DMCM: A Knowledge-Based Cost-Estimation Tool

Norman Crowfoot, Scott Hatfield, and Mike Swank,
Xerox Corporation

Knowledge-based systems have traditionally been implemented in *vertical application areas*, which are characterized by a deep knowledge in a narrow domain. We developed and fielded a successful knowledge-based tool that is characterized by shallow knowledge of a wide range of cost-estimation problems. Our system is used to estimate piece-part manufacturing costs and has broad knowledge of many commodities and their corresponding manufacturing processes.

We believe our design, manufacturability, and cost model (DMCM) system is also unique for the following reasons:

First, the system has enabled business process changes, such as allowing cost estimation to be done at an earlier state of the design process.

Second, the cost estimate is now continually refined. The cost estimate is updated as newer and better data become available. This process is repeated at all stages of the design.

Third, a high level of integration is achieved with existing design information stores, such as a commercial computer-aided design (CAD) system and existing corporate databases.

Fourth, a conventional, commercial database is used to provide flexible data storage and a repository for persistent objects.

Fifth, accelerated implementation allowed the application to be fielded into an integrated business process.

Sixth, simultaneous implementation of multiple cost models maximizes payback in a competitive situation.

This chapter describes our experiences in developing DMCM, including the technology used to reason in the design domain, a detailed description of the application, business process results, and long-term plans.

Knowledge-Based System Competency Center

DMCM is one of several systems developed by the Knowledge-Based System Competency Center (KBSCC) at Xerox Corporation. KBSCC is a corporate-level group of knowledge practitioners located within the Corporate Information Management Group. This department was formed four years ago, motivated by a concern that although Xerox is well known for its knowledge-based system research, this important technology was not being used for solving internal business problems.

In practical terms, *competency center* means a group responsible for identifying and adapting both research work and commercial products into workable solutions for real business problems. This process is commonly called *technology transfer*. Specifically, the center does little basic research but adapts internal and external research and applies it to internal problems.

An important, basic working concept of the center is the formation of working partnerships between developers of a new application and the important users of the application. Far too often, knowledge-based system technology is introduced with a technically well-designed application program that fails to address the structure of the business process it is being introduced to. These attempts almost invariably fail; without altering the receiving structure, technology transfer is almost always doomed.

To assist the technology transfer process, we develop each application with a partnership structure; each partnership is made up of members of the competency center, the sponsoring information-management groups, and the end users who are interested in having the application delivered. This partnership approach provides increased resources for implementation of the application. It also fosters ownership of the new product with both the future users and the long-term owners of the application process.

Task Description

Xerox is primarily known as a supplier of dry-process, plain-paper photocopiers. Indeed, the company pioneered the concept of the easy-to-use, plain-paper copier with the Xerox 914 copier in 1959.

Traditional copiers focus high-intensity light onto an original, imaging the resulting image onto a light-sensitive, electrostatically charged photoconductor through a system of lamps and lenses. The latent image is then developed with particles of toner, and the resulting image is transferred to paper. A final step fuses toner particles into the paper. Additional subsystems are commonly provided to handle the input originals and the resulting output copies.

New copier technology replaces many of the traditional light lens components with digital scanning, modulated lasers, and networked devices. Even with this modern approach, internal image-processing and paper-handling steps remain much as they have always been.

Because of the interactions of the many systems within a modern copier, their design rivals the complexity of other design tasks. Modern copiers use a wide variety of interconnected technologies, including chemical, mechanical, optical, thermal, pneumatic, electric, and software.

Ever-increasing market competition is a fact of life in our business. The process of designing and manufacturing low-cost, quality products is essential to the financial success of Xerox. As a corporation, we discovered more than 10 years ago a marked cost differential between products designed and manufactured by Japanese companies and those produced by U.S. production facilities. Analysis of these differentials demonstrated that early cost estimations are essential to cost management in that a large part of the total eventual cost is designed in at an early stage in the product development cycle.

The Original Design Process

Our design process at Xerox is much like those used in other large engineering organizations. A Product Delivery Team (PDT) is organized and charged with producing a design for a specific new copier or related family of products. The PDT chief engineer is responsible for the product delivery as well as meeting functional, quality, and cost requirements. PDT oversees the design until it is well into production.

Many specialists participate in the design of a new product. These roles are the traditional ones, such as chief engineer, cost engineer, design engineer, designer, and manufacturing engineer. These interactions are diagrammed in figure 1.

Role of Cost Estimation

The actual cost estimations for a designed part are performed by a small number of individuals (approximately 100 across the corpora-

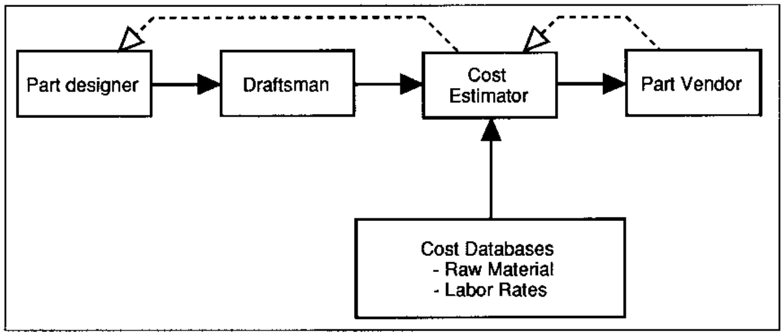


Figure 1. Original Design Process.

tion). These cost engineers match the classic profile for a domain expert. Most of them have many years of experience and specialize in a few commodities, having worked previously in such areas as manufacturing, purchasing, or the design community or as external suppliers.

Further, because the estimation resource is scarce, estimation is typically done late in the design cycle, if at all. There is no time to iterate the design with design alternatives by the time the design reaches the cost engineer.

This environment is the reason that DMCM was developed and deployed. A major design goal was to make DMCM applicable throughout the product development life cycle. In the early stages of product formulation (the concept phase), the tool provides coarse cost estimates for components and supports cost and bill-of-material roll up. In the later stages, as the design matures, the tool evaluates designs for conformance to established design standards, generates detailed cost estimates for component designs, and suggests specific external vendors based on vendor-specific knowledge.

Previous Cost Models

The use of knowledge-based systems in the engineering domain is not new. A number of general approaches to the problem are described in Brown and Chandrasekaran (1986), Dixon and Dym (1986), Dixon and Simmons (1984), Dym (1987), and Forbus (1988). Furthermore, a number of actual systems are described in Hatfield et al. (1987); O'Brien et al. (1989); and Mittal, Dym, and Morjaria (1986).

A study of some of the shortfalls of similar systems was conducted during the feasibility study phase of the DMCM project. Of course, there is the obvious problem that design is a complex domain and that manipulating geometric information is still a difficult task for a computer system. Some of the reasons that were identified during this study are described in the following paragraphs:

Failure to integrate existing databases: Many expert systems are implemented on stand-alone machines that do not communicate with existing company data resources. Although this stand-alone implementation might have been forgivable in an age when AI projects were built on specialized hardware using languages such as Lisp, it is no longer acceptable for production-quality systems.

In many systems, input data are entered through a keyboard or as one-time bulk loads of data from an existing database. In the former case, the labor intensity of the direct key entry can cause the data to become stale. In the latter case, the viability of the data is a function of the transfer cycle. In both cases, the correctness of the data erodes with time.

Such systems become outdated, and users are unlikely to spend the energy to audit data for obsolescence. Unintegrated systems are especially onerous in a domain as dynamic as cost estimation, with commodity prices constantly fluctuating.

Failure to include part geometry: Because of the stand-alone nature of many previous attempts, all part geometry information was reentered by the knowledge-based system user or was not used at all. Frequently, this duplicated information already existed in traditional CAD databases. This duplication introduces errors, additional work, and inconsistencies.

Failure to preserve the reasoning history of decisions: Reasoning systems fielded to date typically do not retain the reasoning processes that lead to their previous conclusions. Thus, one is prevented from auditing these results, comparing design decisions, and building a knowledge base of designed parts. Because of the transient nature of most commercial reasoning tools, reasoning history is lost after the estimation session is ended. Something as permanent as a database is generally not considered for reasoning patterns.

Failure to form partnerships with information-management departments: Primarily because of their stand-alone and experimental nature, previous knowledge-based system applications are not integrated into the working information-management infrastructure. Because the personnel in information-management organizations know the location and connections to databases and other information sources that are required, we included these personnel in the DMCM project at an early stage. They enhanced the success rate of fielding DMCM. Early partnership involvement with information management also allowed us to plan for long-term maintenance and support activities long before these critical processes were needed.

Too detailed knowledge of a functional area: Another concern was the great depth of knowledge encoded in many knowledge-based systems. The engineering mind set finds it more satisfying to build a wonderful system for design paper-feed rollers than to solve the general problem of parts cost estimation. Much research effort has been expended in developing these deep solutions, but few successful systems have resulted (Gael and Pirollo 1989; Gray 1988; Hoeltzel, Chieng, and Zissimides 1987; Mittal and Araya 1986; Mittal, Dym, and Morjaria 1986; Mostow 1985; Tat Chan and Paulson 1987). We believe there are systemic reasons for this failure to produce useful systems. Two things happen when using these deep reasoning approaches: (1) a few problem areas are analyzed and mechanized to the exclusion of others and (2) much time is expended in getting the model exactly right because of the detailed nature of the project. By the time the model is correct, the problem or the technology has changed.

With DMCM, we consciously chose to incorporate shallower knowledge: It requires less creativity, is more invariant, is more likely to be useful over a long period of time, and applies to a wider range of applications.

As additional design knowledge is incorporated into DMCM, we will evaluate the useful lifetime of this knowledge. In many cases, the technologies used in successive generations of products change dramatically, greatly reducing the utility of static knowledge-based systems. One can easily end up having invested two years in modeling knowledge with a three-year life expectancy.

Application Description

This section describes the DMCM application in terms of the delivery platform, the organization of the software shell and the structure of the knowledge bases used. Early results are described in Hatfield and Crowfoot (1990).

Delivery Platform

When DMCM development started, a related effort was under way within Xerox to standardize an engineering workstation. Initial requirements were unclear except that this software run on UNIX and be an open system. DMCM was developed to accommodate almost any such platform.

Hardware: DMCM was implemented for the Xerox engineering workstation environment: a set of Sun SPARC workstations connected through TCP-IP, SNA, and XNS networks to centralized database servers.

Software: Standard, commercial vendor-supplied software was used

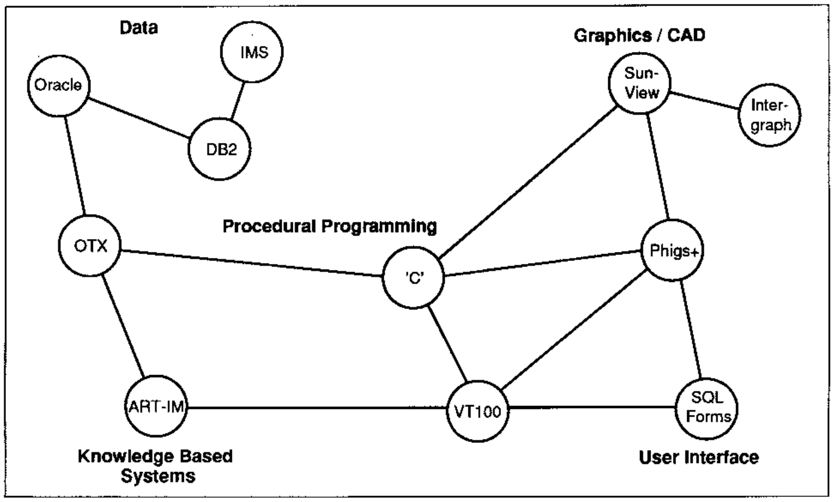


Figure 2. DMCM *Shell Organization.*

where possible. DMCM is currently implemented to run on three graphic user interface desktop environments: SUNVIEW; OPENWINDOWS, Sun's implementation of OPENLOOK; and a classic teletype interface for backward compatibility. See the section on the user interface for details.

Shell Organization

Like many knowledge-based system applications, DMCM is built using a shell that we constructed. Simply stated, a *shell* is a set of building blocks that facilitate presentation and development of the knowledge base itself. See figure 2 for an overview of the tool components that make up the DMCM shell.

Knowledge Base Structure

DMCM accommodates a variety of knowledge models partitioned by commodity. Examples of commodities include plastic injection-molded parts, sheet metal parts, and electromechanical devices. Each knowledge model is a separate, loadable module.

The primary mechanism used to describe costing knowledge is ART-IM (automated reasoning tool for information management from Inference Corporation) rules. Each commodity is costed by a set of some 100 to 200 rules. Constant tabular information, such as press capacities, are kept in facts. Alterable cost information, such as commodity costs, are maintained in a remote relational database to be fetched as needed.

Several ART-IM schemas are used to hold deduced knowledge during the course of a cost estimation. As costs are calculated, they are written to the relational database. This approach allowed us to provide knowledge about a number of commodities in a modular way, strictly separating knowledge of how a commodity functions from its specific physical attributes and current cost.

The types of knowledge that the system can deduce include the following:

> How much force can a 2-inch plastic gear 1/4-inch thick typically tolerate according to Xerox design specifications?
>
> How much money do I save if I reduce the material thickness of this part?
>
> How large a machine press will be required to form this part? What will the cycle time be? How much might a supplier charge to use such a press?
>
> Which supplier, A or B, would most effectively produce this shaft? What should it cost from supplier A? From supplier B?

Much of this design knowledge is represented as rules to calculate forces, quantities, and clearances. This knowledge was copied from the preferred design practice documentation within Xerox. The cost-comparison knowledge is obtained from running the calculations several ways and taking the minimum.

Little knowledge of the part's eventual use is considered because this information is generally not necessary to derive a cost. DMCM also does not currently reason over collections of parts. Examples of knowledge that is not included in the system are as follows:

> Given an input torque and an output torque, consider all the possible approaches to gearing this drive assembly.
>
> Will this base plate support the weight of the system? Is the flexure small enough to maintain the drive rollers in close enough tolerance?
>
> How will my gear function within my design? Which way will it rotate? Have I allowed enough clearance with the meshing part?

Within each commodity, we built knowledge bases that contain logic in each of the following categories:

Geometric reasoning: Because the geometric information present in our CAD drawings is limited to wire-frame stick figures, rules are added to the knowledge bases to convert simple mouse picks into information. Examples of this conversion include determination of various part dimensions, part volume, and number of holes (see Computer-Aided Design Integration).

Cost estimating: DMCM cost estimates include raw material rates;

manufacturing rates; tooling cost; duty, insurance, and freight; and all the conversion coefficients that must be applied (for example, currency conversion).

Manufacturing process specification: To develop accurate cost estimates for a component, DMCM must understand the manufacturing process behind the component. Here, we estimate what it will take to manufacture a part in terms of raw materials, labor rates, and cycle times. These output can form the basis of the manufacturing engineer's process plan for making a part. They might also support capacity planning.

Design standards: Some of the ways that DMCM aids in disclosing design standards include knowledge of how to select appropriate materials for a given application, what standard dimensions are, and what standard finish specifications are.

Cost-saving opportunities: DMCM supports several modes of identifying cost-saving opportunities. First, the DMCM user can alter one or more parameters and rerun the cost estimate in a what-if mode. Second, the user can request a range of external vendors to be considered. DMCM automatically performs the what-if analysis and returns the minimal cost supplier. Finally, DMCM suggests alternate choices whenever it determines a lower-cost alternative. Some of these potential what-if opportunities involve lowering the cost of the part, such as by suggesting thinner materials.

Application Use and Payoff

The development and fielding of DMCM has yielded both direct benefits and indirect benefits from changes in the business process. Changes because of DMCM are happening rapidly because the process of purchasing parts is changing as a direct result.

Benefits

A cost-benefit study was performed during the DMCM concept phase. This detailed study identified many of the specific cost benefits that are now being realized. These benefits are both tangible, measurable cost savings and intangible alterations to the way Xerox does business.

The tangible benefits include (1) labor savings provided by improved efficiency in actually developing cost estimates (the demonstrated time savings for cost engineers is 50 percent, which works out to $1.1 million annually); (2) more efficient designs in terms of material use, manufacturing times, and tooling costs, yielding direct cost reductions; (3) higher-quality products as a result of better conformance to internal design standards; and (4) shorter product development cycles because of fewer redesigns and quicker cost estimations.

Tangible savings in the last three categories are measured by estimating how much closer to the industry benchmark Xerox is able to make its parts. Although it varies by commodity, the study concluded that 40 to 50 percent of the gap between Xerox and the industry benchmark could be removed with DMCM. Multiplied by the amount that Xerox spends on each commodity yields the total annual savings. Naturally, this savings is a softer value, as well as being sensitive, but it is in excess of $20 million annually.

In addition, a number of business processes have improved as a result of implementing the DMCM system. In particular, some of the benefits are (1) working inventory reductions based on using the system to drive toward standards and common parts early in the design cycle, (2) improved use and awareness of design for manufacturing considerations by disclosing manufacturing processes and requirements while the design is still on the drafting board, and (3) improved processes for source acquisition and inventory management by disclosing to the product delivery organization objective measures of cost (eventually, the DMCM cost equations will include such items as logistics costs, service costs, refurbishment costs, facilities costs, and design costs).

Process Change Agent

DMCM was sold to management based on the direct time savings it afforded the cost engineering community. However, it also benefits the whole organization by changing the purchasing cycle and the way in which piece parts are acquired. We consider DMCM to be a vital agent, creating process change. These benefits can be seen as a Trojan horse in the sense that these process changes might only be recognized after the user community has been sold by the direct benefits.

Figure 3 diagrams the desired *end goal*, or ideal state of the design process and the part-costing process. (See figure 1 for the original process.) One of the major process changes we observed with the introduction of DMCM is the increased time spent in contact between PDTs and external part vendors.

Because cost engineers have additional time, they are spending it to get better and more current cost estimates. DMCM output sheets are commonly carried into price negotiation sessions with external part vendors.

A major effort is under way to produce custom cost estimate reports for each major Xerox part supplier based on vendor-supplied costing

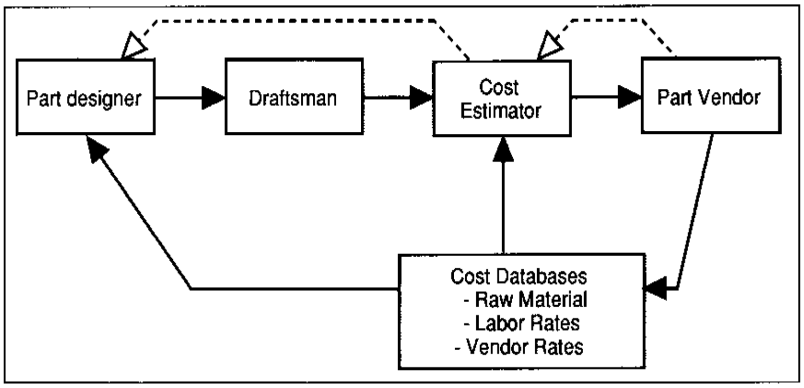


Figure 3. Altered Design Process.

information. In this manner, we can help each of our suppliers reach a higher level of quality and cost effectiveness.

These changes in the way Xerox does business are not an accidental by-product but, rather, the reason for the application in the first place. We believe strongly in planning for and expecting these process changes because they are so often associated with the acceptance of an application. It is possible and, probably, likely that these process changes will deliver more improvement than the direct benefits.

A sample of a DMCM printed report is shown in figure 4. Notice that the detailed cost estimates are given with respect to a selected vendor against a benchmark goal.[1] DMCM is able to cost an estimate against a selected set of possible outside vendors, selecting the particular vendor best capable of cost effectively producing the part.

User Feedback

The current DMCM user population within Xerox is approximately 25 cost engineers. Serious rollout began in January 1992 with the inception of monthly user classes. About 100 cost estimates are done each week with DMCM.

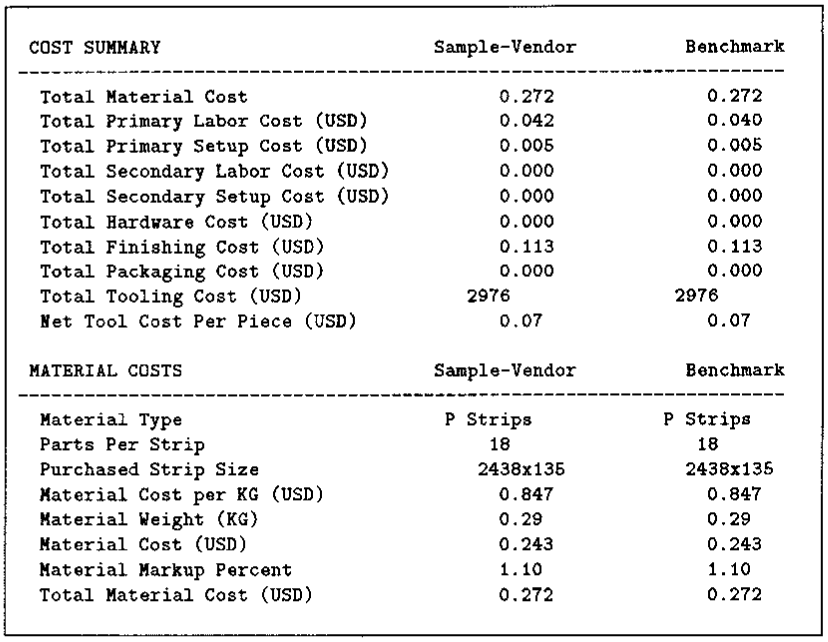
User feedback to DMCM is positive. In addition to the benefits observed by management, end users like the system for the following reasons:

First, access to company data is now easier.

Second, the system requires many times fewer input than prior models by using CAD files directly.

Third, they have the ability to automatically locate design violations.

Fourth, the need to constantly refer to manuals for design standards is eliminated.



```
COST SUMMARY                                 Sample-Vendor       Benchmark
-----------------------------------------------------------------------------
  Total Material Cost                            0.272             0.272
  Total Primary Labor Cost (USD)                 0.042             0.040
  Total Primary Setup Cost (USD)                 0.005             0.005
  Total Secondary Labor Cost (USD)               0.000             0.000
  Total Secondary Setup Cost (USD)               0.000             0.000
  Total Hardware Cost (USD)                      0.000             0.000
  Total Finishing Cost (USD)                     0.113             0.113
  Total Packaging Cost (USD)                     0.000             0.000
  Total Tooling Cost (USD)                    2976              2976
  Net Tool Cost Per Piece (USD)                  0.07              0.07

MATERIAL COSTS                               Sample-Vendor       Benchmark
-----------------------------------------------------------------------------
  Material Type                                P Strips          P Strips
  Parts Per Strip                                18                18
  Purchased Strip Size                          2438x135          2438x135
  Material Cost per KG (USD)                     0.847             0.847
  Material Weight (KG)                           0.29              0.29
  Material Cost (USD)                            0.243             0.243
  Material Markup Percent                        1.10              1.10
  Total Material Cost (USD)                      0.272             0.272
```

Figure 4. Portion of Sample Vendor Report.

Fifth, they have the ability to generate hypothetical designs easily and then get objective metrics back (design cost) to evaluate alternatives.

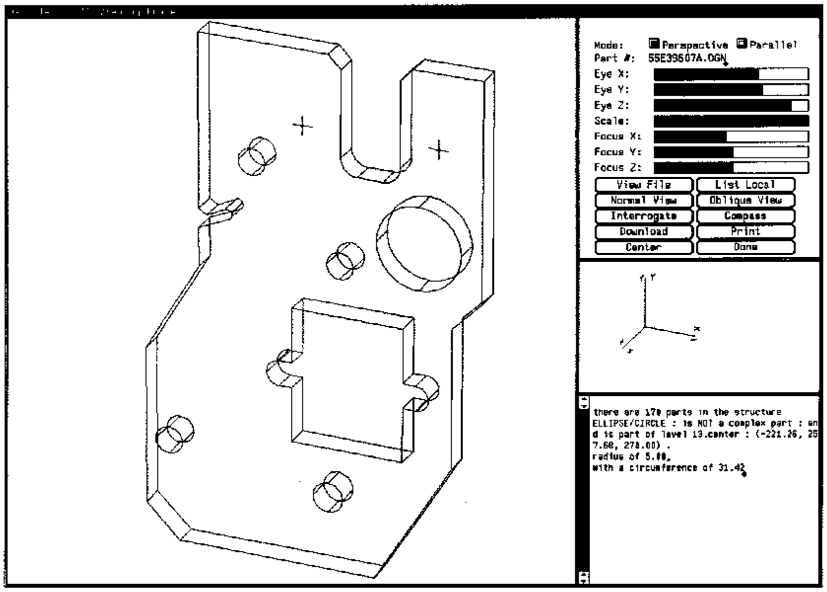
Based on this feedback and progress to date, we anticipate that our full benefits case will be achieved.

Application Development and Deployment

DMCM was developed with four knowledge engineers working over an 18-month time frame. There were three software development phases: (1) implementation of the application shell, (2) implementation of the three initial knowledge bases, and (3) application field testing and rollout to the field.

Implementation Issues

Shell Language: The C language was chosen for the DMCM shell for its ability to integrate with the reasoning system, the UNIX environment, and the graphics tool kit. An additional benefit is the wide degree of portability that the C language affords. C++ was not chosen because of its immature state of development. However, nothing in the DMCM im-



Mode: ■ Perspective ☐ Parallel
Part #: 55E39607A.0GN
Eye X:
Eye Y:
Eye Z:
Scale:
Focus X:
Focus Y:
Focus Z:

```
there are 170 parts in the structure
ELLIPSE/CIRCLE : is NOT a complex part : an
d is part of level 13.center : (-221.26, 25
7.60, 270.00) .
radius of 5.00,
with a circumference of 31.42
```

Figure 5. *DMCM Shell Integration with Computer-Aided Design.*

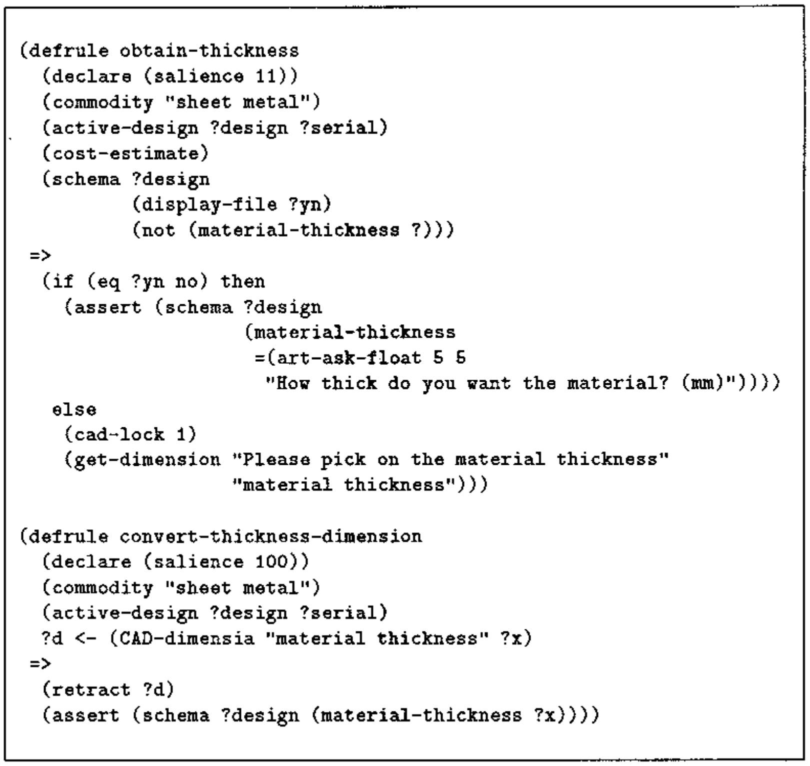
plementation precludes going to C++ at a later date.

Database: ORACLE is used for the relational database. The graphics and geometry presentation are implemented with Sun's PHIGS+ software.

Reasoning system: ART-IM was chosen for the AI reasoning system (Inference 1988) because of its ability to integrate with existing applications as well as the ease of extending the reasoning environment with new C-coded primitives. Such operators were added to allow remote database access; move data files through XNS, TCP-IP, and SNA connections; and support persistent objects.

AI language integration: To facilitate development of a production-quality knowledge-based system and ensure its long-term maintainability, we chose not to develop an internal language to represent DMCM's knowledge. Instead, we use a standard tool in our center, ART-IM. Using a standard tool provides our knowledge engineers with the greatest productivity and simplifies the maintenance task for the maintainers in traditional information-management departments. Use of a standard tool also affords increased flexibility in moving knowledge engineers between projects.

This tool has most of the features that are needed in an AI tool. The features we feel are important include a full-featured rule syntax for



```
(defrule obtain-thickness
  (declare (salience 11))
  (commodity "sheet metal")
  (active-design ?design ?serial)
  (cost-estimate)
  (schema ?design
          (display-file ?yn)
          (not (material-thickness ?)))
 =>
  (if (eq ?yn no) then
    (assert (schema ?design
                    (material-thickness
                     =(art-ask-float 5 5
                      "How thick do you want the material? (mm)"))))
   else
    (cad-lock 1)
    (get-dimension "Please pick on the material thickness"
                   "material thickness")))

(defrule convert-thickness-dimension
  (declare (salience 100))
  (commodity "sheet metal")
  (active-design ?design ?serial)
  ?d <- (CAD-dimensia "material thickness" ?x)
 =>
  (retract ?d)
  (assert (schema ?design (material-thickness ?x))))
```

Figure 6. Example of Geometric Reasoning Rules.

specifying rules and constraints; easy integration with the C language (a requirement for a system that is highly interconnected); a powerful object representation to support modeling of material families, designs with their features, and the like; and commercial support that includes help, documentation, and training for system maintainers.

Integration Issues

Integration with the existing infrastructure proves to be complex and challenging, even for a simple costing system. However, we believe that integration is extremely important and is fundamental to the acceptance of the system. The areas DMCM had to integrate with are detailed in the following subsections.

CAD Integration. The majority of our hardware designs are stored as simple three-dimensional, wire-frame drawings in a format described by our primary CAD vendor (Intergraph). As a part of the DMCM shell, functions are included to copy these drawings from a remote drawing

```
(defrule pl-determine-press-prices
  (declare (salience 10))
  (pl-phase price $?)
  (active-design ?design ?serial)
  (schema ?design (economics ?ver) (manuf-site ?src))
  =>
  (uiprint "\n\nGetting press rates")
  (bind ?sql (string-append
      "select machno, "
      "vnd_cycle_labor_rate + vnd_cycle_burden_rate "
      "from xp_lr_detail where "
      "machno>='HB' and machno<'HC' and "
      (sprintf "ver='%s' " ?ver)
      (sprintf "and src='%s';" ?src)))
  (query-facts ?sql "pl-press-rate-raw"))

(defrule pl-determine-press-convert-to-string
  (declare (salience 100))
  ?fact <- (pl-press-rate-raw ?id ?rate)
  =>
  (if (symbolp ?id) then
    (retract ?fact)
    (bind ?id-new (string-append ?id))
    (assert (pl-press-rate-raw ?id-new ?rate))))
```

Figure 7. Example of ART-IM *Rules Reading Database.*

archive to the workstation. The CAD drawing images are then displayed on the workstation. Many questions posed by the knowledge bases can be answered directly from the display because dimensional information is available by selecting features of the drawing with the mouse. Figure 5 depicts the CAD interaction frame that is part of the DMCM shell.

As mentioned previously, the geometric information that we obtained from the CAD drawings is largely limited to three-dimensional, stick-figure line segments. Additional information is required to reason about abstract concepts, such as width, thickness, volume, and placement. These more abstract concepts are derived from asking questions of the user and having the user answer specific questions by pointing to the appropriate feature with the mouse. The portion of the knowledge base shown in figure 6 determines the stock thickness required for a sheet metal part. The rule obtain-thickness fires when the material-thickness slot is not present within the current design schema. If a display file is not active, the user is simply prompted for the thickness value.

If a CAD display file is available for the part, the second portion of the rule requests the drawing window to be shown, and the user is prompted to click on the material thickness with the mouse. When the dimension is indicated, the shell code asserts the value as the labeled fact (CAD-DIMENSIA "material thickness" <xxx>). The second rule, convert-thickness-dimension, retracts the fact and converts the dimension information into the required slot value.

Work is under way to extend DMCM's reasoning about the parts that are presented. Research such as Cunningham and Dixon (1988), Gero (1985), Kapur and Mundy (1988), Morjaria, Mittal, and Dym (1985), and Wu has documented the importance of using geometric information in the design and analysis domains, citing a number of approaches.

Database Integration. From its earliest inception, DMCM was strongly connected with existing databases, both as a reader and as a writer.[2] The relational database server technology chosen for DMCM is ORACLE. Functions were installed in the shell to enable ART-IM rules to easily create SQL queries.

Figure 7 demonstrates how this task is accomplished. The rule pl-determine-press-prices forms a query by building a query string. This string is passed to a routine that returns each row satisfying the query as a separate fact. The second rule, pl-determine-press-convert-to-string, converts each returned press into a more useful internal form.

With a large, multinational corporation, a number of existing databases must be accessed. Grouped then under the general topic of databases were the following interface challenges:

Corporate databases: Connections were needed to approximately one dozen existing databases. Most of these databases are treated in a read-only manner. These databases are of several different styles, including IBM's IMS-DC, DB2, and ORACLE, and run on several different platforms.

DMCM-specific data: Because we controlled the placement and format of our project data, we standardized on retaining DMCM-specific data in ORACLE files resident on a departmental UNIX processor. These relational ORACLE tables hold cost information, intermediate working data, and the state of ART slots (see Implementation Issues) after a cost estimate is complete.

Persistent objects: ART-IM was augmented with persistent objects to support the reasoning process, allowing DMCM to reason over design objects later from any workstation sharing the department's database. A number of browsing activities are also supported (Nguyen and Rieu 1987; Waldron and Chan 1988). We found that moving schemata[3] between ART-IM's virtual memory and rows in the ORACLE database was a practical solution because it promoted the reuse of objects and provided multiple use, locking, and rollback and recovery of a commercial database. Our technique supports the sharing of design objects across multiple designs. All design decisions are recorded in the design schema slots so that they can be preserved in the database, reviewed, and reused. These design decisions serve as an important history for each part, preserving the initial design choices.

Query Facility: We found that working with a commercially available database such as ORACLE enhanced our productivity as developers. Tools such as the forms package SQL*FORMS reduced the effort required to develop routines to browse and edit many of the records used by a complex knowledge base, such as physical material properties, standard labor rates, and currency conversions. Also available are productive approaches to generating the standard reports that users request.

Integration with IMS-DC. The principal output of DMCM is the estimated part cost. This information is entered into an existing, MVS-resident IMS database.

Early in the development of DMCM we asked the existing information-management support group to estimate the programming resources necessary to construct a remote interface into the MVS-IMS database. When this estimate exceeded one-third of the DMCM programming budget in both time and expense, we took another tack.

Because of the rapid development time frame of DMCM, we elected to interface to existing IBM 3270 transaction screens programatically. We constructed an emulator for the IBM 3278 terminal, using SUN LINK's program interface to the channel data stream. This emulator is used to log DMCM into a remote IMS-DC machine, initiate the proper transaction, complete the screen forms, and transmit the results back to DMCM with no change to existing host programs.

User Interface

The initial graphic user interface desktop environment for DMCM was SUNVIEW. We converted DMCM to operate in the X WINDOW–compliant OPEN WINDOWS environment. OPEN WINDOWS is Sun's implementation of OPENLOOK. Sun's XVIEW software library is used for its widgets. MOTIF was not chosen because of licensing considerations.

In addition, a classic teletype interface is supported for backward compatibility with older, character-mode terminals. Graphics are not supported in this teletype mode.

In many systems, a large fraction of the total development time is spent constructing the user interface. The decision was made to devote a short period of up-front time to build, debug, and document a library of calls that provide most of the DMCM user interface.

From any knowledge base or any C-language routine, the knowledge programmer can request that information be displayed to the user or obtained from the user. Furthermore, these calls have been developed to be knowledgeable about the desktop environment the user is operating in and to perform the appropriate behavior.

Additional environments can then be supported in an incremental fashion by coding a handful of user interface primitives in the new environment. Today, DMCM supports SUN VIEW and character-oriented terminals. In the near future, it will support OPEN WINDOWS PEX (X WINDOWS) with the same primitives.

Model Validation

Initial validation of the cost models was performed during a two-month field test. During this period, a set of six cost engineers estimated their designs both with DMCM and their previous methods. Many adjustments were made to the models following this test.

Because many Xerox parts are purchased externally, the cost engineering group adjusts their parameters on an ongoing basis when external bids are received. In the case of specific external suppliers, part buyers obtain shop labor rates and other cost information from the specific vendor, with the provision that this information is not to be shared with competing vendors.

Deployment

The proof of any application is in its deployment and acceptance by the user population. DMCM was initially released to the field in June 1991. Use during the first six months was primarily by the original expert departments. Three fractional, enhancement releases followed during the remainder of the year.

As mentioned earlier, the current DMCM user population is approximately 25 engineers. The estimated user population for 1992 is 60 users, with growth to 200 expected in 1993. Geographic distribution of use centers initially on Xerox domestic manufacturing centers in New York and California. International users will be located in Holland and Great Britain.

Initial rollout was hampered by the unforeseen relocation of our commodity experts into a different office facility. This move hampered the installation of network-connected workstations because of an older data communications infrastructure. This problems is one of the many real problems that were encountered and overcome.

Maintenance and Continuing Support

We feel that it is easy for a highly skilled group of bright practitioners to produce a showy concept prototype. Getting an application installed, documented, and accepted by a user community, as well as integrated into the corporation, is more difficult, which is the heart of the technology transfer problem.

Continuing Partnerships

A major theme in the development effort was continuous partnership relationships with our customers and cosuppliers. Of prime importance to us has been our cosuppliers, the information-management staff of the Xerox Product Design and Manufacturing Division.

The relationship holds the key to the technology transfer process for several reasons: Most importantly, the information-management groups manage almost all the data DMCM needed, both for its input and its output. The information-management groups also control the existing mainframe programs wrapped around these data. Information-management groups also house people wanting to develop their knowledge-based system skills. Thus, working in a continuing, close partnership with the cosuppliers in such areas as maintaining and extending DMCM into additional commodities benefits all concerned.

Resource Management. Convincing traditional information-management line managers that new resources should be allocated to these ends, getting the line items into continuing budgets, and holding these commitments against the constant buffeting of head count and budget issues are continuing problems. In a dynamic company, the personnel turnover in management positions almost ensures that a completely new cast reviews each year's budgets.

In the case of DMCM, staffing the positions was a problem because appropriate skills were lacking in the internal programmer population. External hires were placed on the project, worked on the development of the code, and then were transitioned into the information-management organization. In this position, they are (1) maintaining DMCM, (2) extending it to additional cost models, (3) continuing the product rollout, and (4) serving as the basis for a knowledge-based system competency center within the information-management organization.

Follow-On Work

The information-management knowledge-based system group continues to work closely with the corporate competency center to develop additional knowledge-based systems. Because these individuals were a part of the original development team, this work follows naturally. One follow-on project is already on the drawing board: The manufacturing operations adviser (MOA) is a system that picks up where DMCM leaves off, adding manufacturing knowledge across multiple parts to more effectively schedule a floor of manufacturing machinery.

Plans are being made to extend DMCM with additional knowledge models to cost additional commodities important to Xerox. The planned commodities include wiring harnesses and printed wire board assemblies.

Acknowledgments

We would like to acknowledge a few of the people who contributed to the development of DMCM: Mark Maletz, Lynn Heatley, Dhimant Master, Jeff Leonard, and Sergio Rubio of the Knowledge-Based System Competency Center and Vince Romano and John Beasman of the Xerox Materials Management Group were all instrumental to our success.

Notes

1. As part of ongoing quality practices, Xerox maintains a measure of industry's best practices and attempts to achieve these measures.

2. In fact, this integration with existing corporate data assets helped gain the acceptance of DMCM.

3. For the purist, *schemata* is the plural form of schema, ART-IM's term for an internal, slot-oriented data structure, or frame. A schema is roughly equivalent to the object in an object-oriented programming environment.

References

Brown, D., and Chandrasekaran, B. 1986. Knowledge and Control for a Mechanical Design Expert System. *IEEE Computer:* 19(7): 92–100.

Cunningham, J., and Dixon, J. 1988. Designing with Features: The Origin of Features. In Proceedings of the 1988 ASME International Computers in Engineering Conference and Exhibition, 237–243. New York: American Society of Mechanical Engineers.

Dixon, J., and Dym, C. 1986. Artificial Intelligence and Geometric Reasoning in Manufacturing Technology. *Applied Mechanics Reviews* 39(9): 1325–1330.

Dixon, J., and Simmons, M. 1984. Computers That Design: Expert Systems for Mechanical Engineers. *Computers in Mechanical Engineering.*

Dym, C. 1987. Issues in the Design and Implementation of Expert Systems. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 1(1): 37–46.

Forbus, K. 1988. Intelligent Computer-Aided Engineering. *AI Magazine* 9(3): 23–36.

Gael, V., and Pirollo, P. 1989. Motivating the Notion of Generic Design with Information-Processing Theory: The Design Problem Space. *AI Magazine* 10(2): 18–36.

Gero, J., ed. 1985. *Knowledge Engineering in Computer-Aided Design*. Amsterdam: North-Holland.

Gray, M. 1988. An Intelligent Design Machine: Architecture and Search Strategies. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 2(2): 105–122.

Hatfield, S., and Crowfoot, N. 1990. Knowledge-Based Systems in Cost Estimating and Design. In Proceedings of the Autofact '90, 21-1–21-10. New York: American Society of Mechanical Engineers.

Hatfield, S., Tuchinsky, P., and McWhortery, R. 1987. BRAKES: An Expert System for Engineers. In Proceedings of the Expert Systems for Advanced Manufacturing Technology Conference.

Hoeltzel, D.; Chieng, W-H.; and Zissimides, J. 1987. Knowledge Representation and Planning Control in an Expert System for the Creative Design of Mechanisms. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 1(2): 19–137.

Inference Corporation. 1988. ART-IM, Integrated Reasoning Tool for Information Management, DPU-C15-R2-AA, Inference Corporation, Los Angeles, California.

Kapur, D., and Mundy, J., eds. 1988. *Geometric Reasoning*. Cambridge, Mass.: MIT Press.

Mittal, S., and Araya, A. 1986. A Knowledge-Based Framework for Design. In Proceedings of the Fifth National Conference on Artificial Intelligence, 856–865. Menlo Park, Calif.: American Association for Artificial Intelligence.

Mittal, S.; Dym, C.; and Morjaria, M. 1986. PRIDE: An Expert System for the Design of Paper-Handling Systems. *IEEE Computer,* 19(7): 102–114.

Morjaria, M.; Mittal, S.; and Dym, C. 1985. Interactive Graphics in Expert Systems for Engineering Applications. In Proceedings of the 1985 ASME International Computers in Engineering Conference and Exhibition. New York: American Society of Mechanical Engineers.

Mostow, J. 1985. Toward Better Models of the Design Process. *AI Magazine.* (6): 44–57.

Nguyen, G., and Rieu, D. 1987. Expert Database Concepts for Engineering Design. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 1(2): 89–101.

O'Brien, J.; Brice, H.; Hatfield, S.; Johnson, W. P.; and Woodhead, R. 1989. The Ford Motor Company Direct Labor Management System. In *Innovative Applications of Artificial Intelligence*, eds. H. Schorr and A. Rappaport, 333–347. Menlo Park, Calif.: AAAI Press.

Tat Chan, W., and Paulson, B. 1987. Exploratory Design Using Constraints. *AI EDAM* 1(1): 9–71.

Waldron, M., and Chan, C. 1988. Object-Oriented System for Component Selection. In Proceedings of the 1988 ASME International Computers in Engineering Conference and Exhibition, 7–62. New York: American Society of Mechanical Engineers.

Wu, Y. 1987. Automated Design and Sketching of Mechanisms Based on Specified Design Requirements by Employing Expert System Methodologies. Ph.D. diss., Dept. of Mechanical Engineering and Applied Mechanics, Univ. of Rhode Island.