systems Engineering Group

Dr. C. Chen
Florida International University

CAME Industry Meeting
August 1995

NIST

Outline

* Background
* Objectives
* Manufacturing Data Package Elements
* Overview of Proposed Methodology
* Issues and Questions

MEL

Background

- The 1993 CAMSE Workshop identified that the inability to reliably predict manufacturing system behavior was the single most important problem faced by manufacturing engineers.

- The March 1995 CAME Forum meeting prioritized machine shop processes, identified types of errors found in manufacturing data packages, and developed a categorization of those errors.

- Followup work at NIST has resulted in a proposed methodology for detecting errors in manufacturing data packages.

Objective

- To define a methodology for the validation of manufacturing data that may be applied using either manual methods or automatic systems to help ensure that machined parts are produced correctly the first time.

TYPES OF MANUFACTURING DATA

1. Manufacturing order
2. Routing sheet
3. Stock material specification
4. Intermediate workpiece geometry & shape information
5. Operation sheet
6. Machine setup sheet
7. Workpiece setup sheet
8. Tool list
9. Fixture list
10. NC program

ENG. DATA WHICH SUPPORT MODELING AND VALIDATION OF MANUFACTURING DATA

1. Product data
    - assembly model
        - administrative data
        - component models
        - mating definition
    - component model
        - modeling (design) feature
        - geometry
        - topology
        - tolerance
        - technical note
        - administrative data

2. Manufacturing resources
    - material
    - machine
    - cutting tool
    - fixture
    - operator

3. Production data
    - inventory
    - customer order
    - production plan
    - production schedule

4. Manufacturing system and resource functional models

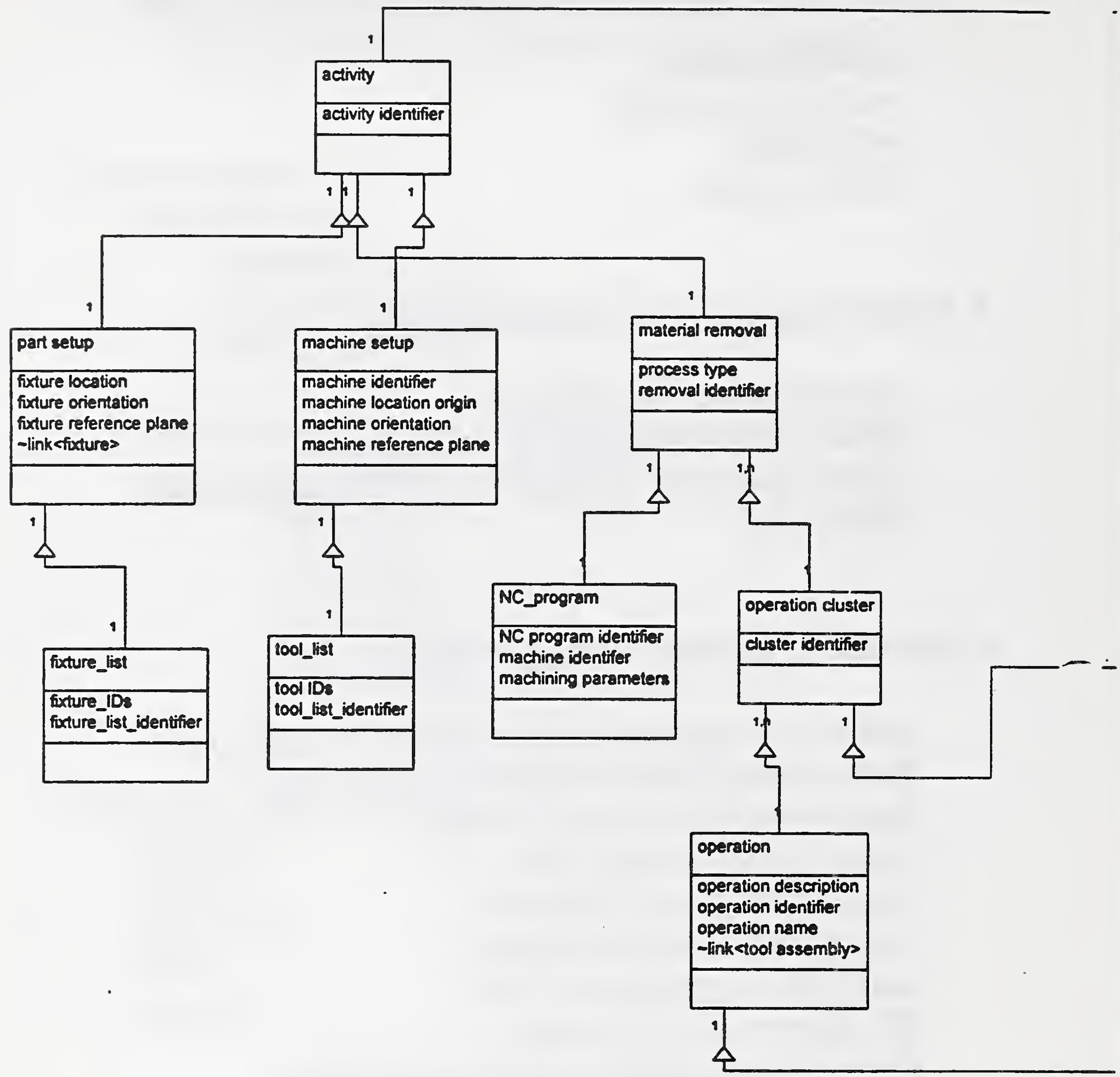
- resource models
- workstation models
- cell models
- factory models

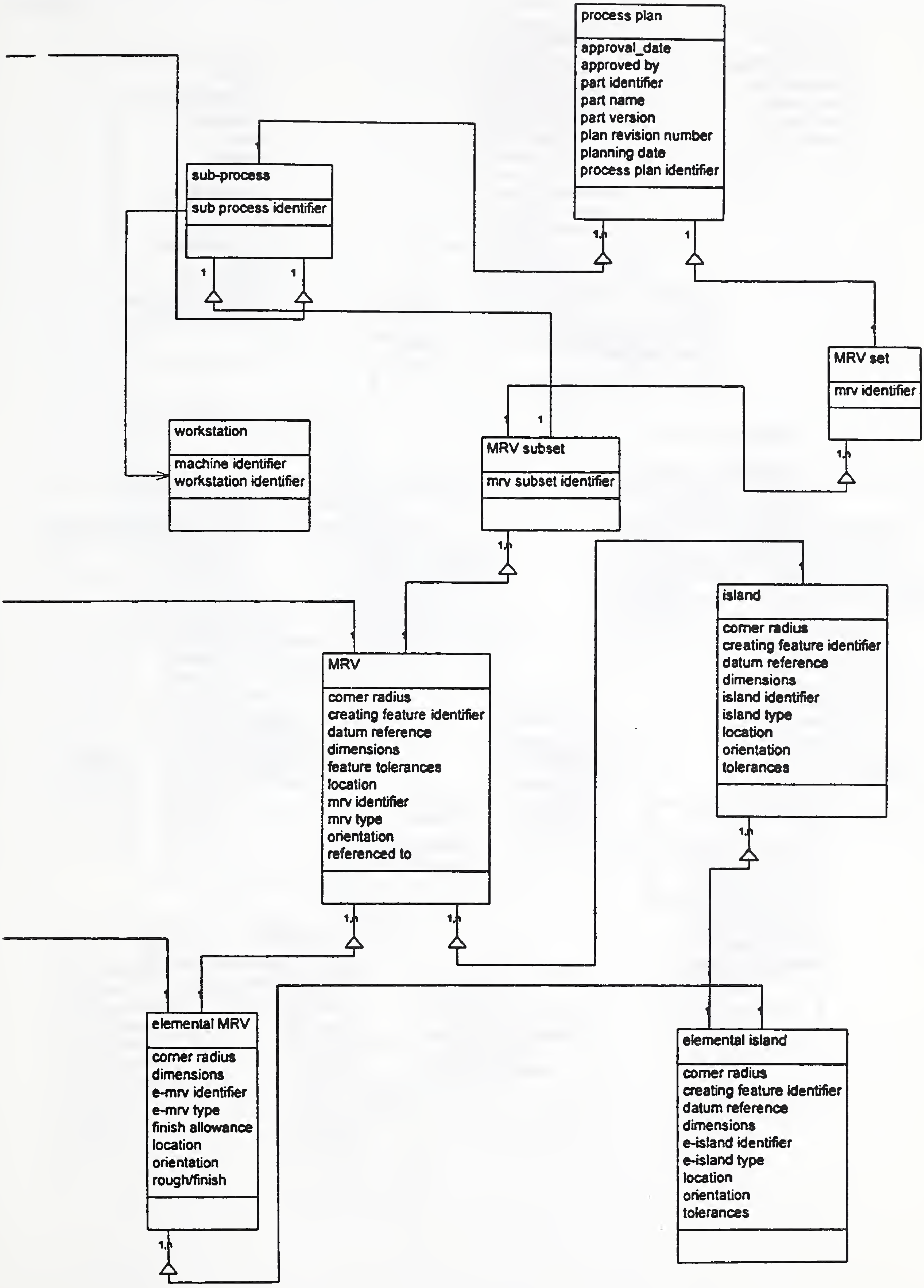
5. Engineering process management data

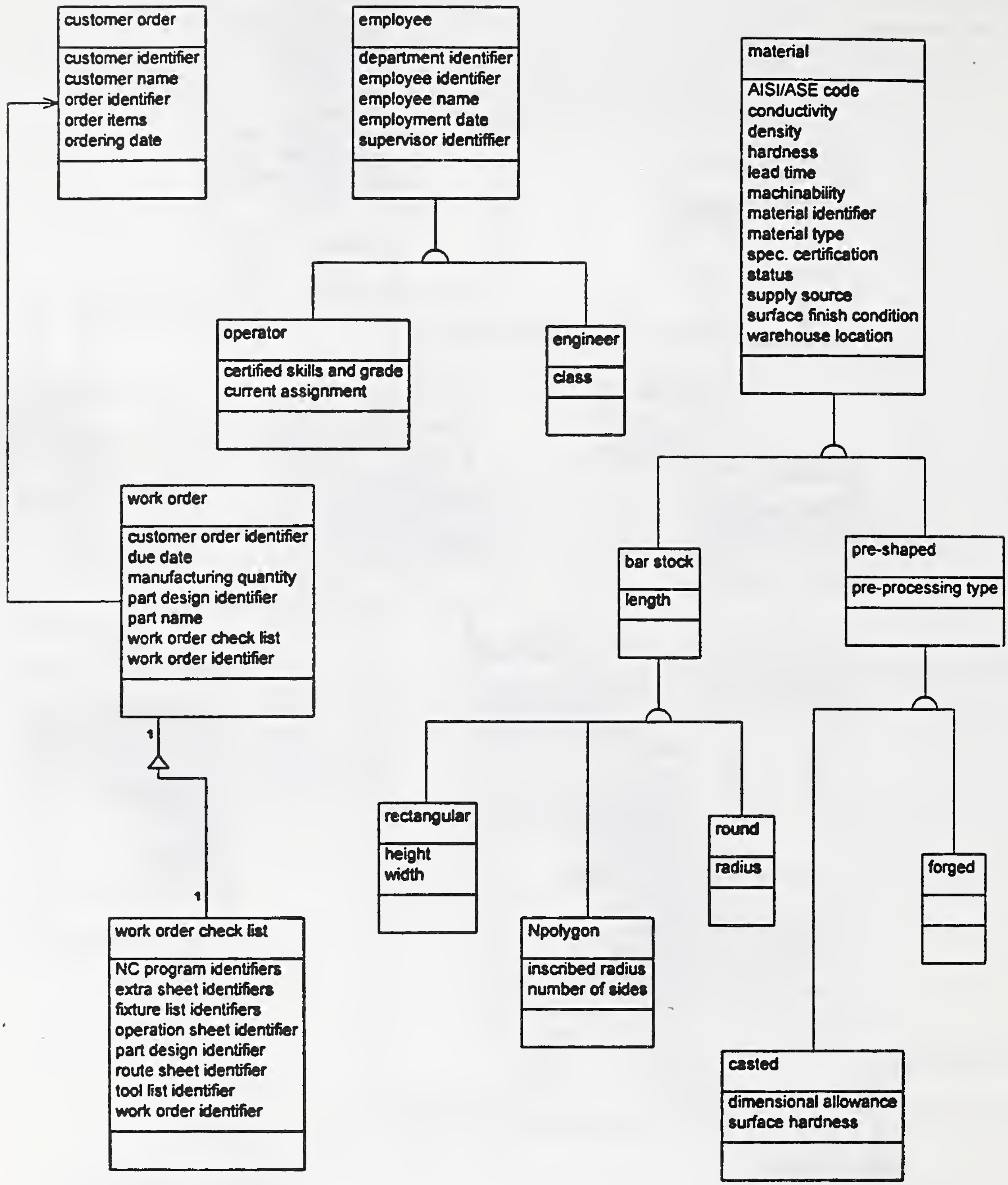
- engineering activity diagram
- engineering information flow diagram
- engineering process control model (business model)
- decision log

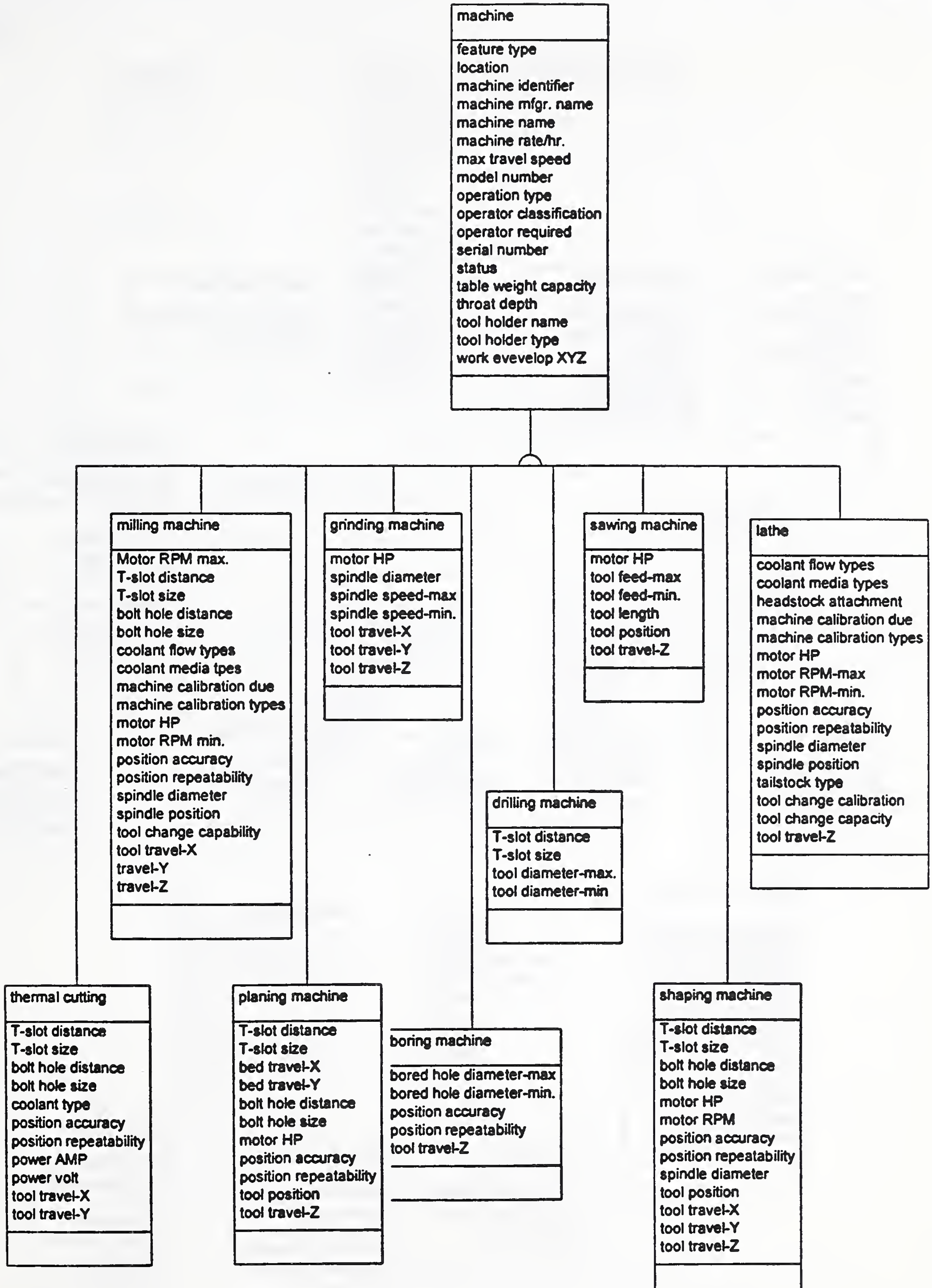
6. Manufacturing engineering knowledge

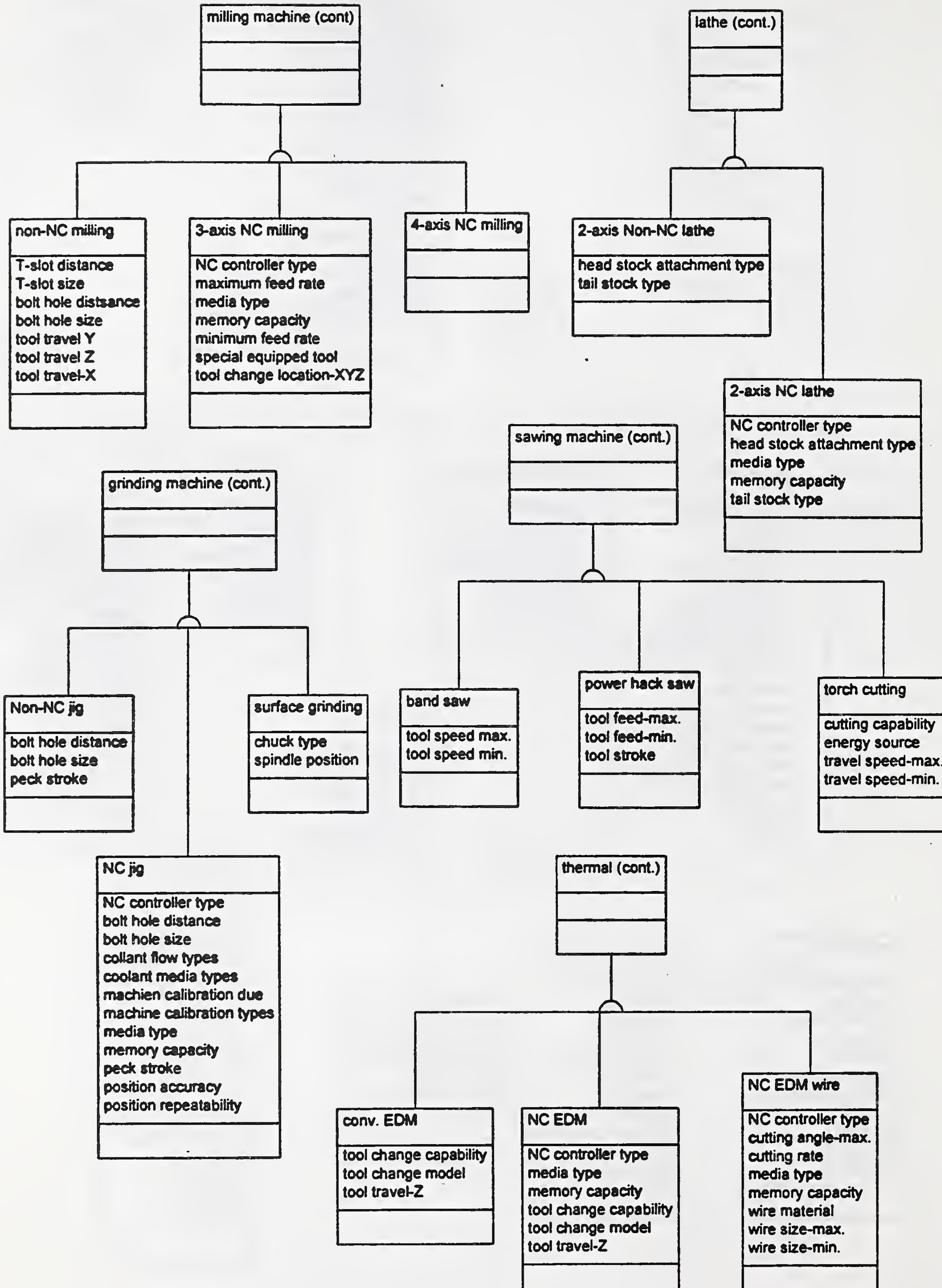
- manufacturing feature classification & definition
- feature recognition & extraction knowledge
- feature transmutation knowledge
- process selection knowledge
- resource selection knowledge
- tooling and fixturing strategy
- operation selection knowledge
- NC programming strategy
- processing parameter selection knowledge

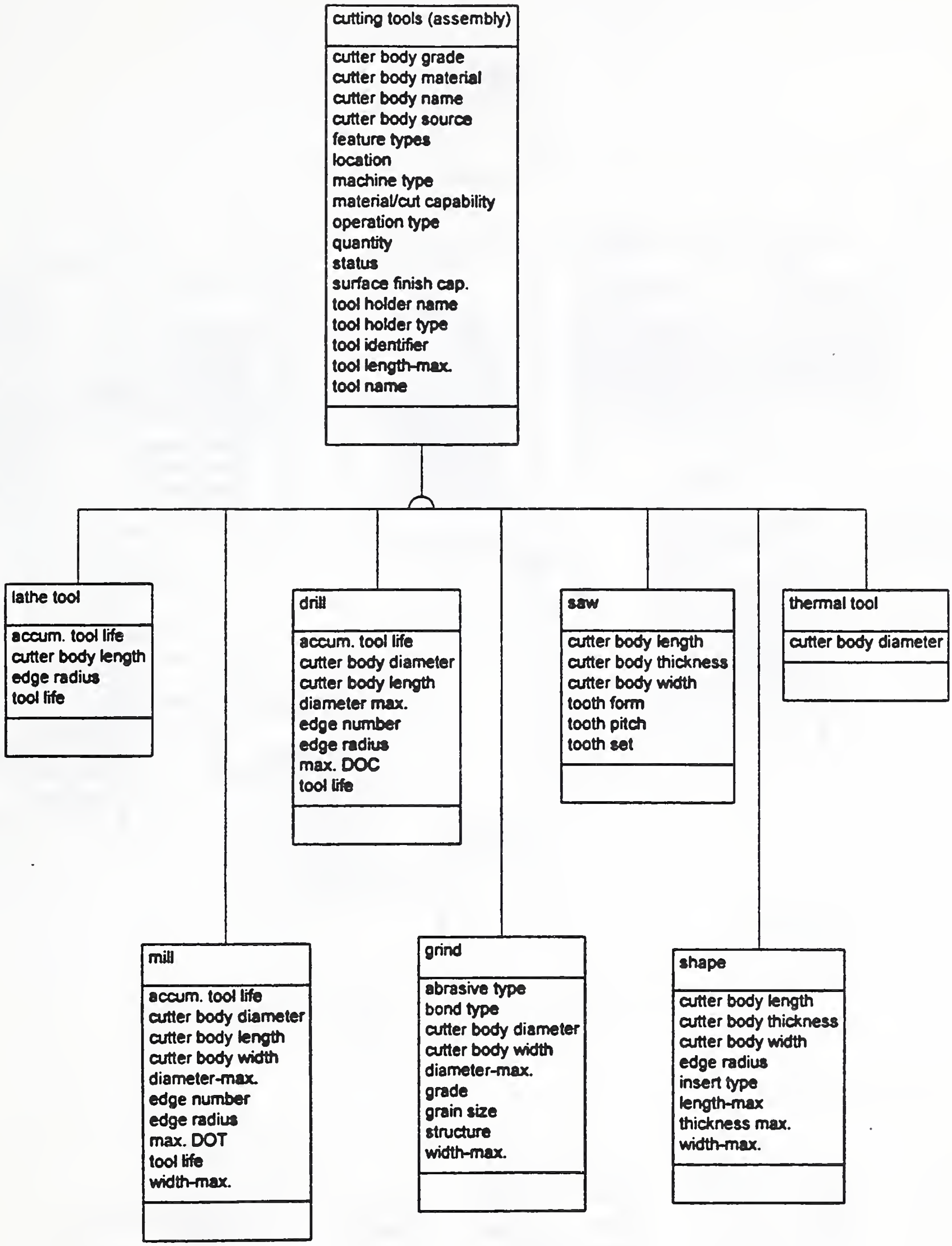

activity
activity identifier
part setup
fixture location
fixture orientation
fixture reference plane
~link<fixture>
machine setup
machine identifier
machine location origin
machine orientation
machine reference plane
material removal
process type
removal identifier
fixture_list
fixture_IDs
fixture_list_identifier
tool_list
tool IDs
tool_list_identifier
NC_program
NC program identifier
machine identifer
machining parameters
operation cluster
cluster identifier
operation
operation description
operation identifier
operation name
~link<tool assembly>

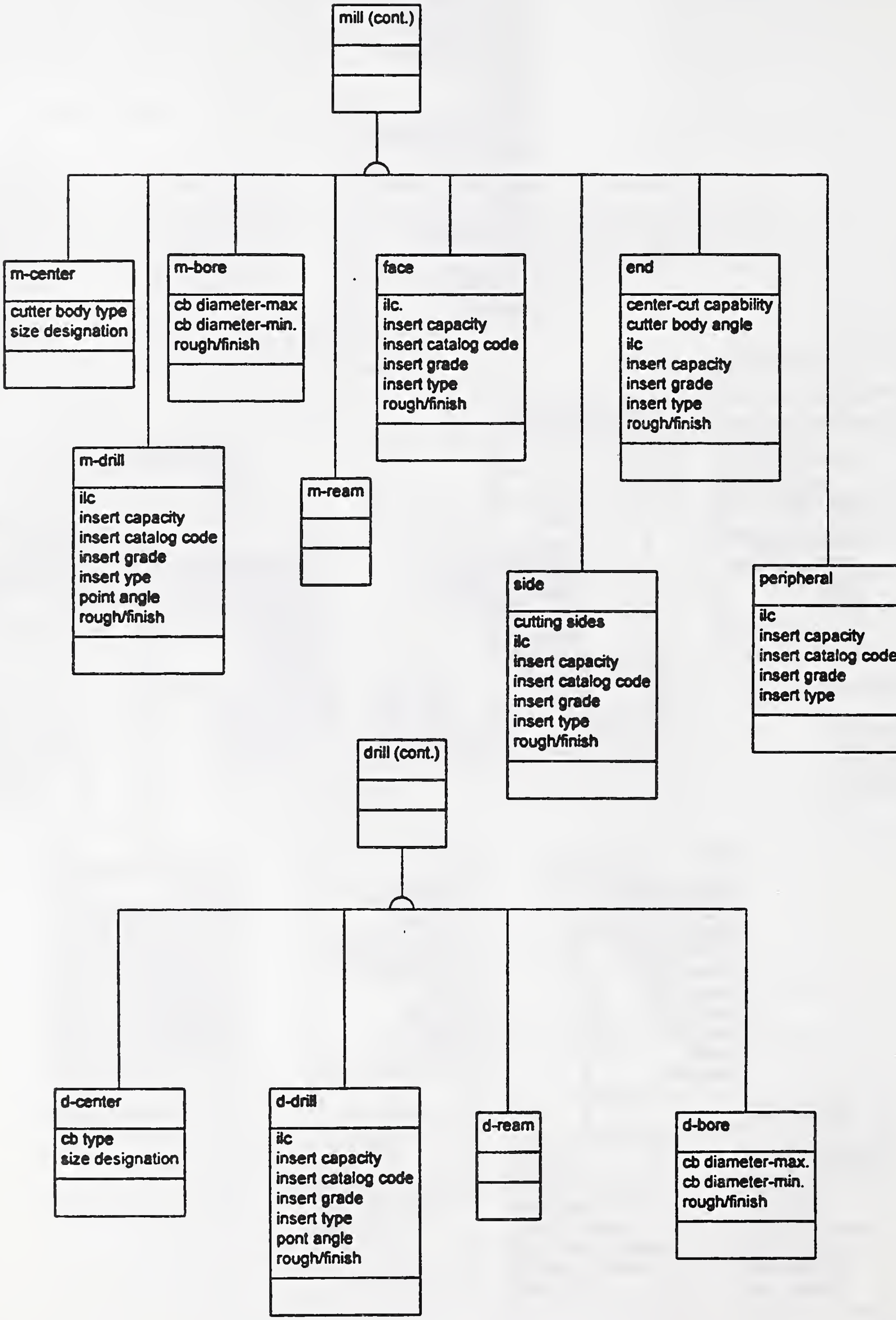

process plan
approval_date
approved by
part identifier
part name
part version
plan revision number
planning date
process plan identifier
sub-process
sub process identifier
workstation
machine identifier
workstation identifier
MRV subset
mrv subset identifier
MRV set
mrv identifier
island
corner radius
creating feature identifier
datum reference
dimensions
island identifier
island type
location
orientation
tolerances
MRV
corner radius
creating feature identifier
datum reference
dimensions
feature tolerances
location
mrv identifier
mrv type
orientation
referenced to
elemental MRV
corner radius
dimensions
e-mrv identifier
e-mrv type
finish allowance
location
orientation
rough/finish
elemental island
corner radius
creating feature identifier
datum reference
dimensions
e-island identifier
e-island type
location
orientation
tolerances
1
1,n

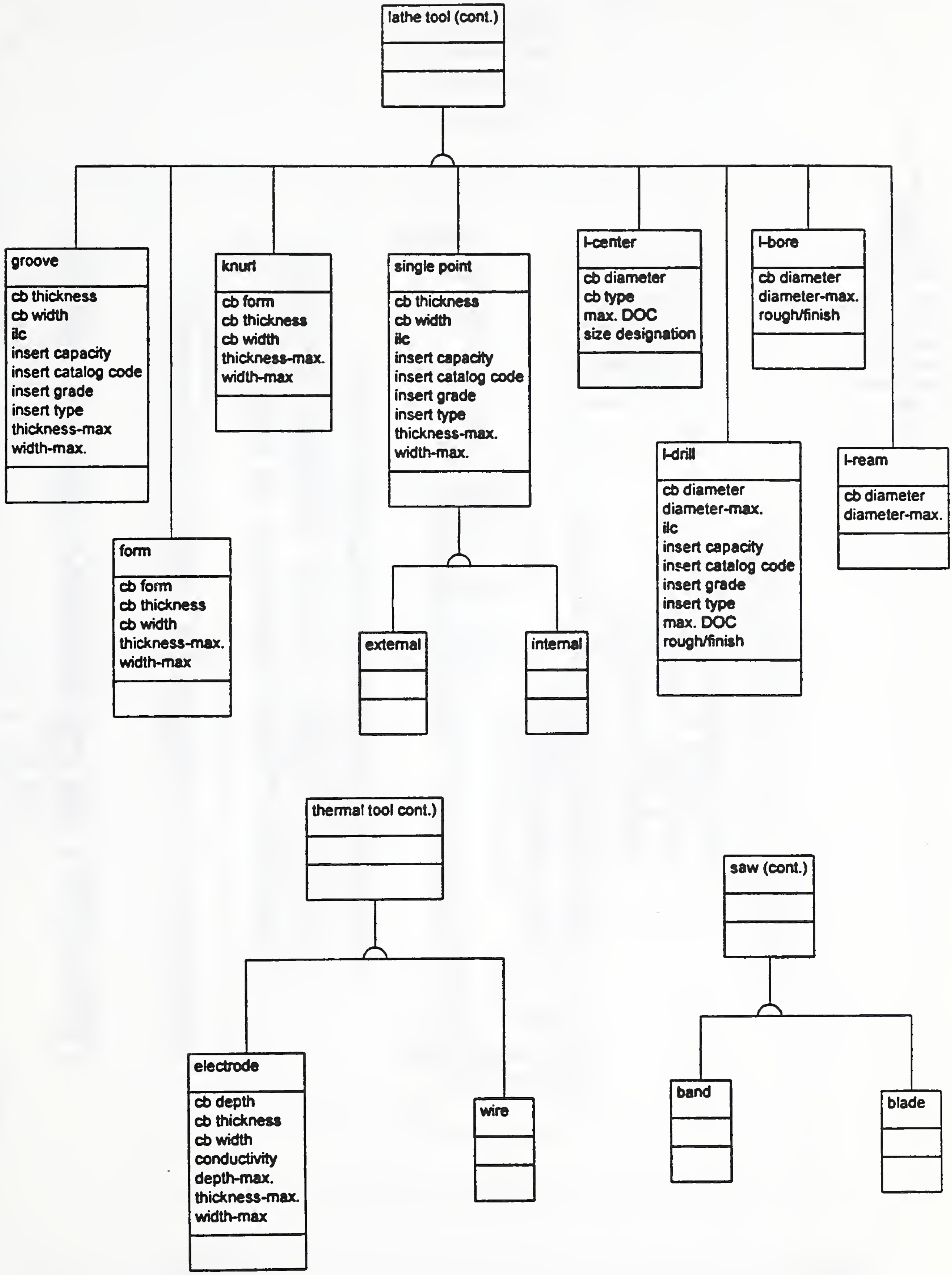

customer order
customer identifier
customer name
order identifier
order items
ordering date
employee
department identifier
employee identifier
employee name
employment date
supervisor identiffier
material
AISI/ASE code
conductivity
density
hardness
lead time
machinability
material identifier
material type
spec. certification
status
supply source
surface finish condition
warehouse location
operator
certified skills and grade
current assignment
engineer
class
work order
customer order identifier
due date
manufacturing quantity
part design identifier
part name
work order check list
work order identifier
bar stock
length
pre-shaped
pre-processing type
1
1
rectangular
height
width
round
radius
forged
Npolygon
inscribed radius
number of sides
work order check list
NC program identifiers
extra sheet identifiers
fixture list identifiers
operation sheet identifier
part design identifier
route sheet identifier
tool list identifier
work order identifier
casted
dimensional allowance
surface hardness


machine
feature type
location
machine identifier
machine mfgr. name
machine name
machine rate/hr.
max travel speed
model number
operation type
operator classification
operator required
serial number
status
table weight capacity
throat depth
tool holder name
tool holder type
work evevelop XYZ
milling machine
Motor RPM max.
T-slot distance
T-slot size
bolt hole distance
bolt hole size
coolant flow types
coolant media tpes
machine calibration due
machine calibration types
motor HP
motor RPM min.
position accuracy
position repeatability
spindle diameter
spindle position
tool change capability
tool travel-X
travel-Y
travel-Z
grinding machine
motor HP
spindle diameter
spindle speed-max
spindle speed-min.
tool travel-X
tool travel-Y
tool travel-Z
sawing machine
motor HP
tool feed-max
tool feed-min.
tool length
tool position
tool travel-Z
lathe
coolant flow types
coolant media types
headstock attachment
machine calibration due
machine calibration types
motor HP
motor RPM-max
motor RPM-min.
position accuracy
position repeatability
spindle diameter
spindle position
tailstock type
tool change calibration
tool change capacity
tool travel-Z
drilling machine
T-slot distance
T-slot size
tool diameter-max.
tool diameter-min
thermal cutting
T-slot distance
T-slot size
bolt hole distance
bolt hole size
coolant type
position accuracy
position repeatability
power AMP
power volt
tool travel-X
tool travel-Y
planing machine
T-slot distance
T-slot size
bed travel-X
bed travel-Y
bolt hole distance
bolt hole size
motor HP
position accuracy
position repeatability
tool position
tool travel-Z
boring machine
bored hole diameter-max
bored hole diameter-min.
position accuracy
position repeatability
tool travel-Z
shaping machine
T-slot distance
T-slot size
bolt hole distance
bolt hole size
motor HP
motor RPM
position accuracy
position repeatability
spindle diameter
tool position
tool travel-X
tool travel-Y
tool travel-Z

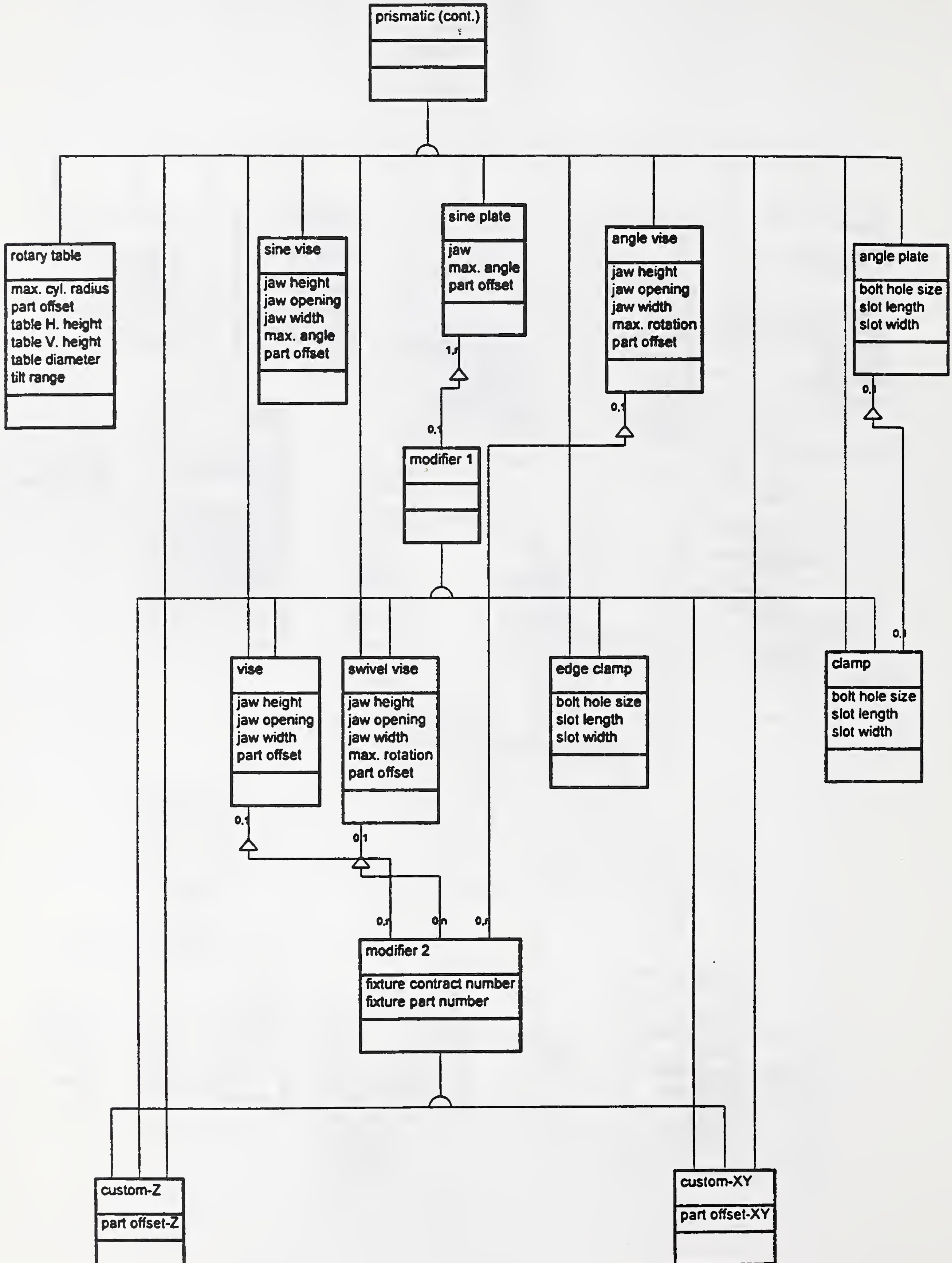

milling machine (cont)
non-NC milling
T-slot distance
T-slot size
bolt hole distsance
bolt hole size
tool travel Y
tool travel Z
tool travel-X
3-axis NC milling
NC controller type
maximum feed rate
media type
memory capacity
minimum feed rate
special equipped tool
tool change location-XYZ
4-axis NC milling
lathe (cont.)
2-axis Non-NC lathe
head stock attachment type
tail stock type
2-axis NC lathe
NC controller type
head stock attachment type
media type
memory capacity
tail stock type
grinding machine (cont.)
Non-NC jig
bolt hole distance
bolt hole size
peck stroke
surface grinding
chuck type
spindle position
NC jig
NC controller type
bolt hole distance
bolt hole size
collant flow types
coolant media types
machien calibration due
machine calibration types
media type
memory capacity
peck stroke
position accuracy
position repeatability
sawing machine (cont.)
band saw
tool speed max.
tool speed min.
power hack saw
tool feed-max.
tool feed-min.
tool stroke
torch cutting
cutting capability
energy source
travel speed-max.
travel speed-min.
thermal (cont.)
conv. EDM
tool change capability
tool change model
tool travel-Z
NC EDM
NC controller type
media type
memory capacity
tool change capability
tool change model
tool travel-Z
NC EDM wire
NC controller type
cutting angle-max.
cutting rate
media type
memory capacity
wire material
wire size-max.
wire size-min.


cutting tools (assembly)
cutter body grade
cutter body material
cutter body name
cutter body source
feature types
location
machine type
material/cut capability
operation type
quantity
status
surface finish cap.
tool holder name
tool holder type
tool identifier
tool length-max.
tool name
lathe tool
accum. tool life
cutter body length
edge radius
tool life
drill
accum. tool life
cutter body diameter
cutter body length
diameter max.
edge number
edge radius
max. DOC
tool life
saw
cutter body length
cutter body thickness
cutter body width
tooth form
tooth pitch
tooth set
thermal tool
cutter body diameter
mill
accum. tool life
cutter body diameter
cutter body length
cutter body width
diameter-max.
edge number
edge radius
max. DOT
tool life
width-max.
grind
abrasive type
bond type
cutter body diameter
cutter body width
diameter-max.
grade
grain size
structure
width-max.
shape
cutter body length
cutter body thickness
cutter body width
edge radius
insert type
length-max
thickness max.
width-max.


mill (cont.)
m-center
cutter body type
size designation
m-bore
cb diameter-max
cb diameter-min.
rough/finish
face
ilc.
insert capacity
insert catalog code
insert grade
insert type
rough/finish
end
center-cut capability
cutter body angle
ilc
insert capacity
insert grade
insert type
rough/finish
m-drill
ilc
insert capacity
insert catalog code
insert grade
insert ype
point angle
rough/finish
m-ream
side
cutting sides
ilc
insert capacity
insert catalog code
insert grade
insert type
rough/finish
peripheral
ilc
insert capacity
insert catalog code
insert grade
insert type
drill (cont.)
d-center
cb type
size designation
d-drill
ilc
insert capacity
insert catalog code
insert grade
insert type
pont angle
rough/finish
d-ream
d-bore
cb diameter-max.
cb diameter-min.
rough/finish



lathe tool (cont.)
groove
cb thickness
cb width
ilc
insert capacity
insert catalog code
insert grade
insert type
thickness-max
width-max.
knurl
cb form
cb thickness
cb width
thickness-max.
width-max
single point
cb thickness
cb width
ilc
insert capacity
insert catalog code
insert grade
insert type
thickness-max.
width-max.
I-center
cb diameter
cb type
max. DOC
size designation
I-bore
cb diameter
diameter-max.
rough/finish
form
cb form
cb thickness
cb width
thickness-max.
width-max
external
internal
I-drill
cb diameter
diameter-max.
ilc
insert capacity
insert catalog code
insert grade
insert type
max. DOC
rough/finish
I-ream
cb diameter
diameter-max.
thermal tool cont.)
electrode
cb depth
cb thickness
cb width
conductivity
depth-max.
thickness-max.
width-max
wire
saw (cont.)
band
blade


prismatic (cont.)
rotary table
max. cyl. radius
part offset
table H. height
table V. height
table diameter
tilt range
sine vise
jaw height
jaw opening
jaw width
max. angle
part offset
sine plate
jaw
max. angle
part offset
angle vise
jaw height
jaw opening
jaw width
max. rotation
part offset
angle plate
bolt hole size
slot length
slot width
modifier 1
vise
jaw height
jaw opening
jaw width
part offset
swivel vise
jaw height
jaw opening
jaw width
max. rotation
part offset
edge clamp
bolt hole size
slot length
slot width
clamp
bolt hole size
slot length
slot width
modifier 2
fixture contract number
fixture part number
custom-Z
part offset-Z
custom-XY
part offset-XY

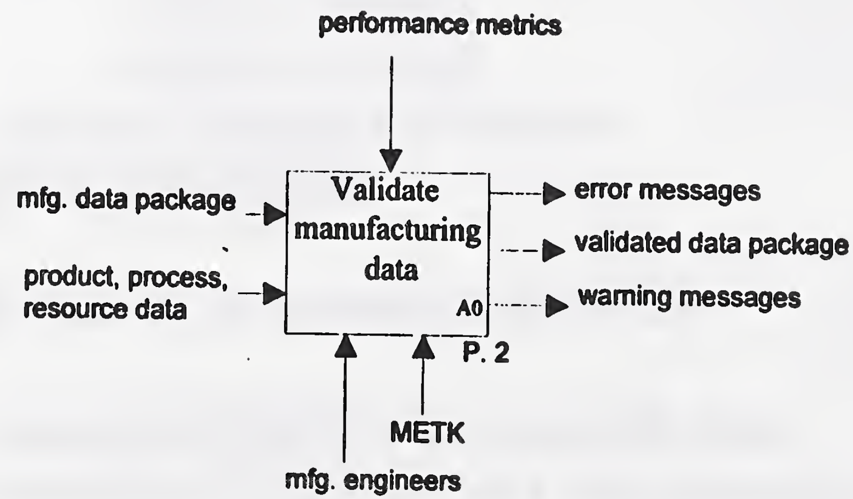
Overview of Proposed Methodology

Requirements for establishing a validation methodology:

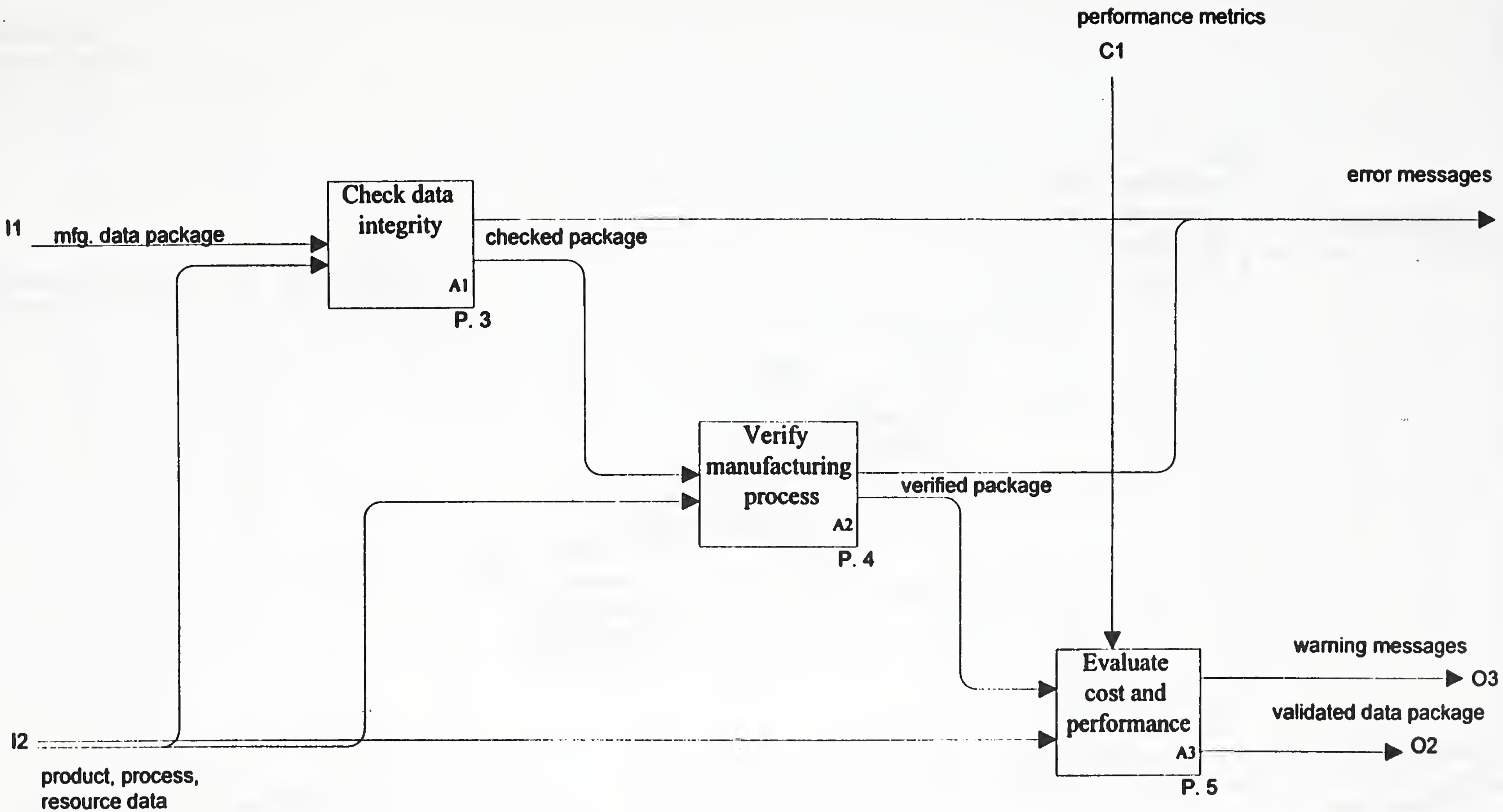
* identification of data elements to be validated
* knowledge of types of errors and where they occur
* established techniques for identifying errors

Key elements of the proposed methodology:

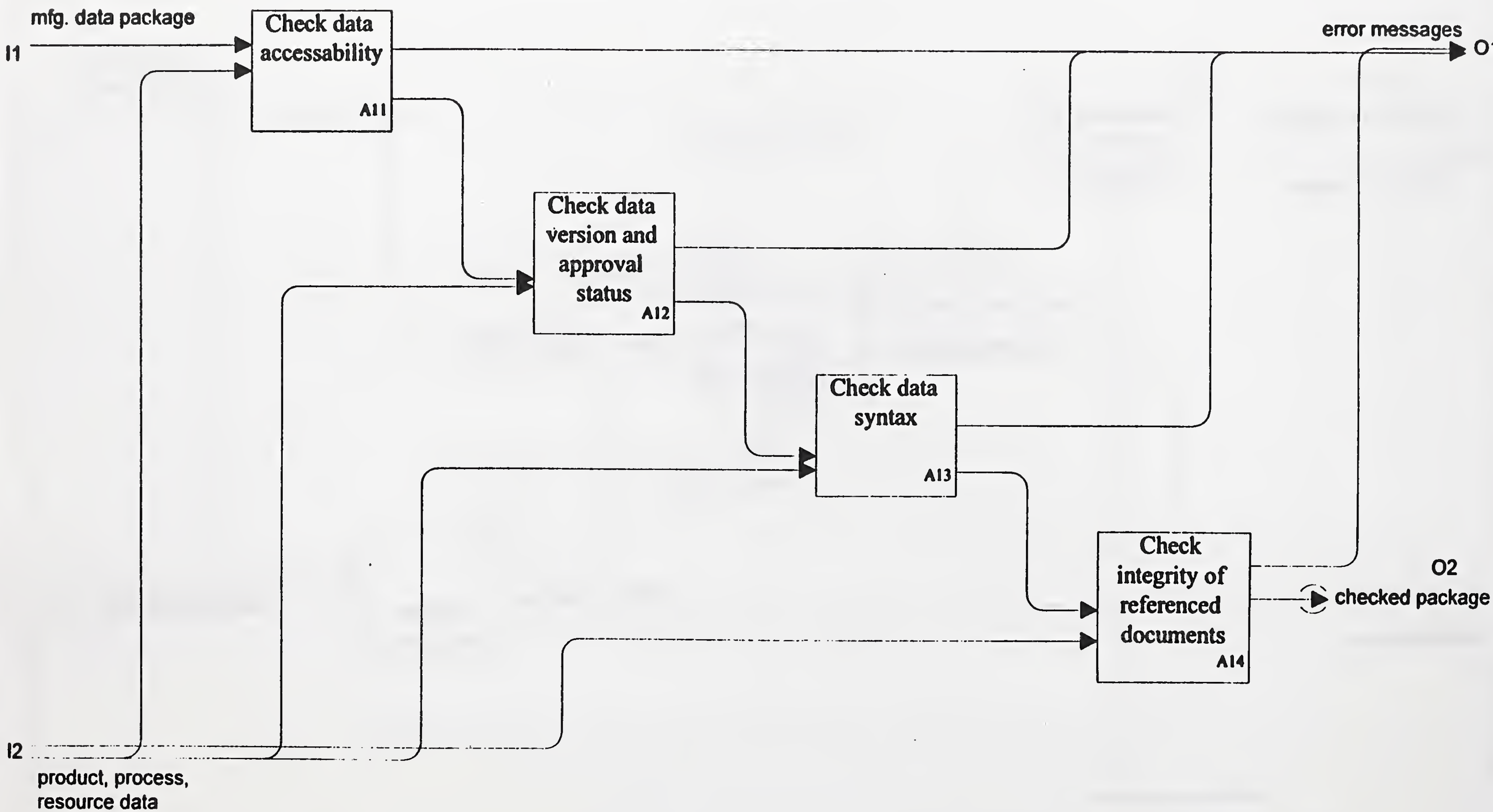
* check data integrity
* verify the manufacturing process
  - resource availability
  - resource capability
  - processing operations
* evaluate cost and performance



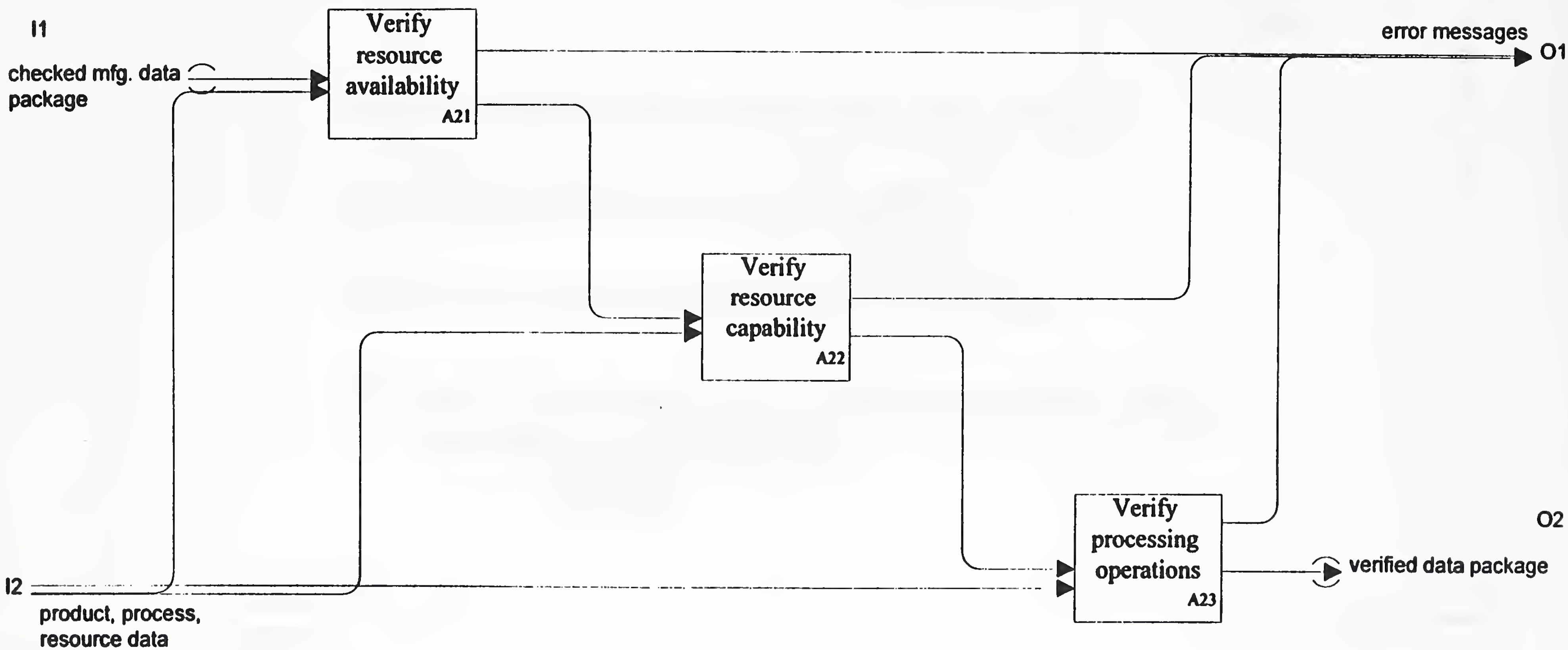
A0: VALIDATE MANUFACTURING DAT.



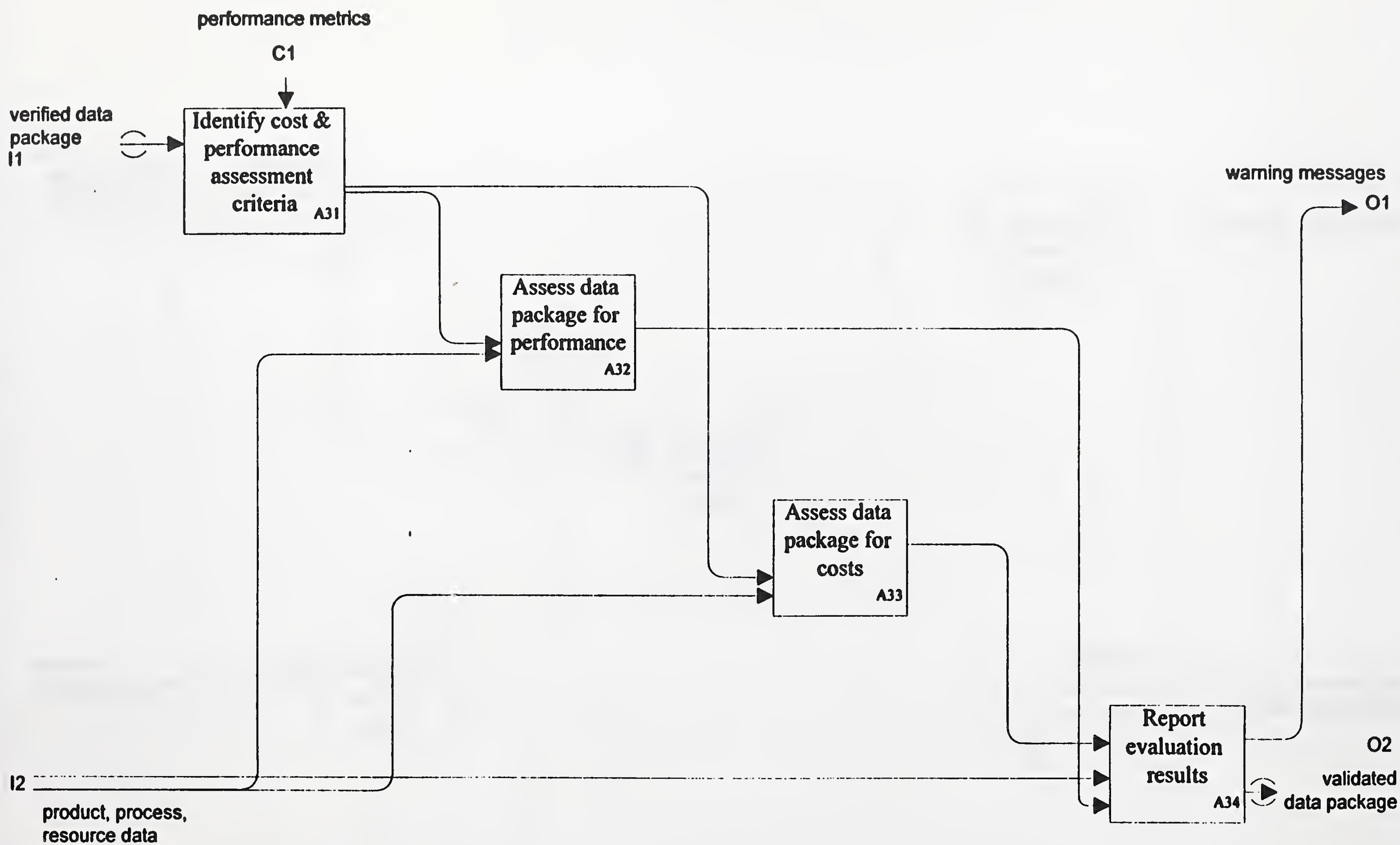
AO: VALIDATE MANUFACTURING DATA



A1: CHECK DATA INTEGR



A2: VERIFY MANUFACTURING PROCESS



A3: EVALUATE COST AND ...RMANCE

Issues and Questions

Are the proposed data package elements correct?

Can the data formats be standardized?

Should the proposed methodology be modified?

How can the methodology be applied incrementally during the data generation process?

ISSUES OF MANUFACTURING ENGINEERING DATA VALIDATION IN CONCURRENT ENGINEERING ENVIRONMENT

C. McLean, S. Leong, and C. Chen
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA
Phone: (301) 975-3511; FAX (301) 258-9749
E-mail: mclean@cme.nist.gov

Abstract

Manufacturing engineering data validation is a critical engineering activity in the product realization process. This paper identifies a set of manufacturing engineering data which is required for production in a machine shop, examines error sources, and proposes a validation methodology for implementation in a computer-integrated concurrent engineering environment. In a sense manufacturing data validation is similar to the practice of inspecting materials and components coming into a shop--the quality of manufacturing engineering data must also be assured before it is released to the shop floor. The ultimate goal of data validation research is to establish techniques that will enable a production facility to produce a product correctly the first time.

Keywords

Engineering information modeling, manufacturing data validation, concurrent engineering, virtual manufacturing.

1 INTRODUCTION

A typical product realization process is divided into three stages: product design, manufacturing engineering, and production. Product design deals with product modeling, functional analysis, and design documentation. Manufacturing engineering specifies the manufacturing procedure and resources required to transform the design into a finished product. Production carries out the engineering plan (product and process design) by coordinating customer orders and resources available to the production system. Among the three, manufacturing engineering has been the most problematic and the least computerized. For the most part, manufacturing engineering still relies on laborious human involvement and is commonly viewed as an art, despite of numerous developments and advances in this area by the CAD/CAM research community in the past decades.

There are few software tools used routinely in industry for automatic generation of manufacturing engineering data. Tools which do automatically generate data typically focus on a narrow portion of the manufacturing engineering problem domain. The main reasons for the lack of tools has been that: 1) there is no effective way of capturing manufacturing knowledge and experience for computer applications, 2) manufacturing engineering data and their representations are not well defined, and 3) manufacturing practices differ significantly among companies. Even fewer computer tools are available for manufacturing data validation. No effective modeling tools exist for capturing engineering and manufacturing resource functionality for data validation. Thus, manufacturing engineering data are often inaccurate and incomplete. Errors sometimes remain undetected until the data is first used on the shop floor, ultimately resulting in data rework, delays in production and product delivery, and higher manufacturing costs. This problem can be critical in production environments where there are long engineering lead times, where engineering data are frequently changed, or where data are shared by a number of engineers involved in product and process development. An automatic data validation tool kit is thus highly desirable, especially when manufacturing engineering data are generated by external resources and the efficiency of a receiving inspection of these external data is a major concern.

The goal of this research effort is development of a manufacturing data validation methodology which, upon its completion, will be able to ensure that the data are complete, correct, and up to date such that the product can be made correctly, as planned at the first time The problem is further complicated in environments where product design and resource availability may evolve constantly, subsequently affecting the validity of downstream manufacturing engineering decisions.

This paper is focused on the modeling and validation of manufacturing process data. The problem domain is limited to the machining job shop environment in which there exist no production lines and no major changes to the production system layout are expected. To outline the manufacturing engineering process in a typical job shop environment and set the scope for further discussion, a brief overview of major manufacturing planning activities is presented in the next section. Section 3 highlights various types of manufacturing engineering data and presents an integrated manufacturing information model. The types of data errors and validation needs are identified in Section 4, followed by a presentation of a data validation methodology in Section 5. A description of the implementation currently under way at NIST is presented in Smith (1995) and is summarized at the end of this paper with concluding remarks.

2 MANUFACTURING ENGINEERING ACTIVITIES

There are three basic functions of manufacturing engineering in a typical manufacturing firm. They are manufacturing administration, manufacturing planning, and process engineering. Process engineering includes design of tooling and production line setups. This paper is focused on the manufacturing planning function and to a lesser degree, the administration function, because they directly contribute to manufacturing data generation and validation.

The modeling of manufacturing engineering activities has been frequently reported in the literature in recent years. Most of these activity models are presented in IDEF0, which organizes activities in a hierarchical structure. For example, in Parker (1994), manufacturing engineering activities were organized into four major tasks: process planning, tooling package development, machining package development, and inspection package development. Process planning was decomposed into three subtasks: resource selection, plan creation, and plan validation and approval. Tooling package development was decomposed into: tooling strategy development, tooling data generation, tooling package verification, and tooling package release and control. Machining package development was decomposed into: NC strategy development, NC machining package preparation, NC data package verification, and NC package release and control. These tasks were further decomposed into more detailed tasks. For example, resource selection consisted of: facility selection, material selection, equipment selection, tooling selection. On the other hand, process plan creation consisted of: in-process shapes/features/attributes generation, process selection, and operations sequencing.

Another manufacturing engineering activity modeling effort can be found in a recent document prepared by NIST (1995). In the NIST's report, five major manufacturing engineering planning activities were identified as follows: 1) determine manufacturing methods, 2) determine manufacturing sequence, 3) develop tooling packages, 4) develop equipment instructions, and 5) finalize the production package. The tasks identified under manufacturing method determination were stock material selection, process selection, major resources selection, and preliminary cost estimation. Under manufacturing sequence determination were: operation specification, operation sequencing, part routing, and plan validation. Under tooling package development were tool selection, tool design, and tool cost estimation. Under equipment instructions development were: in-process part description, tooling requirement specification, operation instruction generation, machine program generation, and equipment instruction validation. Under production package finalization were: final cost estimation, resource package release, scheduling package release, and plan library update.

Both models are intended to capture manufacturing planning activities in the job shop environment. The NIST model however includes, and highlights the importance of, data validation and cost/performance evaluation activities in the planning process. These validation activities may be viewed as an "in-process" validation function. There are additional needs for data validation. For example, a receiving validation is needed when a manufacturing order is being released to the shop floor or external manufacturing data are received.

The manufacturing planning activities are generally inter-related. An upstream decision frequently becomes a constraint to its subsequent decisions, which may also be fed back to preceding activities for design and process changes. For example, a setup decision is a constraint to NC programming, but difficulties found at NC programming may be sent back to the process planner for process modification.

The input data required for these activities include product design, production data, and manufacturing resources. Product design specifies part geometry, form features, material, and tolerances. These product data help the planner narrow the scope of feasible manufacturing processes. Production data allow the planner set a target production quantity and lead time for the process plan. Also it further limits available manufacturing options. Manufacturing resource data such as machines, tools, fixtures, raw materials, and process knowledge are critical to the process decision. The knowledge of resource availability and capability not only enables the planner to make feasible decisions but also improves the decision efficiency by further limiting the scope of feasible solution space; for all planning decisions are made based on available resources, whether they are internal and/or external. However, all the input data are subject to change, which may make a feasible process plan invalid at the time of use. To ensure the validity, some control mechanism needed to monitor and broadcast changes to affected engineering data entities.

Most manufacturing engineering data are still manually generated, even though computer tools are available for assistance. For example, typical process planning systems used in industry still rely on user input for decisions such as feature recognition, process selection, and setup configuration. The planning systems provide a mere working environment for facilitating supplemental planning activities such as plan formatting, plan storage, and data retrieval. For NC programming, APT-based programming systems are typically used to assist in geometry definition, features identification, and tool path generation. Again, in most cases, the user still has to specify part geometry, tool path boundary, and machining parameters. The manufacturing data generated by these planning activities are commonly called routings, operation sheets, material lists, tool lists, fixture lists, machine setups, workpiece setups, tool designs, in-process inspection plans, operator instructions, and NC programs.

3 MANUFACTURING ENGINEERING DATA

Manufacturing engineering data can be broadly classified into two types: product data and process data. Product design data may be documented in CAD models (or data files) and are often translated into engineering drawings for the shop floor. Engineering change orders which record changes to an engineering design may also be included. Primary manufacturing process data are identified as the following nine types:

1. route sheet,
2. stock material specification,
3. intermediate stock shape and geometry,
4. operation sheet,
5. machine setup sheet,
6. workpiece setup sheet,
7. tool list,
8. fixture list, and
9. NC program.

A route sheet specifies a sequence of workstations which each workpiece must visit. It may include both processing stations and queue stations. It may also include scheduling data such as expected arrival time and duration of stay at each station. A stock material specification denotes the initial size and shape of the selected stock material. The selection is done according to the material type and its AISI code specified in the product design. An intermediate stock shape and geometry records the resulting form features and geometry created on the workpiece at each processing step.

Intermediate shape data are critical to workpiece setup and NC programming. To define intermediate shape information for manufacturing, form features are commonly considered as an effective means. An operation sheet contains a set of sequenced machining operations to be performed on the machine with a given workpiece setup. Thus each operation sheet is usually supplemented with a machine setup sheet and a workpiece setup sheet.

A machine setup sheet contains instructions for setting up the machine for the operations specified in the operation sheet. It may include the assignment of cutting tools to specific locations in the tool magazine on the designated machine. If multiple tools are specified in the setup, a tool list needs to be created to list all tools required in this setup. A workpiece setup sheet specifies how the workpiece will be set up on the machine. It may be accompanied by a sketch of the fixturing configuration. If fixture components are used, a fixture list is then required to list the fixture elements to be used for the setup. An NC program is a set of machine instructions prepared for a machining activity. It is machine controller-specific. An NC program is typically prepared for a workpiece setup.

In practice, some of these manufacturing data such as setup instructions and fixture lists may not be made explicitly available and are not formally defined in the manufacturing engineering data packet for the shop floor because they may appear to be trivial and/or tedious. Furthermore, manufacturing process data and formats used in different company may vary considerably. These variations makes manufacturing data exchange and validation extremely difficult. Thus the modeling and standardization of manufacturing data has become a recent research focus in the CIM community. A generic process model called ALPS is presented in Ray (1992). Its application includes modeling of process plans for machining parts. A process plan model specifically developed for NC machined parts can be found in Parker (1994). It attempts to capture all related data entities. By simplifying the above modeling concepts, an object-oriented process data representation schema was proposed and implemented in Sanchez (1994). In the implementation, many data types such as manufacturing features and manufacturing resources were populated and evaluated for their compatibility.

A manufacturing information model has been developed based on the work reported in Parker (1994) and Sanchez (1994) with an emphasis on its compatibility with commercial CAD/CAM packages and current industry practice. Due to limited space, the information model can not be shown here. For the full information model, see Chen (1995). The information model shows that a process plan may have a number of subprocesses, of which each specifies a workstation, a process activity, and a material removal volume (MRV) subset. Each workstation identifies a machine selected to carry out a processing activity. Each processing activity includes a workpiece setup, a machine setup, and the processing task, which is often termed as a material removal activity in the machine shop environment. Each workpiece setup links to a fixture list, while each machine setup points to a tool list, if multiple tools are used. A material removal activity is accompanied by an NC program and a number of operation clusters. An operation cluster denotes a sequence of operations which collectively create a manufacturing form feature (MRV). In other words, an operation removes only a portion of a manufacturing feature (a part of an MRV and called elemental MRV in the figure). Furthermore, each MRV may be constrained by one or many islands, which are converted from protrusions defined in the product model and are treated as physical constraints to the material removal activity. Similarly an elemental MRV may have elemental islands as its constraints. Among the nine manufacturing process data, only route sheets are not explicitly captured in the proposed representation scheme. However, the data required for creating a route sheet such as operations sequence and workstations are available in the model.

4 TYPES OF MANUFACTURING PROCESS DATA VALIDATION

The validity of manufacturing data largely depends on the time-phased cogency of: 1) product design, 2) resource data, and 3) the applied manufacturing engineering knowledge. Because these input data are likely to change over time after decisions are made, the manufacturing engineering data may later become suboptimal or invalid. Thus validation is needed not only at the time of data generation but also at the time of applying these data. Five types of potential data errors and validation needs are identified as follows:

- data integrity,
- resource availability,
- resource capability,
- process validity, and
- cost/performance metrics.

Data integrity deals with the issues of data availability, version control, and data structure (syntax). Data availability checks the existence of each required manufacturing engineering data. Version control ensures that the latest or a correct version of input data is used for generation of manufacturing engineering data. Data structure or syntax ensures checks that data is correctly formatted. A typical data integrity problem is caused by using a wrong version of product and/or process design. For example, an old process plan version may be used to generate NC programs because the NC programming department was not aware of the update.

Resource availability verifies that manufacturing resources specified in the process plan are available. After planning, a selected resource may become unavailable due to reasons such as obsolescence, maintenance, or schedule conflicts. Hence manufacturing data must be re-checked for resource availability before they are released to the shop floor. Process capability is concerned about whether the selected resource has the capability to reliably perform the specified task. Two primary sources of process capability problems are: 1) the resource capability was mis-represented, or 2) the resource's capability has been down-graded (updated) after planning was completed. For example, a machine's repeatability and accuracy may have deteriorated after a period of service.

Process validity is concerned about whether process data such as operation sheets and machine control instructions will perform the task as planned. Typical process validity problems include: 1) inappropriate operation sequence, 2) insufficient setup/teardown instructions, 3) fixturing damage to the workpiece, 4) inappropriate selection of tools, machining parameters, and reference points, 5) collision of a tool holder into the machine tool, fixtures, and/or a workpiece setup, 6) gouging and undercut, 7) workpiece deformation, and 8) thin-wall effects on adjacent form features.

The validation of manufacturing data for cost and performance concerns is different from the other four types of validation. It does not attempt to evaluate the feasibility of the manufacturing engineering data. Instead it is concerned about the optimality of the manufacturing planning decision. It may identify expensive operations, excessive load and unload time, and bottleneck stations. It may also search for less expensive stations.

5 VALIDATION METHODOLOGY

For development of a generic validation methodology, a standard manufacturing engineering data representation is critical. It is a certain requirement for implementation of a computer integrated validation system. In today's manufacturing practice, most data validation is done by the planner who generates the data, and verified (approved) by a supervisor or another planner. Common validation methods are visual inspection, computer graphic simulation, and try-out on a real machine. Although manual inspection and machine try-out are the most common approaches to data validation, significant progress has been made toward development of computer-based data verification techniques.

The development of data validation tools has been largely limited to NC program simulation. Most computer-aided NC programming packages today have some graphic simulation capability for tool path verification. There also exist stand-alone packages for NC program verification, aiming at manually- or externally-generated NC programs. In either case, however, the user still must observe the graphic display and determine whether or not the program is correct, or whether collisions occur. Automatic collision detection capabilities have become available recently in some graphic simulation modeling packages such as Deneb's VNC (1995). Limited capability of operations sequence verification can also be found in recent versions of process planning systems such as ICEM/PART (1994). This is done by checking whether or not the specified removal sequence of manufacturing features violates any physical constraint on the workpiece.

Based on the manufacturing data types and potential errors presented in Sections 3 and 4, the needs for data validation are identified in Table 1. As shown in the table, four manufacturing data types need to be validated for each of the five potential data errors. They are: route sheet, operation sheet, tool list, and fixture list. Machine setup, workpiece setup, and NC program require validation for data integrity, process validity, and cost/performance metrics. Stock material specification needs to be evaluated for data integrity, resource availability, and cost and performance. The only concern with respect to intermediate shape and geometry data is data integrity.

From a data validation point of view, data integrity checks are required for all data types. A resource availability check needs to be applied to those data types which require manufacturing resources. The need for resource capability validation is similar to those for resource availability, except stock material specification. The check for process validity is required for all but stock material specification and intermediate shape data. A cost and performance evaluation can be applied to all the manufacturing data types.

Table 1: Needs for Manufacturing Engineering Data Validation

Data Type	Data Integrity	Resource Availability	Resource Capability	Process Validity	Cost/Perfor mMetrics
Route sheets	x	x	x	x	x
Op. sheets	x	x	x	x	x
Stock specs.	x	x			x
Inter. shapes	x				x
Tool lists	x	x	x	x	x
Fixture lists	x	x	x	x	x
M/T setups	x			x	x
Work setups	x			x	x
NCprograms	x			x	x

It is possible to develop a validation method for each validation need as identified in the table. For example, a validation technique may be desired for checking the availability of resources identified in an operation sheet. One drawback is that there will be many validation packages. It is advantageous to develop a validation tool for each data type for checking all its potential data errors. Such a tool could be easily incorporated into a manufacturing data generation package for an "in-process" data validation. On the other hand, it is also desirable to develop a validation tool for each error type. For example, a validation method could be developed to check only data integrity but for all data types. If so, a logical validation procedure should be to check for: 1) data integrity, 2) resource availability, 3) resource capability, 4) process validity, and then 5) cost/performance.

Data integrity needs to be checked first, to make sure that all required manufacturing data are available and complete; and they are prepared based on the most up-to-date or correct version of input data. Resource availability should be the second step in the validation process. It identifies resources specified in the data and checks if selected resources are available at this time. If they are, a check for resource capability should then be ordered. Otherwise, the problem should be reported and no need to continue for further validation. Resource capability verifies whether each resource can properly perform its intended task. It can be done by checking against its static capability as recorded in the database and may be done independently for each selected resource. An example might be checking to see if each tool in the tool list can properly cut the selected stock material.

Data validity checking is required to ensure that each manufacturing data entity is valid and complete. All manufacturing data may be required for this validation. For example, if a hole is to be drilled on a machine, the validation has to make sure that the hole can be created and precisely located on the workpiece, with the given machine, tools, setup instructions, and fixturing configuration. If an operation sheet is to be evaluated for its process validity, machine setup and workpiece setup need to be first examined, which in turn may retrieve and examine the intermediate stock shape and geometry. In our view, process data validity is the most complicated and challenging validation task. After passing the above four validation tests, the manufacturing process data are considered as valid. The last data evaluation of cost and performance is an attempt to improve its optimality.

For validation of data integrity, a simple data inventory list may be sufficient for checking the existence of each data entity required; on the other hand, an engineering business model may be sufficient for information flow management and version control. For validation of resource availability and capability, a search algorithm will be developed to identify the resources specified in the manufacturing data and verify their existence and capability against the records in the database. For this purpose, a standard manufacturing data representation and a database system will be required. For validation of process validity, computer-based graphic simulation techniques have been widely applied. However, in addition to material flow simulation, various functional models of manufacturing resources and systems need to be created for each application. A computer-based technique for automatic generation of functional models for manufacturing resources such as machine tools and fixturing configurations will certainly improve the validation efficiency and effectiveness. Current simulation capability is still largely limited to statistical data collection and graphic display with only very limited capability of collision detection for NC program verification. Additional capabilities such as material deformation, think wall effects, and tolerance analysis have to be included. Emerging virtual reality techniques could be helpful in construction of virtual machines and manufacturing systems for the proposed data validation.

6 IMPLEMENTATION

Significant progress has been made at NIST toward development of a manufacturing engineering data validation tool kit. Due to the fact that manufacturing data may come from various sources, the need for standard resource and process data models has been recognized. The development of a generic information model is under way. A system architecture and a database management system are being defined to support various engineering activities on different computer platforms and to maintain the vast amount of product, process and resources data. The implementation of the proposed validation methodology is intended to validate manufacturing data at the time when each data entity is created and re-check the data when a manufacturing data packet is being prepared for a manufacturing order.

In addition to the development of a distributed system architecture and manufacturing resource and process data repositories, the implementation effort also includes development of computer-based validation tools for checking data integrity, resource availability, resource capability, and data validity. Development of cost and performance validation tools are also being considered. The system environment is expected to support sharing of various data generated by commercially-available, heterogeneous CAD/CAM systems. The standard information model under development will be used to capture commonly needed manufacturing resources and process data, which will be stored in a distributed database management system and be concurrently accessible by multiple application systems. A number of commercial CAD/CAM systems including Matrix (1994), Pro-Engineer, ICEM/PART (1994), and Deneb's I-GRIP, Deneb VNC (1995b), and Quest (1995a) are currently being integrated to create the concurrent engineering environment.

Matrix is a product data management (PDM) system. It is used to implement an engineering business model for data integrity validation and information flow control. Pro-Engineer is a CAD system used to create test product designs. ICEM/PART is used to interpret a Pro-Engineer model and generate a process plan (operation sheet) for prismatic parts. It will be integrated with other applications to share resource data and store process plans in the database. A validation module will be implemented for checking availability and capability of resources as recorded in the database. Deneb's software packages are initially used to manually create functional models of selected manufacturing systems and resources for process data validation. Automatic modeling of these functional models based on a script will be the next step toward the tool kit development.

7 CONCLUDING REMARKS

Manufacturing engineering data validation is an integrated part of the manufacturing planning process. It is, in our view, the most problematic and the least computerized engineering activity in the product realization process. The main reasons have been: 1) there is no effective way of capturing manufacturing knowledge and experience for computer application, 2) manufacturing engineering data and their representation are not well defined, and 3) manufacturing practices differ significantly among companies. An additional obstacle to validation tool development is that there are no effective tools for creating functional models of manufacturing resources with enough functionality for data validation. Thus, manufacturing engineering data are often inaccurate and incomplete; and errors are sometimes undetected until the data is first used on the shop floor. If only validated data reach the shop floor, many production and delivery delays may be eliminated and higher manufacturing costs may be avoided.

The research effort reported in this paper is aimed at development of a methodology for manufacturing engineering data validation. To this end, nine major manufacturing engineering data types are identified: route sheet, operation sheet, stock material specification, intermediate stock shape and geometry, machine setup, fixture setup, tool list, fixture list, and NC program. Various error sources have been studied and the needs for validation are identified in five categories: data integrity, resource availability, resource capability, process validity, and cost/performance. Validation for data integrity and cost/performance metrics are required for all data types. Resource availability and capability checking should be applied to those data specifying resource usage such as route sheets, operation sheets, tool lists, fixture lists, and stock material specifications. Process validity is the most difficult validation because functional models of manufacturing resources and systems are required to simulate the physical manufacturing process.

The implementation of an engineering data validation system is currently under way at NIST. A number of commercially-available CAD/CAM systems have been assembled and integrated for implementation of a manufacturing engineering data validation tool kit. Among them, Matrix is used for information flow and data integrity control. Deneb's VNC is used to create a functional model of resources for process validation. Quest is used to model material and resource flows on the shop floor. Additional validation tools are being developed for resource availability and capability validation.

Work described in this paper was sponsored by the U.S. Navy Manufacturing Technology Program and the NIST Systems Integration for Manufacturing Applications (SIMA) Program. No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. The work described was funded by the United States Government and is not subject to copyright.

8 REFERENCES

Adra Systems (1994) Matrix System, Chelmsford, MA.

Chen, C. (1995) Information Model for the Manufacturing Engineering Tool Kit, NIST Internal Report, in preparation.

Control Data Corporation (1994) ICEM PART Reference Manual, Volume II, Version 1.2, Control Data, Inc.

Deneb Robotics (1995a) Quest User Manual, Auburn Hills, MI: Deneb Robotics, Inc.

Deneb Robotics (1995b) Virtual Numeric Control, Version 2.1, Auburn Hills, MI: Deneb Robotics, Inc.

NIST (1995) SIMA Background Document, National Institute of Standards and Technology, Manufacturing Systems Integration Division, Gaithersburg, MD, in publication.

Parker, L. (1994) "Product Data Representation and Exchange - Part 213: NC Process Plans for Machined Parts," Gaithersburg, MD: National Institute of Standards and Technology.

Ray, S. (1992) Using the ALPS Process Plan Model, *Proceedings of the ASME Manufacturing International Conference*, Dallas, TX.

Sanchez, M. (1994) Feature-based Process Planning for Machined Components, unpublished Master Thesis, Florida International University, Miami, FL.

Smith, M. and Leong, S. (editors) (1995) *Proceedings of the First Computer-aided Manufacturing Engineering Forum*, Gaithersburg, MD, in publication.

9 BIOGRAPHIES

C. McLean: Group Leader of Manufacturing Systems Engineering, NIST Manufacturing Systems Integration Division. Managed research programs in manufacturing automation at NIST since 1982. Established research program in Computer-Aided Manufacturing Engineering with emphasis on engineering tool integration, manufacturing data validation, production systems engineering, manufacturing simulation, production scheduling.

S. Leong: Joined NIST in 1994, after sixteen years in the manufacturing industry. Established experience at Ford Motor Company, John Deere, and IBM in design, development and implementation of factory automation systems. Spent the last eight years in providing manufacturing consulting services to manufacturers for design and implementation of manufacturing execution and shop floor control systems.

C. Chen: Professor of Florida International University working on IPA as an Industrial Engineer of the Manufacturing Systems Engineering Group in NIST Manufacturing Systems Integration Division. Research interests include engineering information modeling, CAD/CAM systems integration, and production planning and scheduling

Appendix E:

"Implementation Plan and Issues"

Implementation Plan and Issues

Swee Leong
Manufacturing Engineer

CAME Industry Meeting
August 1995

Implementation Plan & Issues

- **Implementation Issues**

-- Manufacturing Data Package

-- Data Model for Baseline System Demo

-- Reference SIMA Architecture

-- Manufacturing Data Management and Control

-- Manufacturing Data State Space Diagram

-- Organization Chart

-- Sample Example: Naval Surface Warfare Center Scenario

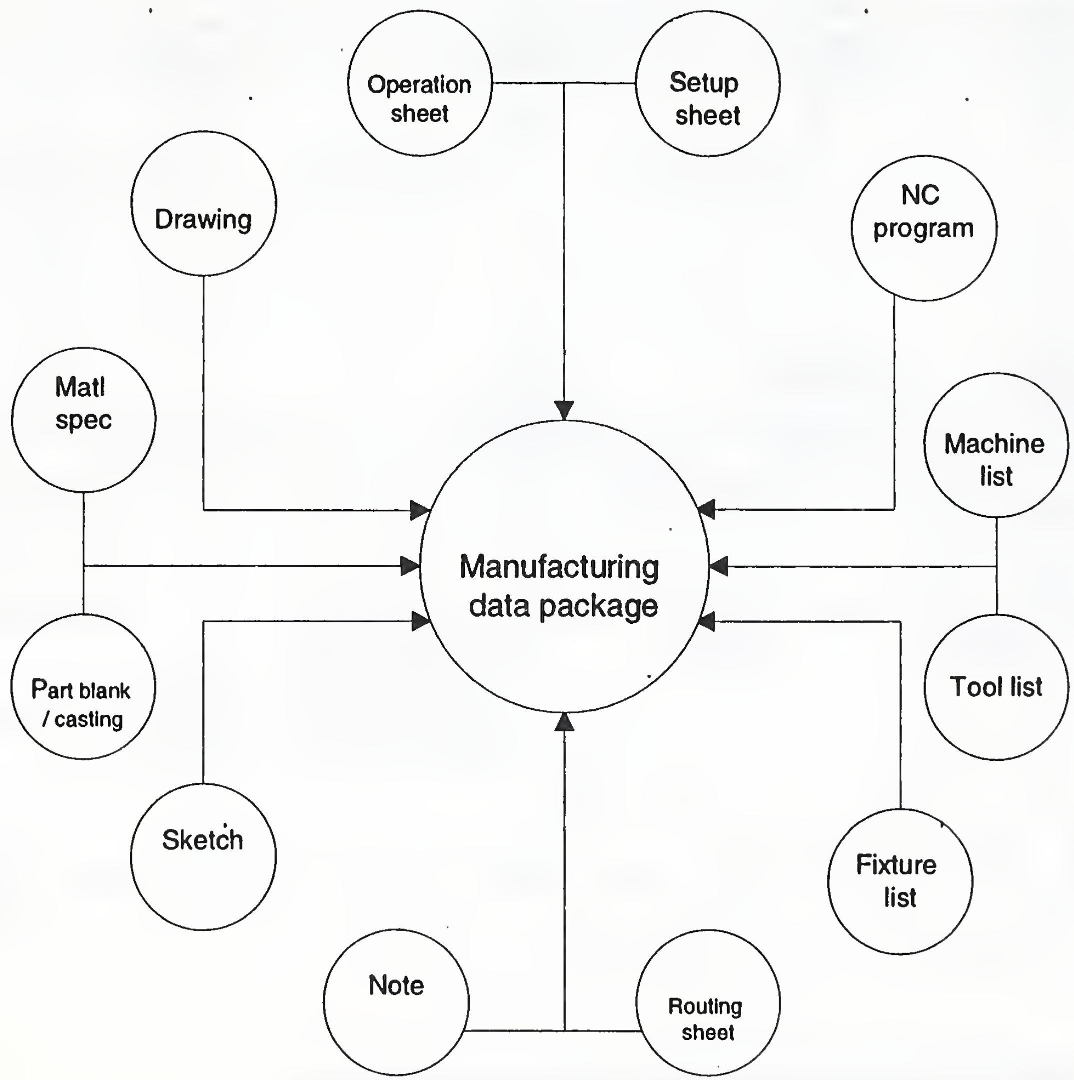
- **Implementation & Rollout Plan / Issues**

-- Prototype System at NIST

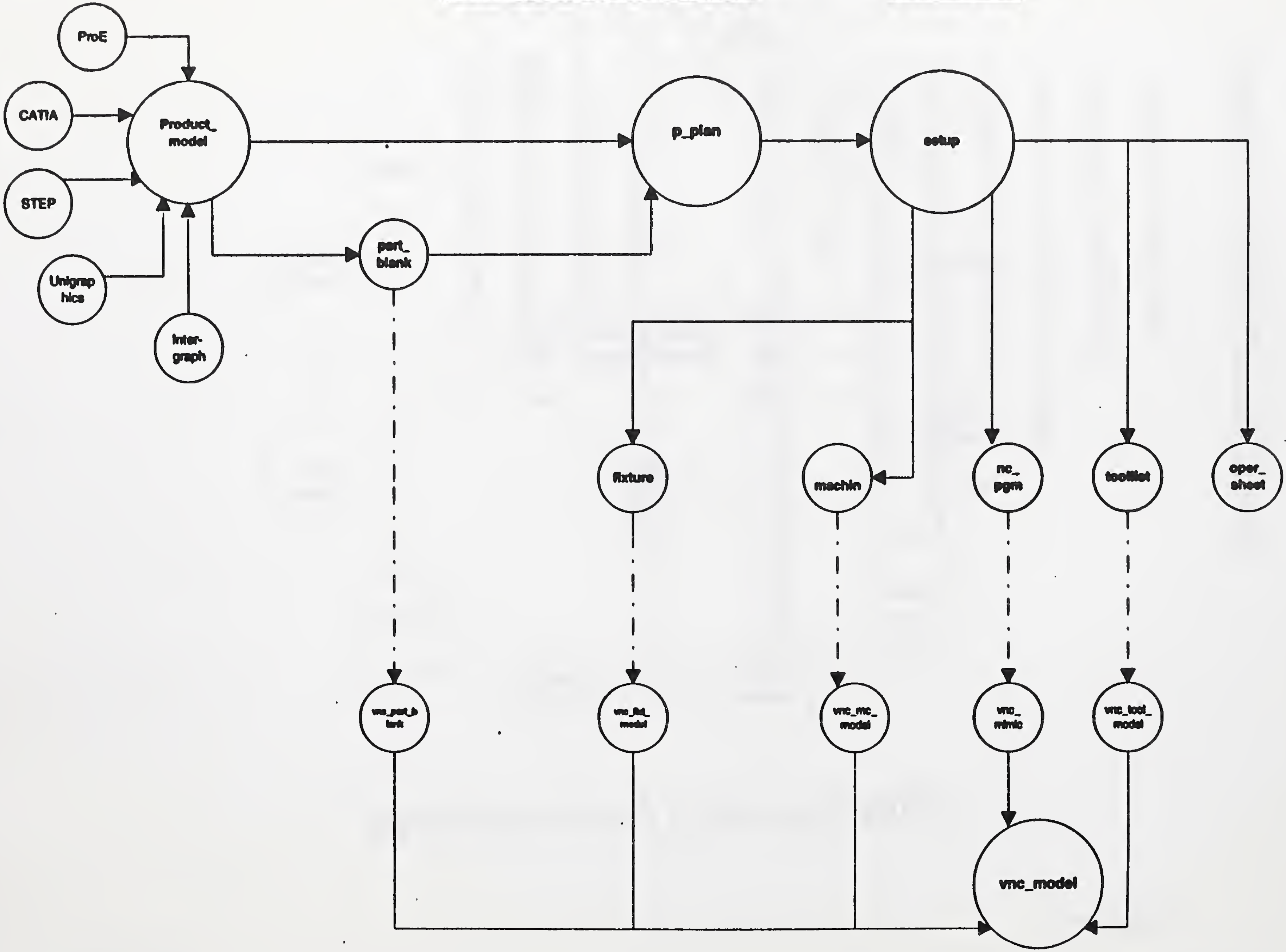
-- Prototype System Roadshow

-- Software Licenses Issues

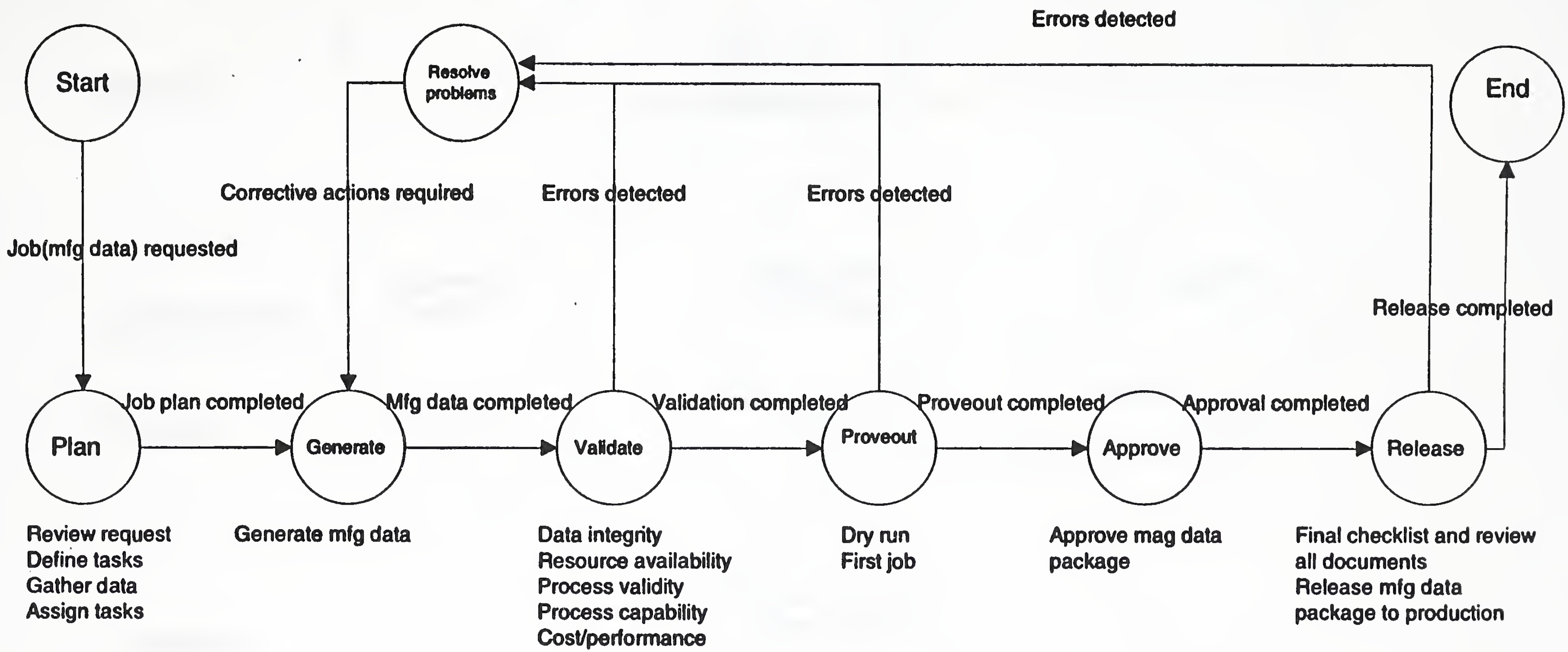
Manufacturing Data Package



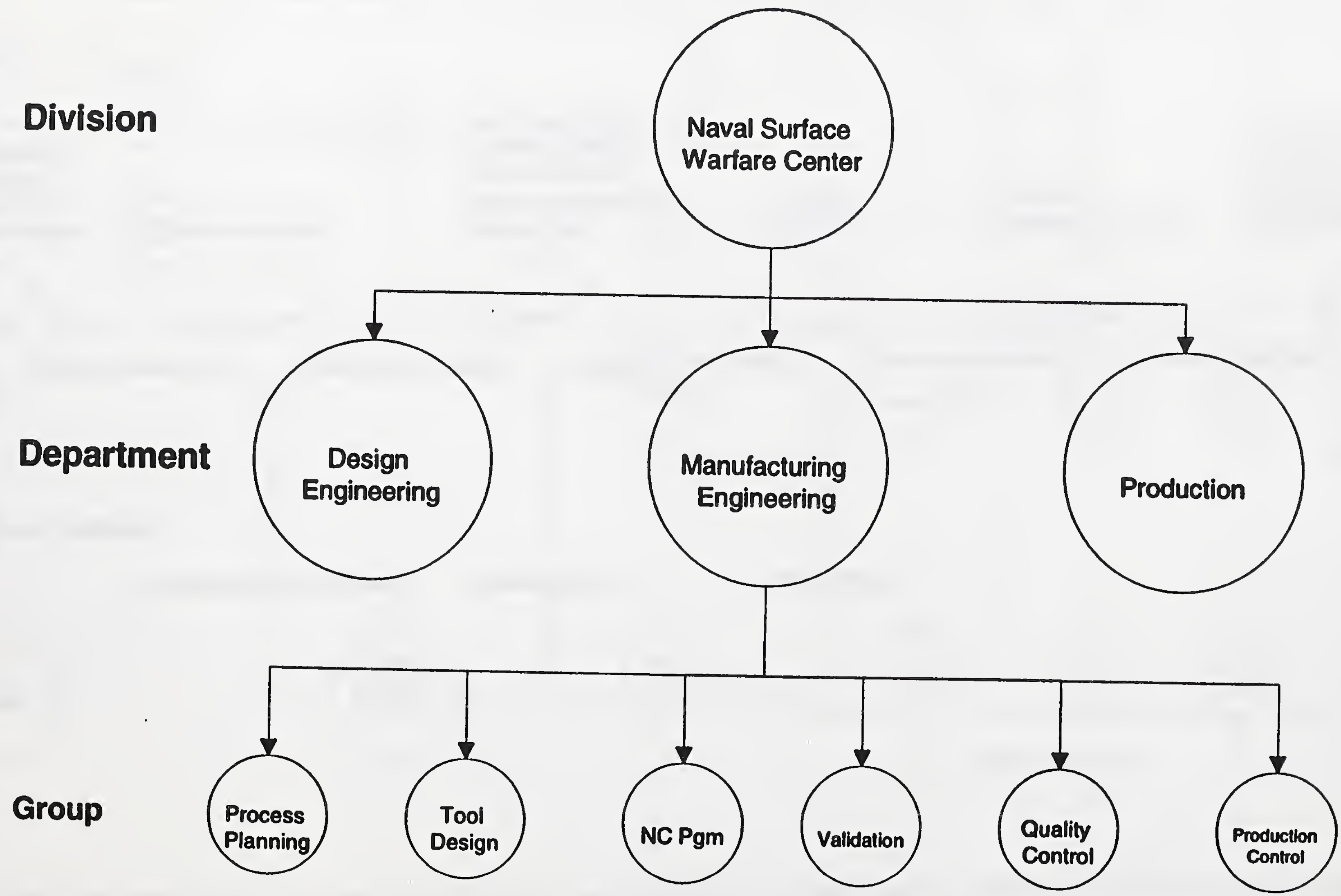
Data Schema for Baseline System Demo



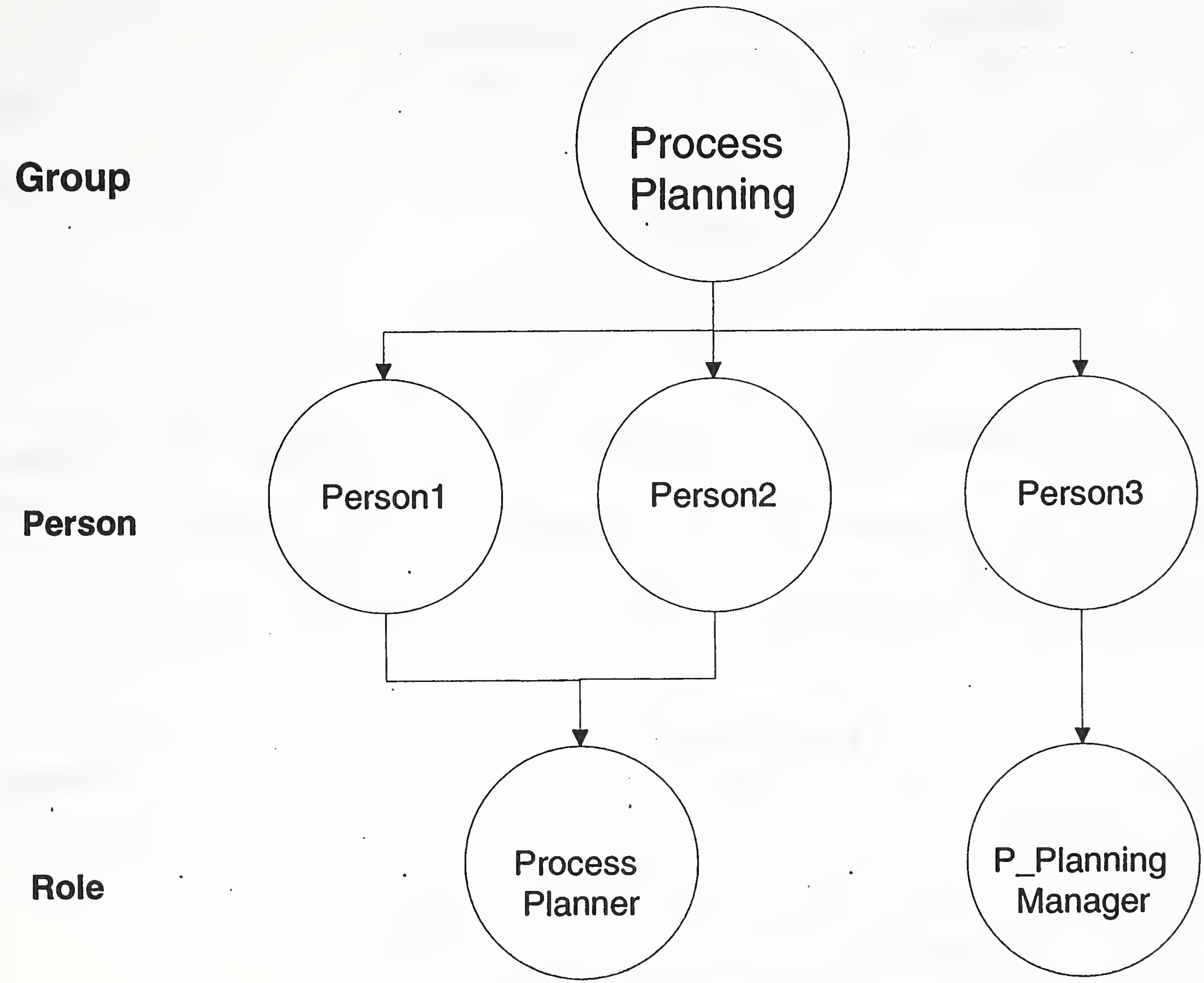
Manufacturing Data Package Preparation and Validation State Space Diagram (Machine shop context)



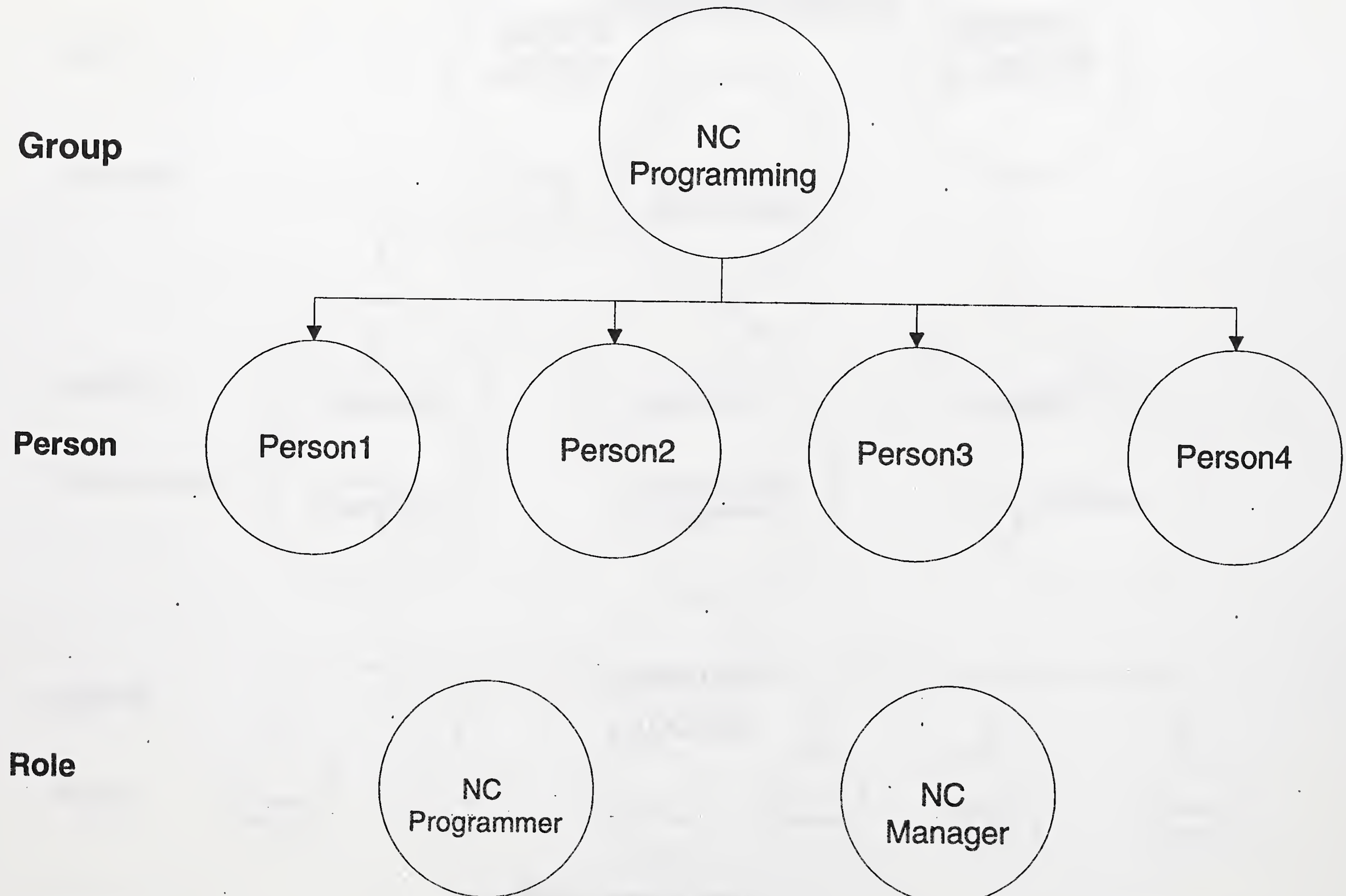
Organization



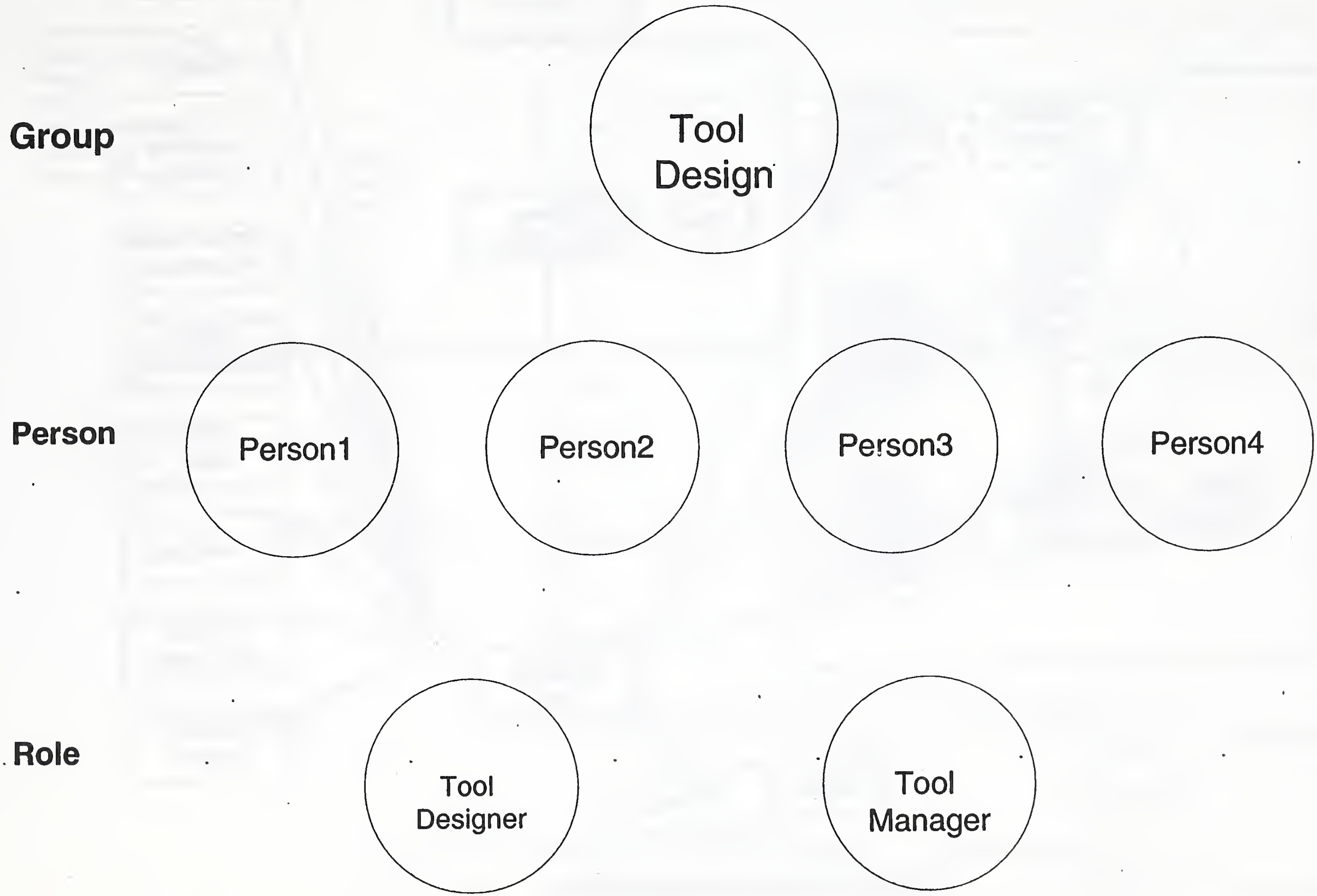
Organization



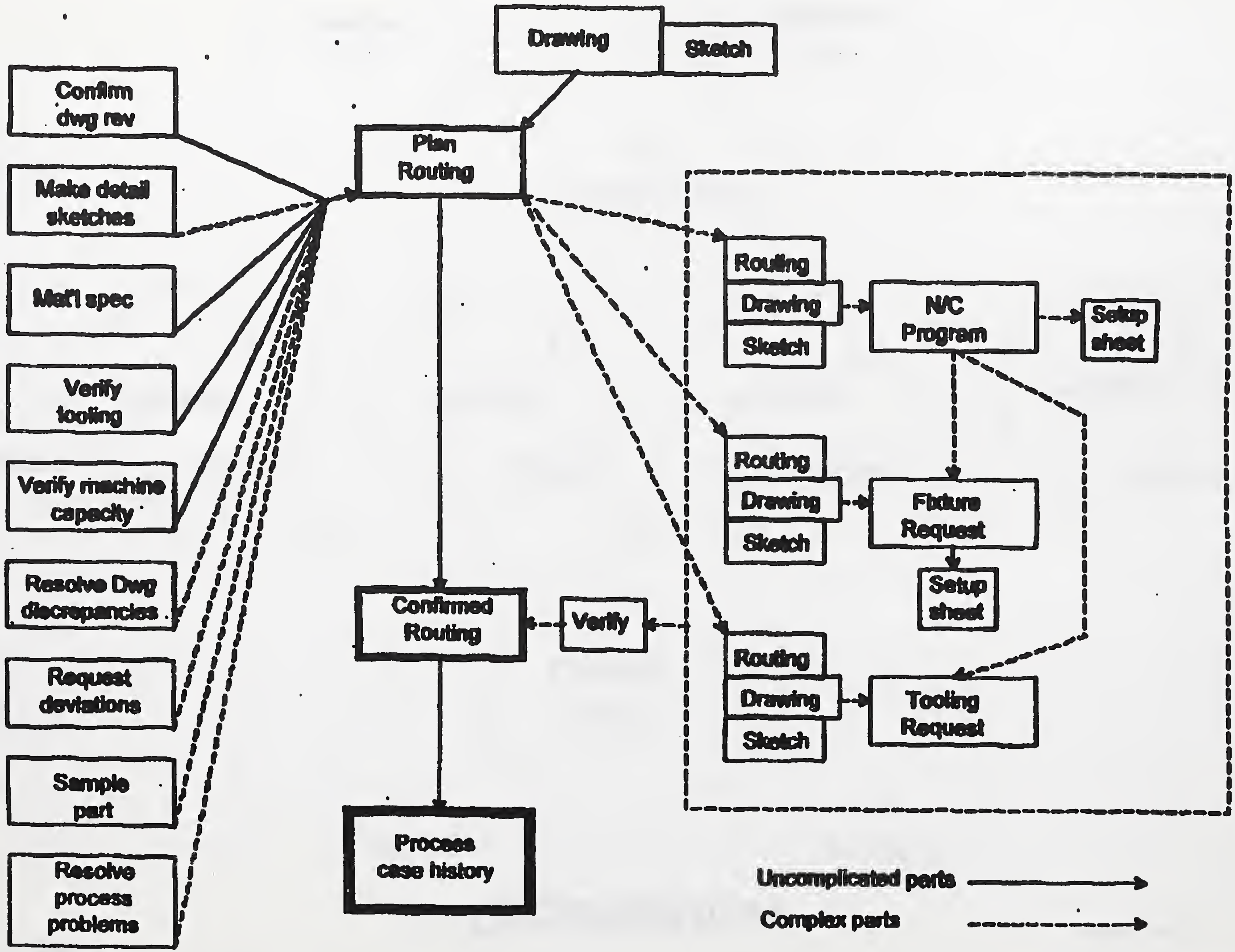
Organization



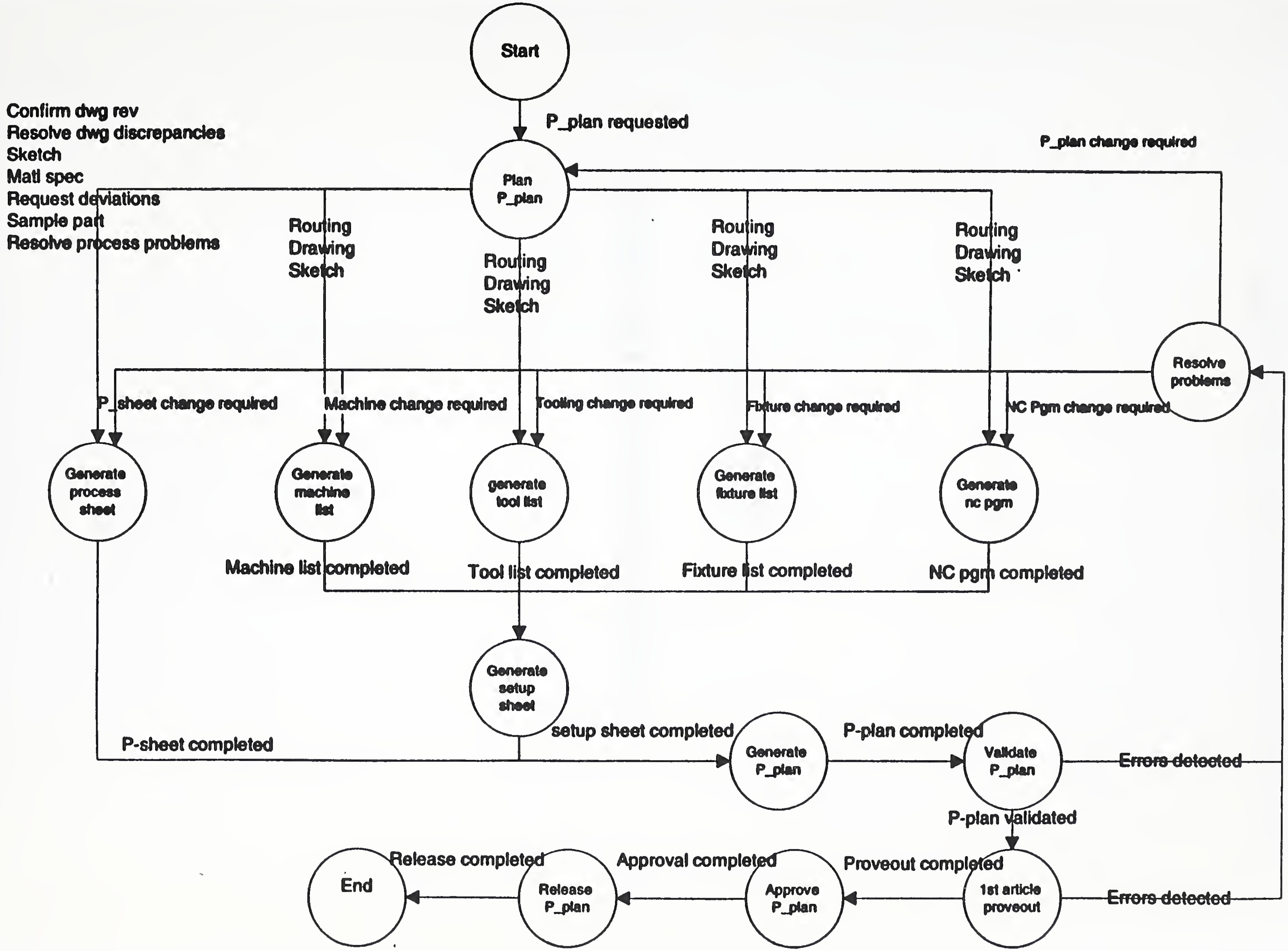
Organization



Request for process routing



Manufacturing Data Package Prepai .ı and Validation State Space Diagram for the Naval Surface Warfare Center



Appendix F:

"CAME METK Architecture Review"

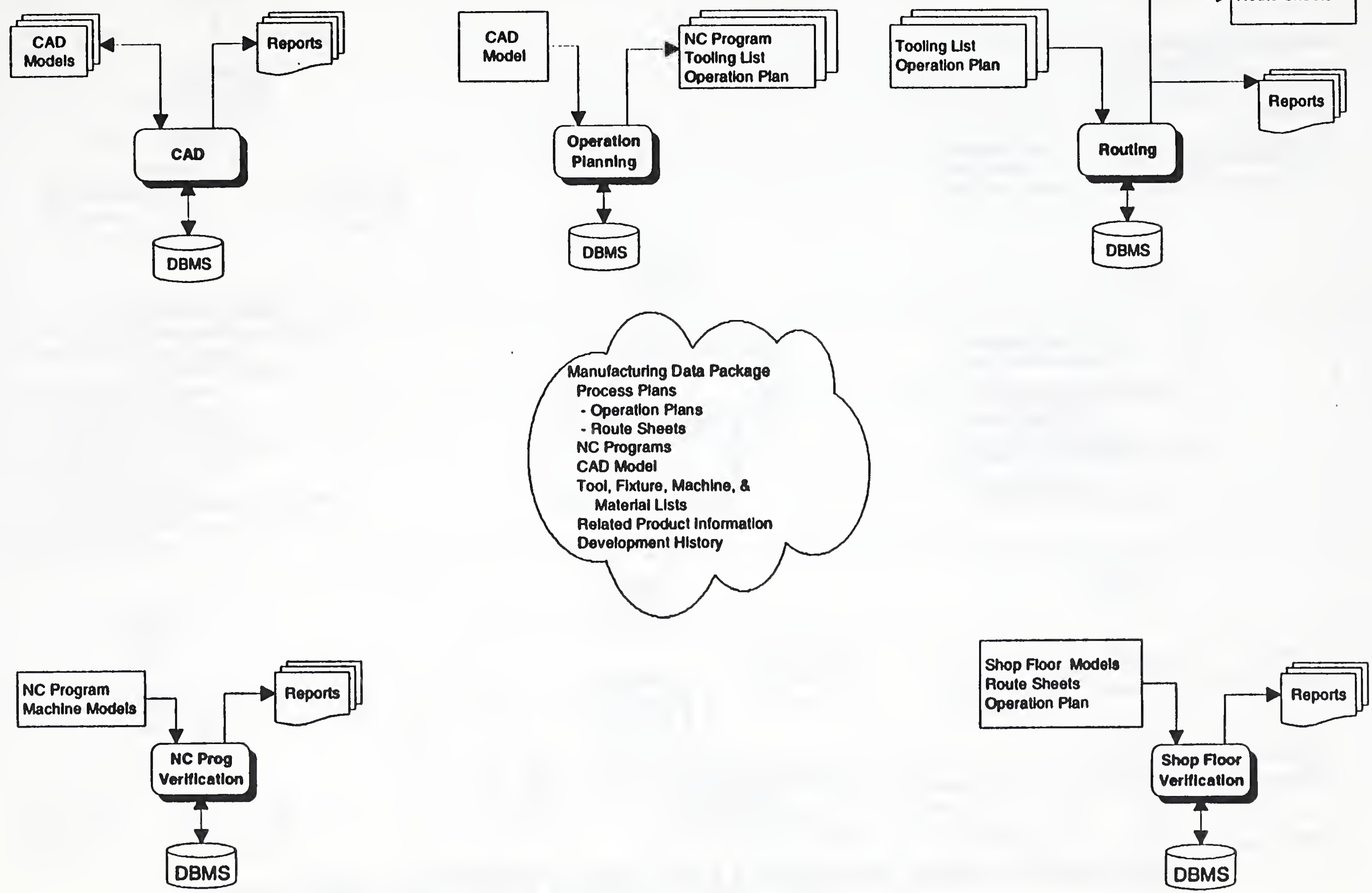
CAME METK
ARCHITECTURAL REVIEW

Frank Riddick
NIST
August 22, 1995

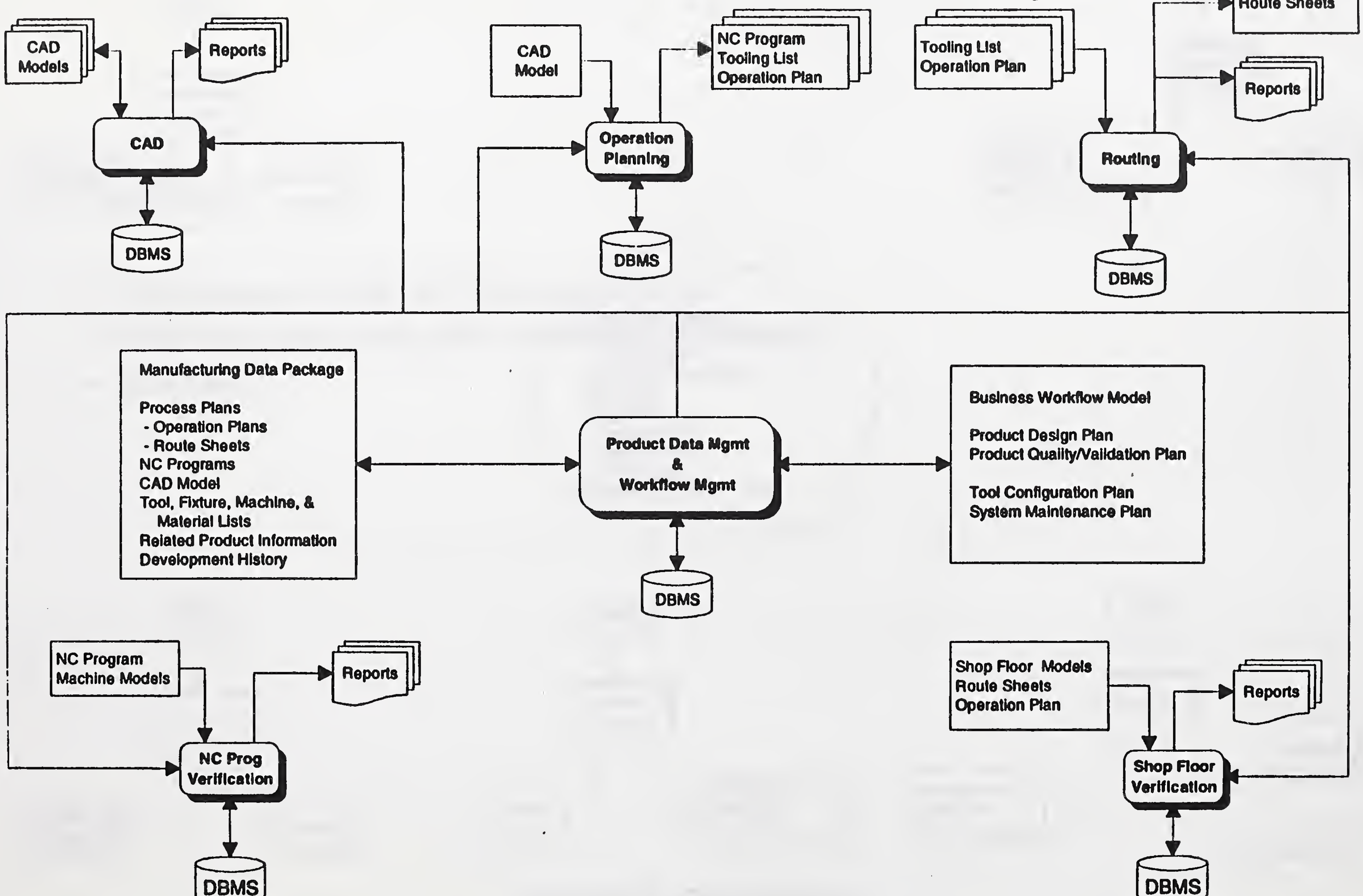
Outline

- Preliminary CAME METK Architecture
- Manufacturing Integration Architectural Issues
- Summary

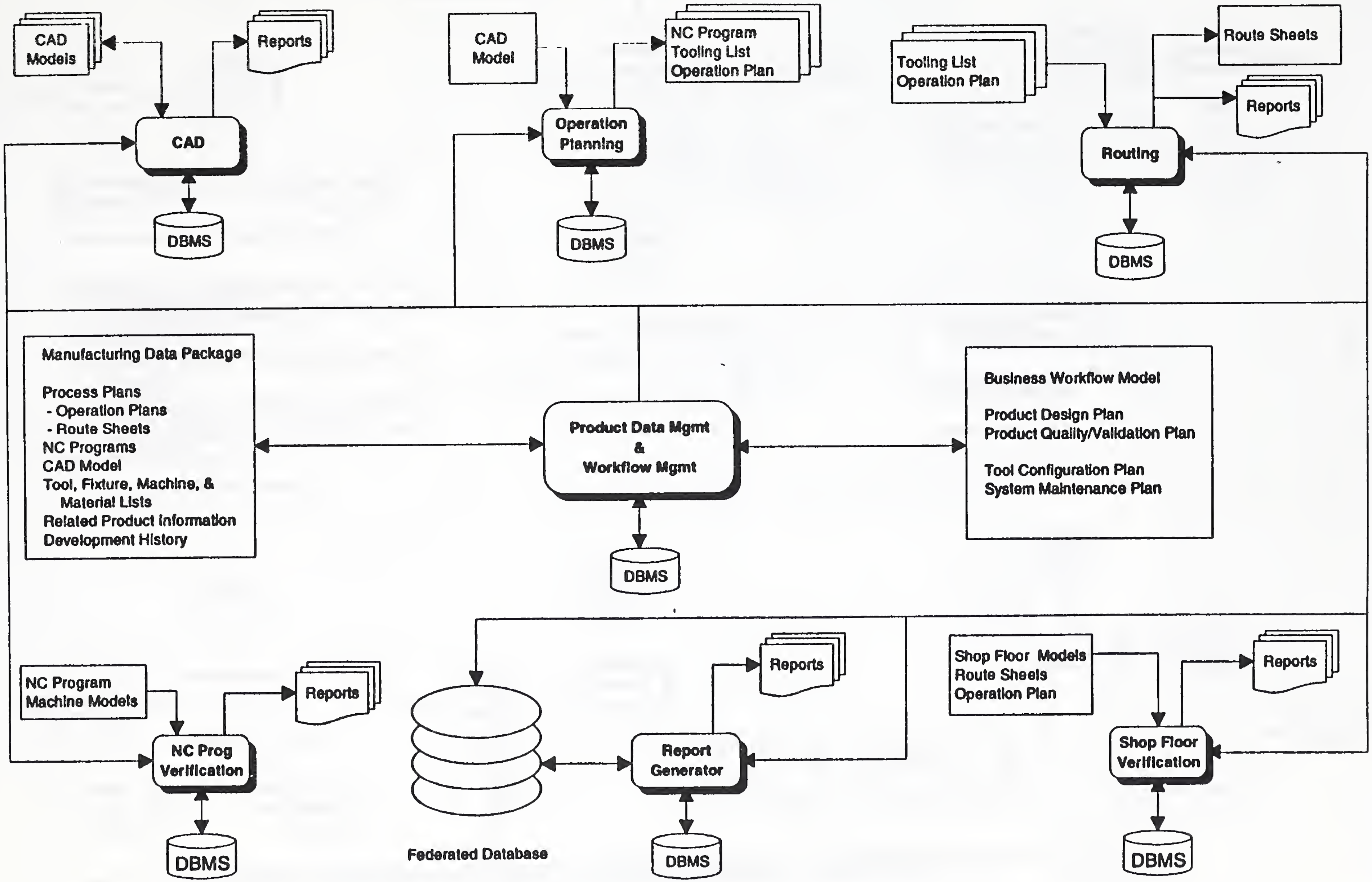
Current situation



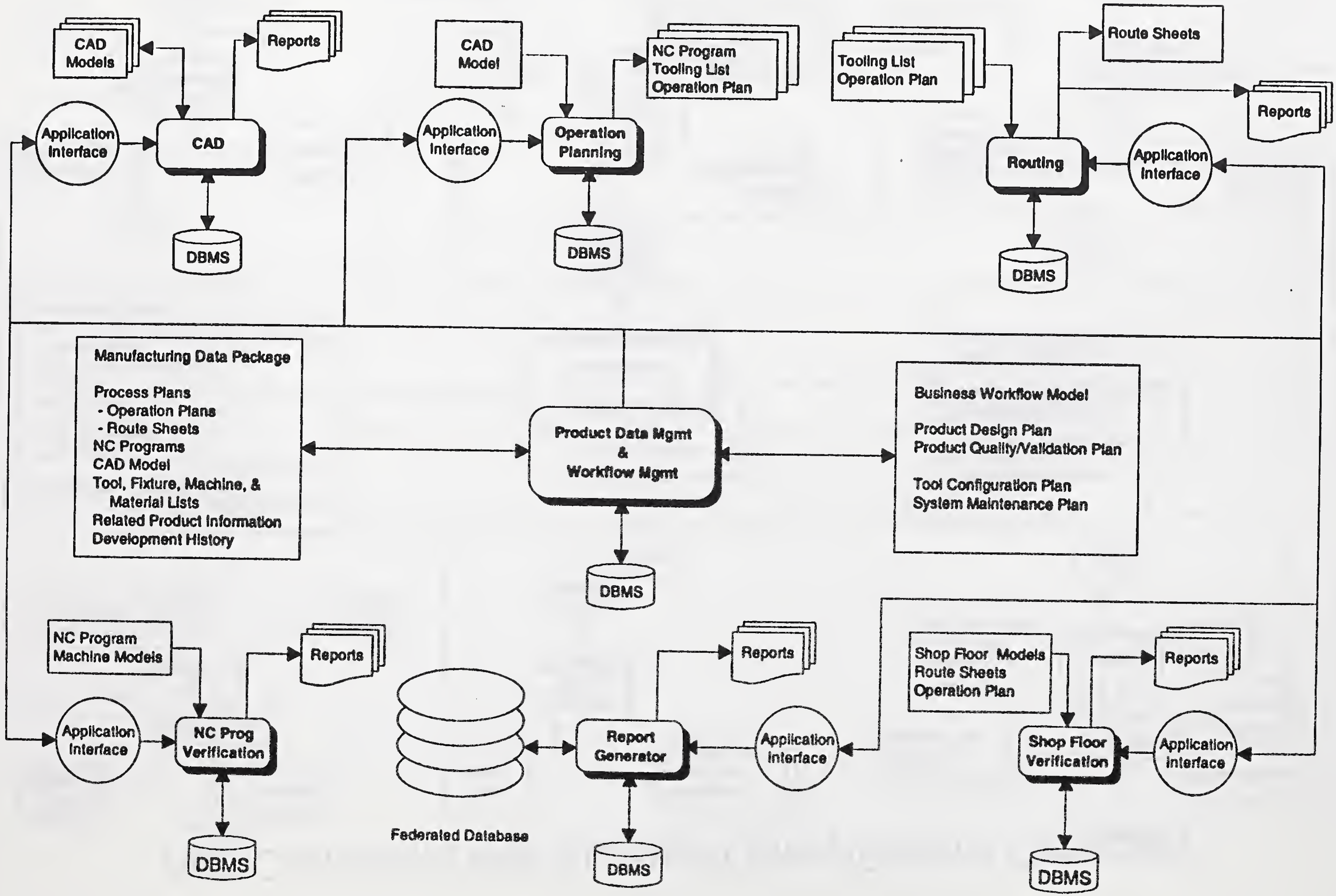
Provide Product Data & Workflow Management



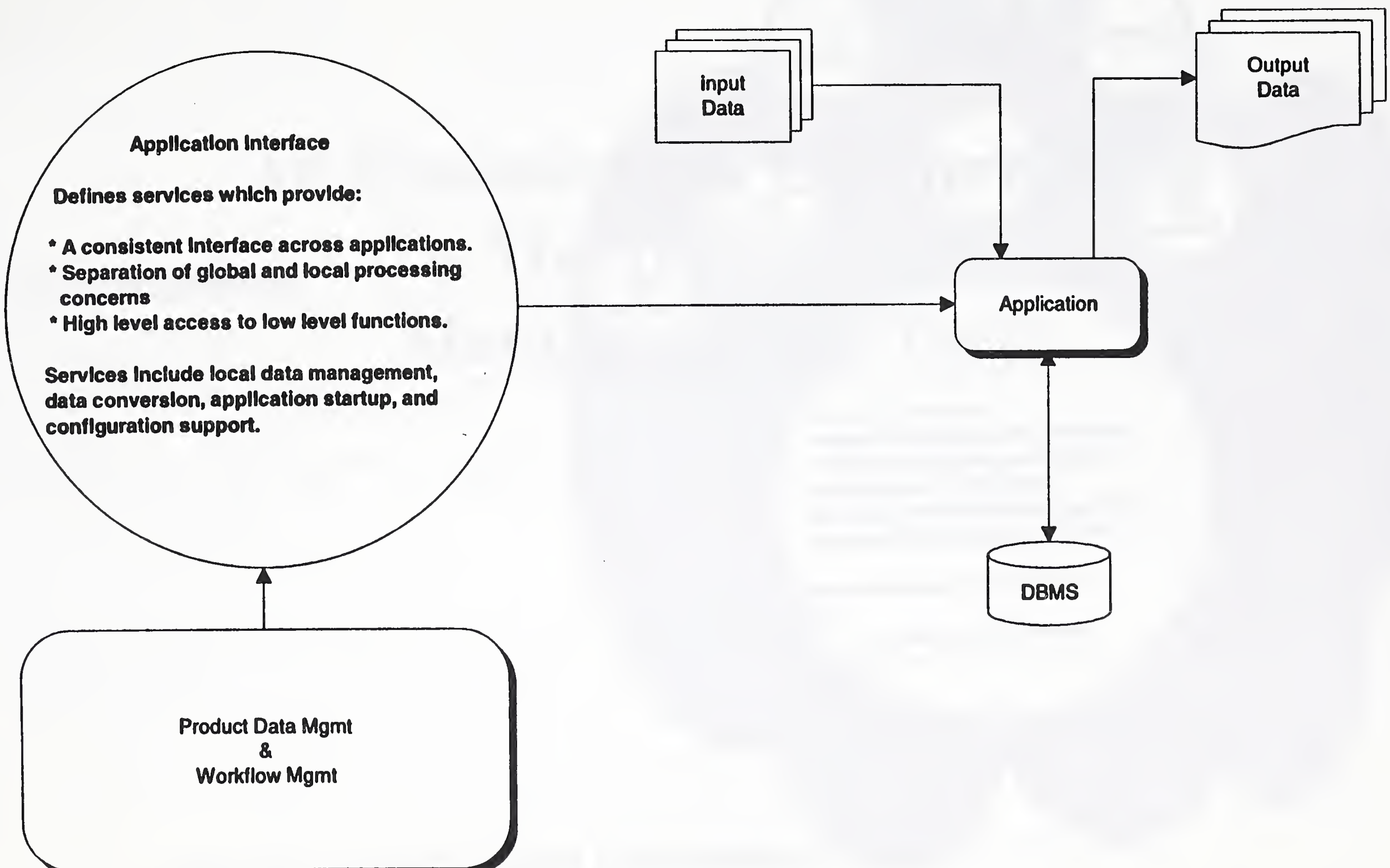
Add Common Data Storage/Query/Report Capability



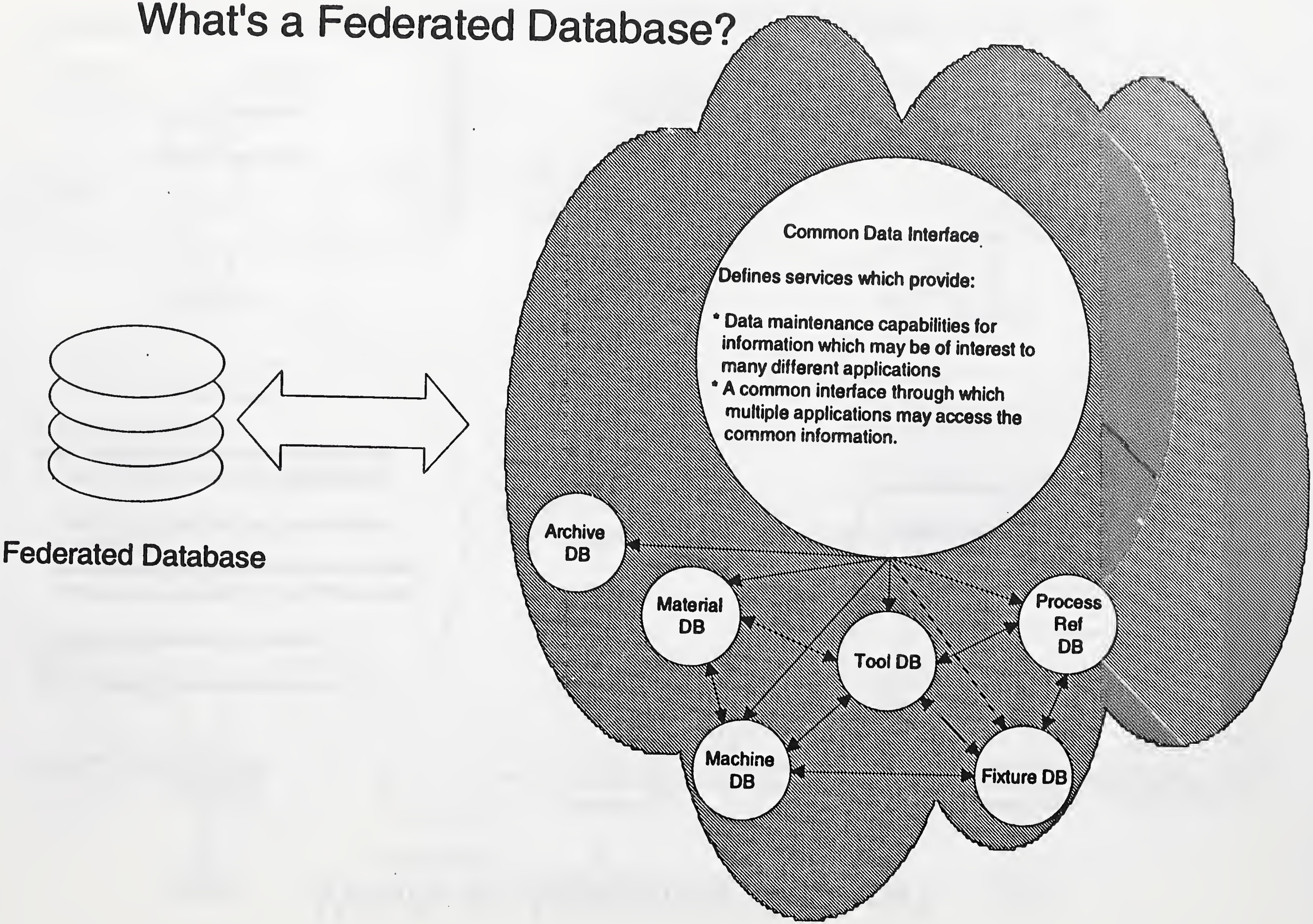
Plus Generalized Application Interfaces = METK



What's an Application Interface?



What's a Federated Database?



An Examination Of Challenges To The Integration Of Manufacturing Tools

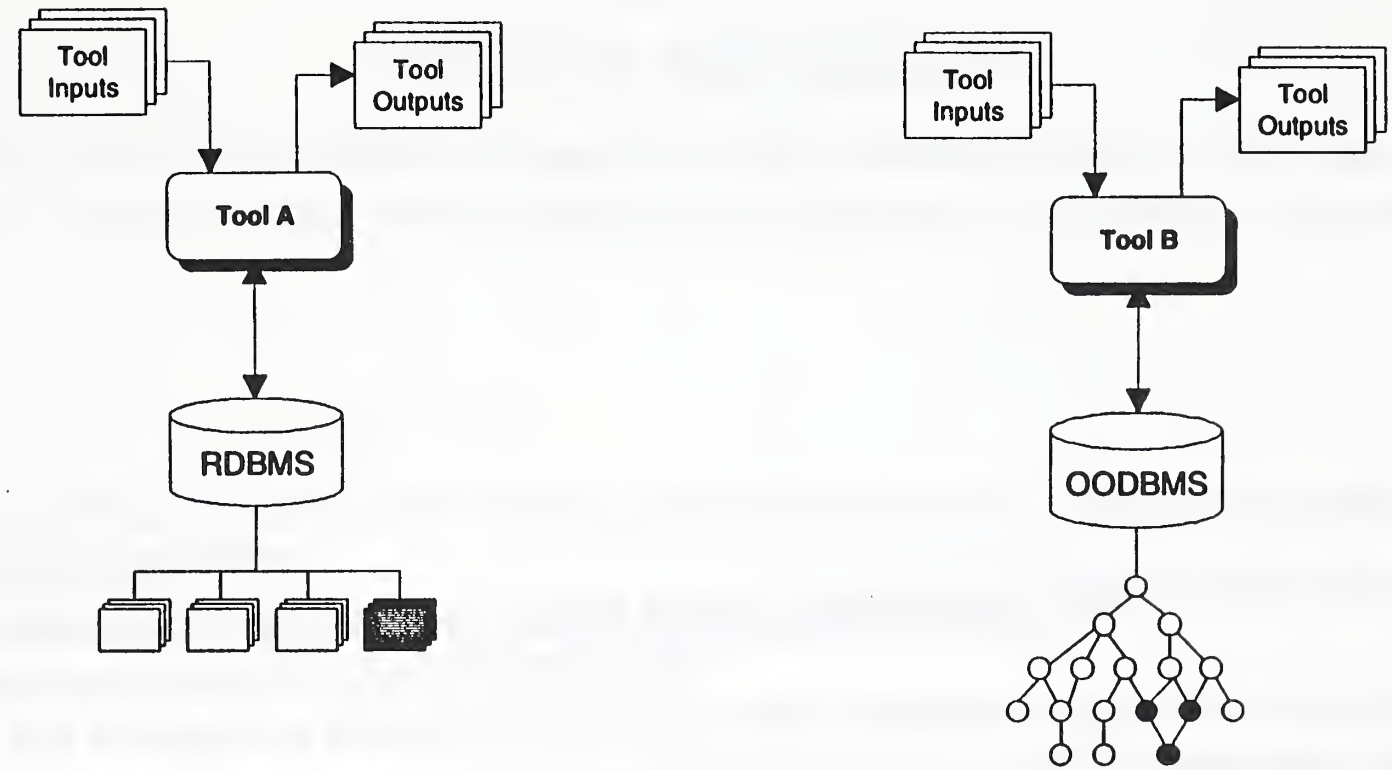
Reference Data Sharing

Reference data describes some characteristic or capability of a physical entity that is of interest to manufacturing tools. It changes very infrequently, if at all. It is used by manufacturing tools as an input to their processing.

Issues:

- Each tool that uses reference data stores it in an internal(proprietary) format.
- Different tools may have slightly(or greatly) different copies of the reference data.
- Each tool provides a different mechanism for getting the data into the tool.

Reference Data Sharing(Continued)



Example:

Tooling and fixture specification information is used by process planning and simulation tools. There is no sharing of the reference data or verification that each tool's copy of the reference data is consistent with another tool's copy.

Reference Data Sharing(Continued)

Possible Solutions:

- Identify types/classes/groupings of reference data that is of interest to different manufacturing tools.
- Define information models and interface specifications for each reference data type.
- Facilitate the creation of standard reference data releases adhering to the interface specifications for each reference data type.
- Provide mechanisms for tools to import and/or dynamically access data from reference data releases.

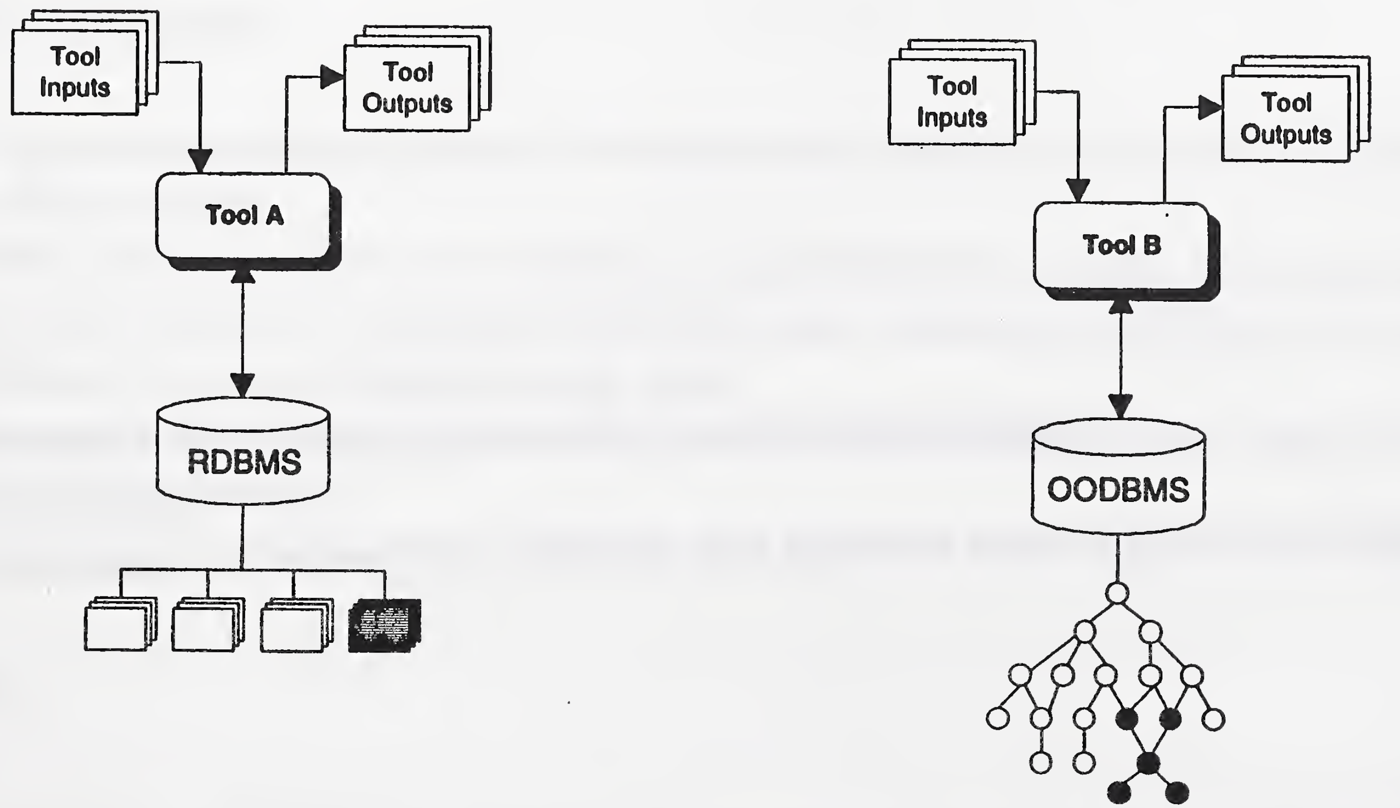
Reference Data Extension

Some tools provide the ability to create or modify reference data. This new information can be viewed as an extension or expansion of the basic reference data set.

Issues:

- This capability increases the chances that different tools will be using different reference data.
- There is no way for other tools to get the new information.

Reference Data Extension(Continued)



Example:

Tooling and fixture specification information can be created and/or modified by a process planning tool, but other tools which might use this information have no way of getting the new information or even knowing that it exists.

Reference Data Extension(Continued)

Possible Solutions:

- Facilitate the establishment of standard reference data releases.
- Provide mechanisms for tools to include extensions/addenda to the reference data releases.
- Provide access mechanisms to reference data so that tools can view the standard reference data release and the addenda as one consistent data store.
- Promote the creation of standard mechanisms for the extension of the standard reference data releases with newly created or corrected information.

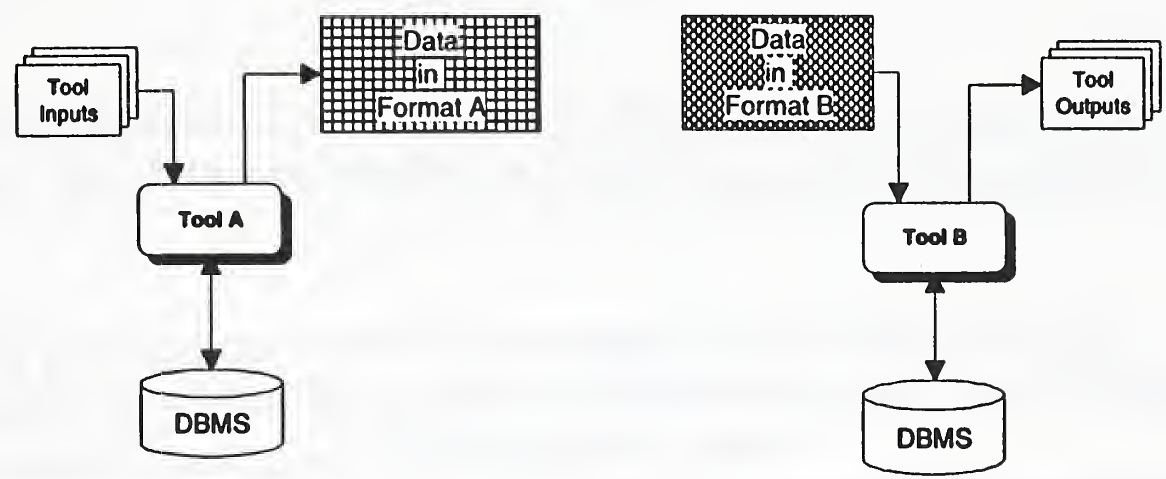
Generated Data Sharing

Often data generated by one tool would be useful as input to other tools. Verification of the results of the development process at each step and as a whole can be greatly facilitated if smooth transition of data from tool to tool can be achieved.

Issues:

- The generated data is often stored in an internal database and is not accessible outside of the tool.
- The tools which could make use of the generated data may not have a way of accessing it.
- In cases where tools provide a means of transferring and receiving generated data, the format of the data generated by the transferring tool may be inconsistent with the format needed by the receiving tool.

Generated Data Sharing(Continued)



Example:

Routing information produced by a process planning tool could be used by a discrete simulation tool(to simulate a part's movement through a production line) but no facility may exist to interchange the data.

Generated Data Sharing(Continued)

Possible Solutions:

- Identify data generated by a tool which may be of interest to other tools.
- Provide mechanisms to export data that may be of interest to multiple tools.
- Provide mechanisms to import tool generated data.
- Propose standard interchange formats for generated data, and provide translators to convert data to and from current proprietary vendor data formats.

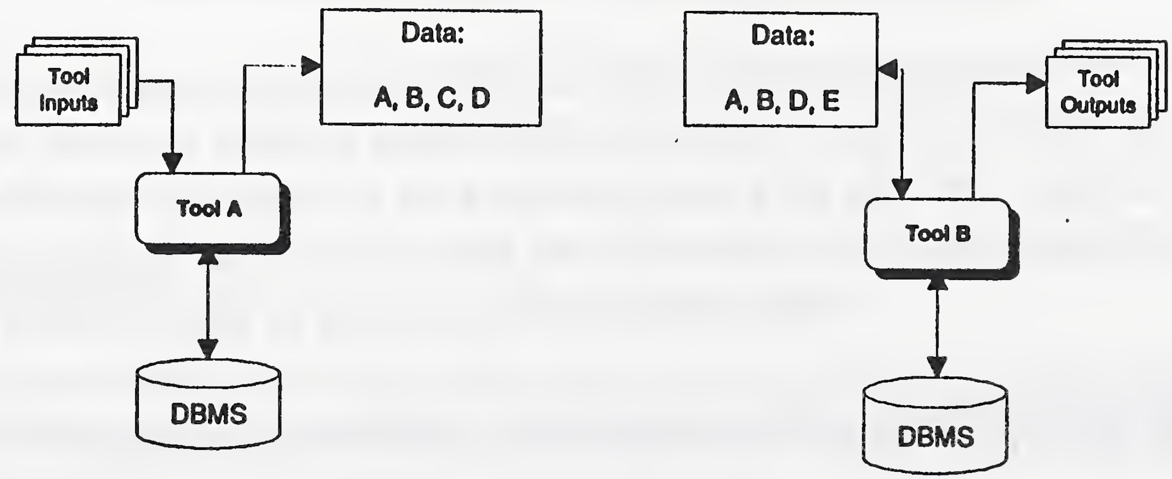
Modification Back-Propagation

It would be advantageous if when a tool modified the generated data of another tool, the originator of the data was notified and any relevant data updated to reflect the modification.

Issues:

- There are no facilities for notifying a tool when modifications to its generated data occur.
- Tools often cannot read in their own generated data.
- If the generated data of one tool can be imported into a second tool, the second tool often converts the data to an internal format for its use. When modifications are made the internal copy is changed, and there is no indication of what would need to be changed in the original generated data to maintain consistency.

Modification Back-Propagation (Continued)



Example:

A process planning system may read in a CAD drawing and modify a tolerance on a feature. No facility exists to notify the CAD system of the change and have it make an appropriate modification.

Modification Back-Propagation (Continued)

Possible Solutions:

- Facilitate the manual transfer of modification information through its support in the configuration management system and through the use of workflow management systems.
- Promote vendor support of automatic notification of modification information through message passing, intelligent agents, and sharing of common data components.

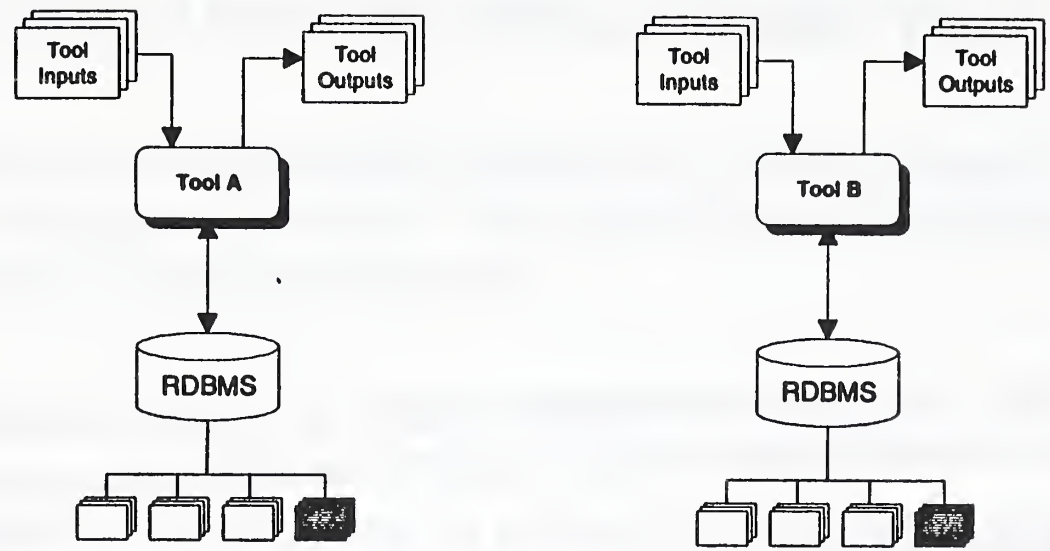
Redundant data storage

Tools which may use the same data often each keep copies of the data.

Issues:

- Few facilities exist for the sharing of the same data by multiple tools.
- Tools may keep both externally accessible and internal copies of the same data.

Redundant data storage(Continued)



Example:

Tooling and fixture specification information is used by process planning, simulation, and cost estimating tools.

Redundant data storage(Continued)

Possible Solutions:

- Identify data elements which are common across different tools.
- Provide a storage management system for common elements.
- Integrate the storage management system with the configuration management system to provide the greatest flexibility in storing the common elements in a distributed fashion.
- Integrate a workflow management system with the configuration management and storage management systems to further minimize unnecessary redundant data storage.

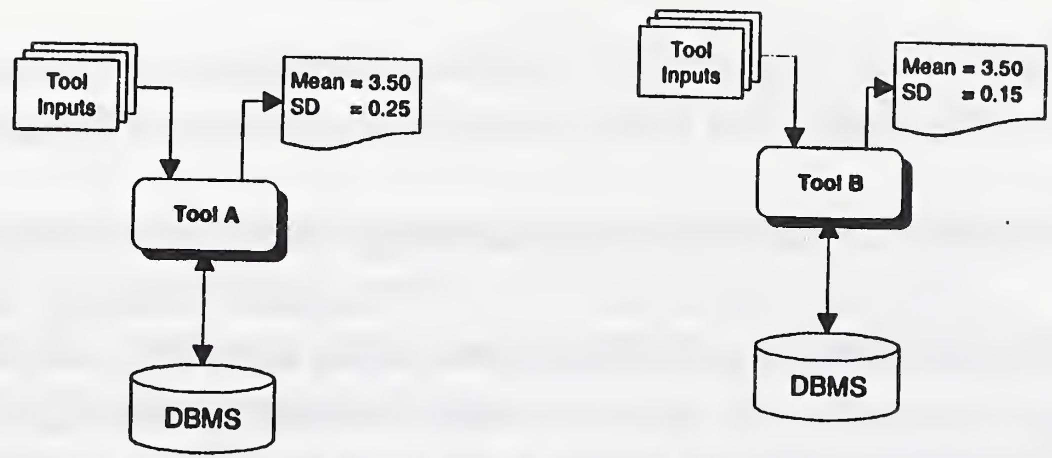
Result Variability Between Similar Tools

It would be advantageous if tools which perform the same function but are from different vendors produce like results. This would allow for the greatest flexibility when picking a suite of manufacturing tools.

Issues:

- Tools using different reference data as a basis for their processing will have a high probability of producing different results.
- The rules/processes/decision logic that tools use for processing are usually different and thus lead to differing results.
- Different tools choose to report different information even when similar information is available.
- The format of results produced by different tools may differ even when the content is the same, indicating variability of results when in fact the results are consistent.

Result Variability Between Similar Tools(Continued)



Example:

Tools which give a statistical analysis of some manufacturing information may give results which seem to have insignificant variability, but which may in fact be significant.

Result Variability Between Similar Tools(Continued)

Possible Solutions:

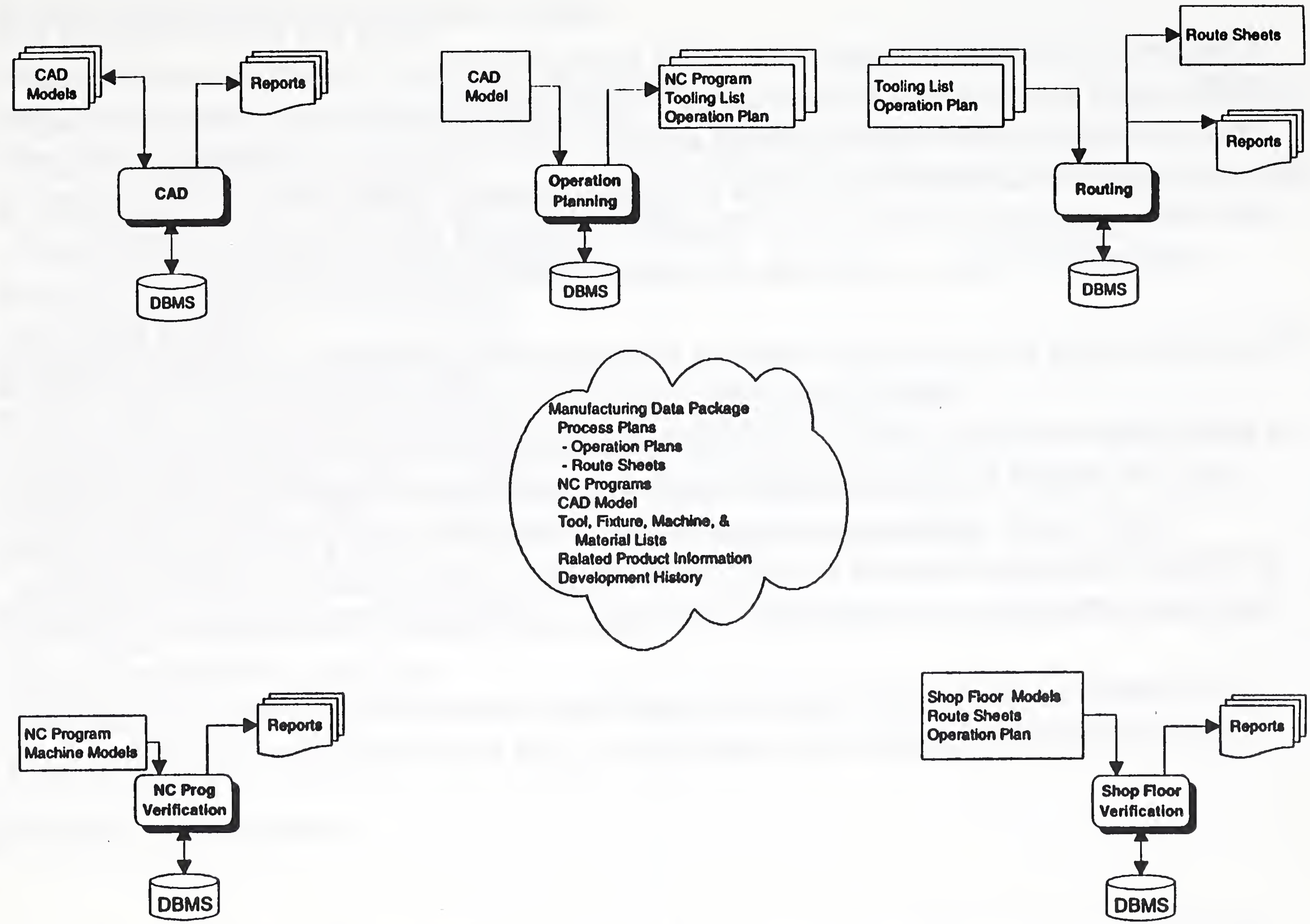
- Facilitate the creation of standard reference data releases and promote their use by all tools.
- Promote the creation of standard knowledge bases to be used as a basis for tool processing and promote their use by tool vendors.
- Promote the creation of standard report formats for important types of tool generated data.

Engineering data package(EDP) validation support

One of the goals in integrating manufacturing tools it to provide better support for the verification of engineering data packages.

Issues:

- The content of an EDP is not well defined.
- Different tools which provide similar functions may not produce all the data needed by the EDP.
- No mechanisms exist to ensure that the parts of an EDP are consistent.
- Some data produced by a tool which is desired to be included in an EDP may only be kept internally.


CAD Models
Reports
CAD
DBMS
CAD Model
Operation Planning
NC Program
Tooling List
Operation Plan
DBMS
Tooling List
Operation Plan
Route Sheets
Reports
Routing
DBMS
Manufacturing Data Package
Process Plans
- Operation Plans
- Route Sheets
NC Programs
CAD Model
Tool, Fixture, Machine, &
Material Lists
Ralated Product Information
Development History
NC Program
Machine Models
Reports
NC Prog Verification
DBMS
Shop Floor Models
Route Sheets
Operation Plan
Reports
Shop Floor Verification
DBMS

Engineering data package(EDP) validation support(Continued)

Possible Solutions:

- Define all of the elements of an EDP. Included in the definition should be the standard reference data releases and generated data formats to be used by all tools operating on the EDP.
- The EDP definition can specify for a core set of elements and optional elements.
- Provide a facility for engineers developing an EDP to include electronic copies of notes, comments, ad hoc drawings, etc., as optional elements of the EDP.
- Establish standard data formats for each data element that is a part of an EDP.
- Facilitate the creation of translation mechanisms to convert tool generated data in an non-EDP supported format into an EDP supported format.
- Establish rules for verifying the consistency of each element of an EDP and for the EDP as a whole.
- Provide a configuration management system for the EDP. The configuration management system should provide an interface for access to the EDP such that the data may be distributed between different tools or processors without each tool having to maintain such knowledge. Change history logging and auditing should also be provided. This will facilitate verifying the correctness of each EDP element, verifying the consistency of the EDP as a hole, and maintaining the correctness and consistency of the EDP.

Other Integration problem areas

- Manufacturing Data Package archiving
- Development history storage/retrieval
- Tool initialization/shutdown
- Distributed work environment complexity
- Data synchronization/data integrity
- Software version compatibility
- Operating system compatibility
- Overlapping tool functionality
- Tool and system performance
- Incompatible information models
- Tool installation/configuration
- Tool & system maintenance
- Graphic output incompatibilities

Summary

- Preliminary CAME METK Architecture

- Manufacturing Integration Architectural Issues

- Solicitation of comments
  - What needed functionality is not provided by the proposed METK architecture?
  - Which issues are not fully addressed by the proposed METK architecture?
  - How does your organization wish to be involved in the development of the final METK architecture specification.

Appendix G:

"SIMA Production System Engineering Environment"

SIMA Production System Engineering Environment

Chuck McLean and Swee Leong
Manufacturing Systems Engineering Group

CAME Industry Meeting
August 1995

NIST

Topics

* SIMA Production Project Goals and Strategies
* Technical Approach
* Production System Engineering Model
* Engineering Functions and Candidate Tools
* Industrial Participants
* University Research Team
* Kick Off Meeting Plans

MEL

SIMA Production Project

Goal:

To develop solutions for integrating production engineering, planning, scheduling, and simulation systems with other manufacturing life cycle applications

Strategy:

- Identify industrial collaborators with diverse production systems
- Leverage on-going research on our OA-sponsored projects
- Develop process and information models for production applications
- Coordinate our efforts and integrate results with other SIMA projects
- Use a product data management system to integrate production engineering and production planning systems
- Test solutions using simulation capabilities established for the AMSANT facilities

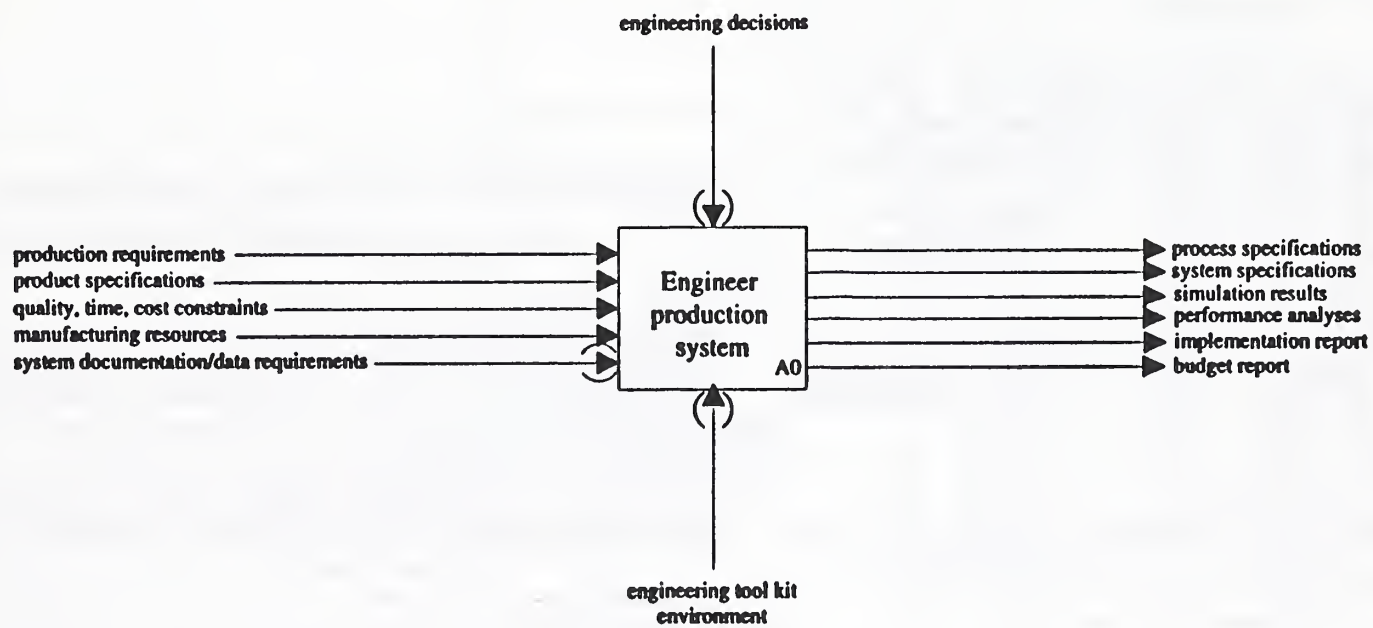
NIST

Technical Approach

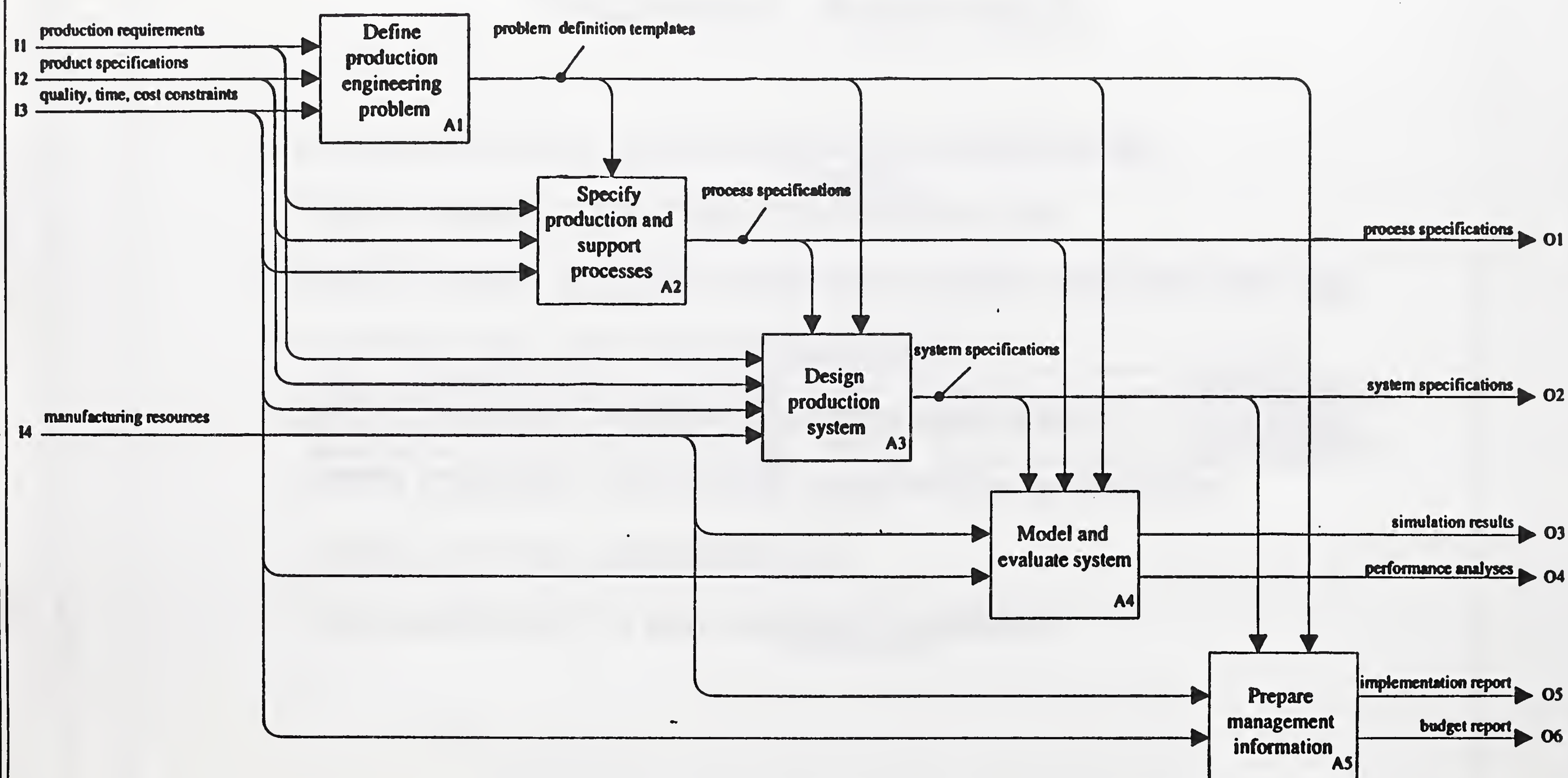
* Develop production system engineering process model
* Select computing platform(s) and candidate tools
* Identify baseline production engineering problem and test case data
* Load test case data into selected tools
* Define information models and links between tools
* Identify integration opportunities and interface specifications
* Integrate and test commercial tools
* Apply environment to new engineering problems

MEL

NODE:	TITLE: Production System Engineerin, .el	DATE: 07/14/9	P. 1



NODE: A0	TITLE: Engineer production system	DATE: 07/14/95	P. 2



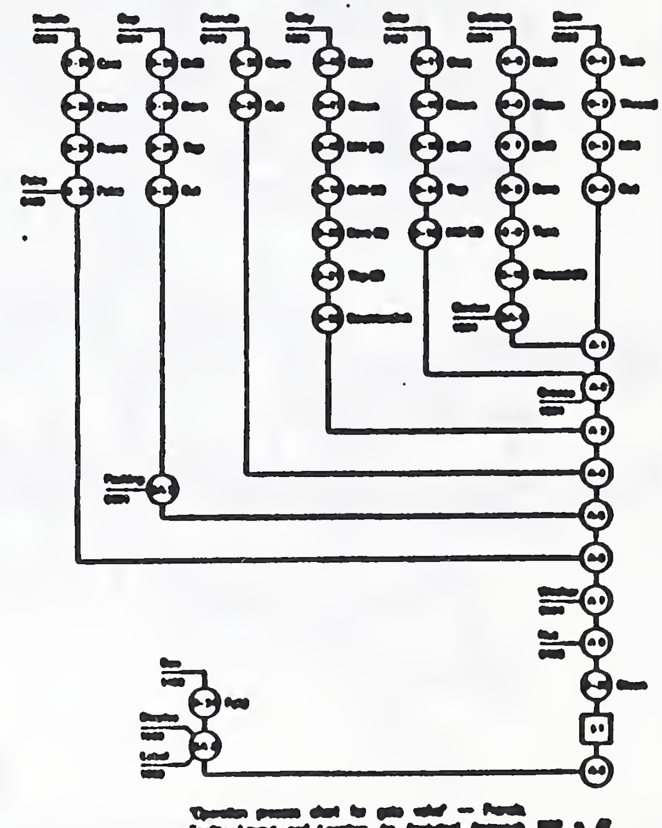
NIST

Process Specification

- Used to develop flowcharts and specifications of the proposed manufacturing process

- Provide functions to create and edit bills of materials, assembly precedence charts, process flow charts, material flow diagrams, pictorial process diagrams, from-to charts, hierarchical decomposition of processes, simplified system schematics, etc.

- Maintain reference library of standard and existing process flowcharts, diagrams, text specifications



MEL

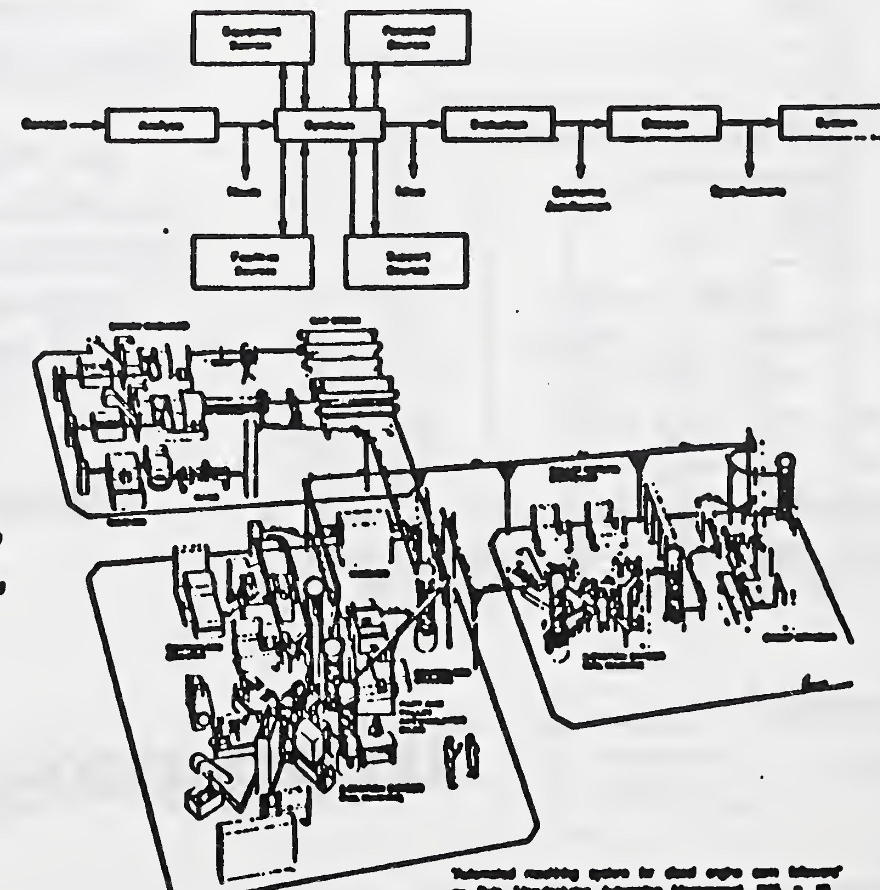
NIST

System Design

- Used to develop designs for production lines, manufacturing cells, material handing systems, workstations, production and support equipment

- Provide functions to create integrated system models and technical specifications

- Maintain reference library of system models, existing designs, generic system configurations, technical standards, etc.



MEL

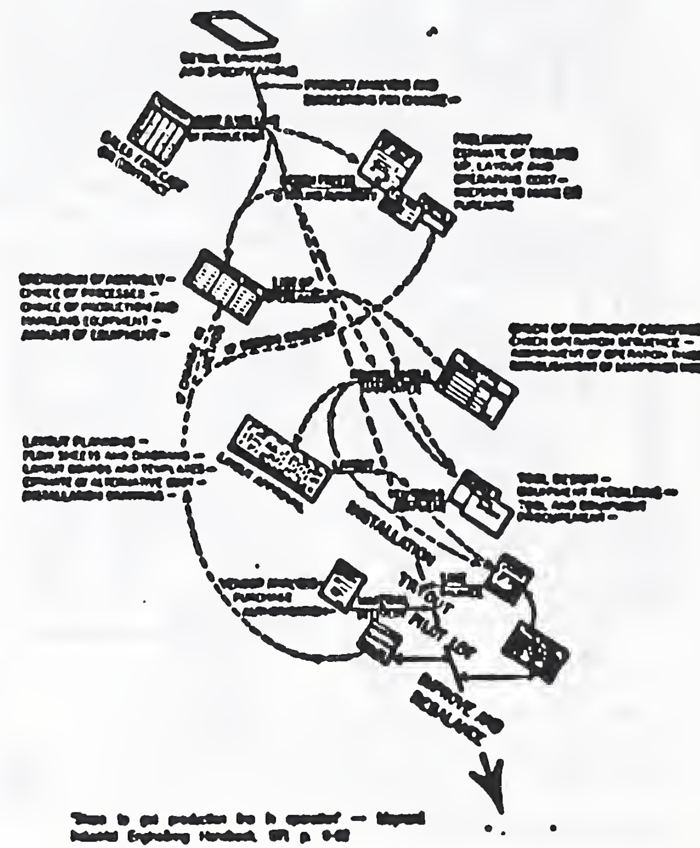
NIST

Production Line Design

– Used to design "production line" type manufacturing systems

– Provide functions to determine rate and volume of production, select processes and equipment, estimate tooling requirements, compute production capacities and operation times, prepare line layout, perform line balancing, plan material acquisition and delivery schedules

– Maintain reference library of systems, line configurations, processing times, performance data, standard line layouts, etc.



MEL

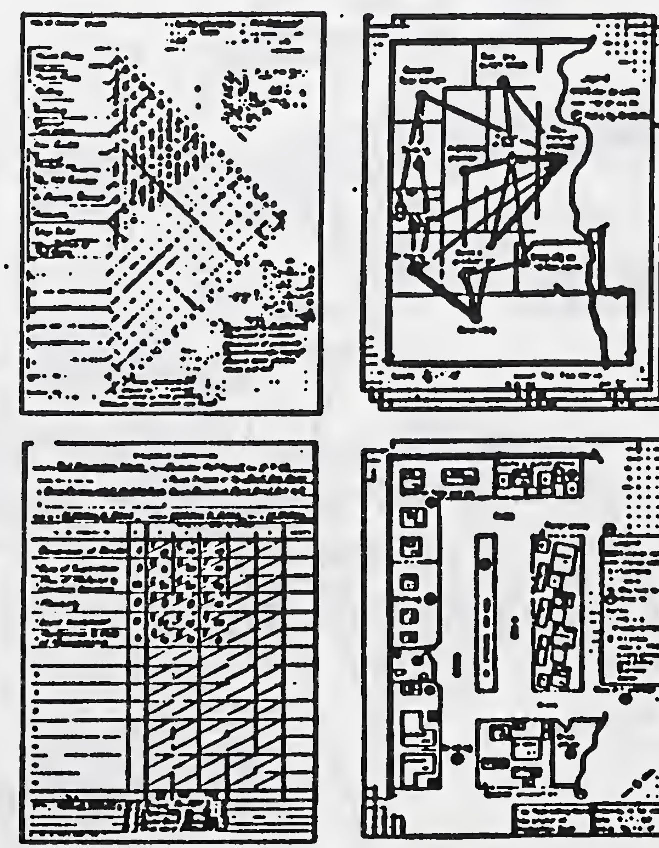
NIST

Plant Layout

– Used to identify facility resource requirements, relationships between requirements, and develop layout plans

– Provide functions to identify activity areas and their features, relationships between areas, total space needs, material flow constraints, and to construct relationship diagrams/layouts

– Maintain reference library of layout data, equipment templates, material handling costs, historical data

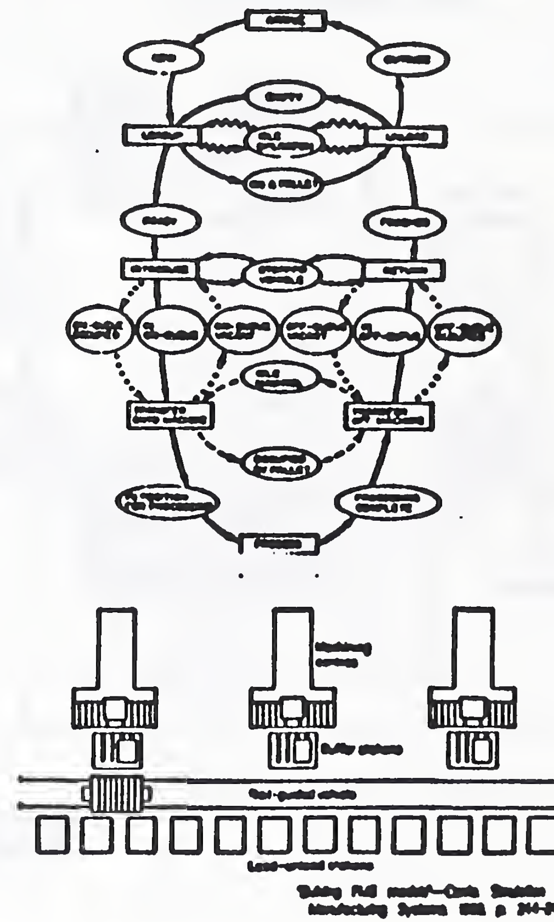


MEL

NIST

Manufacturing Simulation

- Used to model manufacturing system behavior and evaluate performance
- Specify, run, and analyze simulations for throughput, bottlenecks, response to breakdowns, etc.
- Evaluate various system design decisions, e.g. quantity and type of equipment and tooling, storage buffers, workers allocated, scheduling algorithms, etc.
- Maintain simulations of current installed systems, and libraries of: standard models, templates, test data, and industry benchmarks for simulation development and comparison



MEL

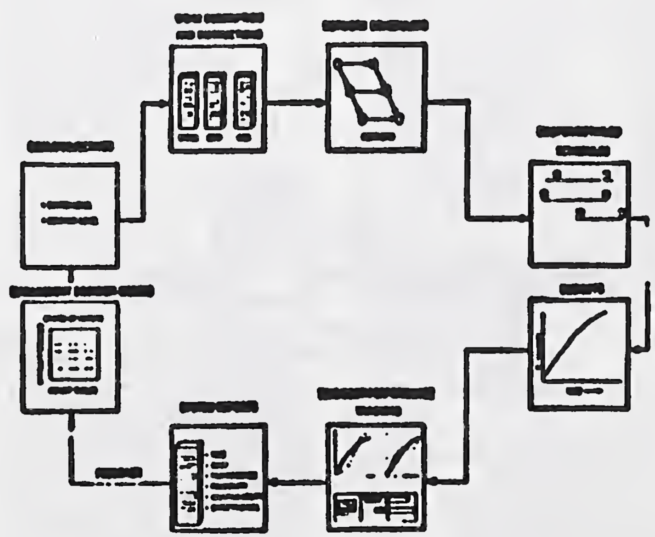
NIST

Project Management

– Used to develop plans, support management decision-making, and track progress on engineering projects

– Provide functions to define goals and objectives, prepare statements of work, generate work breakdown structures (WBS) and budgets, create PERT/CPM and GANTT charts, track costs and performance against time, develop status reports, etc.

– Maintain reference library of plans, standard WBS data, performance history, and budget data



MEL

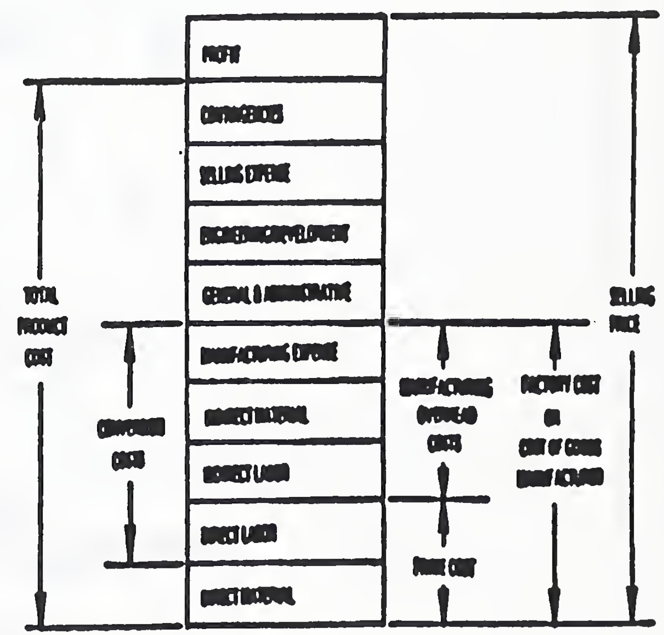
NIST

Cost Estimating

— Used to develop estimates of the costs incurred in manufacturing a product

— Provide functions for retrieving and reviewing product and process data, implementing various cost estimating methods, computing estimated processing times, factoring of learning curves, cross-referencing data based upon group technology classification schemes, etc.

— Maintain reference library of hourly labor and overhead rates, cost of tooling and purchased materials, performance standards, historical cost data



MEL

NIST

Candidate Engineering Tools

* Forms manager and database (MS-Access, . . . ?)
* Process modeling and specification (ABC Flowcharter, . . . ?)
* System engineering (FM/Aspect, . . . ?)
* Mechanical system design (Autocad, . . . ?)
* Plant Layout (FactoryCAD/FLOW, . . . ?)
* Simulation (Promodel, Arena, Quest, IGRIP, . . . ?)
* Cost estimating and budgeting (Excel, . . . ?)
* Project management (Primavera, MS-Project, . . . ?)
* Workflow and product data management (Matrix, . . . ?)
* . . . ?

MEL

NIST

Industrial Participants

* Black and Decker
* Texas Instruments
* GEC Marconi
* Promodel
* Systems Modeling Corporation
* CIMTechnologies
* Framework Technologies
* . . .

MEL

University Research Team

* Production process modeling and simulation
  - Charles Harrell (Brigham Young University - Promodel Corporation)

* System engineering and mechanical systems
  - Bryan Greenway (University of Kansas)

* Plant layout and simulation
  - Charles Parks (Ohio University)

* Project management, cost estimating, and budgeting
  - Charlie Pinkus (California Polytechnic - Pomona)

Kickoff Meeting

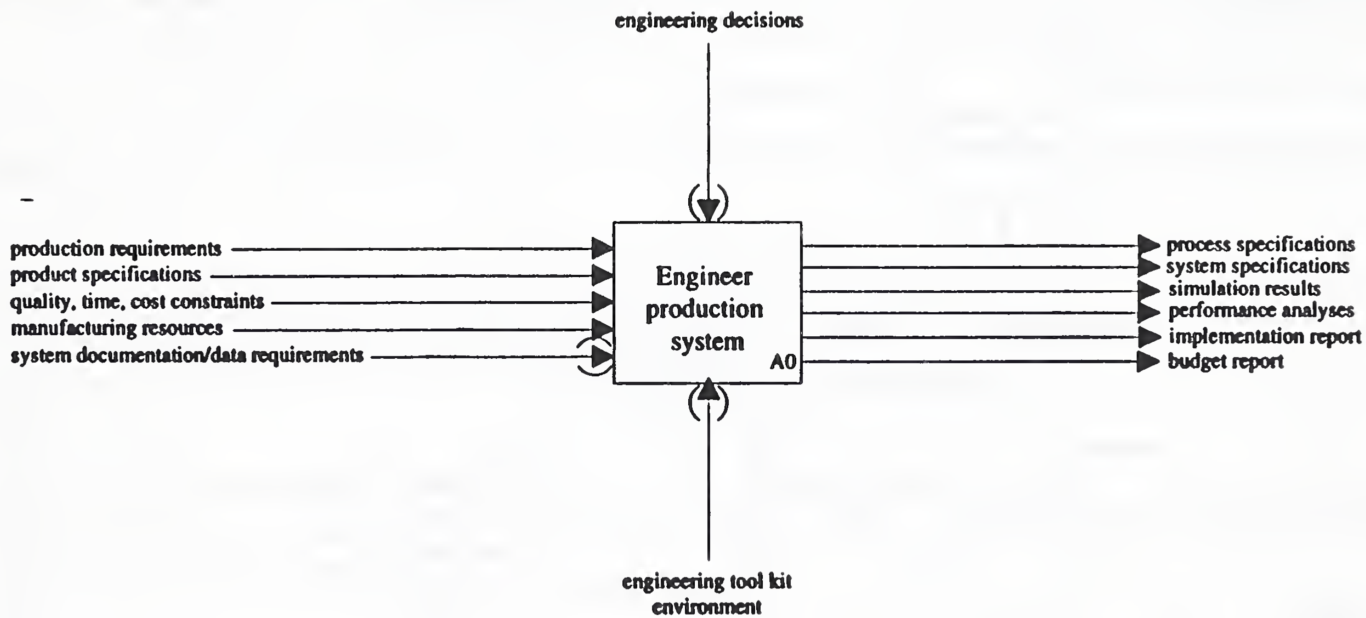
* Black and Decker, Fayetteville, NC - September 18

  - Establish liaisons between teams
  - Review initial software environment
  - Visit production lines
  - Establish baseline data set
  - Develop project plan

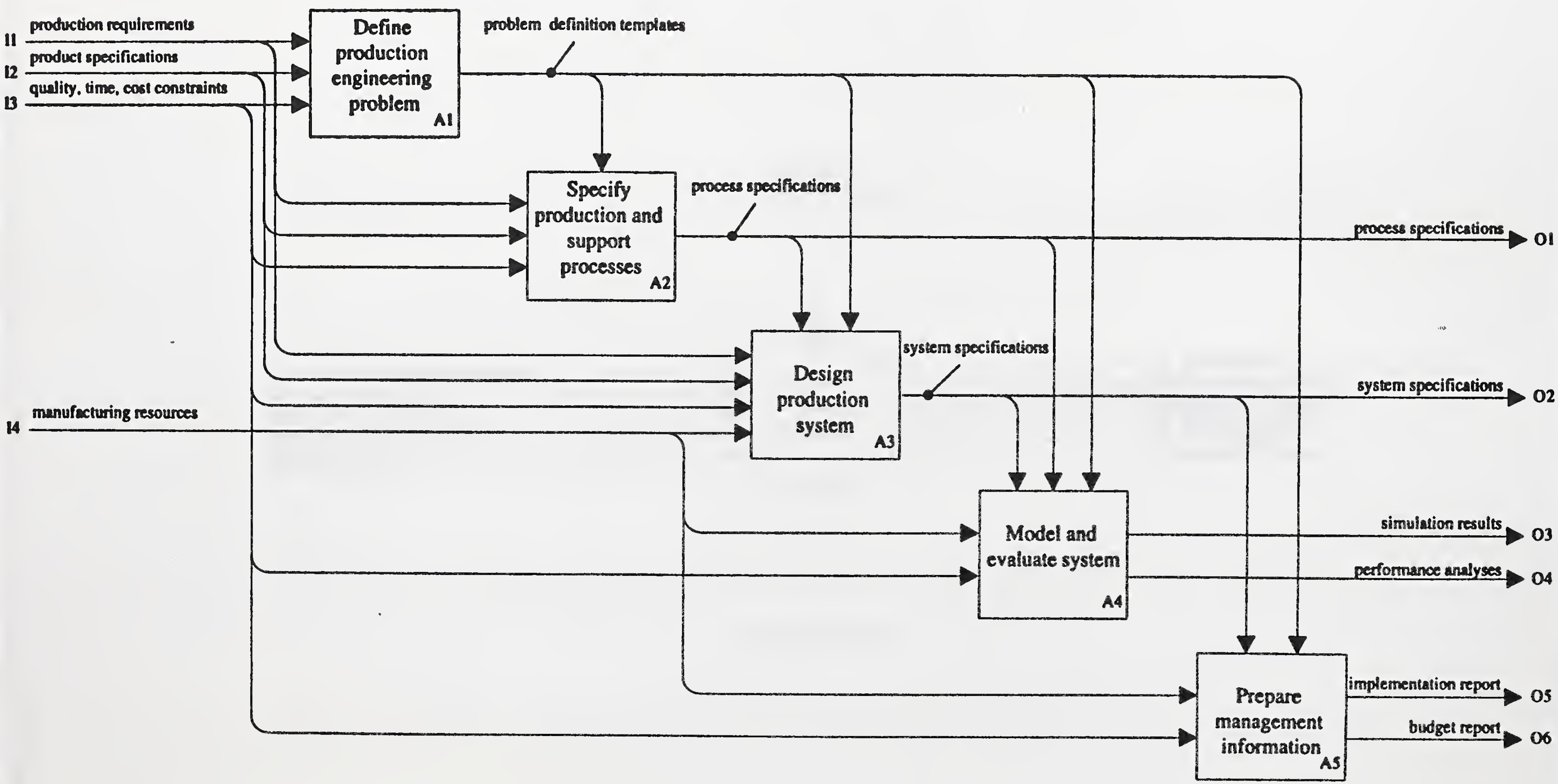
* Engineering tools training, NIST AMSANT Facility - September 19-21

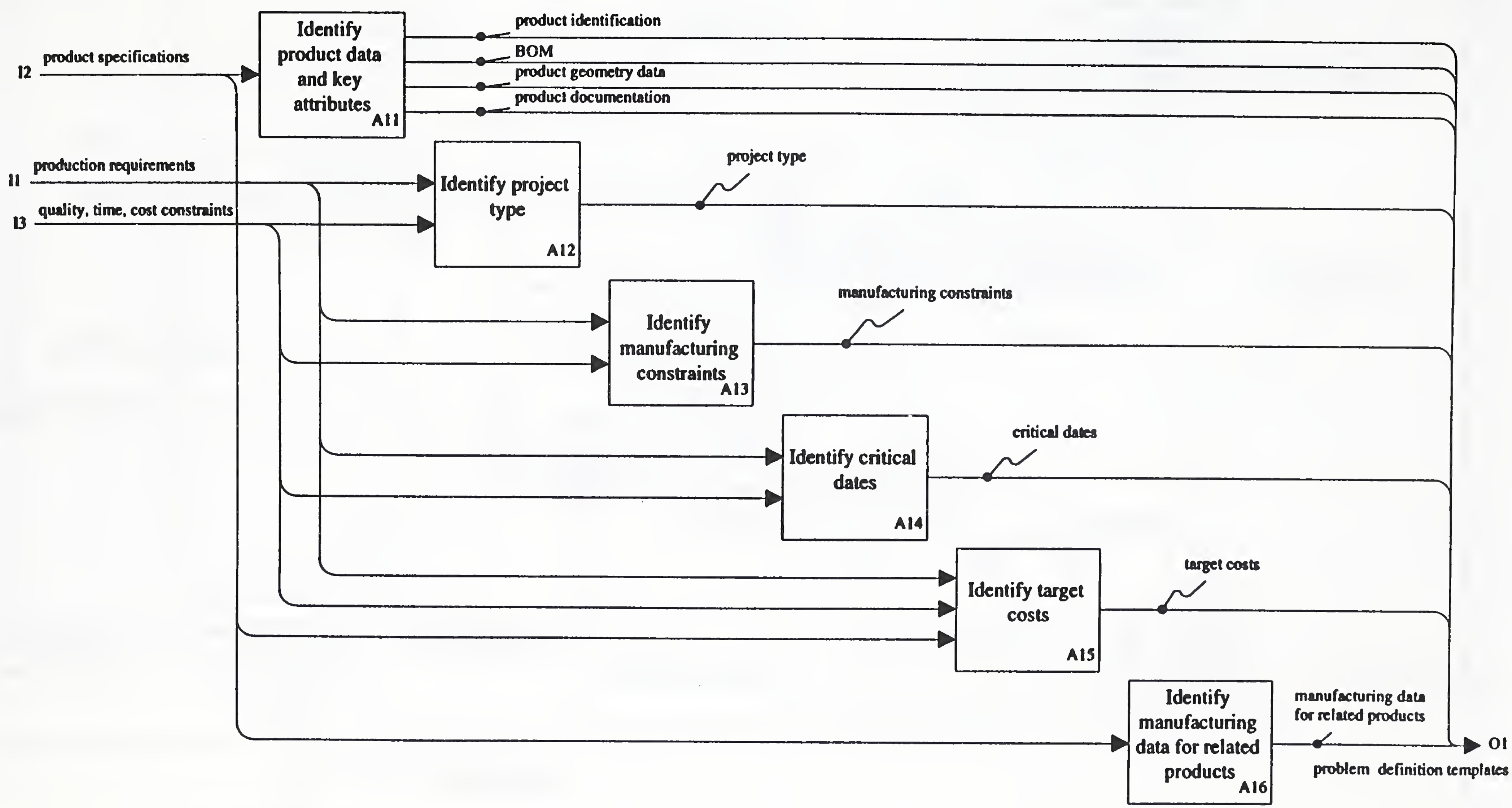
  - CimTechnologies FactoryCAD/FLOW
  - Framework Technologies FM/ASPECT

NODE:	TITLE: Production System Engineeri ›del	DATE: 07/14.	P. 1

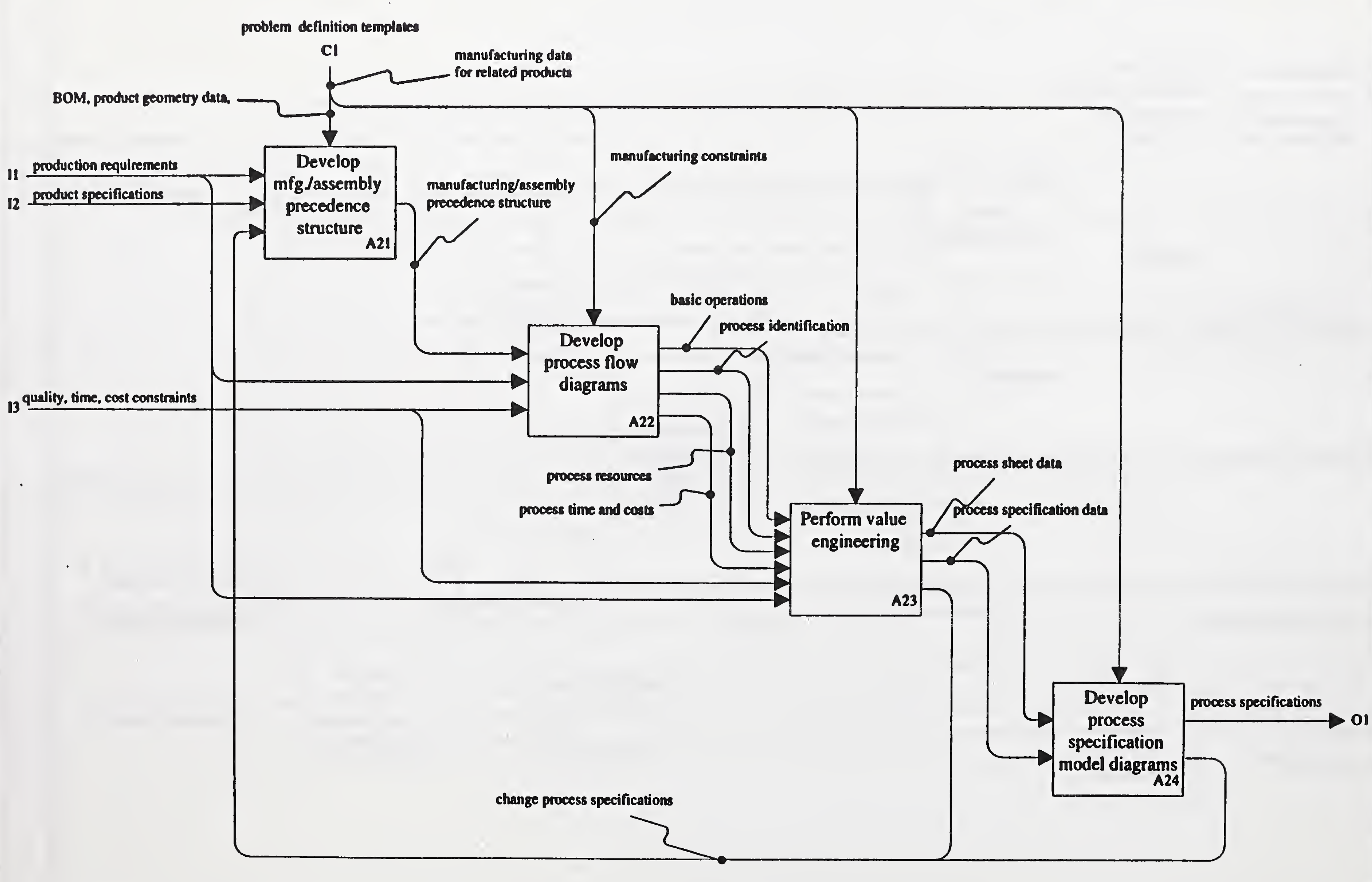


NODE: A0	TITLE: Engineer production system	DATE: 07/14/95	P. 2

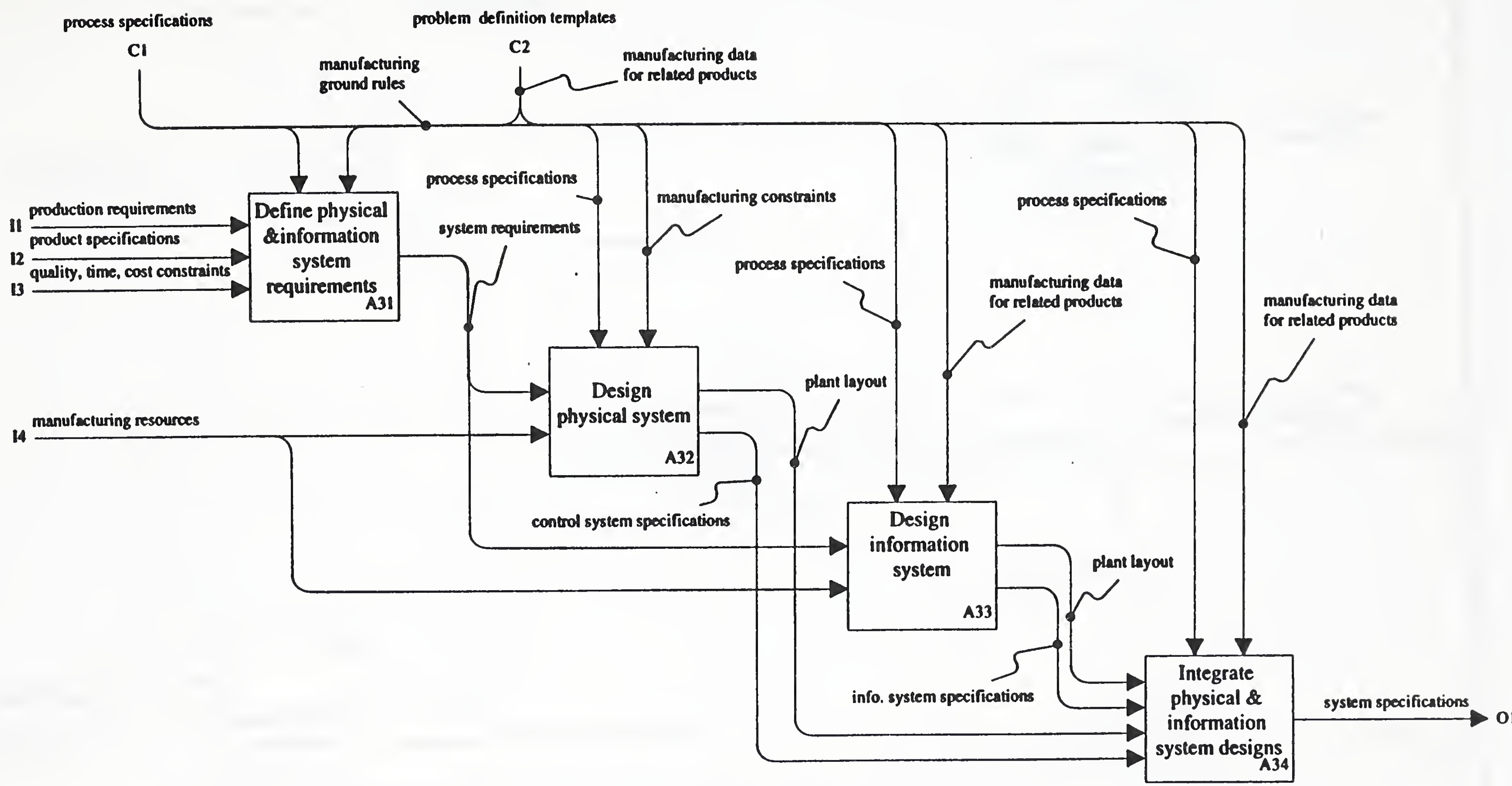



I2
product specifications
I1
production requirements
I3
quality, time, cost constraints
Identify product data and key attributes
A11
product identification
BOM
product geometry data
product documentation
Identify project type
A12
project type
Identify manufacturing constraints
A13
manufacturing constraints
Identify critical dates
A14
critical dates
Identify target costs
A15
target costs
Identify manufacturing data for related products
A16
manufacturing data for related products
O1
problem definition templates

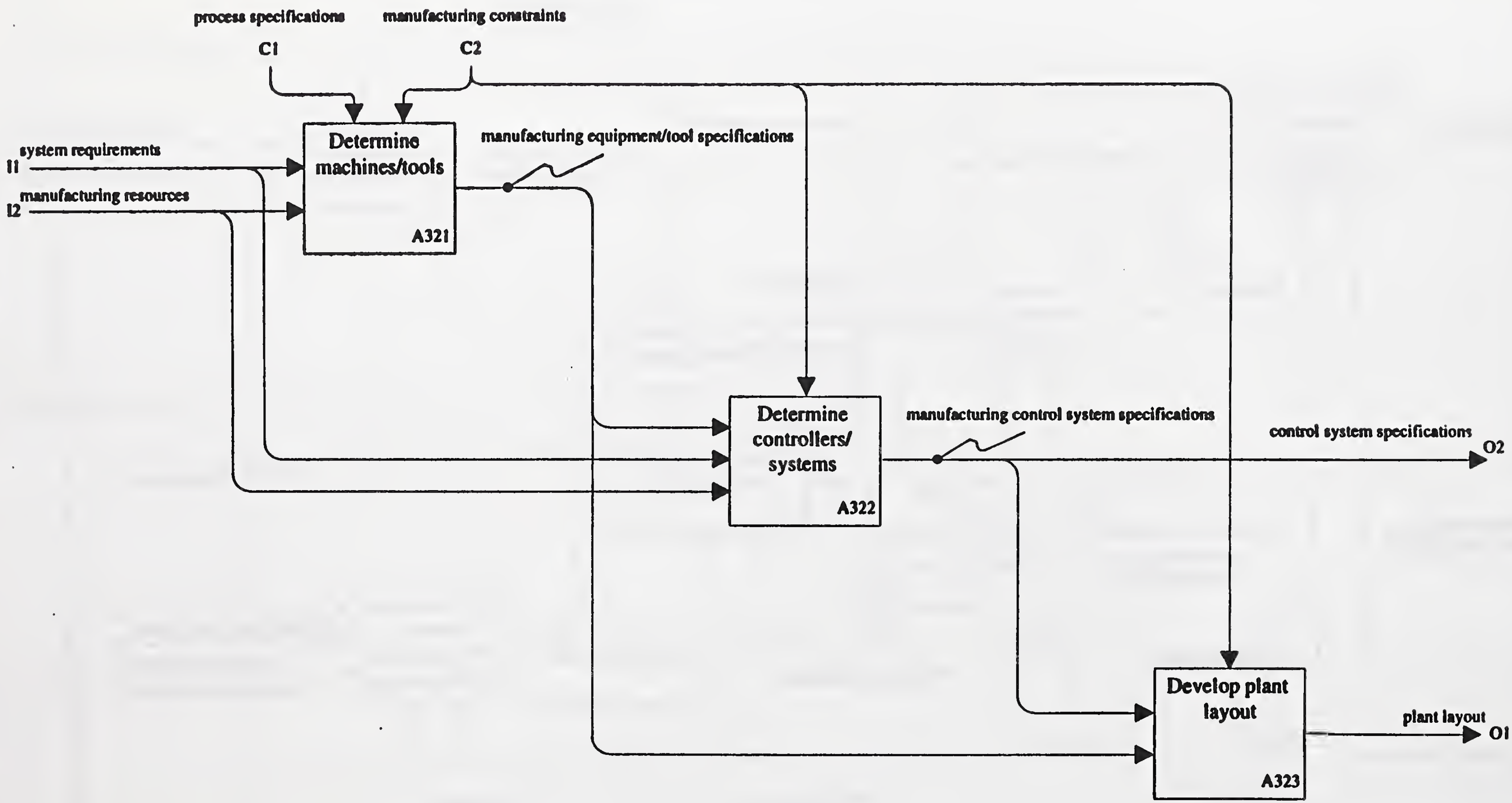
NODE: A2	TITLE: Specify production and support processes	DATE: 07/18/95	P. 4



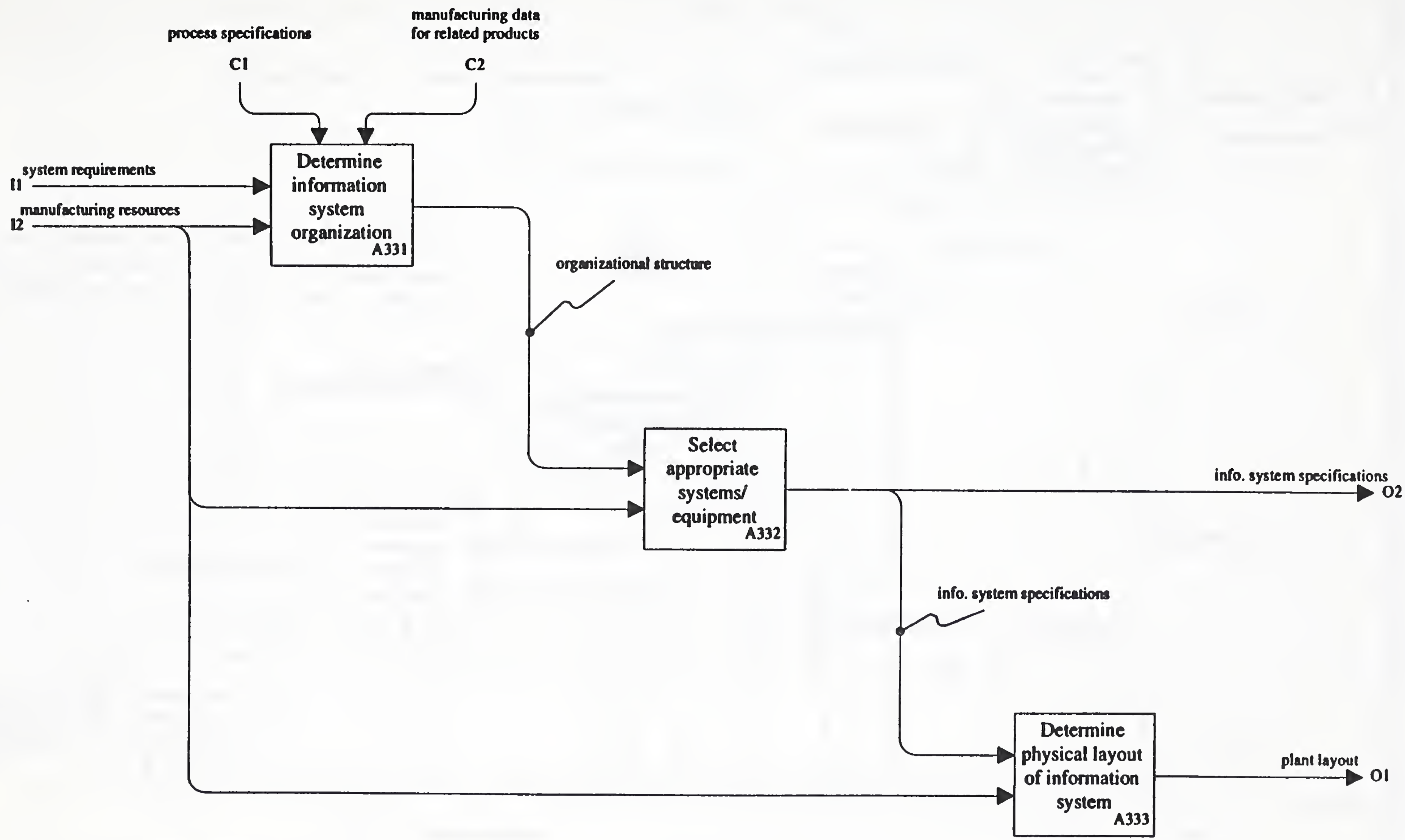
NODE:	TITLE: Design production system	DATE: 07/25/	P. 5



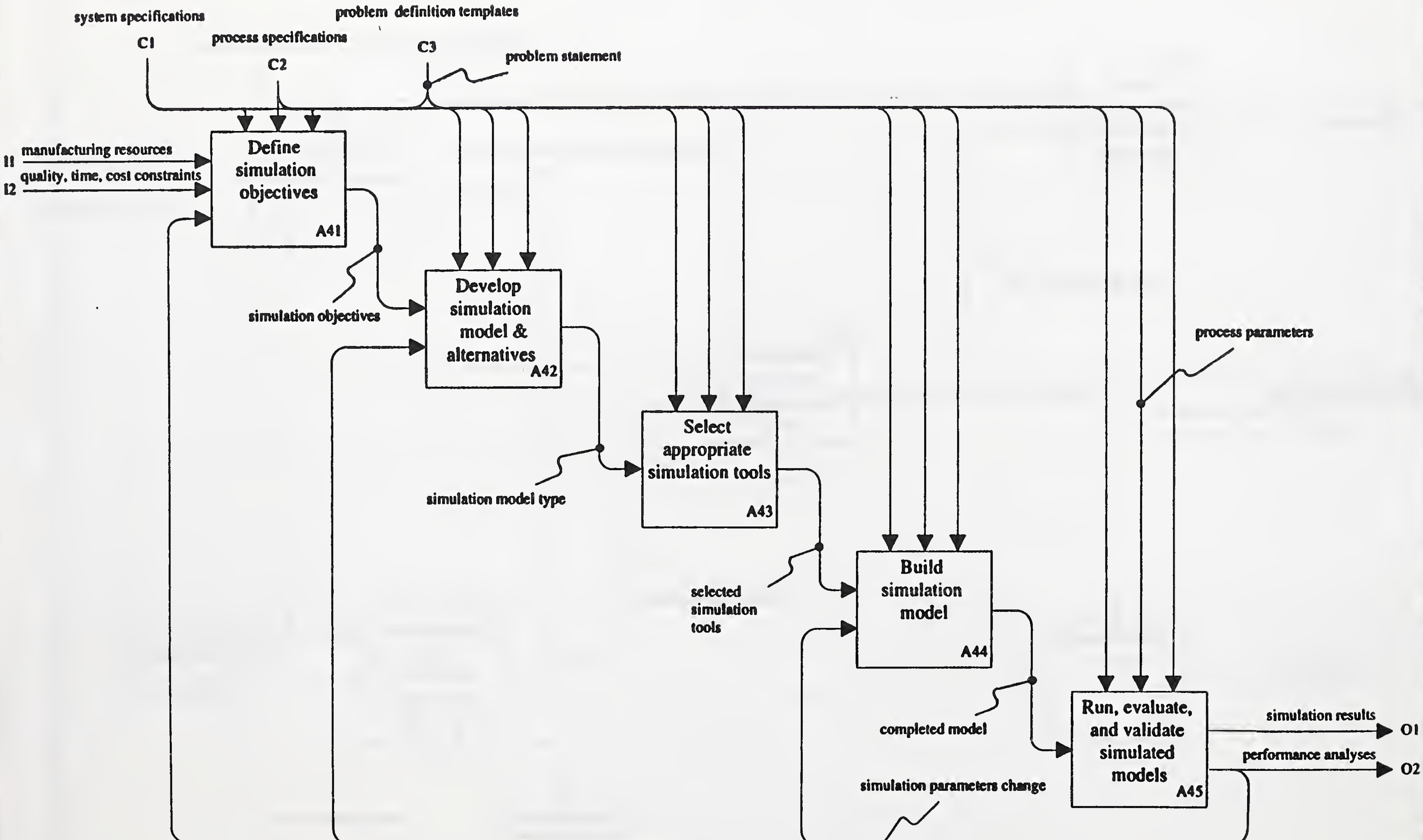
NODE: A32	TITLE: Design physical system	DATE: 08/08/95	P. 6



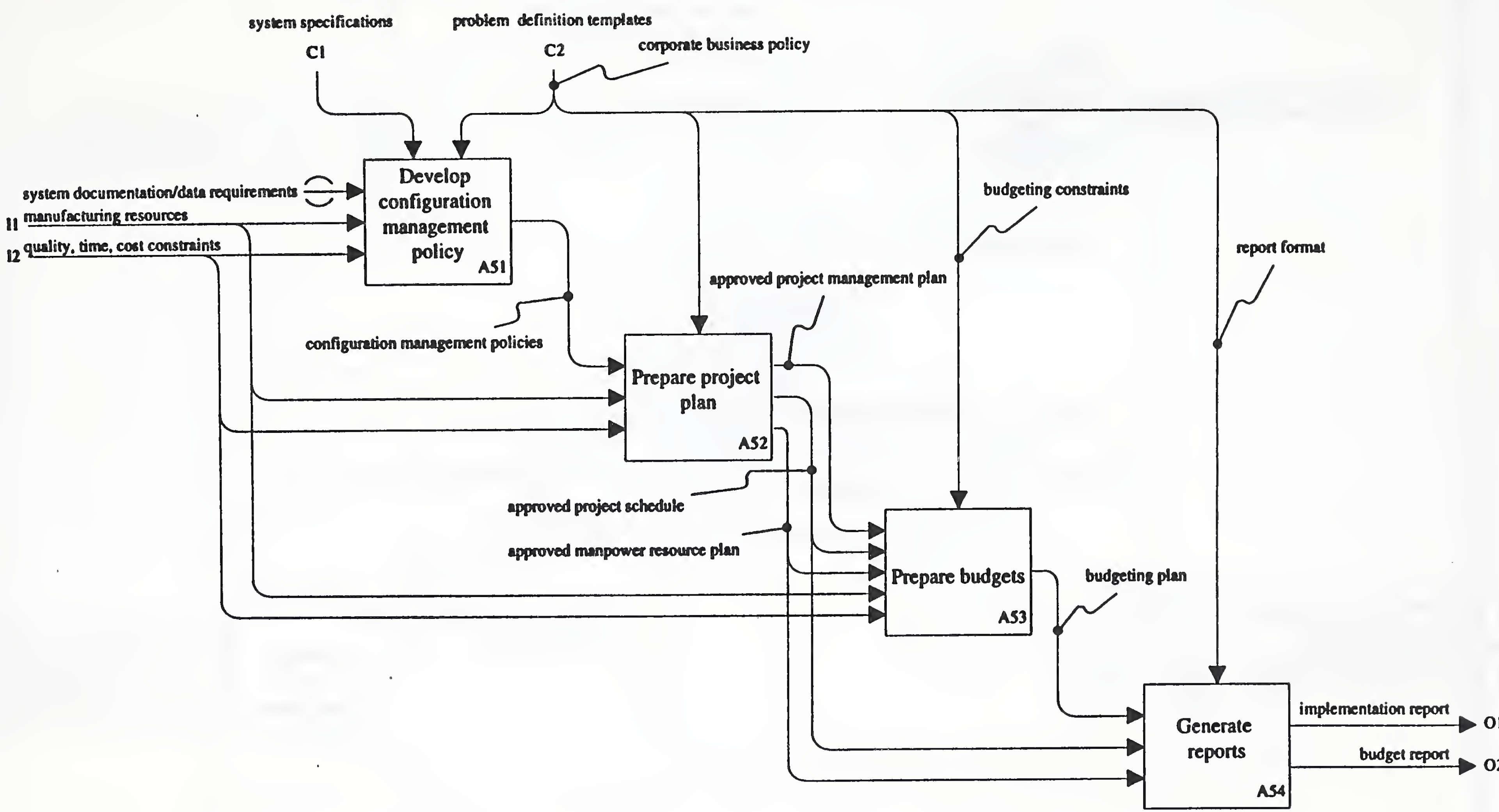
NODE:	TITLE: Design information system	DATE: 08/08/9.	P. 7



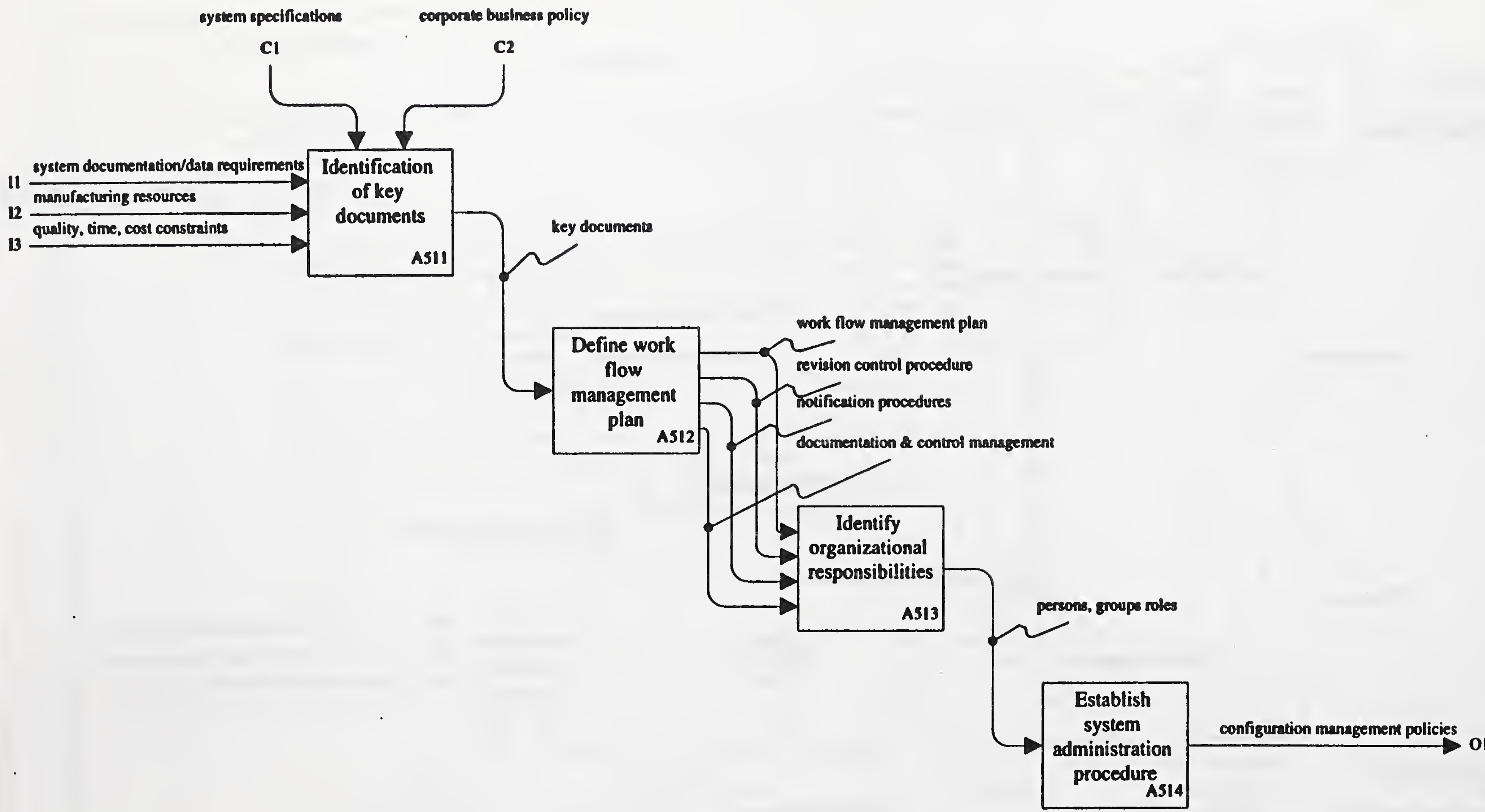
NODE: A4	TITLE: Model and evaluate system	DATE: 07/18/95	P. 8



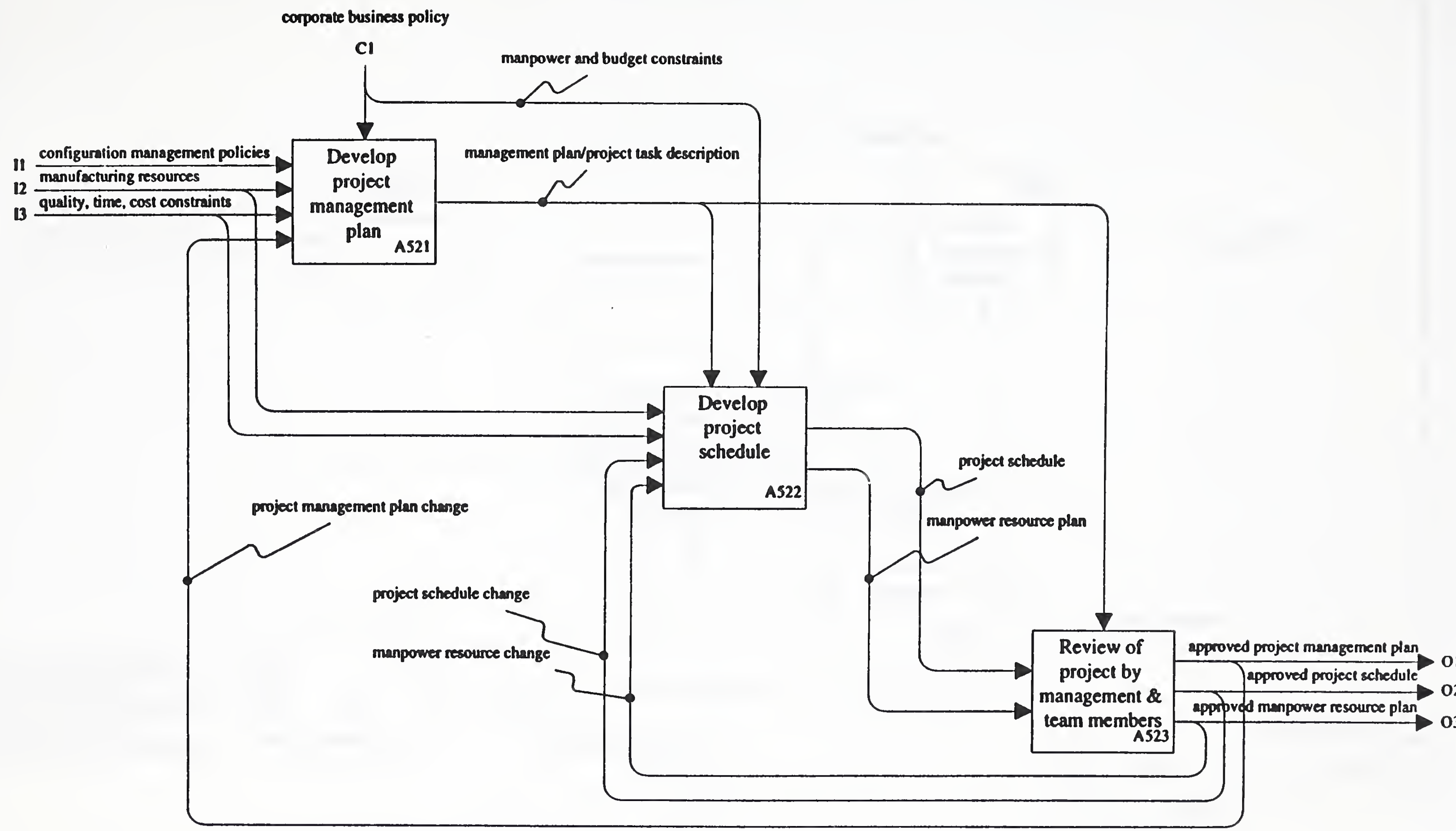
NODE:	TITLE: Prepare management informat	DATE: 07/17/'	P. 9



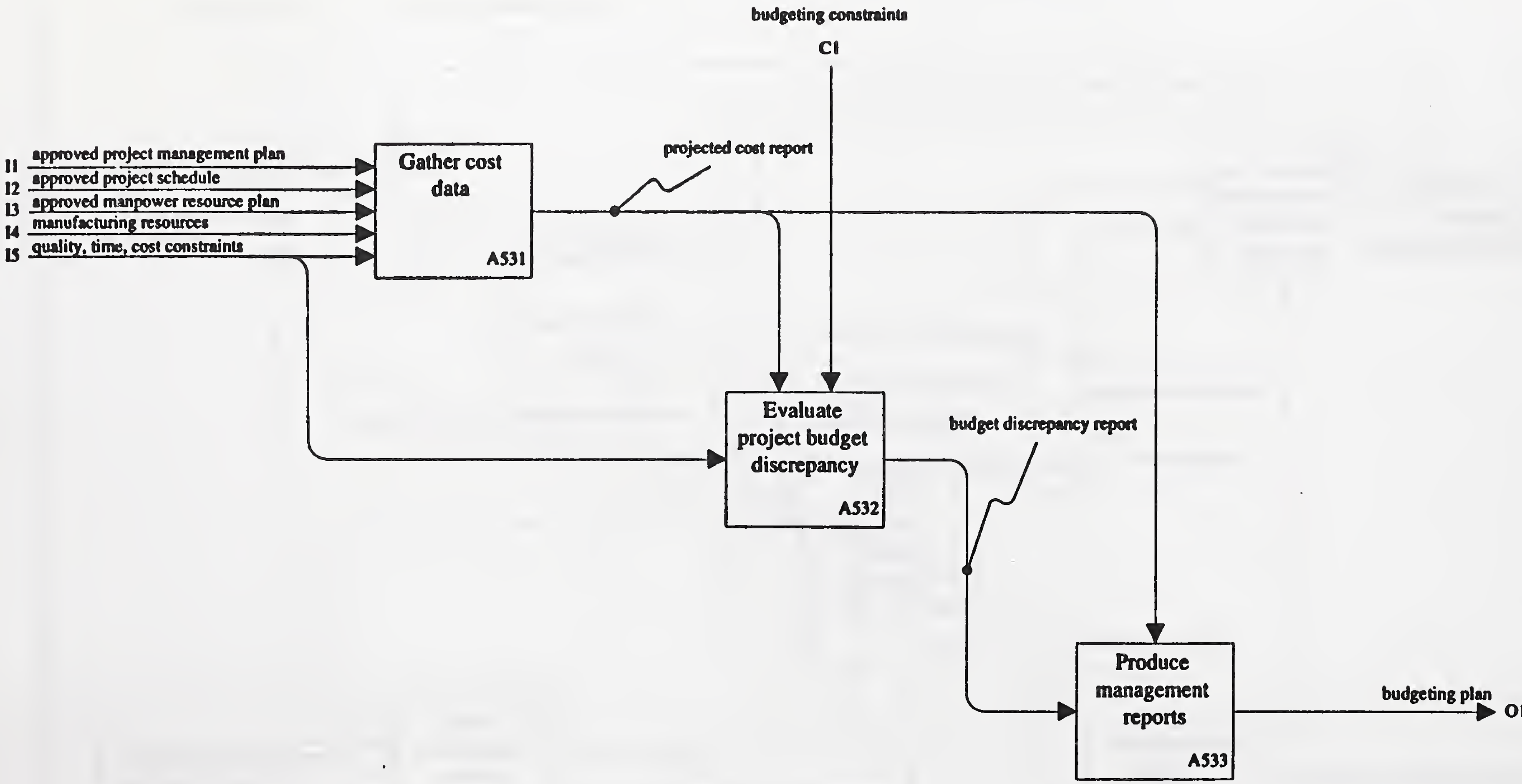
NODE: A51	TITLE: Develop configuration management policy	DATE: 07/17/95	P. 10



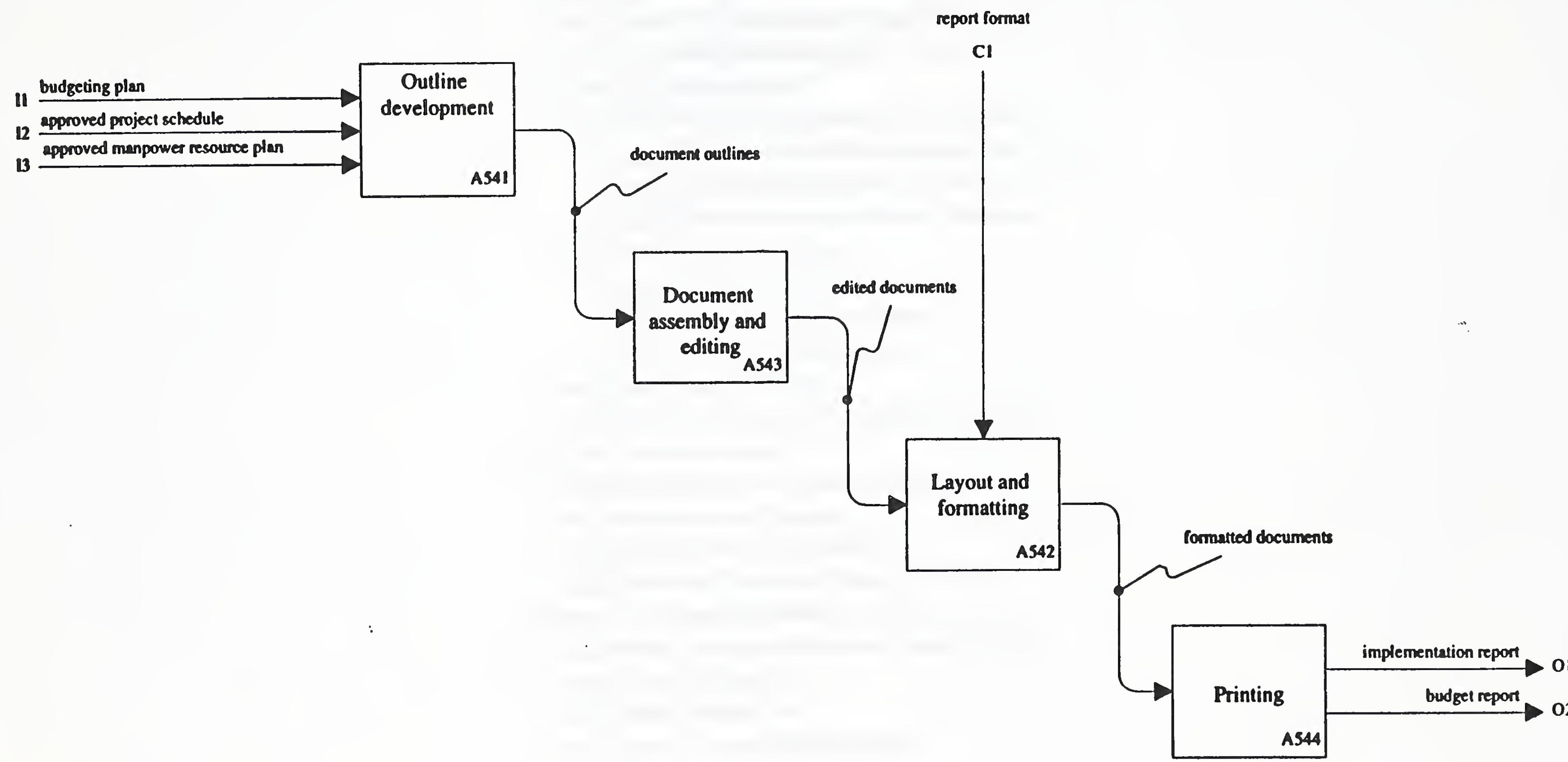
NODE:	?	TITLE:	Prepare project plan	DATE:	07/17	P. 11



NODE: A53	TITLE: Prepare budgets	DATE: 07/17/95	P. 12



NODE:	TITLE: Generate reports	DATE: 07/19/	P. 13



Node Tree for C:\IDEF35\MYMODELS\PSE_06.IDD

- [A0] Engineer production system
  - [A1] Define production engineering problem
    - [A11] Identify product data and key attributes
    - [A12] Identify project type
    - [A13] Identify manufacturing constraints
    - [A14] Identify critical dates
    - [A15] Identify target costs
    - [A16] Identify manufacturing data for related products
  - [A2] Specify production and support processes
    - [A21] Develop mfg./assembly precedence structure
    - [A22] Develop process flow diagrams
    - [A23] Perform value engineering
    - [A24] Develop process specification model diagrams
  - [A3] Design production system
    - [A31] Define physical &information system requirements
    - [A32] Design physical system
      - [A321] Determine machines/tools
      - [A322] Determine controllers/ systems
      - [A323] Develop plant layout
    - [A33] Design information system
      - [A331] Determine information system organization
      - [A332] Select appropriate systems/ equipment
      - [A333] Determine physical layout of information system
    - [A34] Integrate physical & information system designs
  - [A4] Model and evaluate system
    - [A41] Define simulation objectives
    - [A42] Develop simulation model & alternatives
    - [A43] Select appropriate simulation tools
    - [A44] Build simulation model
    - [A45] Run, evaluate, and validate simulated models
  - [A5] Prepare management information
    - [A51] Develop configuration management policy
      - [A511] Identification of key documents
      - [A512] Def k flow management plan
      - [A513] Identify organizational responsibilities

[A514] ish system administration procedure

[A52] Prepare project plan

- [A521] Develop project management plan
- [A522] Develop project schedule
- [A523] Review of project by management & team members

[A53] Prepare budgets

- [A531] Gather cost data
- [A532] Evaluate project budget discrepancy
- [A533] Produce management reports

[A54] Generate reports

- [A541] Outline development
- [A542] Layout and formatting
- [A543] Document assembly and editing
- [A544] Printing

A Process Model For Production System Engineering

C. McLean and S. Leong
Manufacturing Systems Engineering Group
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD, USA 20899
Phone: (301) 975-3511; FAX (301) 258-9749
E-mail: mclean@cme.nist.gov

Abstract

The Systems Integration for Manufacturing Applications (SIMA) Production Project at the U.S. National Institute of Standards and Technology (NIST) is working on the integration of a software tools environment for engineering production systems. This paper describes a process model for the engineering production systems and how that model is being used to integrate the commercial software tools into a workstation environment. The tools used to implement the environment are commercial off-the-shelf software products offered by a number of different vendors. The project is being undertaken as a collaborative effort between NIST researchers, several universities, and U.S. manufacturers.

Keywords

Manufacturing system engineering, process modeling, engineering tool integration

1 INTRODUCTION

Just as computer-aided design and engineering tools have revolutionized product design during the past decade, computer-based tools for production system engineering could revolutionize manufacturing. The major problem today is the lack of software integration--engineers need to move data between tools in a common computing environment. A current NIST study of engineering tools has identified more than 400 engineering software products marketed today, almost all of which are incompatible with one another. Unfortunately, the interface and database standards do not currently exist that would enable the construction of integrated tool kits.

Tool kit environments are needed which integrate clusters of functions that manufacturing engineers need to perform related sets of tasks. The Production System Engineering environment under development by NIST and collaborators will provide functions to specify, design, engineer, simulate, analyze, and evaluate a production system. Other functions included within the environment are project management and budgeting. Examples of production systems which may eventually be engineered using this environment include: transfer lines, group technology cells, automated or manually-operated workstations, customized multi-purpose equipment, and entire plants. The initial focus for this project is on small production lines used to assemble power tools.

The NIST focus for this project under the Systems Integration for Manufacturing Applications Program, Barkmeyer (1995) and the NIST/Navy Computer-Aided Manufacturing Engineering Program, McLean (1993) is on providing the models, integrated framework, operating environment, common databases, and interface standards for a wide variety of emerging tools and techniques for designing manufacturing processes, equipment, and enterprises. This paper outlines a process model which has developed for production system engineering and the tools which are being used to implement the model in an integrated computing environment. Section 2 presents an overview of the model. The Sections 3 through 7 describes the second level of decomposition of the model into: problem definition, process specification, system design, modeling and evaluation, and engineering project management. Section 8 briefly describes the effort to develop and integrated environment and outlines future work.

2 OVERVIEW OF THE PROCESS MODEL

A process model is one of several models that are needed to implement an integrated engineering tools environment. The process model defines the functions that tools must perform in order to engineer a production system. The model also defines inputs, outputs, controls, and mechanisms for carrying out the functions. The process model is a key reference document for defining the data flows and interfaces between the modules in the integrated environment.

The process model for production system engineering has been developed using Integrated Definition Method (IDEF0) modeling techniques and the Meta Software Design/IDEF tool, see Meta (1994). The model defines the tool kit functions and data inputs/outputs for each function. Detailed information models are under development which further specify each data input and output identified in the process model. The information models are being used to implement shared databases, exchange files, messages, and program calls for passing information between the commercial software tools.

The zero level of the model identifies the production system engineering function, its inputs, and its outputs. The first level of the model decomposes the engineering process into five major functions or activities: 1) define the production system engineering problem, 2) specify production processes required to produce the product, 3) design the production system, 4) model the system using simulation and evaluate its performance under expected operating conditions, and 5) prepare plans and budgets. Inputs to the production system engineering function include:

- production requirements,
- product specifications,
- quality, time, and cost constraints, and
- manufacturing resources.

Outputs of the function include:

- process specifications,
- simulation models,
- performance analyses,
- system specifications,
- implementation plans, and
- budgets.

Figures 1 and 2 illustrate the first two levels of the IDEF0 model. The model further decomposes each of these functions and data flows into sub-levels. Brief summaries of the sub-levels are presented in the sections that follow.

3 ENGINEERING PROBLEM DEFINITION

The first step in engineering the production system is clearly identifying the problem which is to be solved. Problem definition data will influence how all of the other production engineering functions are carried out. This activity is primarily one of gathering and organizing data from a number of different sources. Ultimately data gathered as a part of this activity would be recorded in template forms, imported from other applications, and maintained in a shared database. Critical data which must be identified to initiate the engineering process includes:

- Product data and key product attributes - product name, part number, model number, description, functionality, product structure (bill of materials), material composition, dimensions, weight, reference drawings, part geometry models, part family or group technology classification codes, technical specifications, reference documents,

- Production system and engineering project type - new production system (e.g., plant, line, cell), modification to existing system (i.e., product or process changes), relocation of system to new site, phaseout of a production system,

- Manufacturing constraints and issues - market forecast and production rates required (minimum, normal and peak production rates in units/hour, units/shift, units/day, units/year), production capacity, level of automation versus manual operation expected, information and control system requirements, target production site(s), floor space limitations, quality and yield requirements, safety stock requirements, storage availability, known environmental or safety hazards, production plant calendar

- Critical milestone dates and schedules - production ramp up plan, target dates for: system requirements specified, system design completed, requests-for-proposals issued, systems installed, testing completed, training completed, system operational, post production support, system phaseout,

- Expected or estimated costs - product price, manufacturing cost, system implementation, operating costs,

- Manufacturing data for related products - production engineering data for this or previously manufactured products (in some cases all outputs from previous engineering projects), competitor products and sites, possible benchmarking sites.

With the exception of critical milestone dates, most of the information outlined above may at some point be used by the next function, i.e., the specification of production and support processes. All data may be used directly by other downstream functions, if appropriate. During the course of the production system engineering process, downstream functions may provide feedback suggesting changes to the problem definition data.

4 PRODUCTION AND SUPPORT PROCESS SPECIFICATION

The second phase of the production system engineering activity is to develop a process specification for the production and support operations required to manufacture the product, see Tanner (1985), Salvendy (1992), and Sule (1994). Data developed during this phase will ultimately take the form of directed graphs and/or flowcharts. Nodes in the graphs will contain attributes which identify processes and their parameters.

A manufacturing/assembly precedence structure diagram is developed from the product geometry data and bill of materials. From the precedence structure, processes and processing precedence constraints may be derived. The derivation process may be based on human experience and intelligence, or implemented as a rule-based expert system. Data developed by this function includes:

- Process identification - process name, process type (operation, storage, inspection, delay, transportation, information, or combined activity), process parameters,

- Process resources - input product components, output product (subassembly or part identifier), tooling and fixtures, staff and job skill requirements, process by-products and hazards,

- Process time and costs - process duration, estimated process cost, product value-added.

This process is recursive--high level processes are decomposed into subprocesses until all basic or primitive operations are specified. Constraints on groups of processes and operations are identified and precedence relationships are specified.

Process specifications are perhaps best represented as diagrams and/or tables. Graphical editing functions and human interaction are normally required to layout diagrams in an understandable form. Large diagrams may be unwieldy and should be decomposed into multiple levels of sub-diagrams.

Other process specification data which may be developed as part of this phase include:

- activity relationship matrices are defined which describe how different processes relate to each other, e.g., required proximity or location.
- specification of requirements for processes, tooling, job skills, timing and line balancing, quality control, process audits,
- development of process and inspection plans, process description sheets,
- development of time standards for operations,
- ergonomic analyses of manual tasks,
- value engineering analysis (i.e., determination of job activities/steps which can be eliminated).

Processing scenarios may also be defined which describe how production will be carried out before, during, and after the new production system is implemented.

Process specifications next must be reviewed and revised to correct errors, inconsistencies, etc. Feedback requesting changes to the problem definition, as the process specification is developed. As the system design is developed in the next phase, feedback may be provided indicating required changes in process specifications.

5 PRODUCTION SYSTEM DESIGN

The third phase of the engineering process is production system design. This activity includes the design of the physical processing systems, material storage and delivery systems, and information management/control systems for the production system. The production system design problem is addressed in Sule (1994). The mechanical assembly system and flexible manufacturing system problems are described, respectively in Nevins (1989) and Draper (1984). Facility layout is presented Apple (1977) and Francis (1992). Manufacturing system architecture, design, and specification development processes are defined in Rechtin (1991), Bertain (1987), Rembold (1993), Compton (1988), and Purdy (1991).

A generic decomposition of production system design is: 1) define system requirements for each process, 2) assign requirements to system modules, 3) develop system operating scenarios for the modules, 3) identify candidate systems/machines/tooling for each module, 4) evaluate alternative technologies and candidate offerings, 5) determine number of systems required based on processing cycle time and required throughput, 6) conduct system build or buy analyses, 7) select systems for acquisition, and 8) developed detailed design for overall system based upon build and buy decisions.

Other related activities outside of the scope of system design are: 1) procurement of systems, i.e., preparation of request-for-proposals, evaluation of proposals, and awarding of contracts, 2) system development, i.e., building of modules, unit testing, and integration testing of built and bought modules, 3) system operation, and 4) post production support.

The generic production system design process can also be viewed in terms of the specific types of systems involved, i.e., process, logistics support, and information. The remainder of this section briefly summarizes considerations associated with the design of each of these elements of the overall production system.

The design of the processing system involves: the selection of a hierarchy of processing systems to implement the modules (including plants, centers, lines, cells, stations, equipment, devices, and tooling), assignment of processes to the systems, estimation of resource utilization levels, and balancing of production systems.

The design of the logistics systems can be divided into two related problems: production material logistics and plant logistics. Production material logistics includes: determination of production material requirements (raw materials, components, packaging, carriers), estimation of consumption rates, determination of sourcing strategies (make-or-buy analyses and supplier selection), lead times, and shipping (air/land/sea) methods for source materials.

Plant logistics concerns the systems which move and store materials within the facility. Plant logistics involves: determination of floor space and volumetric requirements for each process/machine/system, identification of production and tooling material storage requirements (i.e., loading docks, staging areas, centralized storage areas, line side storage), selection of storage systems (i.e., automated storage and retrieval systems, manual storage systems, production line buffers and feeders), specification of material flow through the facility (i.e., raw materials, components, work-in-process, and finished goods from the dock to lines through lines and back to dock), selection of material handling systems (e.g., hand truck, fork lift, conveyor, AGV), determination of stock replenishment strategies, design of safety and environmental systems, development of physical plant layout in two and three dimensions, and evaluation of logistics system for further production capacity growth capabilities.

Production information systems may include: monitor and control systems, communications, display and user interface systems, database management systems and their databases, data collection systems, production information systems, peripheral devices (e.g., printers, magnetic scanners, monitors, bar code readers, infrared tracking systems), production accounting and reporting, SPC/SQC systems, time and attendance recording, and preventive/corrective maintenance support systems. The information systems design activity includes: requirements specification, architecture development, process and information modeling, detailed design, interface specification, integration and test planning, and user documentation development.

The output of the production system design phase are detailed system specification documents. The phase may provide feedback to problem definition and process specification phases indicating changes which must occur as a result of design analyses. The next phase is the simulation modeling of the system which has been specified by production system design.

6 SYSTEM MODELING AND EVALUATION

Once a design, or partial design, for the production system is specified, it should be modeled and evaluated using simulation technology. The purpose of this phase is to better understand the dynamics of the proposed system and help ensure that it satisfies the constraints outlined in the problem definition phase. Inputs to this phase are derived from all of the previous phases. Pegden (1990), Askin (1993), and Carrie (1988) describe the simulation modeling process. Knepell (1993) describes the evaluation and validation of models.

The first step in developing a simulation model for the system is to define a problem statement and simulation objectives, i.e., what is expected to be learned from the simulation model. The types of alternative models to be considered and constructed need to be identified, e.g., discrete event simulation, material flow, system mechanics and kinematics, ergonomic, and/or manufacturing process. Appropriate simulation tools must be selected based on the types of models to be constructed. Next, system performance measures must be identified. Some examples of performance measures include: throughput, cycle time, work-in-process, machine downtime, and machine utilization.

Next, the system simulation model elements and their behaviors must be specified. Model elements used will depend on the types of simulations to be constructed. Elements of these models may include the attributes associated with: manufacturing resources, servers, queues and selection criteria, workpieces/loads/objects, arrival distributions, processes, system movements and material flows, timing distributions, failure and repair rates, etc. The information needed to derive the model elements will be drawn from problem definition, process specification, and system design data. The actual simulation models may then be constructed using the selected simulation tools.

Another critical activity in the modeling and evaluation phase is the development of test data for the simulation runs. This activity includes: identification of data sources, gathering of test data, formatting and loading the data, and determining the number of simulation runs required to produce significant results. Once the simulation has been constructed and the test data has been loaded, the models can be run and evaluated.

The simulations must be validated, i.e., it is necessary to determine whether results are believable based on experience, other data, etc. There are two aspects to this problem: 1) does the simulation program behave as expected, and 2) does the outcome reflect reality. If the results are not correct or creditable, either the simulation must be fixed, models modified, or the test data may need to be changed. Some examples of evaluations that may be performed on the results include: verification of the accuracy of model, analysis of errors and failures, bottlenecks, throughput, flowtime, expected yields and quality, interference problems, collisions, etc.

After the results of the simulation are reviewed, it may be necessary to revise design specifications and the system models, process specifications, or even basic assumptions spelled out in the problem definition. Some of the results of simulation, e.g., timing data, may be fed forward in to the engineering project management phase.

7 ENGINEERING PROJECT MANAGEMENT

Another parallel phase in production system engineering is the development of engineering project management data. Project management and budgeting is described in Kerzner (1984). These functions include: development of project plans, preparation of budgets, establishment of configuration management controls, and generation of reports. Principal inputs to this activity include: problem definition and system design specification data. Timing information may be drawn from simulation results.

Project planning involves defining the production system engineering project in terms of: phases, tasks, resources, and timing data. Possible phases may include: feasibility study, planning, needs and requirements analysis, detailed design, acquisition and installation, testing, training, pilot and full production operation, and phaseout. Critical milestones are identified as part of the phase definition activity.

Each major project phase is specified in terms of tasks and sub-tasks. Task precedence constraints and overlap options are identified. Required resources associated with each task are identified. Staff responsibilities are specified on each task. Resource balancing may be required. Timing information is also estimated for each task, including: expected or required start, end dates, estimated task durations and lead times. From this data, schedules may be generated and critical paths determined.

Cost factors and their analysis is an extremely important part of the system design and implementation process. Malstrom (1984) provides detailed guidance on manufacturing cost engineering processes that can be used to develop cost estimates and budgets. Budget cost categories that may be considered include: project phase, planning, labor, tooling, capital equipment, projected maintenance, information and control system, operational, training, licensing and inspection, construction, installation, material (components, consumables), overhead (utilities, labor multipliers, area usage), and rental costs.

The budgeting process includes: gathering of cost data, entering data into spreadsheets or databases by budget categories, projecting estimates where data is unavailable, generating summaries by categories, and producing budget reports. Budgeting data is review for significant deviations from targets and opportunities for savings are identified. Budget data is then used to generate feedback, if required to the problem definition and production system design phases.

Another critical activity included in this phase is the configuration management of engineering data and project documents. Principles of configuration management are outlined in Daniels (1985). This activity includes: identification of key documents, definition of revision control/review/promotion policies and procedures, identification of organizational responsibilities, establishment of notification procedures for project staff, establishment of security policies and access control mechanisms, and the placement of documents and data under configuration management.

The final activity in the management area is generation and publication of reports that summarize the results of each of the other phases. Functions included in this activity include: outline development, document editing and assembly, layout and formatting, and printing. This activity draws input from all of the other functions in this phase and the other phase.

Once plans, budgets, configuration management policies, and reports are completed they need to be reviewed to ensure that they are realistic and meet the requirements established in the problem definition phase. If not, either the plans need to be changed or information must be fed back to problem definition and/or system design to re-scope the system.

8 INTEGRATED ENGINEERING TOOLS ENVIRONMENT

The process model for production system engineering is being implemented as an integrated tools environment through the collaborative efforts of NIST, academia, and industry. Academic collaborators include: the University of Kansas, California Polytechnic University, Ohio University, and Brigham Young University. Black and Decker Corporation is collaborating on the production system engineering process and providing test data on production lines. Although a number of engineering tool vendors have provided software for integration into the environment, the final selections of software tools has not been completed.

The production engineering environment is being implemented on a high performance personal computer. Commercial software tools used in the implementation of the engineering environment include: a business process re-engineering/flowcharting package, a plant layout system, a computer-aided design system, a manufacturing simulation system, a spreadsheet tool, a project management system, and a relational database management system. Other tools are under consideration for incorporation into the environment at a future time.

The interoperability of the commercial engineering tools that are available today is extremely limited. As such, users must re-enter data as they move back and forth between different tools carrying out the engineering process. Project collaborators will: define generic information models for production system engineering data, specify interfaces for integrating tools, develop prototype integrated environments and shared databases, and implement test case production system engineering projects. Examples of the types of shared data under consideration by the collaborators for the common database includes: production requirements, product specifications, process specifications (diagrams, flowcharts, plans, routings, operation sheets, programs), equipment specifications, budget spreadsheets, project plans, simulation models and model elements, setup illustrations, plant layouts, information models, interface specifications, system descriptions, estimated yield data, process capabilities, and quality data.

A long term objective of the project is to improve the productivity of users by creating an integrated environment where changes to data and decisions automatically percolate through the various tools contained within system. Project results will provide a basis for defining interface standards that will facilitate the integration and interoperability of commercial tools in the future.

Work described in this paper was sponsored by the U.S. Navy Manufacturing Technology Program and the NIST Systems Integration for Manufacturing Applications (SIMA) Program. No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. The work described was funded by the United States Government and is not subject to copyright.

9 REFERENCES

Apple, J.M., (1977) *Plant Layout and Material Handling*, John Wiley and Sons, New York, NY.

Askin, R.G., Standridge, C.R. (1993) *Modeling and Analysis of Manufacturing Systems*, John Wiley and Sons, New York, NY.

Barkmeyer, E.J., Hopp, T.H., Pratt, M.J., Rinaudot, G.R., editors (1995) *Background Study: Requisite Elements, Rationale, and Technology Overview for the Systems Integration for Manufacturing Applications Program*, NIST Technical Report, Gaithersburg, MD.

Bertain, L., Hales, L. (1987) *A Program Guide for CIM Implementation*, Society of Manufacturing Engineers, Dearborn, MI.

Carrie, A. (1988) *Simulation of Manufacturing Systems*, John Wiley and Sons, Chichester, Great Britain.

Compton, W.D., editor (1988) *Design and Analysis of Integrated Manufacturing Systems*, National Academy Press, Washington, DC.

Daniels, M.A. (1985) *Principles of Configuration Management*, Advanced Applications Consultants, Rockville, MD.

Draper Laboratory Staff (1984) *Flexible Manufacturing Systems Handbook*, Noyes Publications, Park Ridge, NJ.

Francis, R.L., McGinnis, Jr., L.F., White, J.A. (1992) *Facility Layout and Location: An Analytical Approach*, Prentice-Hall, Englewood Cliffs, NJ.

Kerzner, H. (1984) *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, Van Nostrand Rheinhold, New York, NY.

Knepell, P.L. and Arangno,D.C. (1993) *Simulation Validation: A Confidence Assessment Methodology*, IEEE Computer Society Press.

Malstrom, E.M. (1984) *Manufacturing Cost Engineering Handbook*, Marcel Dekker, NY.

McLean, C.R. (1993) "Computer-Aided Manufacturing Systems Engineering" in *IFIP Transactions B-13 Advances in Production Management Systems*, North-Holland, Amsterdam, Netherlands.

Meta Software Corp. (1994) *Design/IDEF User's Manual and Tutorial For Microsoft Windows*,

Meta Software Corp., Cambridge, MA.
Nevins, J.L., Whitney, D.E., (1989) *Concurrent Design of Products and Processes: A Strategy for the Next Generation in Manufacturing*, McGraw-Hill, New York, NY.
Pegden, C.D., Shannon,R.E., Sadowski,R.P. (1990) *Introduction to Simulation Using SIMAN*, McGraw-Hill, New York.
Purdy, D.C. (1991) *A Guide to Writing Successful Engineering Specifications*, McGraw-Hill, New York, NY.
Rechtin, E., (1991) *Systems Architecting: Creating and Building Complex Systems*, Prentice-Hall, Englewood Cliffs, NJ.
Rembold,U., Nnaji, B.O., Storr, A. (1993) *Computer Integrated Manufacturing and Engineering*, Addison-Wesley, Wokingham, England.
Salvendy, G., editor (1992) *Handbook of Industrial Engineering*, John Wiley and Sons, New York, NY.
Sule, D.R., (1994) *Manufacturing Facilities: Location, Planning, and Design*, PWS Publishing Company, Boston, MA.
Tanner, J.P. (1985) *Manufacturing Engineering: An Introduction to Basic Functions*, Marcel Dekker, New York, NY.

10 BIOGRAPHIES

C. McLean: Group Leader of Manufacturing Systems Engineering, NIST Manufacturing Systems Integration Division. Managed research programs in manufacturing automation at NIST since 1982. Established research program in Computer-Aided Manufacturing Engineering with emphasis on engineering tool integration, manufacturing data validation, production systems engineering, manufacturing simulation, production scheduling.

S. Leong: Joined NIST in 1994, after sixteen years in the manufacturing industry. Responsible for the design, development and implementation of factory automation systems at Ford Motor Company, John Deere, and IBM. Provided consulting services to manufacturers for design and implementation of manufacturing execution and shop floor control systems since 1987. Research interests include manufacturing engineering tools integration, engineering data validation, and production system engineering.

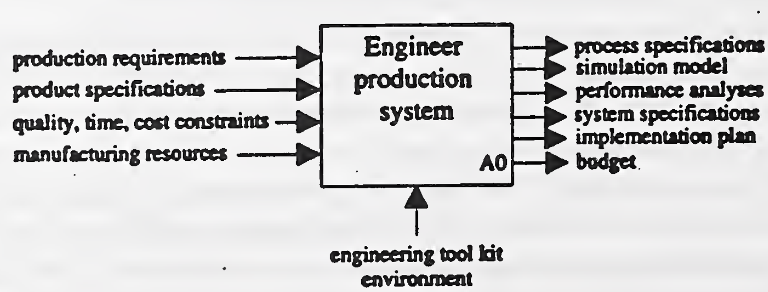


Figure 1. Top level of the production system engineering IDEF model

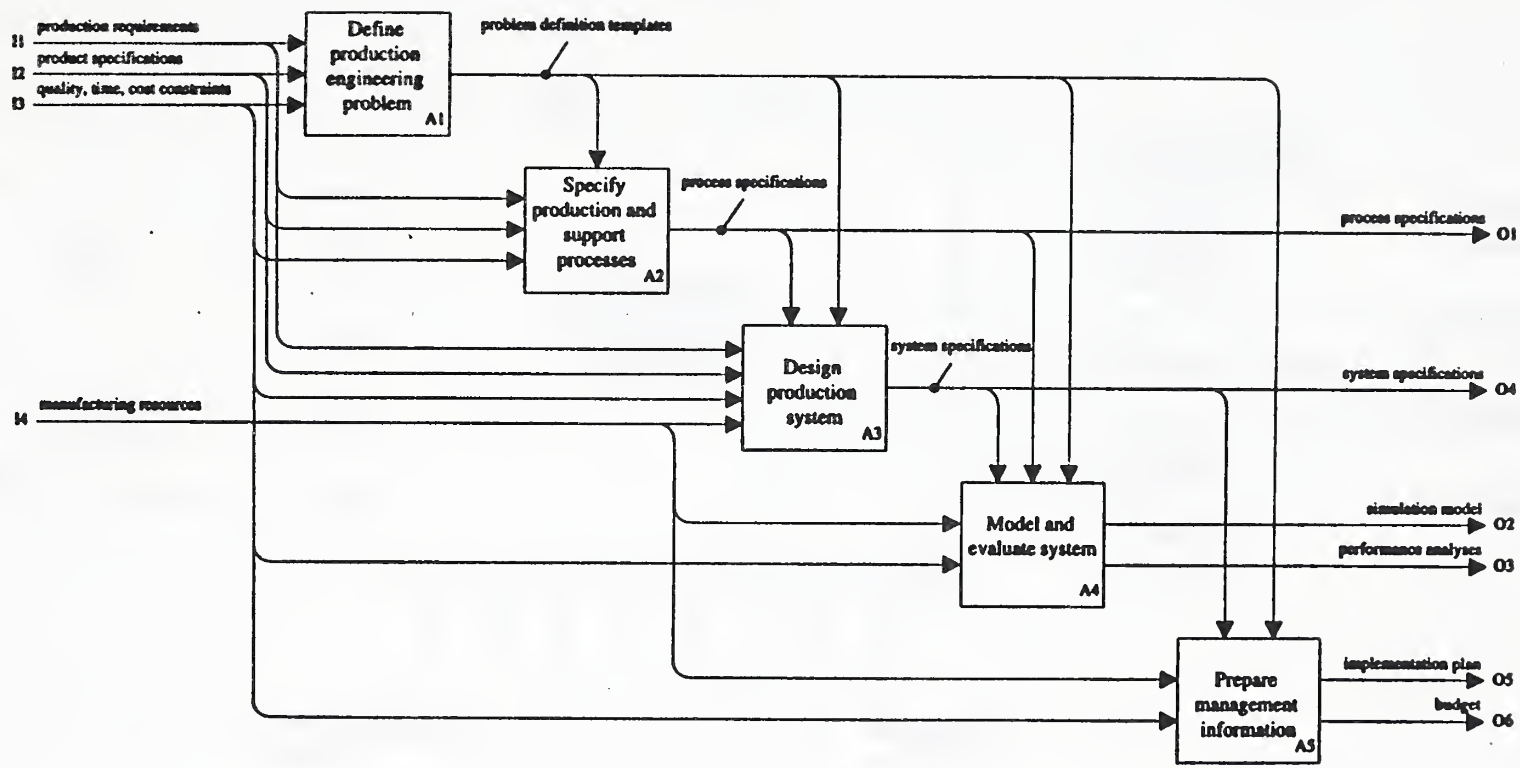
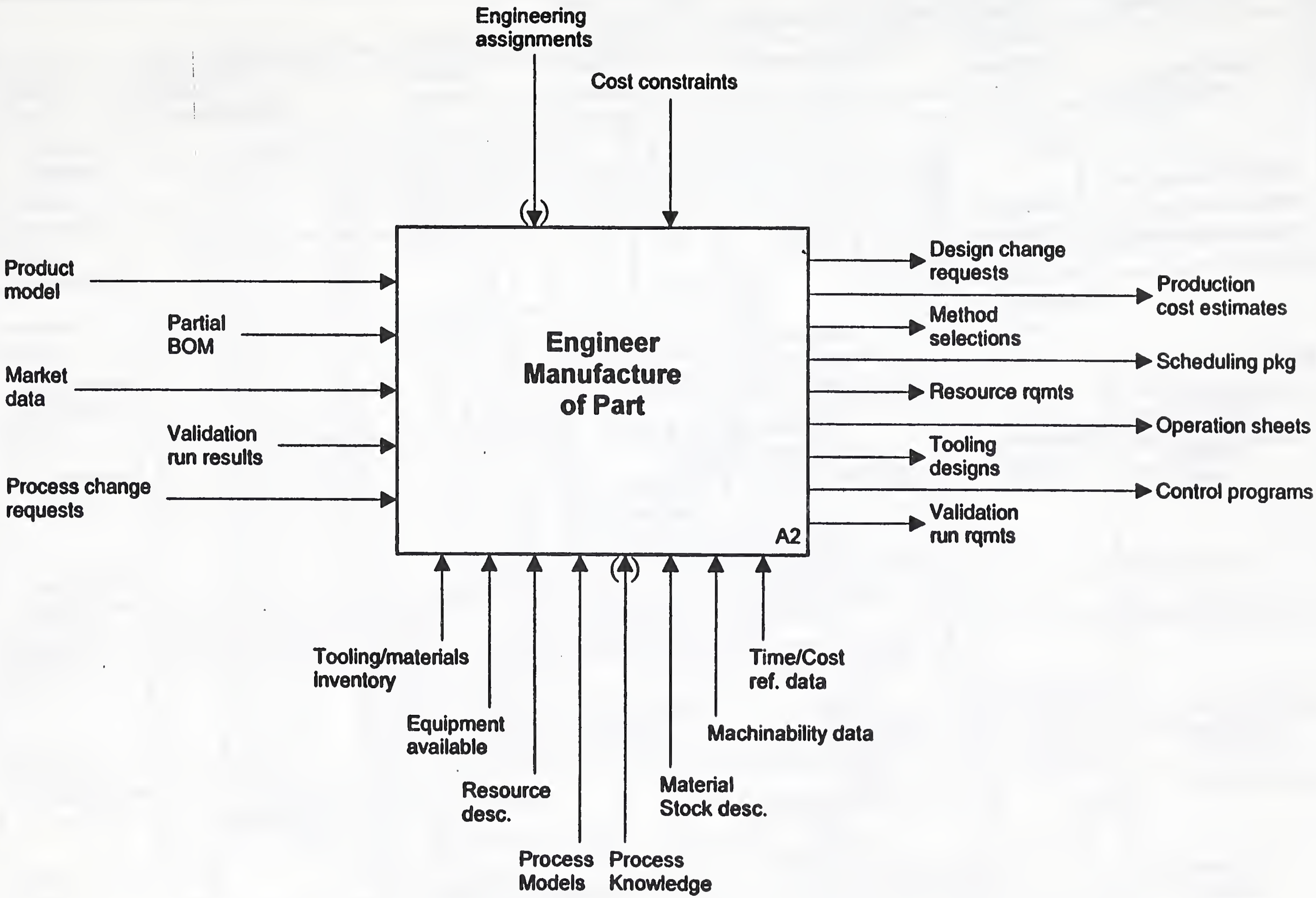
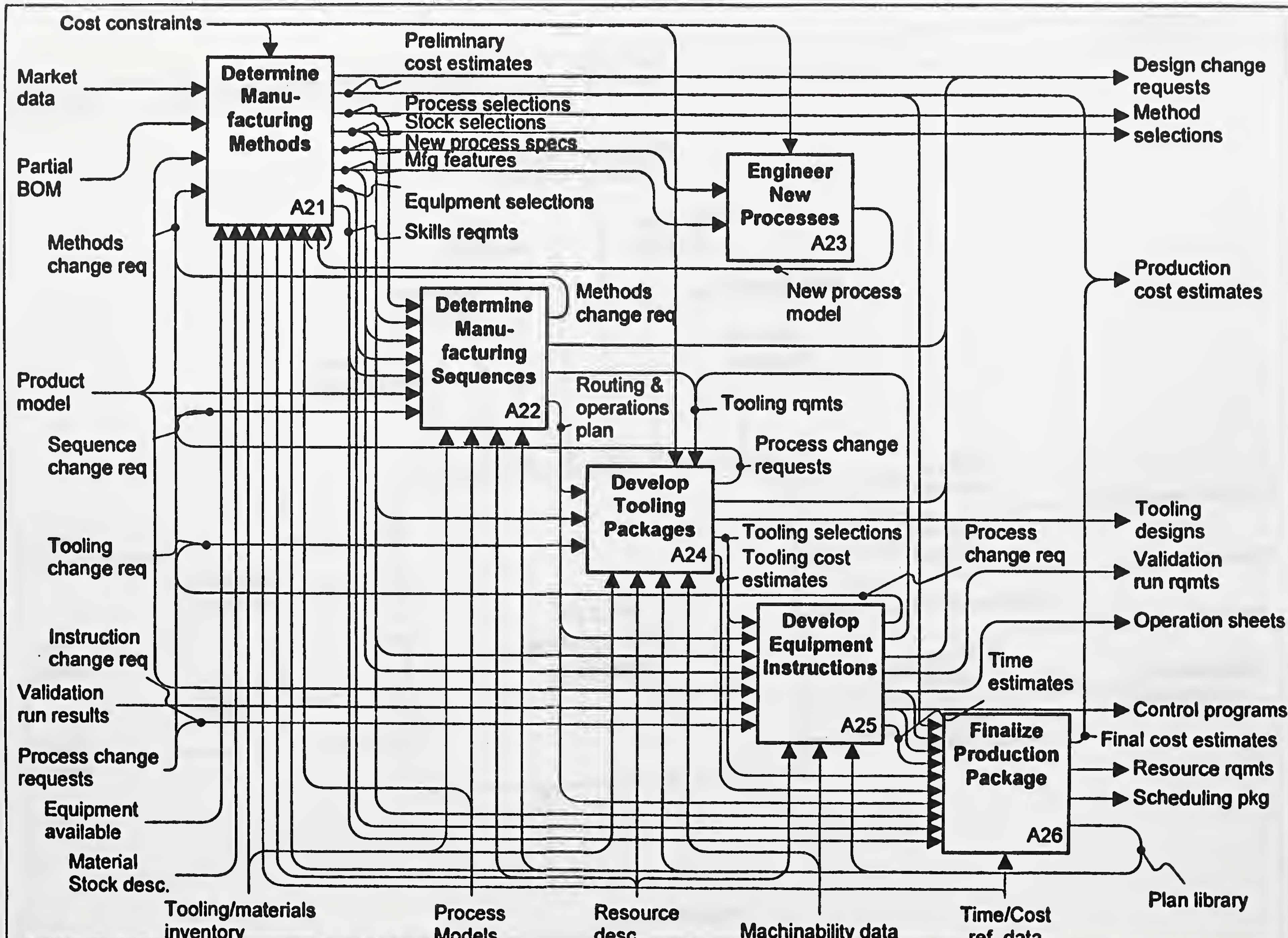


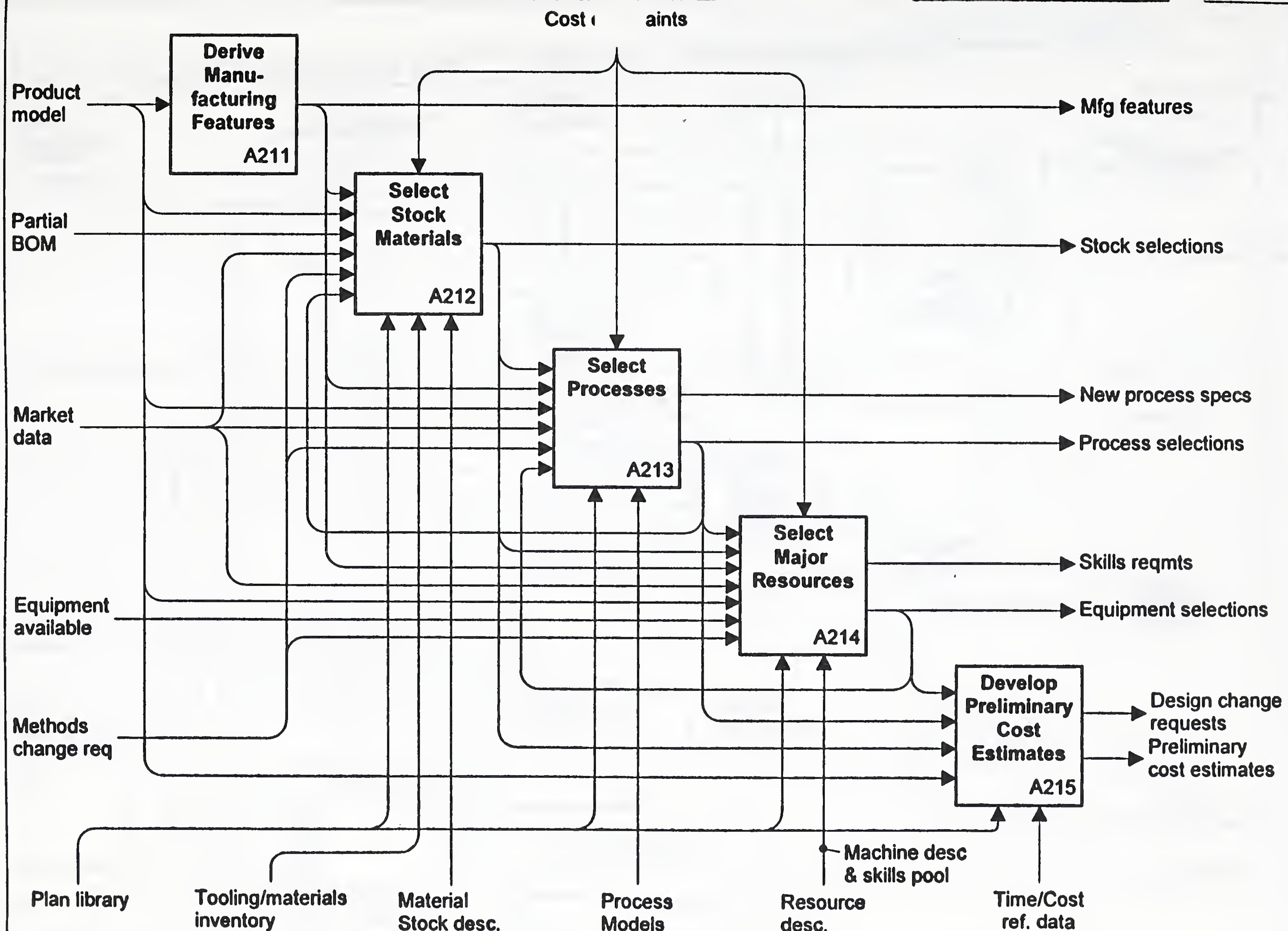
Figure 2. First level of decomposition of the production system engineering model



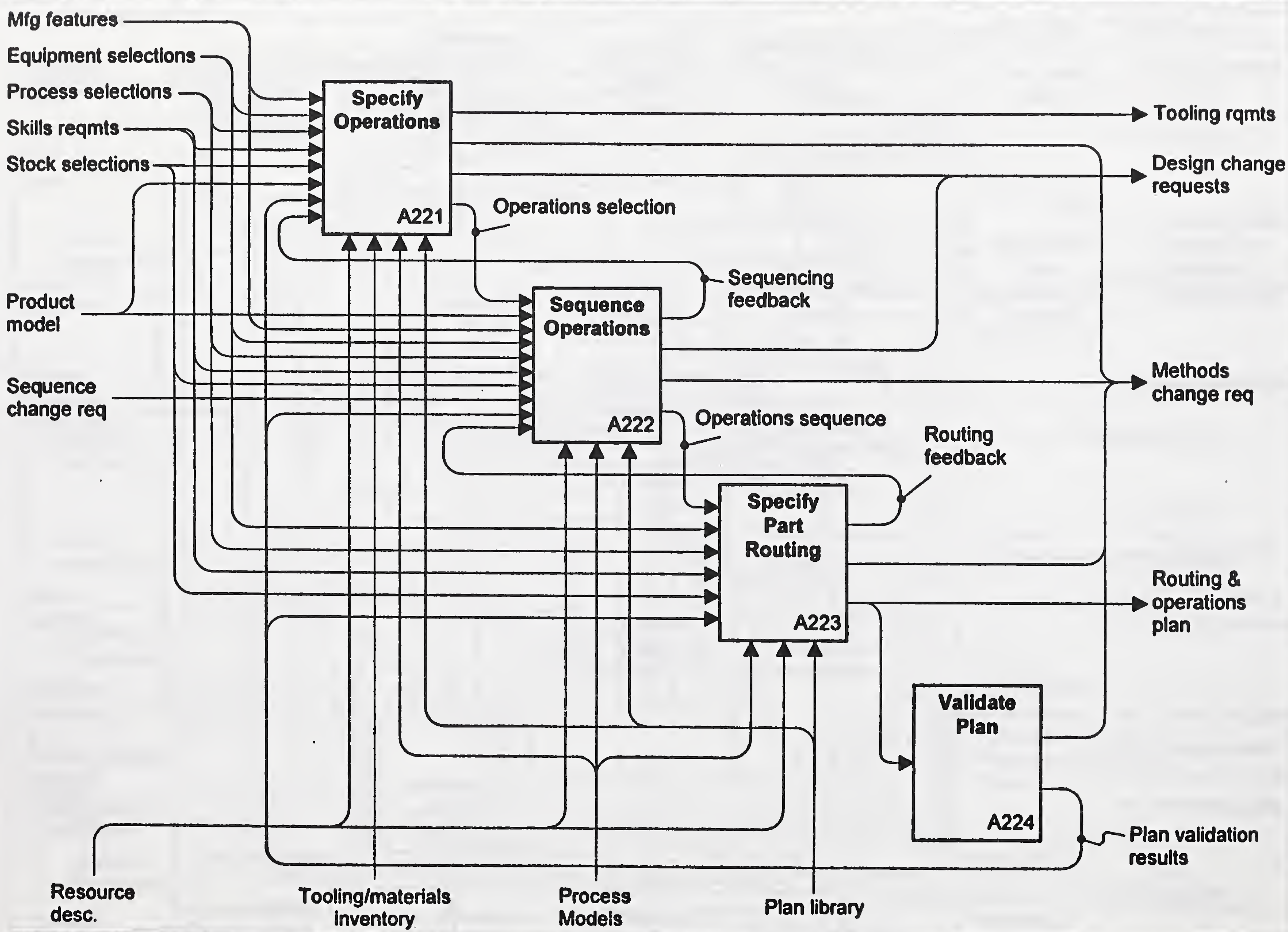
Engineering assignments
Cost constraints
Product model
Partial BOM
Market data
Validation run results
Process change requests
Engineer Manufacture of Part
A2
Design change requests
Production cost estimates
Method selections
Scheduling pkg
Resource rqmts
Operation sheets
Tooling designs
Control programs
Validation run rqmts
Tooling/materials inventory
Equipment available
Resource desc.
Process Models
Process Knowledge
Material Stock desc.
Machinability data
Time/Cost ref. data

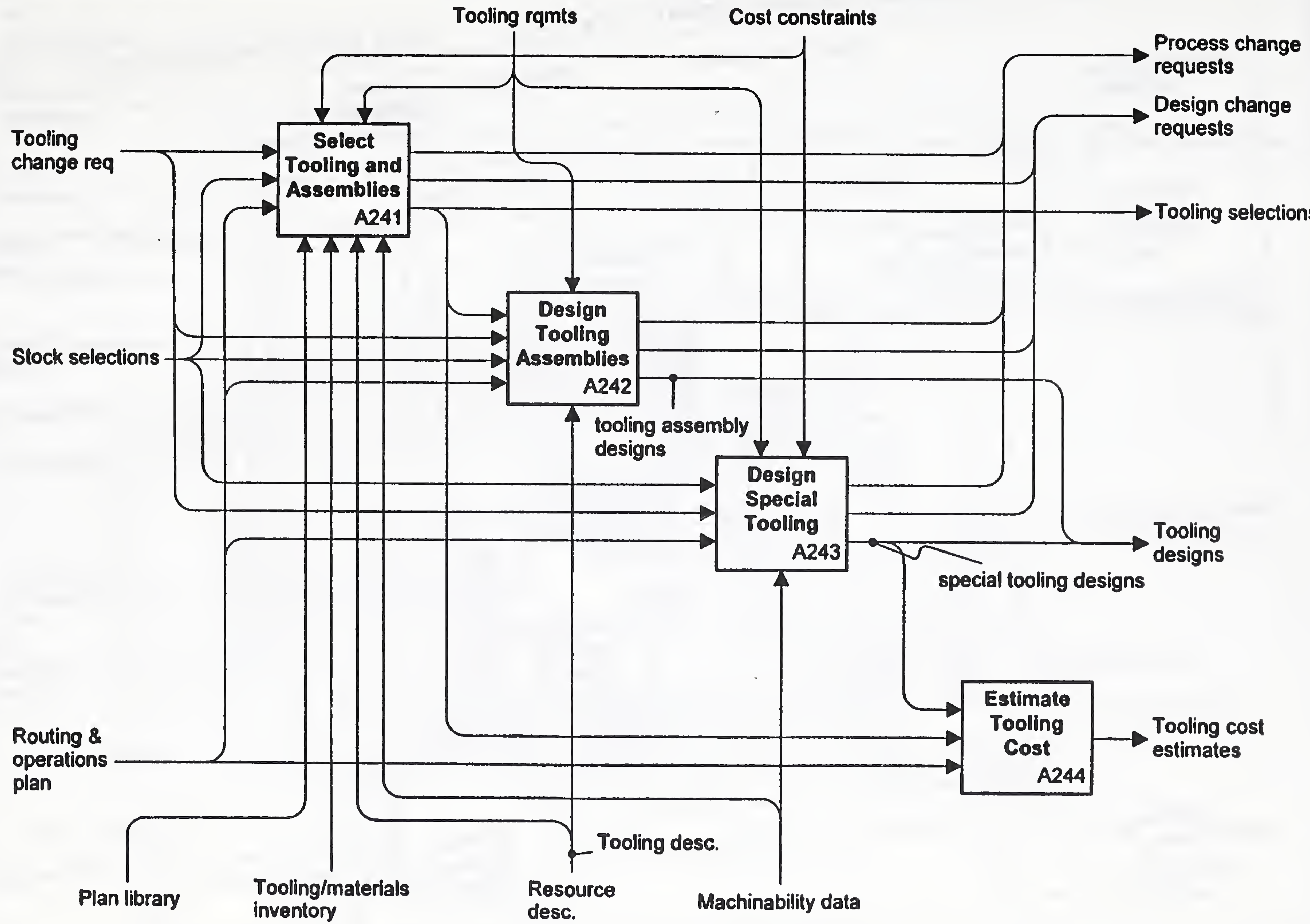


Cost constraints
Market data
Partial BOM
Methods change req
Product model
Sequence change req
Tooling change req
Instruction change req
Validation run results
Process change requests
Equipment available
Material Stock desc.
Tooling/materials inventory
Process Models
Resource desc.
Machinability data
Time/Cost ref. data
Plan library
Determine Manu-facturing Methods
A21
Preliminary cost estimates
Process selections
Stock selections
New process specs
Mfg features
Equipment selections
Skills reqmts
Engineer New Processes
A23
New process model
Determine Manu-facturing Sequences
A22
Methods change req
Routing & operations plan
Tooling rqmts
Process change requests
Develop Tooling Packages
A24
Tooling selections
Tooling cost estimates
Process change req
Develop Equipment Instructions
A25
Time estimates
Finalize Production Package
A26
Design change requests
Method selections
Production cost estimates
Tooling designs
Validation run rqmts
Operation sheets
Control programs
Final cost estimates
Resource rqmts
Scheduling pkg

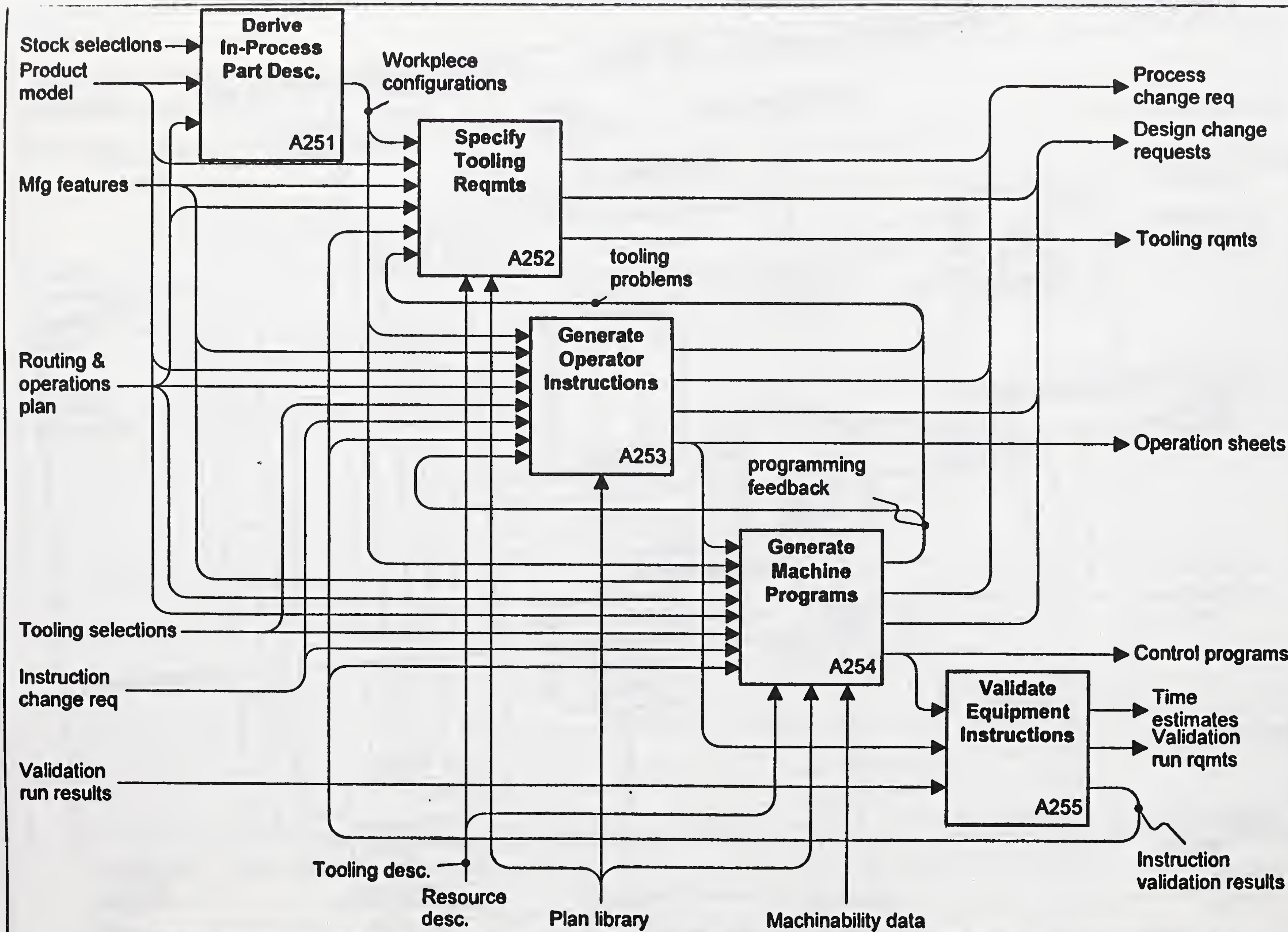


07/17/95	SIMA Manufacturing Activity Model	Node: A21	Determine Manu- facturing Methods	Page: A2-3

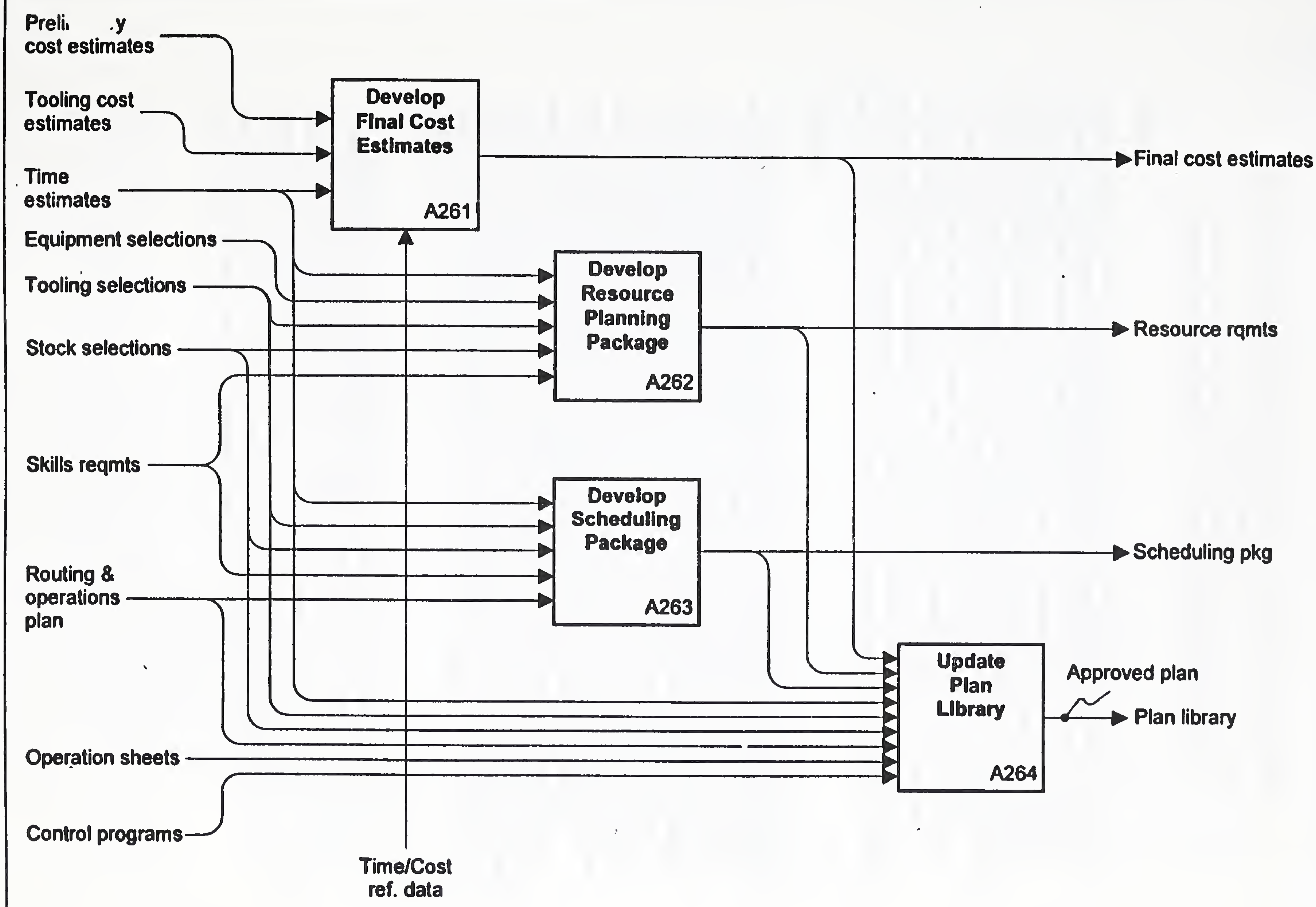

Mfg features
Equipment selections
Process selections
Skills reqmts
Stock selections
Product model
Sequence change req
Specify Operations
A221
Operations selection
Sequencing feedback
Sequence Operations
A222
Operations sequence
Routing feedback
Specify Part Routing
A223
Validate Plan
A224
Tooling rqmts
Design change requests
Methods change req
Routing & operations plan
Plan validation results
Resource desc.
Tooling/materials inventory
Process Models
Plan library



Tooling rqmts
Cost constraints
Process change requests
Design change requests
Tooling selections
Tooling change req
Select Tooling and Assemblies A241
Design Tooling Assemblies A242
Stock selections
tooling assembly designs
Design Special Tooling A243
Tooling designs
special tooling designs
Estimate Tooling Cost A244
Tooling cost estimates
Routing & operations plan
Tooling desc.
Plan library
Tooling/materials inventory
Resource desc.
Machinability data



07/17/95	SIMA Manufacturing Activity Model	Node: A25	Develop Equipment Instructions	Page: A2-F


Preli. .y
cost estimates
Tooling cost
estimates
Time
estimates
Equipment selections
Tooling selections
Stock selections
Skills reqmts
Routing &
operations
plan
Operation sheets
Control programs
Time/Cost
ref. data
Develop
Final Cost
Estimates
A261
Develop
Resource
Planning
Package
A262
Develop
Scheduling
Package
A263
Update
Plan
Library
A264
Final cost estimates
Resource rqmts
Scheduling pkg
Approved plan
Plan library

Manufacturing Engineering Toolkit Prototype Demonstration

Michael J. Iuliano
Manufacturing Systems Integration Division, Manufacturing Engineering Lab
National Institute of Standards and Technology , Gaithersburg MD, USA

Abstract

A computer-aided Manufacturing Engineering Toolkit (METK) prototype is currently under development at the United States National Institute of Standards and Technology (NIST) as a part of the Computer-Aided Manufacturing Engineering (CAME) project which is jointly sponsored by the U.S. Navy Manufacturing Technology program and NIST. The toolkit consists of commercial-off-the-shelf (COTS) CAD/CAM applications housed together on a high speed computer workstation. The METK is envisioned to be an integration of these applications to support sharing of data between the applications. Current system includes a product data management application, a CAD application, a generative process planning application, and a suite of manufacturing simulation applications used to plan/evaluate manufacturing system components from the machine tool to the shop floor level. This tool kit will be used in manufacturing data validation as a part of the overall product planning process required to manufacture a part. This paper describes a demonstration of the METK prototype. Overall objectives of this effort include specification of integration interfaces and a methodology for manufacturing validation.

Introduction

A demonstration of the toolkit applications has been prepared to illustrate the functionality of a prototype METK. The demonstration is comprised of two scenarios. The first scenario involves tasks performed to validate an engineering data package specified to manufacture a small prismatic product. This scenario of the demonstration consists of creating a solid model geometry in the Pro-Engineer CAD application. Using the CAD geometry as input into the generative process planning application ICEM PART. ICEM PART then creates a process plan and stores the information in the ORACLE database. A CNC program is also produced by the ICEM PART application. Interface software is then executed to extract the machine tool, cutting tools, raw stock and fixture information from the database. This interface software was developed by Robert Judd, Ohio University under the Intelligent Machining Workstation project. This interface is currently implemented as UNIX shell scripts which queries the Oracle database for the appropriate information, creates the directory structure needed by DENEB VNC, and constructs a simulated workcell in VNC. The workcell consists of a pre-developed kinematic VNC model of the EMCO 100 milling workstation we are using in the demonstration, a blank fixtured to the machine table, geometric models of the tooling mounted on the machine, and the appropriate CNC program. See Figure 1 for a depiction of the EMCO 100. The demonstration then executes DENEB VNC to simulate the machining process. VNC is used to help identify any errors in the CNC program, any tool crashes or part gouges.

The second scenario of the demonstration simulates the workflows between factory workstations used to create the prismatic product. A virtual factory is being modeled in DENEB Quest. Each workstation in the virtual factory will perform processes that represent manufacturing processes. The processes were selected by industrial participants at the first technical meeting of the Computer Aided Manufacturing Engineering Forum on March 21-22 1995. The inclusion of additional workstations in the virtual factory which perform other types of processes will be considered as the needs of the forum participants change over the life of the project. The current concentration of the Quest model development is the workflow required to produce the prismatic product used in the first scenario of the demonstration.

Demonstration

The product to be manufactured is a small rectangular prismatic workpiece with over thirty manufacturing features or patterns of topology and geometry consisting of holes, notches, slots and pockets. The workpiece material is plastic. In ICEM PART, features are volumes of the workpiece to be removed by a sequence of machining operations. ICEM PART recognized all of features and specifies setups, machining operations, tooling, and the tool paths necessary to manufacture the product. In this case, ICEM PART specified two setups using shank end mills, machine taps, a twist drill and a center drill for machining.

The simulation workcell model was run in DENEB VNC and a collision occurred between the tool holder and the workpiece when the 1/4 inch twist drill was machining two of the holes on the workpiece. The holes are recessed in a pocket that is not wide enough to accommodate the width of the toolholder holding the 1/4 inch drill.

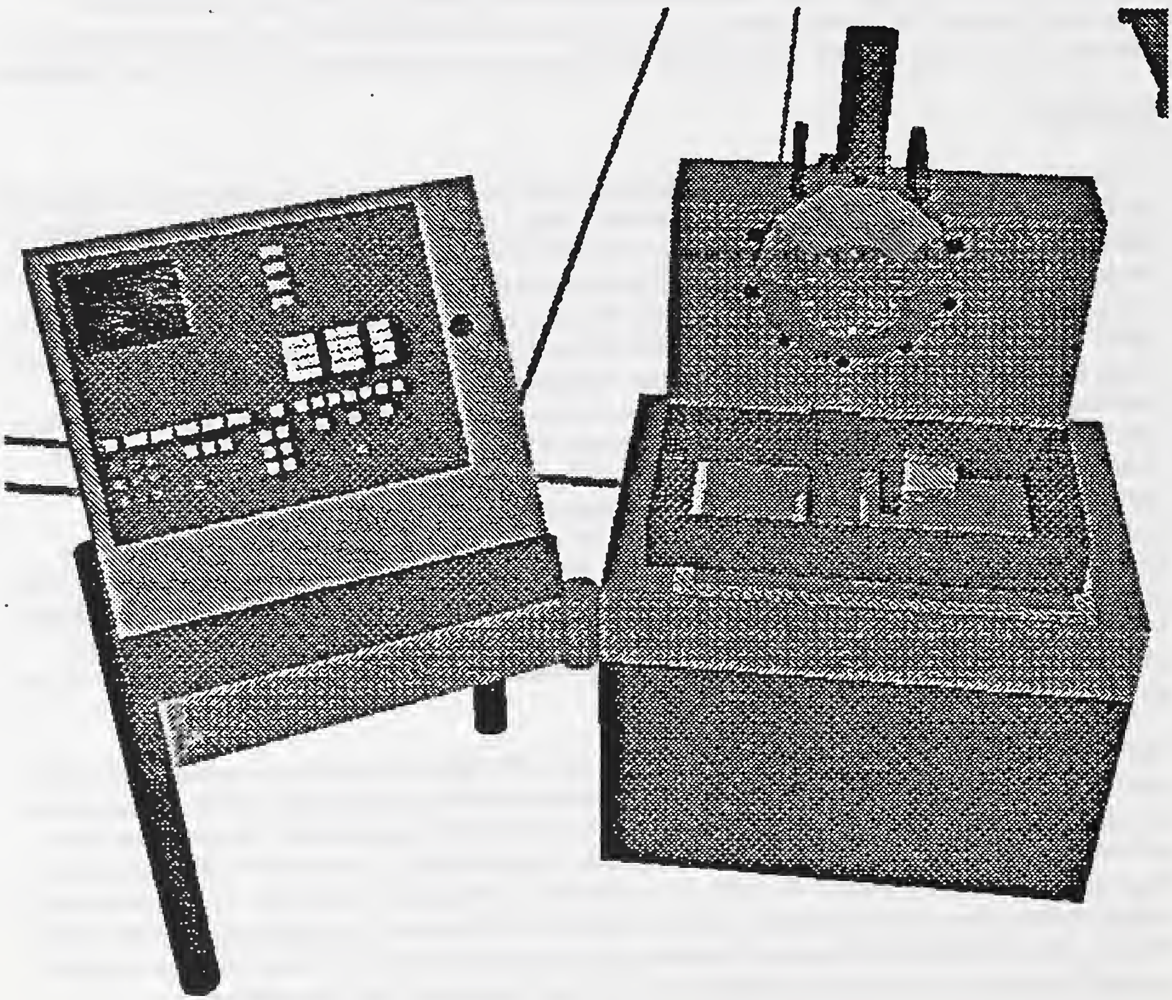


Figure 1 - EMCO 100 Vertical Milling Machine

ICEM PART is executed again, this time the user manually overrides the specification and specifies three setups to machine the workpiece. The third setup has the workpiece flipped over so the holes that caused the collision in the two setup process plan could be drilled from the underneath of the piece. The three setup data is generated and translated to the DENEB VNC workcell model. The manufacturing simulation is executed with three setups and the product is manufactured correctly in the simulation.

This emphasizes a key point the METK project is trying to get across: In the context of data validation, the integrated toolset of software applications cross-check each other for consistency and accuracy to ensure in the end, a better, more reliable engineering data package hits the machine shop floor the first time. The result should be the real life workpiece can be successfully machined the first time thereby reducing the time and money expenditure for producing the machined part.

The virtual factory being modeled in DENEB Quest currently consists of the following manufacturing areas: tool room, shipping, receiving,, heat treat, paint, manufacturing/engineering/administrative offices, and three machining areas. The tool room contains a tool assembly station, a fixturing station, tool crib, and a shop floor supervisor's office. The shipping/receiving areas have raw storage, tables, a scale, a bandsaw and forklift. The heat treat area contains two ovens. The paint area contains paint tanks, a paint robot under a paint hood, and pallets. Area 1 contains three EMCO 100 vertical milling machines that are used to machine the prismatic part we are manufacturing in the demonstration; a parts washer, two Mandelli horizontal vertical mills, three T30 Cincinnati Millacron milling machines, a coordinate measuring machine (CMM), tables and pallets. Area 2 contains two lathes, three grinders, and a finisher. Area 3 contains a laser cutter/punch, a laser punch, a press bender, bandsaw, drill press, jigbore, and a belt sander. See Figure 2 for a depiction of the three EMCO 100 milling machines as they sit on the virtual factory floor.

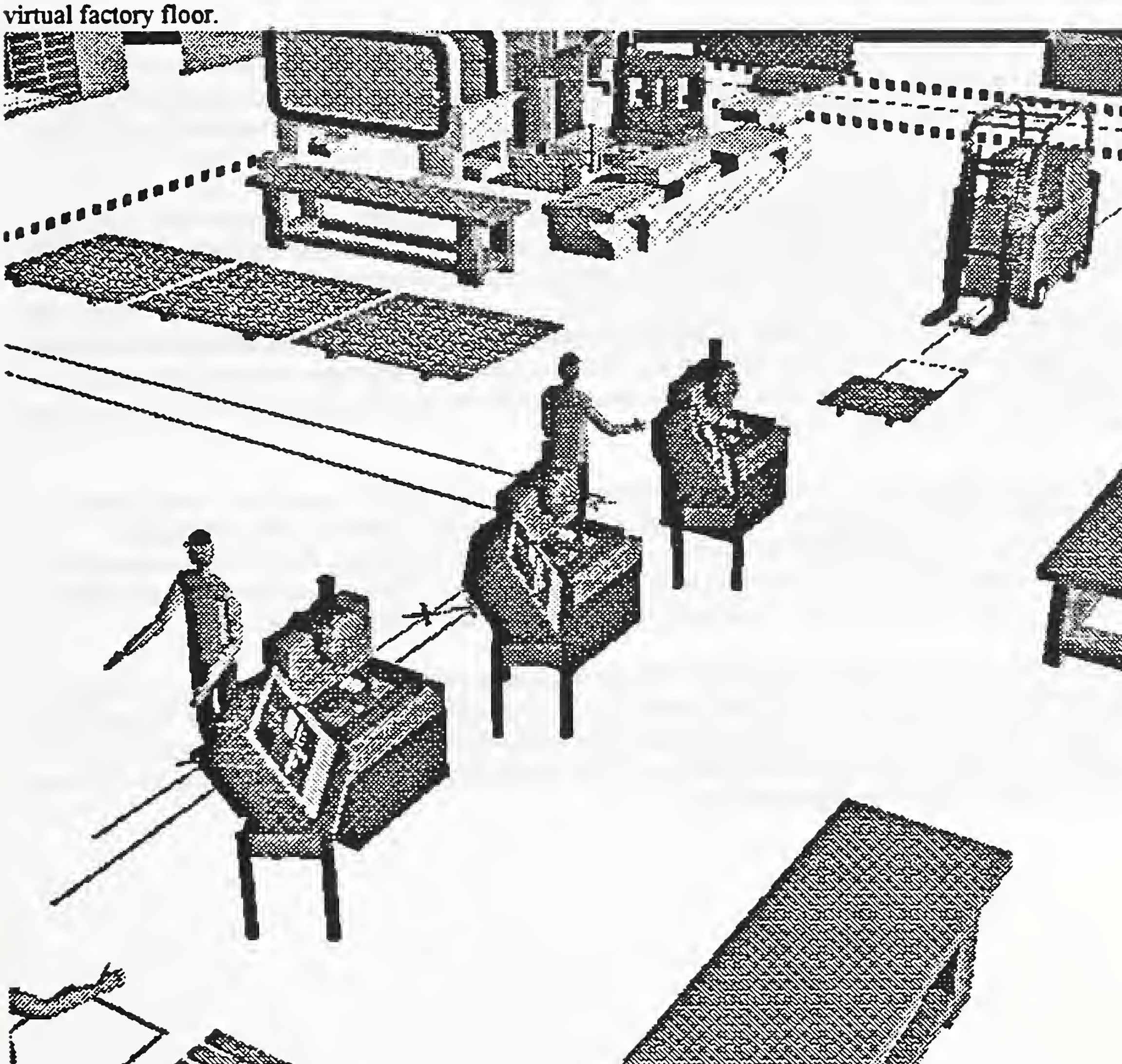


Figure 2 - Three EMCO 100 Vertical Milling machines as they reside on virtual factory floor.

The concentration during development of the virtual factory has been to model the workflow required to produce the prismatic product used in the demonstration. The Quest model simulates the following workflow between workstations:

1. Raw bar stock arrives at receiving area .
2. The raw bar stock is cut in receiving by a bandsaw.
3. The cut stock is loaded in a box and forklifted over to area 1 for machining.
4. The cut stock is unboxed and loaded on the vertical milling EMCO 100 workstations (One workstation for each setup required to machine the product as specified by the ICEM PART generative process planning application).
5. After the products have been milled, they are boxed and sent on to the remaining workstations in the workflow:
6. CMM for quality assurance and gauging. Then washing, heat treating, painting, and shipping.

Software/Hardware

The METK prototype currently consists of the following software: Pro-Engineer is the CAD application, Matrix is the product data management application, ICEM Technologies PART is the generative process planning application, DENEB Quest and VNC are the manufacturing simulation applications used for data validation.

Pro-Engineer is a CAD system that can be used to create product designs. Once the product is designed, and output file that completely describes the geometry can be produced. We are using Pro Engineer now, but other CAD systems are envisioned to be integrated in later versions of the toolkit.

Matrix is used to implement a engineering business model for data integrity and information flow control The business model of a product identifies the states a product passes through for production, what data comprises the product at each state, and the requirements for moving the product to the next state. Matrix also has version control of data at each state.

ICEM Technologies PART is a generative process planning application. It uses a knowledge base of feature definitions, jigs/fixtures , machine tool, cutting tool, methods, and scenario information implemented in an Oracle database. PART accepts the CAD product design data as input and uses the knowledge base to create a process plan for producing the product. This knowledge base is updateable. Version 1.2.100 of ICEM PART is currently being utilized in the prototype toolkit.

DENEB Quest is a simulation application used for: analyzing production scenarios, product mixes and failure response for machines and labor; factory layout; throughput; and production costs. DENEB VNC is a simulation application for visualizing and analyzing the functionality of a machine tool, it's CNC controller, and the material removal process to optimize machining. Quest version 2.1 and VNC version 2.1 are currently being used in the toolkit.

These applications reside together and execute on a single UNIX based Silicon Graphics workstation. The workstation is configured as follows:

Onyx Extreme Deskside Workstation
- 200 MHz dual R4400 processor
- 128 megabyte RAM
- 4 megabyte secondary cache
- 2 gigabyte internal DAT tape drive
- 4 gigabyte SCSI-2 internal disk drive
- internal CD ROM
- dials and button box
- 21 inch Multisync Granite monitor.
- IRIX 5.3 operating system

This workstation is located in the AMSANT facility at NIST. This workstation is connected to the Internet and therefore capable of file transfer protocol (FTP) to accommodate transfer of data files from other sites participating in the project.

Conclusion

The METK prototype will help develop a better understanding and help further identify functional requirements for the individual manufacturing application so it will also help identify integration problems existing between applications that prevent data sharing. Once these integration issues are identified, they can be addressed and technical solutions can then be proposed. One issue is that the input data from one manufacturing application must be able to be input to subsequent versions of the application, i.e, provide upward compatibility. If data is generated for a particular version of an application, that data should not be thrown away, it should be able to be used in later versions of the application. Another issue is data format. If an application generates data in a specific format, will that format be readable by other applications in the toolkit.

Appendix H:

"Integration Aspects of the METK Project"

Carnegie Mellon University
Software Engineering Institute

Integration Aspects of the METK Project

Alan W. Brown
(awb@sei.cmu.edu)

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA15213

August 1995

Sponsored by the U.S. Department of Defense

Notes:

These are slides and notes for the workshop in Gaithersburg, MD on 22/23 August 1995.

These slides aim to provide some background in the area of tool integration, and to point to specific issues that must be addressed in the METK project. Then, an outline of a plan for the next few months is described as the basis for discussion.

Carnegie Mellon University
Software Engineering Institute

General Integration Issues

Spectrum of Integration Approaches

Current Reality

Integration as a Design Activity

Notes:

There are a number of general integration issues which the METK project faces. In these next slides a few of these issues will be explored.

Note that much of the basis for these issues comes from our experience in the Computer-Aided Software Engineering (CASE) domain. However, it is our belief and expectation that the same issues are being faced in the Computer-Aided Manufacturing Engineering (CAME) domain.

Carnegie Mellon University
Software Engineering Institute

Spectrum of Integrated Approaches -1

Independent tools
- individual tools developed in isolation
- integration is ad hoc involving user workarounds

End-user integration solutions
- large organizations invest in specific integration of externally and internally developed tools
- large investment in developing and maintaining the integrated solution

CASE coalitions
- vendors provide specific integrations with strategic partners (e.g., various design and analysis tools)
- involves further customization by end-users

2

Notes:

We can briefly examine the architectural approaches toward CASE tool integration that are common. This will give a flavor of the current state-of-the-practice in this area.

The obvious baseline is that no integration is in place. The tools are acquired individually, and no explicit attempts have been made to integrate them. This is typical when different departments in the same organization have purchased tools, when managers and engineers have separate environments, or when different projects assemble their environments with little or no concern for others.

In some cases large organizations have spent many millions of dollars developing their own in-house toolkit environments. As well as the development cost they have the on-going burden of maintenance and upgrades. While this approach offers greater control of the environment, many companies are regretting this choice as it diverts large amounts of resources from the main activities of the organization (manufacturing, telecommunications, or whatever).

The coalition approach is an attempt by tool vendors to react to user needs by forming specific alliances with other tool vendors. They amend their tools to make them work together in a more consistent way. The interfaces and protocols they use may or may not be made public.

Carnegie Mellon University
Software Engineering Institute

Spectrum of Integrated Approaches -2

Tools integrated into a framework
- use of a common integrating mechanism (e.g., a relational database, PCTE, BMS, CORBA)
- use of common protocols, standards, and interchange formats (e.g., EDIF, PDES, STEP)

"Full" integration
- in practice, a large, complex development environment is a combination of all of the above
- possibly unified through an overall integration strategy
- directed by focussing on the development processes being supported by the end-user organization

3

Notes:

An approach which is beginning to reach operational practice is the use of a framework product. Two approaches are evident — a database approach as exemplified by use of relational and object-oriented databases, and a message passing approach as seen in the SoftBench and ToolTalk products.

In addition to the framework products, additional standards and interchange formats are used. Examples are the X window system to provide visual consistency across an environment, and the EDIF as an import/export facility between design tools.

In practice the toolkit environments often seen are a hybrid of these approaches. Pragmatic reasons dictate that this is so. For example, in buying tools and framework products from different vendors there are inevitably different ways that will be more appropriate for connecting them. Legacy tools and data, and the immaturity of much of the technology, are two more reasons that contribute to this hybrid approach. It seems as though this will be true for at least the next few years.

Carnegie Mellon University
Software Engineering Institute

Current Reality

Mostly "point-to-point" integrations of tools

User-developed scripts and filters

"Vendor pacts" producing tool clusters

Limited practical experience with framework products

Lack of measures and metrics to determine effect of tools and integration strategies

The "frameworks wars" and "standards wars" continue

A lot of maneuvering, activity, and promises

Notes:

Unfortunately, while useful progress has been made, there is still a significant way to go. In particular, the state-of-the-practice as it is commonly found is not very advanced. Point-to-point integrations of tools and the use of scripts and filters still predominate. These ad hoc approaches are allowing people to develop and maintain quite large and complex systems, although often at great cost.

The next generation of technology that will help to relieve the burden of development has been much talked about, but has yet to see serious use. Initial products look promising, but their immaturity is leading a lot of people towards a "wait and see" attitude.

Carnegie Mellon University
Software Engineering Institute

Integration as a Design Activity

For most purposes, the notion of integrating tool X with tool Y is meaningless without a context in which to perform the integration.

There may be many different kinds of integration contexts based on different end-user process needs.

- **a project goal (e.g., reducing data duplication)**
- **a project deliverable (e.g., documentation standard)**
- **a project process (e.g., a manufacturing process)**

We can view integration as a design activity aimed at satisfying the development process needs.

Notes:

An approach which we are interested in pursuing is that integration is a design, or engineering activity. The importance of this approach is that it forces you to consider integration in a different light to the other approaches. Notably, it leads to consideration of:

— designing a toolkit environment with integration in mind

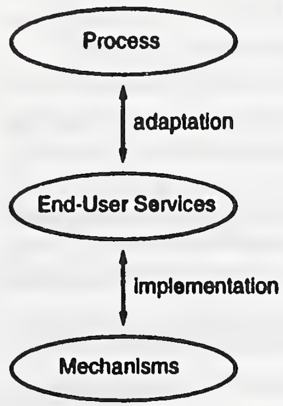
— matching integration aspects to the needs of an organization

— trade-offs that are necessary between costs and benefits of the different integration strategies that could be adopted

This leads to considering a method for tool integration. A major element of this method is matching the technology to the needs of an organization as expressed in the understanding of their end-user processes. In particular, they will have a goal for their integrated solution that must be met in order for the integration to be deemed to be successful. Then, decisions that are made in terms of approach to integration can be considered in the light of achieving this goal.

Carnegie Mellon University
Software Engineering Institute

A 3-Level Model of Tool Integration



CM example

bug tracking procedures and distribution of new system releases

transactions, workspaces, repository management, system build, etc.

file system, make, shell scripts, Andrew File System (AFS), etc.

Notes:

We need a way to begin to think about different aspects of the tool integration problem. A useful approach is to separate issues on three different levels — process, end-user service, and mechanism.

At the mechanism level we have to be concerned with the physical characteristics of the technology we have available — the architectures of the tools, the framework and infrastructure technology, the ad hoc "glue" that must be developed, and so on. The bottom line is that these components must be made to work together.

At the process level we have the procedures and policies of an organization that must be supported by the integrated environment. It is these process elements that provide the context (i.e., the requirements and constraints) that determine how the tools should interact at the mechanistic level.

At the end-user services level we can describe the functionality provided by the physical components without being distracted by architectural details of those component. This allows us to discuss the conceptual integration of the functionality in support of an organizations process needs. This intermediate level provides a buffer that bridges the gap between the detailed mechanistic aspects of the components, and the high level needs and constraints of users.

Carnegie Mellon University
Software Engineering Institute

Who "Owns" the Tool Integration Problem?

Process	**Technology consumers (end-user organizations)**
End-user services	**Technology integrators (environment integrators)**
Mechanisms	**Technology producers (framework and tool vendors)**

Notes:

One way in which this distinction between process, end-user service, and mechanism is useful is in considering who "owns" the integration problem. We can consider different aspects of the problem and see where the primary responsibility for that aspect lies.

At the process level it is technology consumers that have the major role. It is the end-user organizations that have in depth knowledge of their processes and practices, and provide the criterion for establishing the success or failure of an integrated toolkit environment.

At the end-user services level the technology integrators usually hold the key. They will be familiar with the range of technology components available in the marketplace, and have a general understanding of an organization's process needs. Matching the technology to those needs is their main goal.

At the mechanism level technology producers such as tool vendors have a deep understanding of the inner workings of their tools, their relationship with framework products, and so on. For example, amendments to tools to make them more usable, or easier to integrate, can typically best be handled by the tool vendors themselves.

Carnegie Mellon University
Software Engineering Institute

Integration Summary

A 3-level model of integration can help in understanding previous approaches to tool integration.

It can also be used as part of an analytical technique for examining integration in a toolkit environment.

We can take a view of integration as a design activity.

This leads to the possibility of defining a method of assembling a toolkit environment that supports the integration needs of an organization.

We are continuing to explore these notions to refine our method based on an "integration by design" approach.

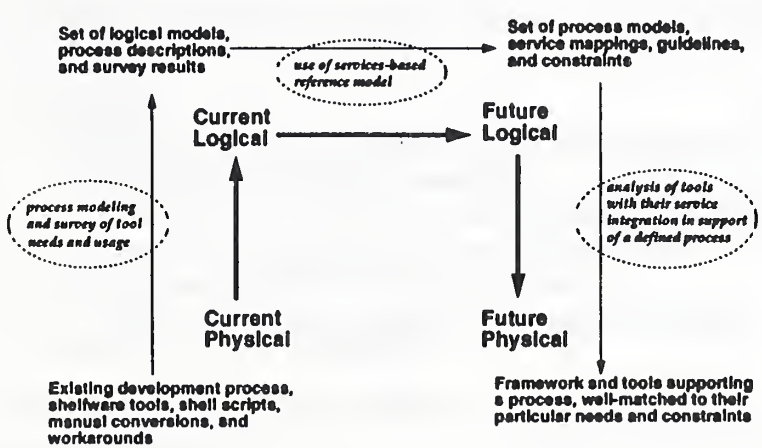
Notes:

We have applied the "integration by design" approach in a number of examples at the SEI, mostly in the CASE domain.

We have found benefit from this approach by using it as the basis for analyzing existing systems, and for developing a design technique that helps in the construction of toolkit environments.

Carnegie Mellon University
Software Engineering Institute

A Design Approach to Integration



Notes:

We can consider an example of a method for designing a toolkit environment to achieve an integrated result.

In this example, we follow the classic design life-cycle of modelling the current physical situation, abstracting the current logical situation, devising the future logical situation, and then implementing that in a future physical situation.

To help us in each of these steps we have products to produce, and methods and techniques that help us produce those products.

An important idea is that this method allows the design process to be documented. In particular, the decision points concerning what needs to be done, and how it will be achieved, will inevitably need to be revisited. This leads to an interactive process where many steps in this method will be revisited and decisions will be changed. By documenting this process there is the raw material to allow such controlled change to take place, and for decisions concerning integration of tools to be made within the context of the overall needs, constraints, and goals of an organization.

Carnegie Mellon University
Software Engineering Institute

Specific METK Issues

Short-term vs. long-term needs

Coordination of participants

Notes:

Here we identify a number of specific issues that must be considered in carrying out the integration work in the METK program.

Each of the items listed here is discussed in more detail in the subsequent slides.

Carnegie Mellon University
Software Engineering Institute

Short-Term vs. Long-Term Needs

Develop an integrated prototype using Part, Deneb, and Matrix in the next 6 months.

Develop guidelines and any necessary infrastructure for tool integration over the next 3 years.

Early decisions about tool integration strategy must be made for the prototype.

May need to revise the prototype based on longer-term activities.

Notes:

At present there is some tension between the immediate, short-term need to produce an integrated prototype using the 3 selected tools, and the longer-term need to consider general integration issues and offer advice and consultancy.

At present, the longer-term objectives (i.e., over the next 3 years) revolve around producing guidelines and advice for tool integrators in the manufacturing engineering domain. This may involve development of some common data structures, data formats, or interface agreements that will aid tool integration in this domain.

However, in the short term (i.e., in the next 6 months) there is a very focused objective: produce an integrated prototype using the Part, Deneb, and Matrix tools.

While the short and long-term objectives are related, the immediate tension arises due to the fact that simple, pragmatic integration decisions must be made concerning the three tools at hand. These decisions may prove to be poorly made once the more extensive aspects of tool integration have been considered in detail.

Coordination of Participants

A wide variety of organizations is involved in this project.

The immediate need is to connect vendors, integrators, and end-users for the integrated prototype, and to agree on:

- roles and responsibilities
- a common operational scenario
- the system architecture
- a detailed schedule for development, integration, and testing

Notes:

There are a lot of people who are involved in this project. The first major task is to try to make sure all the various players understand their role, and what is expected from them.

Once this is achieved, it will then be necessary to quickly reach agreement on the basic infrastructure of the solution: concept of operation, major system components, interfaces, data structures, transfer formats, etc.

Once this is achieved a detailed schedule and milestones can be produced. This, of course, should then be closely monitored as the work progresses.

Short-Term Agenda -1

Aim to demonstrate substantial progress before the end of 1995.

Objective:

- Produce a demonstrable prototype integration of the Part, Deneb, and Matrix tools that:
  - illustrates the value-added of integrating the tools
  - provides a forum for collaboration between vendors, integrators, and end-users
  - can be used as a test case for examining longer term integration issues

Notes:

We have a very short timescale for work on the integrated prototype. This will force a number of early decisions to be made.

By producing this prototype we can get immediate end-user feedback on the integration of the tools, begin the task of getting the various players to work together, and provide grist for the longer-term integration mill.

Carnegie Mellon University
Software Engineering Institute

Short-Term Agenda -2

Strategy:

- To achieve this objective it is necessary to:
  - focus on a small number of particular scenarios of tool interaction in well-bounded domains
  - make assumptions about the operating context, roles and responsibilities of different users, inputs and outputs of the prototype, and so on
  - develop a system architecture that defines the principle components of the system, the interfaces among those components, and the allocation of system functionality to those components

Notes:

To make reasonable progress the boundary must be drawn quite tightly around the integrated prototype. The decisions made in this regard must be well-documented so that the decisions can be analyzed once the longer-term integration strategies are defined.

Carnegie Mellon University
Software Engineering Institute

Short-Term Agenda -3

Tactics:

- In the near term we will begin by:
  - describing in detail a particular operational context to be supported by the integrated tools
  - examine the interfaces, data structures, and control mechanisms used by the three tools
  - document the assumptions and context which limit the application and applicability of the prototype
  - establish the roles and responsibilities, and produce a detailed schedule for all players in the development of the prototype
  - agree on the basic system architecture for the integration of the tools including the shared data structures, key common events, and expected input and output formats from the tools

Notes:

Work has begun in most of these areas. The main need now is to coordinate this work, and to reach a point where all parties agree and sign off on the results.