Converting computer-integrated manufacturing into an intelligent information system by combining CIM with concurrent engineering and knowledge management

Biren Prasad
CERA Institute, Tustin, California, USA

Keywords
Computer-integrated manufacturing, Information systems, Simultaneous engineering, Knowledge management, Information technology, Product development

Abstract
Some industrial organizations using computer-integrated manufacturing (CIM) for managing intelligent product and process data during a concurrent processing are facing acute implementation difficulties. Some of the difficulties are due to the fact that CIM – in the current form – is not able to adequately address knowledge management and concurrent engineering (CE) issues. Also, with CIM, it is not possible to solve problems related to decision and control even though there has been an increasing interest in subjects like artificial intelligence (AI), knowledge-based systems (KBS), expert systems, etc. In order to improve the productivity gain through CIM, EDS focused its information technology (IT) vision on the combined potential of concurrent engineering (CE), knowledge management (KM) and computer-integrated manufacturing (CIM) technologies. EDS – through a number of IT and CIM implementations – realized that CE, KM and CIM do go hand-in-hand. The three together provide a formidable base, which is called intelligent information system (IIS) in this paper. Describes the rationales used for creating an IIS framework at EDS, its usefulness to our clients and a make-up of this emerging IIS framework for integrated product development.

Industrial Management & Data Systems
100/7 [2000] 301–316
© MCB University Press
[ISSN 0263-5577]

Introduction

The product development environment typically suffers from a number of shortcomings. Some are partly due to the lack of integrated tools that information technology (IT) management has to deal with (Stark, 1992; Tonshoff and Dittmer, 1990) regularly. While others are partly due to the diverse nature of an enterprise's business operations (Pawar and Riedel, 1994; Dong, 1995; Bauman, 1990). Too often, tool-related shortcomings are caused by inappropriate or inadequate computer groupware or information aids – such as hardware and software tools to needed database management tools, knowledge-ware, intelligent technologies and standardization (Althoff, 1987). Technology is used here in a generalized sense, similar to its definition in *Webster's Dictionary* (1990) – "the totality of the means employed to provide objects necessary for human sustenance and comfort." For example, by standardizing the design plans, tools and databases of all departments, Toyota enabled design work to overlap between stages. Downstream processes were started while upstream design plans were still being completed (Okino, 1995).

Sources of shortcomings in CIM operations

The operational shortcomings of computer-integrated manufacturing (CIM) result from four main sources (Prasad, 1996):

1. *Process stagnation*: Process stagnation examples include tradition (for example, why fix if it is not broke), legacy systems, business operations, management, technical, or operational 3Ps – policy, practices and procedures (Barclay and Poolton, 1994).
2. *Influence of infra-structural factors*: Examples include factors such as a company's culture, mindset, legacy database, and human factors (Dimancesen, 1992).
3. *Communication roadblocks*: Lack of familiarity, product experience, and training among the CIM teams are some typical examples of communication roadblocks (Albin and Crefeld, 1994).
4. *Organizational roadblocks*: Lack of management support, confidence, and commitment to apply CIM in full force (not haphazardly) throughout an enterprise are some typical examples of organizational roadblocks (Bajgoric, 1997).

The manufacturing industry today is deeply in a paradigm shift (Prasad, 1995b):

From an "economy of scale":

- to an "economy of information" (Kimura, 1994);
- to an "economy of flexibility (agility)" (Mortel-Fronczak *et al.*, 1995); and
- to an "economy of intelligent manufacturing" (Lim, 1993; Prasad, 1997).

While the emphasis is constantly changing from legacy computer systems to new CIM initiatives, most computer-aided design (CAD), computer-aided manufacturing (CAM), CIM and computer-aided engineering (CAE) tools – commonly called C4 (CAD/CAM/CIM/CAE) systems – are in a state of flux. Examples of new CIM initiatives include systems engineering, integrated product development (IPD), knowledge-based engineering (KBE) (Abdalla and Knight, 1994), total quality management (TQM), computer-aided logistics systems/electronic data interchange (CALS/EDI) (Bauman, 1990), etc. This change (from the legacy systems to new CIM initiatives) is putting additional pressure on the C4 tools. These C4 tools are constantly required to provide an up-to-date knowledge, not just the information or data, at the right place, at the right time, with the right amount, and in the right format. These tools are continuously processing a variety of information and knowledge transactions at many different places during a product lifecycle evolution, which also needs to be accessed by other team members at many more places and applications. To allow an effective and efficient processing of knowledge transactions during product realization, C4 tools are being redesigned to reflect an organization's collaborative and competitive posture. Standardization – as in common systems (Althoff, 1987), common methods (Larsen and Alting, 1992), and common processes (Jones and Edmonds, 1995) – is becoming increasingly more important. The quest for C4 standardization is rapidly spreading to all disciplines, organizations and structures.

As a result, desirable characteristics in C4 tools are changing from their original needs (Alting, 1986) as "design tools":

- to data exchange tools (Tonshoff and Dittmer, 1990);
- to distributed computing tools (workstations, mainframe, database) (Stark, 1992);
- to work-group computing tools available globally across the networks, local area network (LAN) wide-area network (WAN), etc. (Willett 1992; Kimura, 1994).

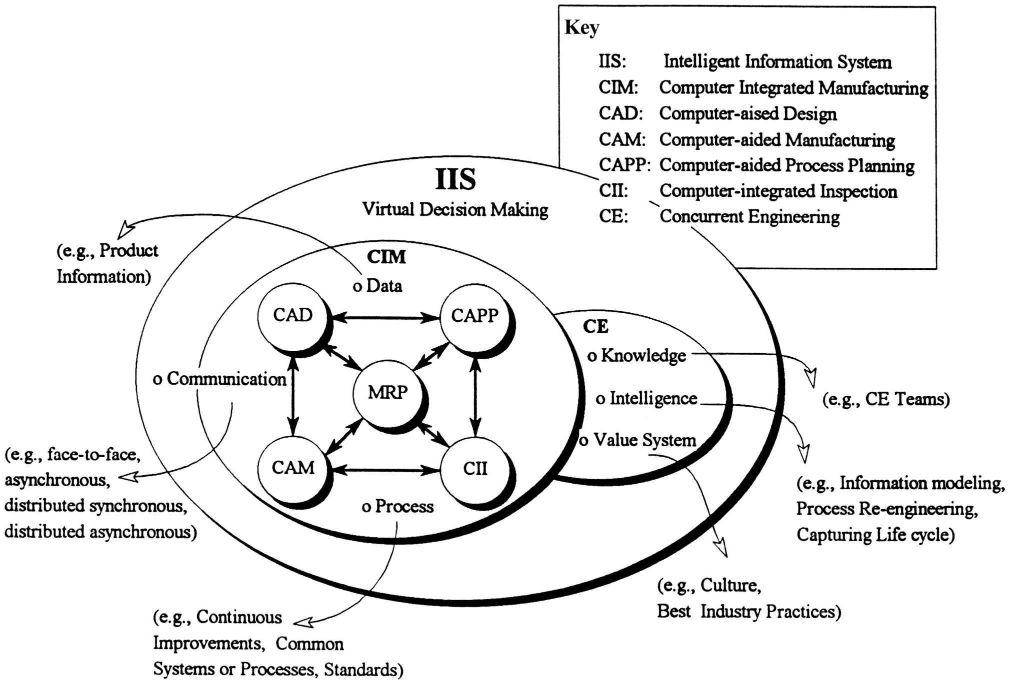
Toyota, for example, by unifying these characteristics across all organizational areas, all departments and work-groups working on the product, has reduced its average automobile time-to-market period from 30 months to 18 months (Okino, 1995; Shina, 1994). In recent years, there is an increased emphasis on the use of new or emerging feature-based standards during data exchange (Hummel and Brown, 1989; Jones and Edmonds, 1995). Product data exchange using STEP (PDES)/standard for the exchange of product model data (STEP) is being implemented in newer CAD tools through the use of a series of application protocols (APs). One of the APs addressed by the initial release of STEP is configuration-controlled design (CCD), formerly designated as AP203 (Curran, 1994). CCD represents the dawning of a new era in digital product data exchange as it specifies how solid models are to be communicated. Using this protocol, one CAD system can directly exchange solid models in a standardized format with another dissimilar CAD system.

Most research and development (R&D) efforts toward automation for modern manufacturing have been independently developed. For example, creating faster processors as hardware brains (e.g. silicon graphics) for running high-end graphics applications, as in the 1980s, were independently developed. As a result, these automated applications were often self-assertive. It did not work out a panacea, as initially expected, for reducing design and development lead-time. In the same decade, design grew more complex, and the amount of time required to prepare the corresponding data (inputs) for each tool to be used in the design process also increased. At the present time, highly automated areas in manufacturing include CAD, computer-aided process planning (CAPP), CAM, manufacturing resource planning (MRP), and computer-integrated inspection techniques (CII) (Chang and Wysk, 1985). With such tools, major functions are performed electronically (using compute power) but the data are nearly always passed manually. There is a recent proliferation of "islands of pre- and post-processors" generated from using these tools due to piece-wise growth of the tools themselves and lack of in-house standards for applying them uniformly across the various tool sets. With the dependency on computational and logical techniques, the recent emphasis has been on integrating the existing CAD, CAM, CAPP, MRP and CII systems to provide a CIM environment (see Figure 1). Developments in integration area include initial graphics exchange standards (IGES), design for manufacturing and assembly (DFMA), database management systems (DBMS), PDES, just-in-time (JIT), manufacturing automation protocol (MAP) and cell control software (Prasad, 1995a). However, they are currently deemed to be independent contributions to improve productivity or efficiency in specific CIM areas or applications.

Today, CIM systems are merely being applied to integration and processing (storage and automation) of data, communication, and processes (common systems and standards). The communication part of CIM design is data or information.

1. *Data or information.* This includes many different categories of product images and information:
   - CAD data;
   - CAM data;
   - CAPP data;
   - CII data;
   - design specifications;
   - the history of production; and
   - interface information in various forms, including electronic, text, raster images, video, audio and their mixture, as well as many different types of paper formats and methods.

Figure 1
Automated areas in manufacturing



2 *Process*. This includes methodologies, such as continuous process improvements (CPI), quality function deployment (QFD), Pugh, Taguchi, TQM, common systems, standards, etc. Embedded within such a vast information base, lies the hidden knowledge about the product realization and the design and manufacturing process knowledge.

CIM, in the aforementioned way, is not able to deal with knowledge management (KM) issues adequately and for solving design and manufacturing problems related to concurrent decision and control even though there has been an increasing interest in subjects such as artificial intelligence (AI), distributed blackboards (DBBs), knowledge-based systems (KBS), expert systems, technical memory, collaborative engineering, etc. (Siong, 1993). The latter slew of tools is more cognitive, collaborative and distributive in nature and is quite potent for decision making required during concurrent engineering (CE) and KM (Zhang and Zhang, 1995). The paper describes how CE and KM functionalities can be combined together with CIM to provide a suit of product design and manufacturing capabilities that cut across entire organizational lines globally. The paper is based on the author's experience in implementing CIM to a large automotive manufacturing client of EDS.

Basis of decision making in CE

The concept of CE was initially proposed as a potential means to minimize the product design development and delivery (PD^3) time (DARPA, 1987). Since then, many interpretations of "concurrent engineering (CE)" have emerged in literature (Zhang and Zhang, 1995). Today, CE is much more encompassing. Expectation ranges from a modest productivity improvement to a complete push-button type automation, depending upon the views expressed. CE is a parallel approach – replacing the time-consuming linear process of serial engineering and expensive prove-outs (DARPA, 1988). It is intended to elicit the product developers, from outset, to consider the "total job" (including company's support functions).

CE has a major impact on the process set-up, and on the way an organization conducts the PD^3 business. As shown by Zhang and Zhang (1995) and Prasad (1996), CE replaces the traditional sequential "over the wall" approach to a simultaneous design and manufacture approach with parallel, less interrelated processes. It aims at reducing the total effort in bringing the product from its concept to delivery, while meeting the needs of both the consumers and industrial customers.

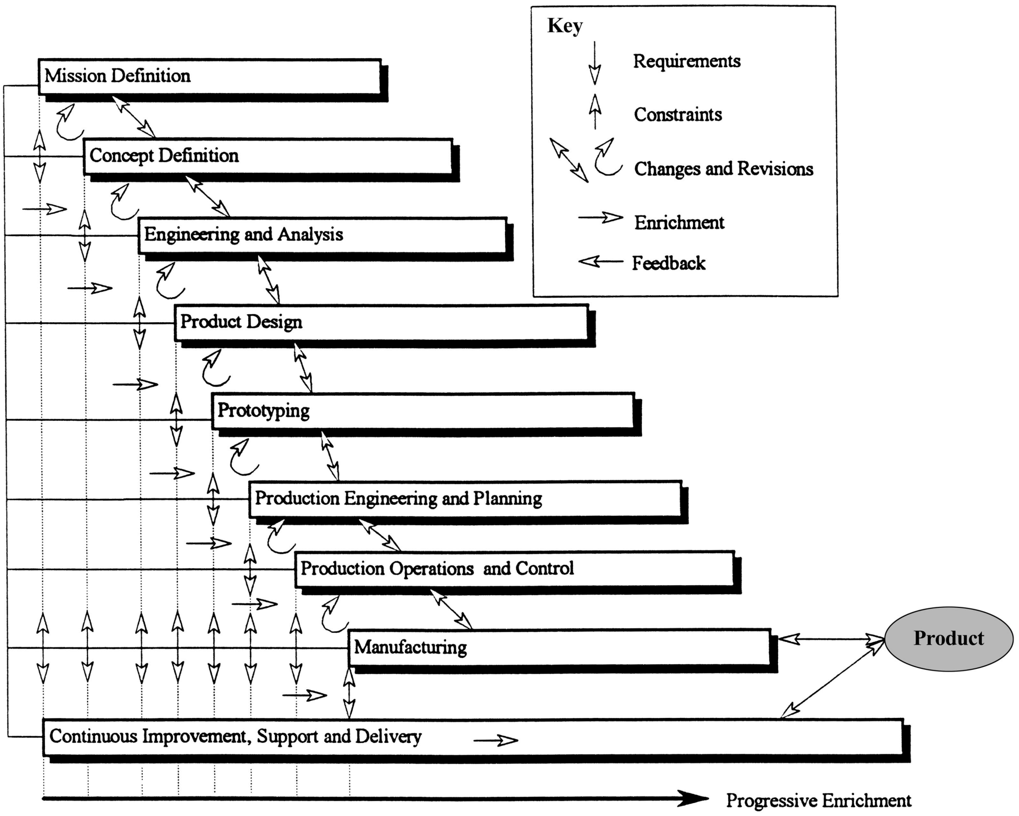
The four major phases of the product design and development (as shown in Figure 2.26 (Prasad, 1996)) have been detailed into nine tracks (shown in Figure 2) running in parallel. Figure 2 shows the different tracks of the development process. These tracks are: mission definition, concept definition, engineering and analysis, product design, prototyping, production engineering and planning, production operation and control, manufacturing, and finally, support and delivery. The continuous improvement – "support and delivery" – is an ongoing coordination track, which runs for the full lifecycle. This track provides, besides normal project management functions, sequencing, cooperation and central support to the other tracks. These tracks are not unique to any particular product, steps and overlaps may differ from product to product.

Business drivers for CE

Prasad (1996) has chosen to divide forces that influence a CE domain into seven agents (called here seven Ts): talents, tasks, teams, techniques, technology, time and tools. One of the primary team issues is the decomposition of tasks. The people's issue is the composition of teams. Teams are often used to cooperatively solve the problem. Technology issues arise due to the drive for competitiveness. Examples of popular technologies in CE are soft prototyping, visualization, product data management, design for X-ability, multimedia, EDI, etc. Tools mean software, hardware, and networks that make CE practical in today's world of multinational corporations, multi-partner projects, and virtual corporations. From the time point of view, CE is an initiative of the product development community that has the goal of reducing the length of the product design and manufacturing cycle time by allowing teams of engineers to develop design modules concurrently from their perspectives (Pennell and Slusarczuk, 1989). Training also plays an important role in CE. A popular word in the business press is reengineering, meaning, in short, revamp the processes by which one satisfies customers' needs.

Timing is an important consideration in a PD^3 system. A lot rides on timing of

Figure 2
Showing concurrency during phases of product design and development process



decision making and problem discovery. Approximately 80 percent of a product's lifecycle cost is driven by decisions made in the first 20 percent of the program effort (DARPA, 1987; DARPA, 1988).

Once a PD^3 process is decomposed into a set of tracks, and a track is decomposed into a set of tasks, they become one full spectrum of steps leading to a product realization. The staggering of their (steps') start points and overlaps is indicative of partial information sharing. Orders are indicative of their precedence. The amounts of overlap between any two consecutive tasks are indicative of the degree of dependency that may exist between them (Krishnan, 1993). In general, there will be greater affinity and dependence between pairs of activities which are adjacent to each other. The further away the activities are positioned from each other, the lesser would be the degree of affinity or the need for information transfer among them. For example, a mission definition track would be more closely related to concept definition but would have very little bearing with activities such as manufacturing. Similarly, a manufacturing track would be closely related to a production operations and control track but would be quite less sensitive to tracks such as engineering and analysis. If the tasks are completely independent, they all can be aligned along the left margin of the diagram, keeping the precedence intact. The time-to-market in that case would be dominated by tasks that take the longest time to finish. This is a case of a true "simultaneity" or "simultaneous engineering" situation.

Some discerning companies which have felt a greater drive to improve the competitiveness urge, are focusing their vision on the combined potential of CE, KM and CIM technologies (Abdalla and Knight, 1994). CE, KM and CIM do go hand-in-hand. Together, they are called intelligent information system (IIS) (Prasad, 1995c). CIM plus CE and KM equal IIS.

$$\mathrm{CIM} \oplus \mathrm{CE} \oplus \mathrm{KM} \Rightarrow \mathrm{IIS} \qquad (1)$$

Some companies consider manufacturing information-intensive. Most companies consider manufacturing knowledge-intensive. Many more believe that intelligent handling of knowledge, not just the information or data through computer techniques or knowledge-management techniques, can yield a better CIM system since it can monitor, detect and correct problems. IIS reduces the need for frequent manual intervention.

Enabling elements of IIS

The major enabling elements of IIS largely applicable to product development are (Prasad, 1997):

- *Seven Ts*. The seven Ts consist of talents, task, teamwork, techniques, technology, time and tools (Prasad, 1995b). Table I lists factors showing scope for the enabling elements and what typical questions to ask to determine its (scope's) importance. The teamwork entails manufacturing support personnel with "X-abilities" talents (expertise) in the product development team (PDT). The teamwork also implies making use of surrogate "X-ability" tools during the early stages of design rather than being called on only when problems crop up, or when the design is set in stone. Tools and technology includes a growing set of inter-operable computer aids for geometric design and prototyping – networked into a highly extensible environment. Techniques imply analysis and design methods (such as software prototyping, design for X-ability, analysis, simulation, multi-media, virtual reality, etc.) to visualize product and process concepts quickly, and in a format that is understandable to all team members. Teams apply these tools and techniques to timely experiment a number of product and process options that are available or feasible. For manufacturers to become more "time competitive," teams need tools to match the tasks. In other words, software and hardware tools, techniques and technology must empower teams at all levels of an organization; either by pro-actively informing the teams of tasks that require immediate attention, or by giving teams timely information to make timely decisions, before tasks become critical.
- *Empowerment*. This entails empowering multi-disciplinary teams so that they can make critical decisions early in the product-definition stage. Often, due to lack of empowerment, teams make decisions later in the process, such as during major design reviews, when the cost of engineering change is much more dramatic.
- *Requirement management*. It deals with attaining a balance between product and process management. Managing process starts with understanding the requirements, interfaces, and plan of manufacture for an existing design to extending support for a new design, if envisioned, while meeting the artifact's major functional goals.

- *Information and knowledge modeling*. It entails the use of various models that electronically represent, in convenient forms, information (knowledge, methods and data) about the product, process, and the environment in which it is expected to perform.
- *Standard means of exchange for data, methods and knowledge*. This is accomplished by a slew of standardized support systems (computers, networks, tools, database, applications, procedures, etc.) to encourage the sharing of data among CE team members. A network of compatible systems quickly relays product design plans, iterations, reviews, and approvals to the CE participants.
- *Information and knowledge sharing architecture*. This includes enabling technologies for CE – multimedia communication, framework integration, enterprise integration and coordination in a distributed synchronous setting. Standardized means of knowledge sharing foster effective communication among the many different personnel teams involved in IPD activities. Examples are: recording of design history, processing design plans, common product design and development process, standardized testing, design

Table I
Seven Ts – the enabling agents

Major enabling agents	Factors showing scope/range	What typical questions to ask to determine its (scope's) importance or contribution
Talents	Expertise (competence), experience, negotiation ability, negotiation power, intelligence quotient (IQ), job skills, education, professional development, job training programs, technical and leadership training, culture/attitude	Is the team competent to do the job? Is the team experienced enough? Is the team able to come to a consensus? Can team resolve its conflict? Does the team apply common sense? Do the team have basic understanding of the engineering concepts? Is culture of the team conducive to cooperation?
Task	Independent, dependent, coupled, size, complexity, novelty, repetitive, hierarchy, product, process and work breakdown structures, numbers, technical risk, etc.	Are the tasks dependent, independent or coupled? What is the project's size? How complex is an activity? Have the tasks been decomposed enough? Are the tasks unique? Are the tasks repetitive? Do we understand the tasks' hierarchy? How big/small is each decomposed task/hierarchy? What are the probabilities for their successful completion?
Teamwork	Cooperation, commitment, motivation, trust, morale, role balance, job rotation, group dynamics, personal satisfaction, empowerment, etc.	Are team members cooperative? Are they committed? Are the team motivated? Do team members trust each other? Do they respect each other? Are they concerned about their personal gains, security? Do the members help each other in needs? Do the teams change hats frequently? Are the teams able to communicate effectively? Do the teams have open and clear channels of communication?
Techniques	QFD, quality engineering, CPI, process re-engineering, Taguchi, robust design, serial, concurrent process, systematic approach, decomposition, integration, concurrent function deployment (CFD), TQM, etc.	Is the team familiar with CE techniques and their usage? Does the team use QFD, quality engineering and CPI principles while doing his or her job? Does the team re-engineer the process before automation? Does the team understand differences between serial and concurrent development? Is the team able to decompose products into hierarchical structure? Does the team understand: concurrent function deployment (CFD)?; total value management (TVM)?; or big picture?
Technology	CAD/CAM/CAE/CIM, JIT, process planning, NC, DNC, workstations, networks, client server, e-mail, product technology, process technology, features, innovation, etc.	Are the teams trained in the use of CAD/CAM/CAE/CIM and JIT software systems? Do they understand their usage in product development? Have they used NC and DNC from the same CAD model environment? Are they computer literate? Are they comfortable working on different workstations, distributed networks, and client server environments? Do the team have e-mail capability? Do they understand the current design technology or its limitation? Do they understand new parametric, feature-based or knowledge-based CAD/CAM systems that are coming to the marketplace?
Time	Start time, finish time, lead time, magnitude, delivery time constraints, productivity, schedules	What is the lead time for doing an activity? What are the start and finish times (schedules) for completing a task, an activity or a phase? How long an activity would take to finish? Can the project be completed on time? Why it takes so long to perform an activity? Are the teams working effectively? Are the tools helpful? Is the process robust enough?
Tools	Office tools, communication tools, networking tools, project management, computer-based design modeling tools, computer aids, product models, process models, enterprise models, codes and standards	What are the available tools? Do the teams have enough compute power? Do the teams share the compute resources? Are the office tools meeting the teams' needs? How are the team communicating? Are the teams networked? Do the teams have cooperative e-mail facilities? Are they able to engage others in (over-the-Net) discussion of parts' features, graphics and video transfer. What collaborative tools are available over-the-Net? Are the relevant design codes and standards available on-line?

review, video conferencing, project management, etc. Mozaic – the Auto-trol Technology's object-oriented CE architecture – is an example of the modularity and architecture that comes from object-oriented technologies. Objects represent the real-world product and process decomposition, tree structures, and their ease of communication across platforms that make them ideal for collaborative environments.

- *Cooperative problem solving – seven Cs*. Cooperative problem solving includes seven Cs: collaboration, commitment, communication, compromise, consensus, continuous improvement, and coordination. It means sharing problem-solving insight or deep knowledge, so that instead of a single individual, the whole team can make joint decisions. Functions that focus and facilitate collaborative discussions involve use of sound analytical basis such as recording of design rationale, electronic critiquing of designs, and planning and execution of design changes.
- *Intelligent decision making*. Decision making can be viewed as a process of creating an artifact that performs what is expected (specified as requirements) in the presence of all sorts of constraints and operating environment that governs its behavior. Depending upon the cognitive knowledge about a product available to a decision maker, design decision may range from cognitive to progressive.

$$\text{Enabling agents} = \cup\,[\{\text{seven Ts}\}, \{\text{empowerment}\}, \{\text{requirements management}\}, \{\text{information and knowledge modeling}\}, \{\text{standard means of exchange for data, methods and knowledge}\}, \{\text{information and knowledge sharing architecture}\}, \{\text{cooperative problem solving – seven Cs}\}, \{\text{intelligent decision making}\}] \quad (2)$$

Although IIS has a major impact on productivity, there are several major barriers that inhibit regaining the full potential of manufacturing competitiveness.

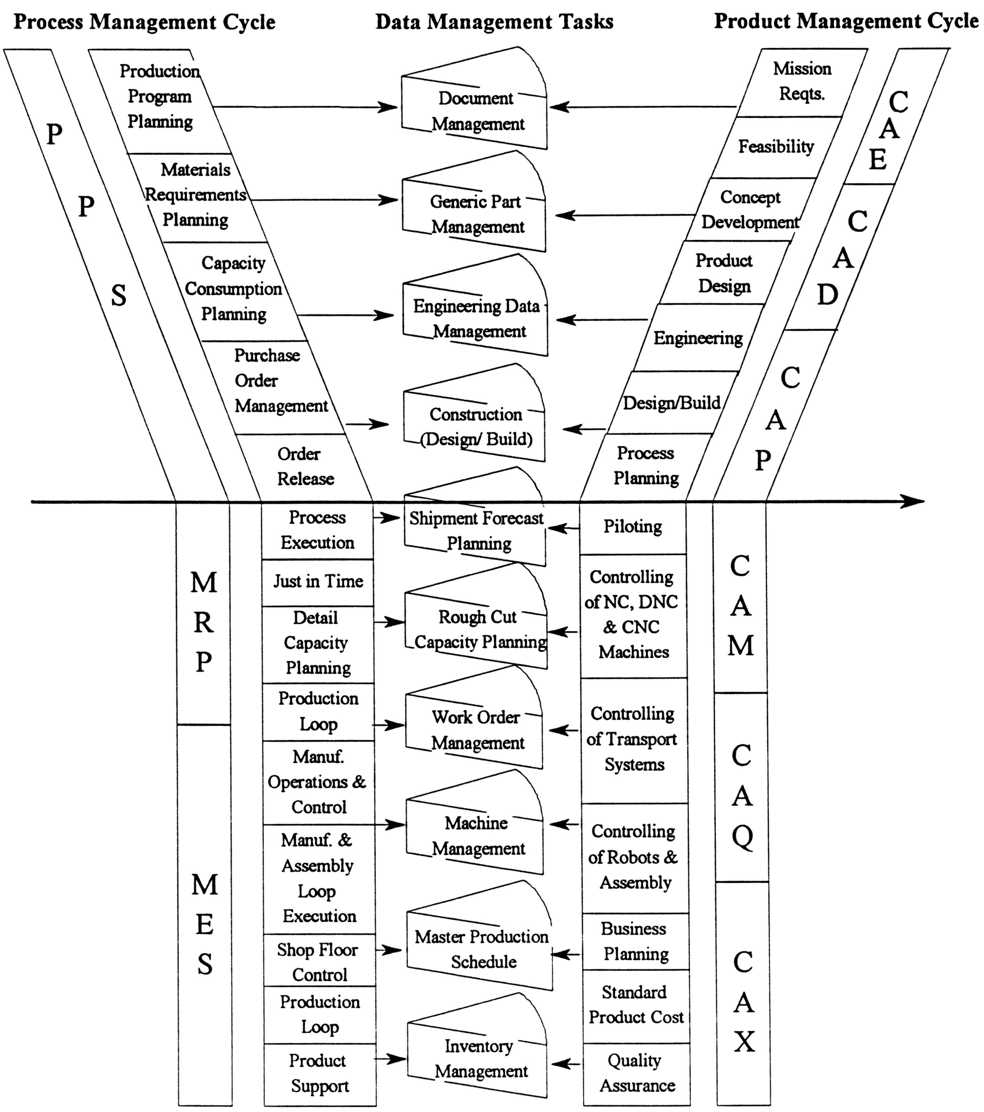
Foundation of IIS

Designing a new and complex system requires dividing the design work-groups into design teams, determining the iterations between the computational and data modules, and reordering the sequence of modules. A module in this context does something. It can be, for example, a computer program or a person to minimize iterations. Very few "cooperative tools" are available to aid the design team in making early decisions regarding proper execution sequences of the design.

More and more people are looking to software to be a teacher, expert advisor, organizer, problem solver and specialized librarian, in addition to its more traditional function of being a "productivity tool." They expect the software to teach them better ways of doing their work. Some are seeking new, better, and more enjoyable work environments (e.g. multi-media, windows, etc.). And a growing number is looking to software to help them do things that they have never done before. Software vendors are also responding with better capabilities, more efficient environments, faster processing, and all-in-one integrated tools.

In the future, an engineer will perform a multitude of highly specialized tasks, each having more than one disciplinary flavor. Intelligent systems that contain both the products' specific functional knowledge (algorithmic and heuristic) and the process-specific facts pertaining to the product manufacturing operation will be used extensively throughout a corporation as shown in Figure 3. Figure 3 shows most of the manual tasks related to product design, engineering, and manufacturing processes computerized into a series of integrated product and process-design modules or knowledge-based tools to support a complete customer-focused manufacturing system. Both purchase order management and inventory management provide direct planned order conversion from MRP/master production schedule (MPS). Both MRP and MPS can be net change or regenerative with available-to-promise for forecasted parts. Production plan forecast targets inventory balancing, smooth production curve, and seasonal product-line planning for both new product rollout and old product phase-outs. They are shown as three vertical parallel rows of functions supporting the product management cycle, process management cycle and data management tasks (see Figure 3). The engineering/manufacturing database in the middle column indicates that product management and production management cycles are sharing the information and knowledge through a common database. A placement of a database-pie along the middle column (in Figure 3) and the corresponding arrows pointing to it are indicative of the information and knowledge flow coming from the two management cycles. The arrows are not indicative of the flow from an element of a management cycle to an element of the engineering/manufacturing database, as it appears to be the case in Figure 3. Purchase

Figure 3
Concurrent intelligent information management systems

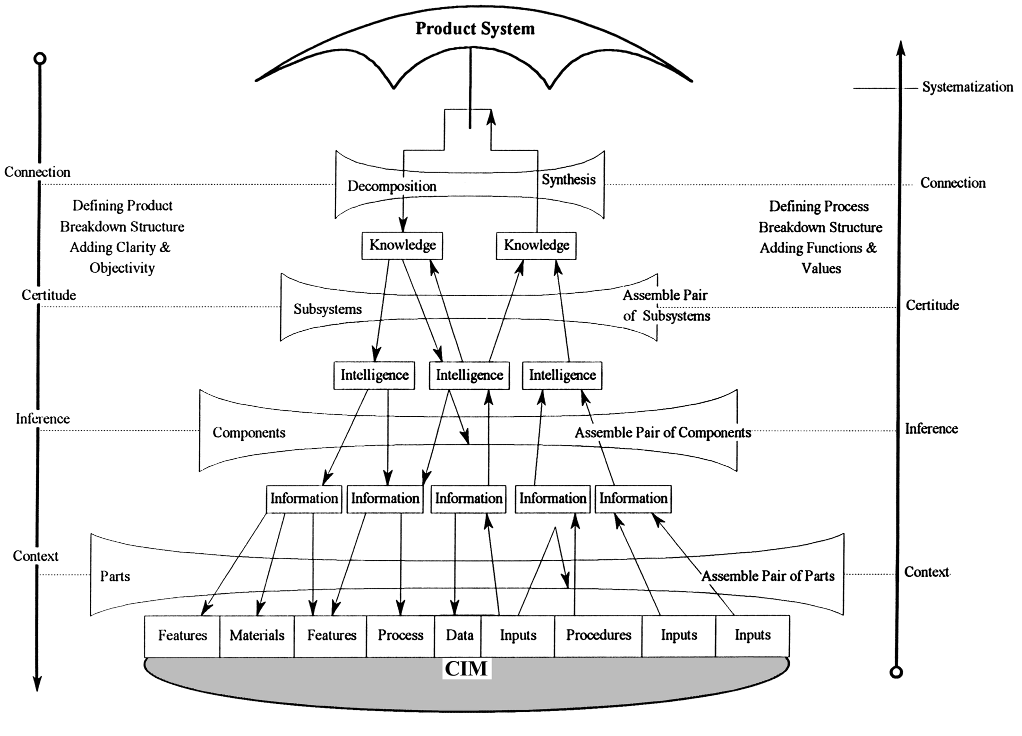


order management can provide a configuration capability to make-to-order, to make-to-stock and to customize sales orders based on models or options. The CE approach emphasizes “operations decision-making” support in contrast to planning emphasis of MRPII, and the more recently developed planning systems concepts such as enterprise resource planning. Inherent in the notion of operations decision-making support is the need for up-to-the-minute feedback on the “execution” of the production plan, that is a real-time manufacturing execution system (MES), fully integrated into the traditional planning, order management and financial functions. Other CAE, CAD and CAX (where X stands for E, D, M, etc.) tools will assist a work-group manager in making early decisions regarding decomposition of a complex design problem (product system) into elements: sub-systems, components, parts, and features, materials, process, data, etc. These tools will guide the decoupling of decomposed elements into independent or semi-independent steps. The tools will recommend, depending upon the elements’ complexity, paralleling steps (concurrent use of a computer program or a person) which require little or no feedback between them. Some tools will also recommend which steps are unavoidably coupled, thus forcing them to be executed in a serial mode. Other tools will allow groups to visualize the relationships among various steps. The tools can also be used to reach consensus regarding which design constraints are common among the steps and which feedbacks are to be allowed. The tools will not only provide a framework for analyzing these alternatives but will actually show how to alter these processes, based on the captured intelligence or knowledge stored as technical memory.

Figure 4 serves as a roadmap in pointing out things in relation to automation of a company's CE environment. It consists of five guiding layers: systematization, connection, certitude, inference and the context. The top layer deals with activities that happen after the completion of planning track and before the product design loop is completed. Members of product design, process design, and production-ready teams participate in engineering the product. Beginning with a feasible product concept from the earlier loop, the detailed sub-system design requirements are specified in this phase. Parts, components and sub-systems are modeled generically. Optimization and design improvement are carried out in an iterative fashion with alternatives further refined through the process outlined in the concept development loop. Each functional requirement, including both product and manufacturing requirements, is satisfied one by one. The product design and development process has six major activities (see Figure 4):

1. *Design and build of "top-down" PtBS tree.* An exploded view of the CAD model can be obtained from the initial assembled model of the product using methods, such as a rule-based or knowledge-based approach. An exploded view represents an ordered disassembled state of a product. The precedence described by the relative location of parts ensures the parts' compatibility.
2. *Allocation of specifications.* Parts can be initially designed using a feature-based product representation scheme, such as descriptive form features, constraint-based, etc. An assembled product model can be created using the inter-part mating information or knowledge specified by the designer or through the feature recognition aspects of the intelligent CAD system.
3. *Data layer.* This layer determines the specification levels from the voice of the customers (VOC). The voice of the customers helps determine the specifications and to categorize them into different levels: systems, sub-systems, components, parts, materials and features, etc. Specifications come in forms such as materials, structures, features, parts, tools and manufacturing processes as shown in Figure 4.
4. *The assembly sequences and development (aggregation) of assembly (or system) plan.* The assembly process involves systematic introduction of parts, or groups of parts, to a fixture or to an assembled component.

Figure 4
Foundation of an intelligent information system (IIS)



The precedence information, or knowledge implicit in an exploded view, can be used to generate the selected assembly sequences. The generated assembly plan must ensure part mating conditions and a set of assembly criteria designed to avoid redundant and awkward assembly sequences.

5 *Validation of the assembled design against the allocated specifications*. An assembly operation is valid if, for example, it establishes planar or cylindrical contacts or does not enclose components. The enclosed situation is most commonly found at the axes' intersection point where the parts after the proposed operation are geometrically inaccessible. An assembly sequence is considered better if it requires the minimum number of changes in the direction of assembly or is geometrically accessible and does not involve awkward assembly operations.
6 *Great product*. The performance of the integrated system is optimized to derive a quantitative set of sub-system functional performances. The performance of each individual sub-system is then optimized by trading in system and sub-system activities. If the resulting feasible system design, sub-system design, component design and part design outcomes are not satisfactory, the aforementioned six-step procedure/activity is repeated. The procedure is continued until a reasonable model (for system, sub-systems and components) is obtained.

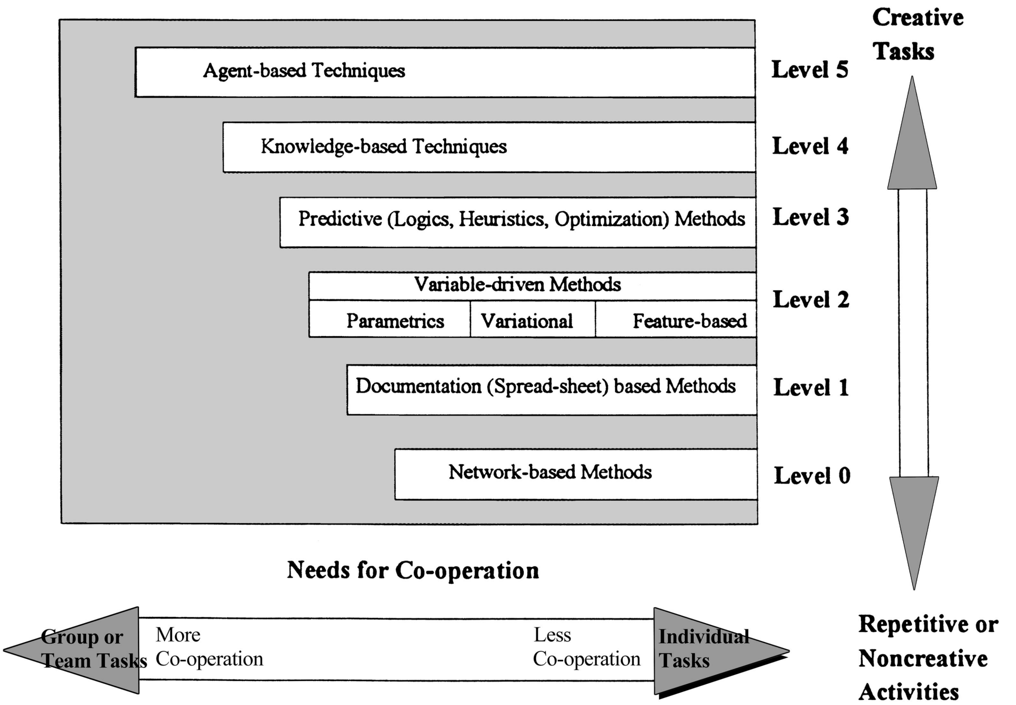
Figure 4 shows the interactions among the aforementioned activities. Sometimes, a given function is achieved by a combination of several sub-systems (e.g. heating usually requires heating elements and a blower). In those cases, the CE team allocates the functional requirements to a set of candidate sub-systems to engineer the performance of an integrated system, as shown in Figure 4. Product engineering activities include embodiment (sub-system design, assembly, prototype design, manufacturing processes), detail design (components, parts), and bill-of-materials (assembly instructions, re-assembly, accessibility, DFX, etc.) (Shilke *et al.*, 1989).

The IIS tools provide means to synthesize – assemble a pair of these data (features, materials, process, inputs, etc.) into a part; a pair of parts into a component, a pair of components into sub-system, and a pair of sub-systems into a working system. Joining the activities in pairs can be continued until all of the features, materials, process, data, inputs, etc. are converted into parts. The multiplicity of boxes for "inputs" and "features" simply indicates that they are obtained from more than one source. Arrows are indicative of directions of information flow. When all parts are converted into components, it constitutes a super-set of data called "information." When all components are reconstructed into sub-systems, the conversion adds "intelligence" to each pair of components. This super-set of information leading to sub-system is called a layer of "intelligence." Finally, when all of the sub-systems are synthesized back into a full system, this synthesis step adds "knowledge" to each pair of subsystems. Information is an enriched set of data. Intelligence is an enriched set of information. And finally, knowledge is an enriched set of intelligence. As teams and their tools need to use data in different places, those data must be linked into corresponding translators that automatically convert them into a more meaningful form, such as new forms of knowledge (see Figure 4). And the way to assure this type of automation is through knowledge-based engineering (Prasad, 1996).

Levels of information enrichment

There are various types of activities that take place in product design and development. On the one hand, there are repeated or non-creative activities that can be performed by a team member or individual. Such activities are routine, teams are familiar with them, and they do not require much collaborative effort. Some are middle-of-the-road activities that may require some degree of intelligence for decision making. On the other hand, there are creative activities that require knowledge beyond one's own disciplines or areas of expertise. Depending upon the levels of activities and need for cooperation, the degree of intelligence required varies. This is shown in Figure 5 where six levels of techniques, or methods required for a class of activity, are identified against the "degree of creativity" and "needs for cooperation." The first of such techniques is "network-based techniques," which can be performed by an individual team or a team-member, and where the activities are routine types. This is identified as level 0 in Figure 5. The next level is level 1. Over time, team members may have discovered heuristics in performing such tasks – what work the best (best practices) and – what to do in what situation (common systems). Such activities are still routine, though in order to reduce the lead-time, some level of intelligence, such as logic and heuristics-based methods, would

Figure 5
Levels of techniques/methods driving cooperation in a CE office



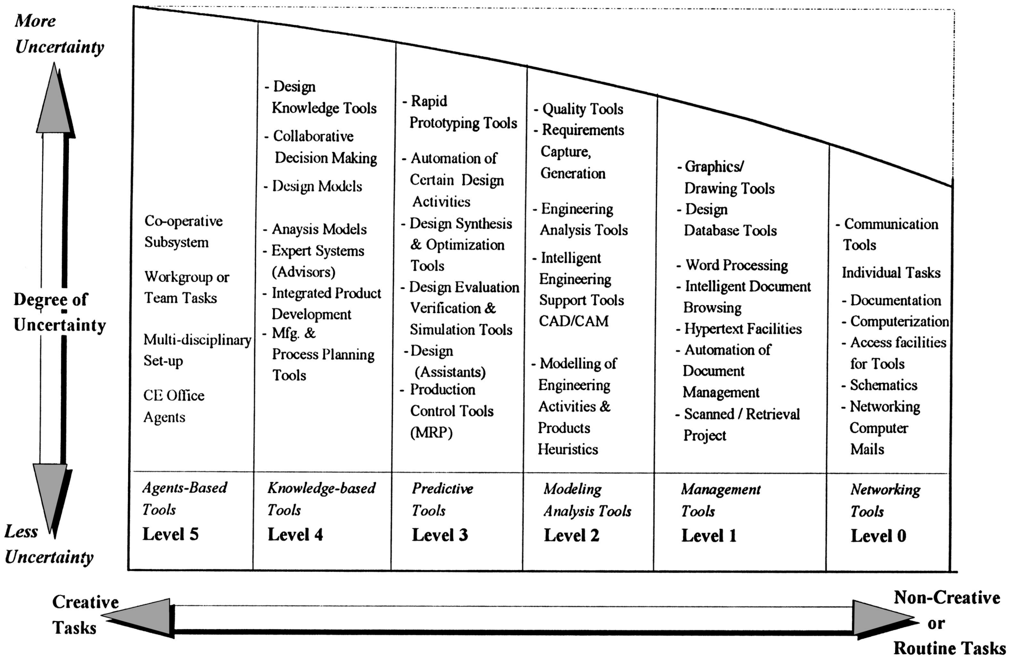
be useful. The need for cooperation increases as one moves away from simple problems to family of part – geometry creation (level 2 activities). The use of variable-driven methods (such as parametric, variational or feature-based) are useful for level 2 to alleviate the boredom tasks of recreating the design details repeatedly, based on geometrical compatibility. There are problems "beyond geometry" whose solutions require non-geometrical knowledge, such as materials substitution, configuration designs, layout designs, knowledge of interaction problems, etc. These are classified as level 3. Knowledge-based techniques are more suitable to provide a level of intelligence to deal with such a "knowledge-rich" class of problems adequately (level 4). On the other end of the spectrum are the agent-based or "multiple knowledge-based" activities (level 5), which require teams with intelligence, ingenuity, and creativity. An individual team with its own knowledge may not be able to comprehend the magnitude of the decision. Most complex decisions are made during teams reviews, or quality network circles, or from similar collaborative sources. The levels of techniques or methods required addressing all these classes of activities are contained in Figure 5. There are six levels of intelligence techniques identified, one for each level of activities, from level 0 to level 5. Level 5 activities are not easily amenable to automation techniques since the possibilities are unlimited. In Figure 6, an attempt is made to classify the range of tools by the "degree of creativity" and "degree of uncertainty" present. The computerized tools required for creative tasks (levels 4 and 5) are of a very different class than those required to solve routine type activities (level 0).

$$\text{Range of tools} \equiv \bigcup\,[\{\text{networking tools}\}, \{\text{work-flow management tools}\}, \{\text{modeling and analysis tools}\}, \{\text{predictive tools}\}, \{\ldots\}, \{\text{knowledge-based tools}\}, \{\text{agent-based tools}\}] \qquad (3)$$

The range of all such tools with product development potential can be classified into the following six difficulty levels:

- *Level 0: Networking tools.* The types of activities that may fall in this category are document computerization and access facilities for text, graphics, schematics and distributed database facilities. Networking tools also include communication tools such as electronic mails, GroupWare and multimedia between, and across, the members of CE teams.
- *Level 1: Workflow management tools.* These control the priority of tasks in a work-group, a unit, a department, or in an enterprise setting. Database tools, such as proven systems database, proven components and part database can be used

Figure 6
Automation levels of computerized tools in a CE office



for this purpose. Other types of tools in this category are: word-processing, spreadsheet, schedules, work-flow charting and time management, browsing, graphics/drawing tools, hypertext facilities, intelligent document management, retrieval and version control, quality tools, etc. The quality tools include an array of conceptual tools, such as cause and effect diagrams, check sheets, histograms, pareto diagrams, control charts, scatter diagrams, matrix charts, SPC, etc.

- *Level 2: Modeling and analysis tools.* Tools of this level should enable the generation, refinement, quantification and prioritization of requirements, such as QFD, objective tree, etc. Such tools are the result of modeling engineering activities, for example, geometric modeling tools such as solid modeling, surface modeling, etc. Tools may also be of product modeling types, such as STEP/Express, using feature-based or similar techniques. It also includes engineering analysis and support tools, such as FEA, mechanism analysis, mathematical calculations, intelligent CAD/CAM, wherein rules of thumb, heuristics, and parametric rules for model creation are captured.
- *Level 3: Predictive tools.* These tools are a result of design evaluation, verification and simulation tools, design synthesis and optimization, and automation of design activities based on parametric, simulations, design assistants, advisors or expert type of systems. Tools that are useful for design evaluation and verifications are: design for X-ability (reliability, serviceability, assembly, disassembly, manufacturability, testability, safety, etc.), failure mode and effect analysis (FMEA), fault-tree analysis, etc. Tools that are useful for design synthesis are: boundary searching, functional analysis, concept selection, feature-based design, design retrieval, materials selections, value engineering, production control tools, etc.
- *Level 4: Knowledge-based (KB) tools.* These tools help teams to apply manufacturing and engineering intelligence to sort out bad alternative design concepts from good ones. KB tools include: design-knowledge tools, collaborative decision-making tools for coordination, and analysis/design models. This also includes design automation based on optimization techniques, or integrated product development, expert systems (advisors), integrated product development, and manufacturing and process planning, etc. The latter includes tools such as: process capability, manufacturing process selection, materials selection, MRP, CAM tools, NC and distributed numerical control (DNC) verification tools, etc.
- *Level 5: Agent-based tools.* Such tools are used when constraints are present, when multiple knowledge sources (product and

process knowledge) are present as agents, and when conflicts occur requiring trade-off. Agent-based tools belong to distributed AI and cooperative knowledge-base fields such as cooperative expert system, CE office agents, etc. GroupWare technology replaces the conference room with the "electronic" white-board.

The levels of information enrichment and degree of cooperation are very much related. Cooperation provides the degree of confidence in the use of data, information, intelligence and knowledge. The maturity of data enrichment yields the "knowledge." Agent-based tools contain the largest amount of cooperative knowledge. The usefulness of tools depends on the collective creativity of the individual teams participating in applying the tools to problem solving. The teams' dependence on cooperative problem solving decreases as we move to lower-level tools (level 3, or level 2 . . . 1) requiring less team cooperation and more individual effort. Level 0 tools, for example, do not require any team cooperation. The applicability of a set of tools at a particular automation level depends on many other factors. The important ones are: degree of certainty, accuracy and completeness of information, and its integrity in current work environment and procedures. It is not difficult to capture the domain knowledge in most routine tasks with a high degree of confidence. Mining of rules in routine tasks is most common in levels 0 through level 2. Level 2 tools allow teams to build a modeling environment and to capture the domain knowledge before any eventual automation of the design activities can take place. The rest of the levels are more suited for specific applications such as family of parts' category involving multiple interactions or disciplines. Higher level (levels 4 and 5) tools are useful when a product, or part, is frequently redesigned for a variety of specifications. Typical examples include: different bore size and stroke length cylinders for 4-cycle, 6-cycle and 8-cycle engines, etc.

IIS functions

From any vantage point in the IIS environment, a team should be able to do one or more of the following:

- Access information or knowledge about previous product or process designs (past histories) instantly.
- Access information or knowledge about X-ability considerations for design including manufacturability, reliability, maintainability, safety, cost, quality, performance, etc.
- Access the most current state of the product or process configuration description as it is being developed within the multi-functional PDT unit.

This information, or knowledge, should be available to team members irrespective of the point of origin on the lifecycle, contributing work-group that he/she belongs to, or his/her geographical location. In addition, low-level automation tools should contain at a minimum the following features:

- An outline of the rationale behind recent changes in the design and newer versions.
- Capability to notify and record design or manufacturing process changes.
- Points to support the capture of design and implementation data.

Future of IIS

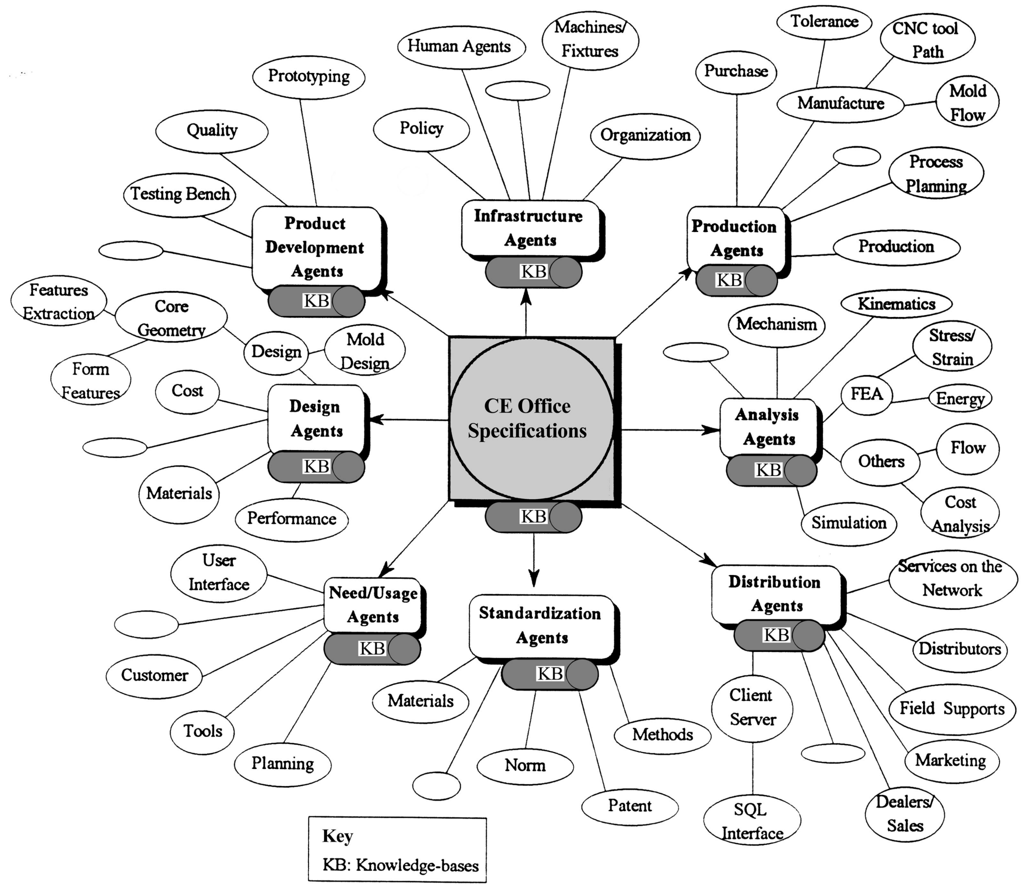
Perceived functions applicable to each division, or group, in the product's lifecycle will be formed as virtual agents of the intelligent system as shown in Figure 7. These virtual agents will aid the CE teams in performing functions through an entire product's lifecycle with accelerated speed and greater accuracy. Typical functions may range from:

- creating infrastructure (infrastructure agents); to
- establishing standards (standardization agents); to
- determining the product need or usage (need/usage agents); to
- designing the product (design agents); to
- analyzing the product (analysis agents); to
- modifying an old design (product development agents); to
- manufacturing (fabrication agents); to
- production (production agents); and finally to
- sales and marketing (distribution agents).

The examples of activities that are performed by each agent in a CE office are listed in Figure 7.

The empty circles indicate there could be more activities that belong to this agent but are omitted from Figure 7 for clarity. KB indicates an associated knowledge base for an agent. These agents will be derived from the in-house (company) experts, who have been designing and developing the products over its lifecycle. The information is organized as intelligent objects of an object-oriented database, so that their characteristics can be interrogated by other agents by sending messages to each other.

Figure 7
Population of agents in a CE office



These intelligent objects will increase the sharing of heuristics, algorithmic and derived knowledge, resulting in reduced product lead-times, improved accuracy, lower costs, and improved customer satisfaction.

The CE designers, manufacturing engineers, and other teams will be able to communicate their ideas early. The environment will contain tools and functions for cooperative problem solving. Collaborative discussions will be carried on interactively over the network. Problem-solving functions, such as critiquing of ideas, recording of design rationale, and simultaneous planning and execution of design modifications among the CE teams, will be done on a real-time basis. Computerized modules will form the nucleus for speedy communication through a concept known as software (or rapid) prototyping.

IIS is a general definition of many intelligent deployment techniques that have been introduced in recent years. These IIS techniques range from soft "prototypes," i.e. generic templates of product/process design, to so-called "hard" prototypes:

- *"Soft" prototypes*: The well-known QFD/house of quality, and IIS/house of values are examples of "soft" prototypes. IIS consists of a series of intelligent soft prototypes that can be used for product development. The DARPA and Air Force have mostly expressed interests in these "intelligent" techniques as a method of risk reduction, allowing new product concepts to be investigated earlier in their design phase by all members of an integrated product development team.
- *"Hard" prototypes*: IIS techniques for "hard" prototypes include applications in modeling "physical phenomena."

Concluding remarks

The paper describes what constitutes an intelligent information system (IIS). The most standard form of CIM commonly provides a battery of tools and systems – computer-aided X-functions (CAXs) and computer-integrated X-functions (CIXs). Where X stands for a typical lifecycle function, such as design (CAD), engineering (CAE), process planning (CAPP), manufacturing (CAM), etc. CE and KM in IIS bring forth three missing links of CIM:

1. *Intelligence*: The intelligence comes from the virtual elements of CE teams.
2. *Knowledge*: The knowledge mainly comes from information modeling (digital models), and "capturing lifecycle intent."
3. *Value system*: Value system deals with items such as culture, best industry practices for embedding a procedural discipline in CIM operations, and acceptable standards in enterprise-level communications.

It is observed working on a large CIM implementation, that the key to IIS success is understanding the obstacles to implementing CE in existing CIM processes and identifying opportunities for product, process and organization (PPO) improvements. The identification of improvement opportunities and the implementation of effective product development process control strategies can be facilitated by the systematic collection and monitoring of relevant in-process metrics.

References

Abdalla, H.S. and Knight, J. (1994), "Expert system for concurrent product and process design of mechanical parts", *Journal of Engineering Manufacture*, Vol. 208 No. 3, pp. 167-72.

Albin, S.L. and Crefeld, P.J. III (1994), "Getting started: concurrent engineering for a medium-sized manufacturer", *Journal of Manufacturing Systems*, Vol. 13 No. 1, pp. 48-58.

Althoff, J.L. (1987), "Integrated information support system, common database model subsystem", *Proceedings of Autofact*, SME, Dearborn, MI.

Alting, L. (1986), "Integration of engineering functions/disciplines in CIM", *Annals of the CIRP*, Vol. 35 No. 1, pp. 317-20.

Bajgoric, N. (1997), "Organizational systems integration: management information systems perspective", *Concurrent Engineering: Research and Applications – An International Journal*, Vol. 5 No. 2, June, pp. 113-22.

Barclay, I. and Poolton, J. (1994), "Concurrent engineering: concepts and practice", *International Journal of Vehicle Design*, Vol. 15 Nos 3-5, pp. 529-44.

Bauman, R. (1990), "CALS and concurrent engineering: business strategy or tool survival?", *Aviation Week and Space Technology*, Vol. 133, July, pp. s1-12.

Chang, T.C. and Wysk, R.A. (1985), *An Introduction to Automated Process Planning Systems*, Prentice-Hall, Englewood Cliffs, NJ.

Curran, L. (1994), "STEP bridges the way to better product modeling", *Machine Design*, Vol. 66 No. 6, pp. 137-42.

DARPA (1987), *Workshop on Concurrent Design, Key West, FL, December 1-3*, Defense Advanced Research Projects Agency, DARPA, Washington, DC.

DARPA (1988), *Workshop on Concurrent Design, Key West, FL, December 2-4*, Defense Advanced Research Projects Agency, DARPA, Washington, DC.

Dimancesen, D. (1992), *Seamless Enterprise: Making Cross-functional Management Work*, 1st ed., HarperCollins, New York, NY.

Dong, J. (1995), "Organization structures, concurrent engineering, and computerized enterprise integration", *Concurrent Engineering: Research & Applications (CERA)*, Vol. 3 No. 3, September, pp. 167-76.

Hummel, K.E. and Brown, C.W. (1989), "The role of features in the implementation of concurrent product and process design", Proceedings of the Winter Annual Meeting of the ASME, San Francisco, CA, December 10-15, in Chao and Lu (Eds), *Concurrent Product and Process Design*, DE Vol. 21, PED Vol. 36, ASME Press, New York, NY, pp. 1-8.

Jones, R.M. and Edmonds, E.A. (1995), "Supporting collaborative design in a seamless environment", *Concurrent Engineering: Research & Applications (CERA)*, Vol. 3 No. 3, September, pp. 203-12.

Kimura, F. (1994), "Virtual manufacturing as a basis for concurrent engineering", *IFIP Transactions and Applications in Technology*, Vol. 17, pp. 103-8.

Krishnan, V. (1993), "Design process improvement: sequencing and overlapping activities in product development", DSc thesis, September, Massachusetts Institute of Technology, Boston, MA.

Larsen, N.E. and Alting, L. (1992), "Dynamic planning enriches concurrent process production planning", *International Journal of Production Research*, Vol. 30 No. 8, pp. 1861-76.

Lim, B.S. (1993), "ICIMIDES: intelligent concurrent integrated manufacturing information and data exchange system – a distributed blackboard approach", in Gu, P. and Kusiak, A. (Eds), *Concurrent Engineering: Methodology and Applications*, Elsevier Science Publishers B.V., Amsterdam, pp. 135-74.

Mortel-Fronczak, J.M. van de, Rooda, J.E. and van den Nieuwelaar, N.J.M. (1995), "Specification of a flexible manufacturing system using concurrent engineering", *Concurrent Engineering: Research & Applications (CERA)*, Vol. 3 No. 3, September, pp. 187-94.

Okino, S. (1995), "Less is more", *Automotive Industries*, Vol. 175 No. 3, March, pp. 81-2.

Pawar, K.S. and Riedel, J.C.K.H. (1994), "Achieving integration through managing concurrent engineering", *International Journal of Production Economics*, Vol. 34 No. 3, pp. 329-45.

Pennell, J. and Slusarczuk, M. (1989), "An annotated reading list for concurrent engineering", *Technical Report HQ 89-034130*, Institute for Defense Analyses, Alexandria, VA.

Prasad, B. (1995a), "A structured methodology to implement judiciously the right JIT tactics", in Taylor and Francis (Eds), *Journal of Production Planning and Control*, Vol. 6 No. 6, pp. 564-77.

Prasad, B. (1995b), "CE plus CIM equals IIS: intelligent information systems", in Rangan, R. (Ed.), *Proceedings of the 1995 ASME Mechanical Engineering Congress and Exposition*, November 12-17, 1995, San Francisco, CA, ASME Press, New York, NY.

Prasad, B. (1995c), "On influencing agents of CE", *Concurrent Engineering: Research and Applications – An International Journal*, Vol. 3 No. 2, June, pp. 78-80.

Prasad, B. (1996), *Concurrent Engineering: Fundamentals*, Vol. 1, *Integrated Product and Process Organization*, Prentice-Hall, Englewood Cliffs, NJ.

Prasad, B. (1997), *Concurrent Engineering: Fundamentals*, Vol. 2, *Integrated Product Development*, Prentice-Hall, Englewood Cliffs, NJ.

Prasad, B., Wang, F. and Deng, J. (1997), "Towards a computer-supported cooperative environment for concurrent engineering", *Concurrent Engineering: Research and Applications – An International Journal*, Vol. 5 No. 3, September, pp. 233-52.

Shilke, N.A., Rhode, S.M., Fruechte, R.D. and Rillings, J.H. (1989), "GM Systems engineering center", GM Project Trilby, *Automotive Engineering*, February, pp. 167-70.

Shina, S.G. (1994), *Successful Implementation of Concurrent Engineering Products and Process*, Von Nostrand Reinhold, New York, NY.

Siong, L.B. (1993), "ICIMIDES: intelligent concurrent integrated manufacturing information and data exchange system – a distributed blackboard approach", *Concurrent Engineering: Methodology and Applications*, Elsevier Science Publishers, Amsterdam, pp. 135-74.

Stark, J. (1992), *Engineering Information Management Systems: Beyond CAD/CAM to Concurrent Engineering*, Van Nostrand Reinhold, New York, NY.

Tonshoff, H.K. and Dittmer, H. (1990), "Object instead of functional-oriented data management in manufacturing – a new approach towards CIM", *Proceedings of the International Conference on Manufacturing Systems and Environment*, pp. 443-7.

Webster, A.M. (1990), *Webster's New Collegiate Dictionary*, G. & C. Merriam Company, Springfield, MA.

Willett, K. (1992), "Networking advances will anchor concurrent engineering", *Electronic Design*, Vol. 40 No. 1, p. 6.

Zhang, H.C. and Zhang, D. (1995), "Concurrent engineering: an overview from manufacturing engineering perspectives", *Concurrent Engineering: Research & Applications* (CERA), Vol. 3 No. 3, September, pp. 221-36.