Design Tradeoffs
For Advanced Manufacturing
Using Cost Modeling

Eva Moy

S.B. Mechanical Engineering
Massachusetts Institute of Technology, 1995

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of
Master of Science in Mechanical Engineering
at the Massachusetts Institute of Technology
February 1998

Signature of author ....
Eva Moy
Department of Mechanical Engineering
3 October 1997

Certified by ....
Timothy G. Gutowski
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by ....
Ain A. Sonin
Chairman, Department Committee on Graduate Students

Design Tradeoffs
For Advanced Manufacturing
Using Cost Modeling

Eva Moy

Submitted to the Department of Mechanical Engineering
on 3 October 1997 in partial fulfillment
of the requirements for the degree
of Master of Science in Mechanical Engineering

Abstract

Cost modeling is an activity that is easy to estimate on the fly, but difficult for a company to standardize both the procedures and the data. In the project, a detailed process plan is created from a physical description of the part, the production requirements, the material properties requirements, and internal rules which govern the interrelationships. The final cost also takes into account tooling and equipment requirements based on the above. The software implementation, using Cognition Corporation's *Cost Advantage* program, groups manufacturing steps by commonalities across various manufacturing technologies.

Thesis supervisor: Timothy G. Gutowski
Title: Professor of Mechanical Engineering

Acknowledgements

I would like to thank the following people for their friendship, advice, and encouragement:

Laboratory for Manufacturing and Productivity

- Prof. Timothy Gutowski, Professor of Mechanical Engineering, MIT
- Prof. Susan Mantell, University of Minnesota, Visiting Faculty at MIT
- Karuna Mohindra, Administrative Assistant, MIT

Draper Laboratory

- Robert Faiz, Research Scientist
- Mark Mello, Research Scientist

Cognition Corporation

- Kevin Sullivan, Software Engineer
- Dave Cronin, Sales
- Kenn Goutal, Customer Support

Friends

- Imran Haq, Nandini Iyer, Ali Merchant
- Maria Raposo and the rest of the *Graduate Student Council*
- Saul Blumenthal, Cristian Gonzalez, Stacey Blau, and the rest of *The Tech*

Table of contents

Introduction

The field of manufacturing includes expertise from many different fields. The title of this thesis, “Design Tradeoffs for Advanced Manufacturing Using Cost Modeling,” reflects the interaction between three major players: the designer, the manufacturing engineer, and the cost estimator. Although they are introduced as adversaries in the following cartoons (see Figure 1 and Figure 2), the goal of this thesis is to develop a framework by which they may cooperate in a concurrent engineering environment. The software implementation plays a large role in the communication between team members, as well as to standardize company practices and provide an institutional history.

This chapter includes:

- Traditional cost estimation
- Goals of the project
- Roadmap through the thesis

FIGURE 1. Cartoon depicting relationship between design and cost estimation.

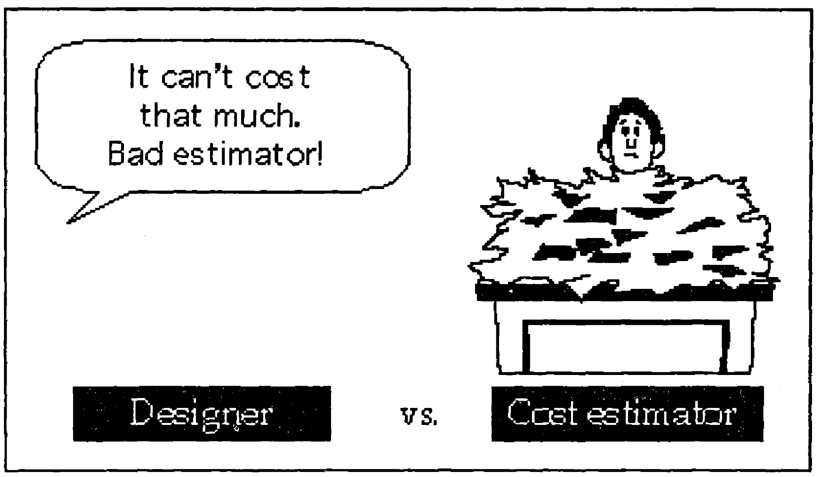
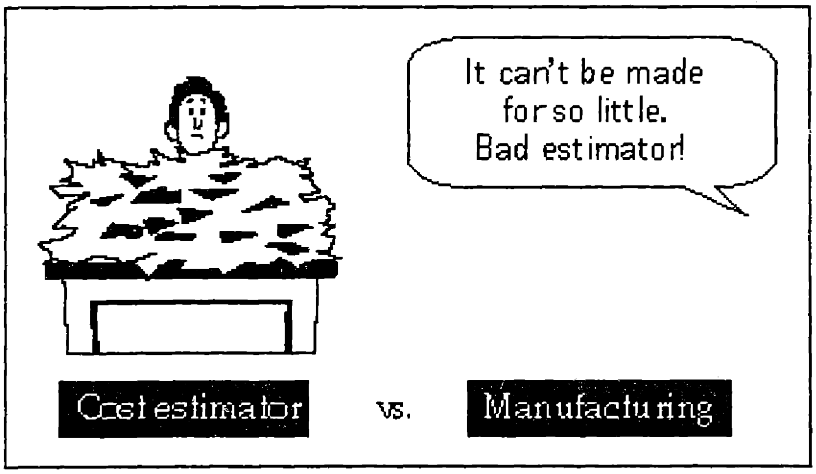


FIGURE 2. Cartoon depicting relationship between cost estimation and manufacturing.

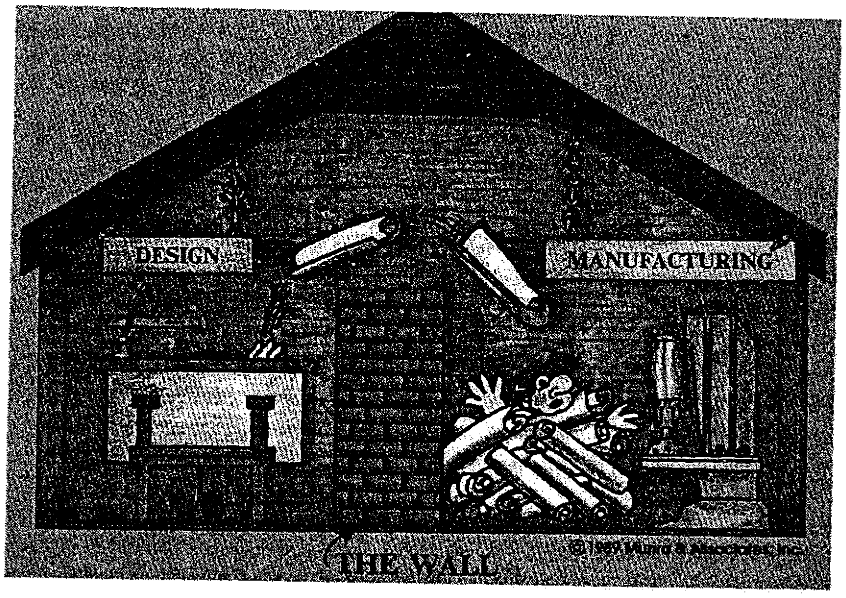


1.0 Traditional cost estimation

Manufacturing requires the close interaction of experts from many disciplines, from designers to process engineers to cost accountants. All of these areas are equally important in creating the best, cheapest part in the shortest time. It is important to know the tradeoffs early in the design process, since the majority of the manufacturing cost is determined early in the design. A wrong decision could result in disastrous capital expenses or the selection of a process which incurs small expenses over a large number of parts. Although there is no "right" decision, there may be many reasonable ones.

The traditional attitude of designers is popularly known as the "over the wall approach," as shown in Figure 3. The manufacturing engineers must deal with design specifications without the chance to give feedback about either the process or the design details.

FIGURE 3. The "over the wall" approach to design. [1]



Boothroyd, Dewhurst, and Knight — leaders in the field of design for manufacturing and assembly (DFMA) — explain their solution [1]:

> One means of overcoming this problem is to consult the manufacturing engineers at the design stage. The resulting teamwork avoids many of the problems that will arise. However, these teams, now called simultaneous engineering or concurrent engineering teams, require analysis tools to help them study proposed designs and evaluate them from the point of view of manufacturing difficulty and cost.

A lack of communication in this team, however, would hinder the optimization of both design and manufacturing as well as disempower the people involved.

2.0 Goals of the project

This section sets forth the long-term, philosophical goals of the project, as well as the proposed implementation solution. There are many other approaches to modeling manufacturing costs, including MIT theses in the Department of Mechanical Engineering, Department of Material Science and Engineering, and Technology and Policy Program. [3] [5] [9] [13] [16] [19] [20] [21] [22] [25] [31] [34] The current work differs from most in that it tries to set up a framework for solving a broader range of problems instead of focusing on one application. The work in this thesis represents a survey of current cost models available in the market, a generalization of the common features, and a first-pass implementation of the resultant software architecture.

Goal: Develop a software implementation independent of application.

This project is not an attempt to create yet another cost model, but a framework by which a person can make consistent, standardized comparisons across mechanical designs, manufacturing technologies, and cost estimation techniques. The model architecture emphasizes the automatic process plan creation process, as well as the feedback mechanism by which one may optimize the process and input variables. Of course, it is important for the model to contain accurate time-cost data, but the emphasis is on rapid preliminary design. In other words, the model should be a designer's tool, not an accountant's tool.

Specifically, we focused on vacuum-assisted resin transfer molding (VARTM) of fiber-reinforced composite parts. Several unique features are incorporated into the cost model. Cost estimates are based on physical parameters such as part complexity and process rates. The model will include a sensitivity analysis which allows simple design and process tradeoffs. The commercial software has an expert system architecture and can access data from databases, formulas, and CAD drawings.

Goal: Evaluate a given design for manufacturing cost effectiveness

Design for manufacturing is an application which suffers from both an overload of knowledge and a lack of knowledge structure. The overload results from the wide variety of expertise and data needed in the areas of design, manufacturing, and cost. The lack of structure results from the traditional barriers between these three disciplines. In addition, data is usually confidential, not well documented, or incomplete due to the low production levels.

An expert system is very useful in consolidating knowledge in a central location and follows in the spirit of concurrent engineering. We can assume that, although a person is not expert in all areas, he can understand the basic principles behind decisions in other areas. In addition, the expert system should be easily adapted to match the pace of change in the manufacturing technology itself.

Finally, a well-documented software application (both on paper and on-line) can serve as a good teaching tool. It is important to emphasize the benefits of the program to non-expert users.

Goal: Evaluate alternative processing technologies

Advanced manufacturing technologies often offer the promise of higher performance -- parts which are stronger, lighter, more resistant to corrosion, etc. than those made by traditional technologies -- at a reasonable cost. Examples include composites (new material), micro-electromechanical sensors (new process), and high speed machining (old technology with improvements). However, the cost of implementing any new technology is a major impediment to both their development and widespread use in industry. In composites applications, an aircraft company may be willing to pay more for a lightweight composite airplane fuselage, but an automotive company may not be willing to pay more for a decorative dashboard panel. On both ends of the spectrum, however, the gain in competitive edge is great if the technology can prove itself to be both cost effective and have improved functionality.

With an early cost model (in the design or pre-production states) the relative ranking of several designs and/or manufacturing technologies is more important than an absolute bottom line cost. Once the decision has been made to go ahead with a product, the designer is concerned with reducing the cost.

Goal: Facilitate standardization of cost estimates in an organization over time.

The program should capture the "corporate memory" — to apply knowledge consistently through an organization and provide access to expert information when the experts themselves are not available. [40] Even so, we must recognize that manufacturing processes are constantly being improved and numerical values updated.

3.0 Roadmap through the thesis

This thesis contains both background information and practical notes about the use of *Cost Advantage*. The reader does not need to read one part before the other, but should have a general familiarity with the vocabulary and concepts associated.

Chapters 2, 3, and 4 provide background information about the various knowledge bases needed in developing this model. Chapters 5, 6, and 7 are the main deliverables. In Chapter 5, we present a "wish list" of software features, based on a survey of other commercial cost estimation products. In Chapters 6 and 7, we organize this list into an formal framework and implement it using the *Cost Advantage* software. Chapter 8 shows some sample output and post-analysis of the program. Chapter 9 concludes with an analysis of the approach taken, a review of the milestones, and suggestions for future model builders.

The appendices provide more practical information for the future model builder. Appendix A contains all oversized figures and tables. Appendix B supplements the *Cost Advantage* User's Manual. The information is specific to our model and addresses problems not solved with *Cost Advantage*. Appendix C is the bibliography.

TABLE 1. Roadmap through the thesis.

	Chapter	Description
1	Introduction	This chapter explains the three major parts of the project title and the importance of good software implementation.
2	Resin transfer molding	General description of RTM. How did we find experts? Details about braiding and tool design. Detailed process plan, and different categorizations for thinking about the same information. Source of time and cost estimations.
3	Cost modeling	General methods of cost estimation, such as complexity, time estimation, comparison with past runs, etc. Discussion on sensitivity analysis methods.
4	Expert systems	Description and brief comparison of types of expert systems. How does *Cost Advantage* fit in? Who is an ideal expert, and how do you get them to cooperate.
5	Software selection	Problems with existing software. List of requirements & decision criteria. Description of software considered.
6	Conceptual architecture	Independent of implementation, what architectural structure would allow the most flexibility?
7	Implementation	Using concepts developed above, translate conceptual architecture into concrete software ideas.
8	Results	Example cost reports and analysis results.
9	Future work	Comparison of milestones and work accomplished. Notes about scheduling. Future implementation such as sensitivity analysis, links with Witness and ProEngineer.
A	Oversized pages	Collection of oversized figures and tables.
B	Model builder's guide	Supplements *Cost Advantage* User's Manual with practical advice about using the program and designing the practical implementation issues.
C	Bibliography	Listed by design for manufacturing, manufacturing processes, cost modeling, and knowledge-based systems.

Chapter 2

Resin transfer molding

Process plan generation cannot be done in a deterministic way if reasonable alternatives exist for producing various details. This is a major problem for newly developed processes. Hence, we will first generate a default process plan based upon the similarity of the parts to ones in the catalogue and an assumed microstructure. This plan will clearly note assumptions and list possible alternatives. The plan will allow user alteration by selecting known options or by direct entry of a new plan. If the plan requires time steps not contained in the database, then these will also have to be entered. As manufacturing procedures become standardized, the default process plans will be modified to reflect these improvements.

Without a sponsoring company with a clear product, manufacturing facility, and commitment, it was very difficult to procure experts who were dedicated to the project. We primarily talked with Draper Supervisor Robert Faiz and visited the manufacturing facility of Emerson & Cuming in Newton, Mass. See Chapter 3 for guidelines for obtaining experts.

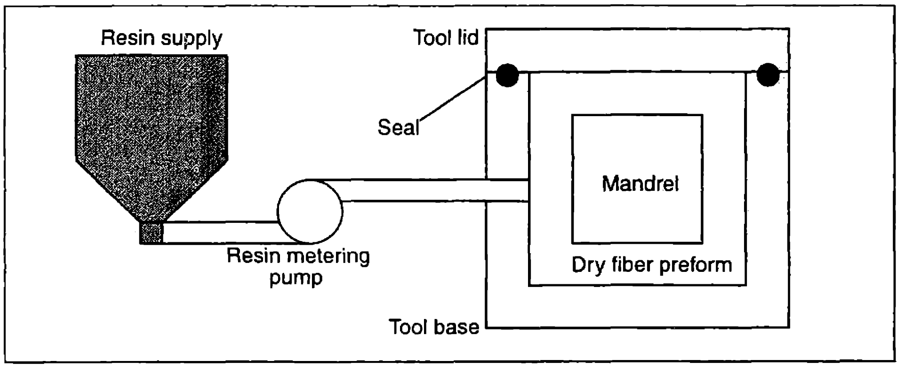
This chapter includes:

- General description of RTM
- Areas of expertise in RTM
- Manufacturing steps

1.0 General description of RTM

Resin transfer molding is a composites processing technology where resin is injected into a dry fiber preform in a mold and then cured. Figure 1 shows a schematic representation of the vacuum-assisted resin transfer molding process we used in our model. Other variants on the process include: resin transfer molding with vent holes, preform molding, structural reaction injection molding, and high-speed resin transfer molding.

FIGURE 1. Main components in a vacuum-assisted resin transfer molding setup for a box beam. Cross-sectional view is shown.



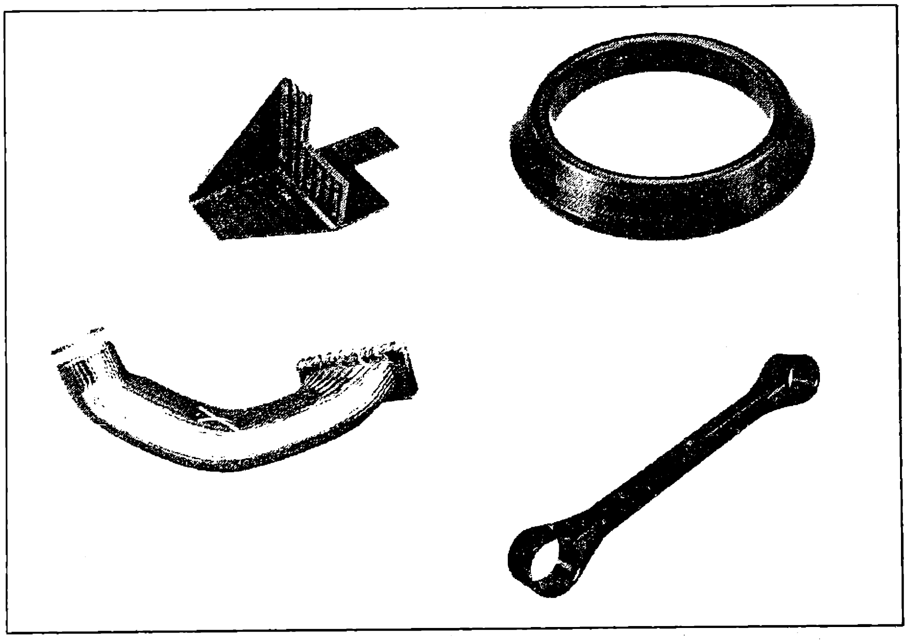
In the software implementation, the first decision would be to choose the high-level manufacturing process (e.g., RTM, sprayup, layup). The current program concentrates on VARTM, but allows this decision for the model user to input manually. Table 1 shows a comparison of manufacturing process based on design guidelines.

TABLE 1. Comparison of manufacturing processes based on design guidelines. [10]
(* Requires staged cure to avoid exothermic reactions in thick parts)

Design parameter	RTM	Sprayup	Layup	SMC
Min inside radius (in)	0.25	0.25	0.25	0.0625
Molded-in holes	no	large	large	yes
In-mold trimming	no	no	no	yes
Core pull, slides, undercuts	difficult	difficult	difficult	yes
Min draft angle (deg)	2-3	0	0	1-3
Min practical thickness (in)	0.080	0.060*	0.060*	0.050
Max practical thickness (in)	0.500	no limit	no limit	1
Thickness variation (in)	0.010	0.020	0.020	0.005
Bosses	difficult	yes	yes	yes
Ribs	difficult	no	yes	yes
Hat section	yes	yes	yes	no
Finished surfaces	2	1	1	2

Figure 2 shows a range of parts which are possible through resin transfer molding.

FIGURE 2. Sample RTM parts made by Fiber Innovations in Norwood, Mass.



2.0 Areas of expertise in RTM

Each of the following sections is complicated enough that we could build an expert system for each one. Although this wasn't included in the implementation phase of the project, we briefly discuss outside work which would be useful in creating rules.

The author of [12] expressed the difficulty in trying to collect expert information for resin selection.

> It came as a surprise to be requested to prepare a procedure for selecting the proper engineering plastic for a given application. Since such decisions are made every day by someone, I assumed it was a fairly routine and repetitive exercise, and only needed to be reviewed, assembled in some logical sequence, and written.
>
> The assumption was incorrect. It was a revelation to learn that, of nearly fifty experienced people interviewed who are close to the resin selection responsibility in all its aspects, no two of them could agree on a logical process for selection; and no one knew of a publication that could. It then occurred to me that the way to develop such a procedure would be to assemble a number of actual developments in engineering plastics from material suppliers, molders, and end users and process based on practical case histories. This assumption, too, was incorrect, for such case histories are either not readily available, or records were never kept, or decisions were made based on pragmatic rather than logical considerations.
>
> Therefore, to prepare this book, I was led to collecting as much pertinent data as practical and to writing a simple suggested route to resin selection. It has been prepared not for the plastic engineer expert in design, but for anyone with an appreciation for simple technical and mechanical considerations.

This same problem is repeated for each area of subspecialty. In this project, we briefly considered several areas which could eventually be included in the longer-term project: resin selection, fiber preform, tool design, cure, and autoclave arrangement. We developed some high-level rules for tool design as a paper exercise and part of a Knowledge-Based Systems (6.871) project

Resin selection. Choose the resin based on performance criteria and processing conditions. Table 2 shows some important questions in the process of resin selection.

TABLE 2. Considerations when choosing resin. From "Engineering Thermoplastic Resins" [12]

Category	Question
General information	What is the function of the part?
	How does the assembly operate?
	Can several functions be combined in a single part?
	Can the assembly be simplified?
	What is the required service life?
	What are the consequences of part failure?
	Are there space or weight limitations?
	What thermal properties are required?
	What electrical insulation properties are required?
	What are the cost limitations?
Codes	Are acceptance codes required?
Environmental	What is the operating temperature range?
	Will the part be exposed to sunlight? weathering? corrosion? humidity?
Mechanical	How is the part stressed in service?
	What is the magnitude of the stress?
	What is the stress vs. time relationship?
	What is the maximum deformation that can be tolerated?
	What are the effects of friction and wear?
	What tolerances are required?
Appearance	Style
	Shape
	Color / Colorability
	Surface finish

Fiber preform. Fiber preforms are made in a process similar to that of sewing clothing. Since the textile industry is well developed, compared to composites technology, many of the processes are well known. From [9]:

> The technology is available to automate the steps to actually produce and cut the fabric. However, after the fabric has been cut to the appropriate shapes, the automation ceases and manual labor is required. A common scene in the apparel industry is rows and rows of people sitting in front of sewing machines sewing shirt sleeves to the shirt.

Although the processes are similar, sometimes the material properties (no pun intended) make a large difference. For example, the clothing fabrics are usually more flexible, stretchable, and have lower elastic moduli than composites fabrics. This translates to more difficulty in handling and stitching.

Tool design. Given the geometry and resin, determine certain design criteria (number of gates, tool configuration, mandrel configuration, heated tool, etc.) based on ease of assembly and cure kinetics. We did find a geometry-based cure model [8], but that would have only been valid for very simple parts.

Cure. Perform cure analysis, using temperature and geometry to control the temperature. This sub-project was not chosen because of the continuous-time component.

Autoclave arrangement. Predicting the heat flow within an autoclave can be a complex thermodynamic analysis that varies with the configuration of parts arranged inside. We can use past information about the configuration and quality of each batch to control the temperature. [39]

3.0 Manufacturing steps

The process steps can be represented either by category or by chronology. The full process plan is shown in Appendix A.

3.1 Process steps by category

We divided the process steps into seven categories, which would be applicable for any manufacturing technology, both composites and otherwise. One benefit of using this "kind-of" classification is that steps involving transportation, inspection, etc. are very similar across manufacturing technologies.

1. Preparation
2. Setup
3. Material handling
4. Operation
5. Transportation
6. Inspection
7. Finishing

3.2 Process steps by chronology

Table 3 shows the manufacturing steps in RTM by chronology. The list was compiled by looking at the example model which came with *Cost Advantage*, thinking through the process steps with Professors Gutowski and Mantell, and speaking with Mr. Faiz from Draper Laboratory.

TABLE 3. Manufacturing steps in resin transfer molding, by chronology.

Major category	Detailed steps
Preparation	Identify required items
Mandrel preparation	Load mandrel piece on overhead hoist
	Clean braiding mandrel
	Apply parting agent on mandrel
	Position heat lamps
	Cure parting agent on mandrel
	Remove heat lamps
	Assemble mandrel
Prepare braider	Setup braiding machine
	Wind tow onto spools
	Replace spools
	Load spools
	Thread tow into braider
	Load mandrel into braider
	Unload mandrel piece from overhead hoist
	Braid preform over mandrel
	Load mandrel piece on overhead hoist
	Remove mandrel from braider
Mold preparation	Load tool onto overhead hoist
	Clean resin transfer mold
	Clean o-ring groove
	Apply mold release to RTM mold
	Position heat lamps
	Cure parting agent on mold
	Remove heat lamps
	Unload tool from overhead hoist
	Setup tool assembly jig
	Assemble RTM mold

TABLE 3. Manufacturing steps in resin transfer molding, by chronology.

Major category	Detailed steps
Capture preform in mold	Set preform into mold
	Unload mandrel piece from overhead hoist
	Secondary preform placement
	Position o-ring mold seal
	Load top plate onto overhead hoist
	Attach top plate to press
	Unload tool from overhead hoist
	Match mold with top plate on press
	Apply clamping force
Injection machine preparation	Setup resin injection machine
	Attach vacuum line
	Attach RTM line
Mold preparation	Preheat platens (and tool)
	Plug heater into mold
	Heat up mold
	Evacuate mold
Resin preparation	Preheat resin (to lower viscosity)
	Mix two parts of resin
	Vacuum de-gas resin
	Load mixed resin onto injection machine
Inject resin and cure	Inject resin
	Cure in RTM mold
Disassemble manifold	Detach vacuum line
	Detach RTM line
	Unplug heater from tool
	Remove RTM mold lid
	Remove tool from heating press

TABLE 3. Manufacturing steps in resin transfer molding, by chronology.

Major category	Detailed steps
Remove part from tool	Move tool into cooling press
	Close top plate
	Cool down part
	Open top plate
	Remove tool from cooling press
	Load mandrel piece on overhead hoist
	Remove part from mold
	Remove mandrels
	Unload mandrel piece from overhead hoist
	Remove o-ring
Post cure	Put part in fixture
	Post cure
	Remove part from fixture
Finishing steps	Load part into overhead crane
	Deflash
	Manual debur
	Setup NC trim
	Load part into NC
	NC trim
	Remove part from NC

Chapter 3

Expert systems

An expert system requires, of course, knowledgeable and cooperative experts. Although we had only limited access to such experts in this project, this chapter will set forth some of the guidelines for obtaining and working extensively with such experts in the future.

In addition to collecting information, we built an expert system to contain the information collected. There are several paradigms for expert systems, based on the desired functionality. We went with a rule-based system, and the rules were formalized into a system and implemented into *Cost Advantage* by Cognition Corporation.

Expert systems are a subset of artificial intelligence. The words "expert system" and "knowledge-based system" are used interchangeably.

This chapter includes:

- Introduction to expert systems
- The ideal expert
- Guidelines for creating rules

1.0 Introduction to expert systems

Knowledge-based systems have many benefits. [35]

- Preserve/replicate knowledge and expertise
- Make knowledge accessible
- Apply knowledge consistently through the organization over time
- Provide an environment for knowledge standardization and growth
- Leverage the expert
- Improve practice; support the average
- Help avoid disasters
- Help manage change
- Manage change and complexity in knowledge-intensive situations and processes
- Solve complex problems that thwarted traditional data processing techniques
- Prototype new approaches easily and quickly

Expert systems can be very powerful, but there is a danger of applying the technology to a task for which is not well suited. It is important to understand the advantages and limitations of expert systems, as well as other alternatives to this computer technology. The model builder should be aware of the reasons against using expert systems, the strongest of which can be resistance from the experts themselves. The program needs to be built and updated over time; benefits should be quantified; and the company should "buy into" its success. Other implementation technology alternatives include [35]:

Search the solution space. This differs from expert systems because it searches through the possible solutions in a methodical way. Expert systems try to make decisions based on reasoning. Examples include adaptive search and optimization and genetic algorithms.

Dealing with uncertainty by assigning mathematical probabilities. This differs from expert systems because of the emphasis on "certainty factors" and relationships with past data. Expert systems try to capture the underlying reasons instead of blindly following previous data. Examples include reasoning under uncertainty and statistical comparison.

The reader should realize that artificial intelligence is a relatively young field and that many research efforts are a demonstration of technology more than a full-blown implementation. However, this has started to change in the recent decades. One other thing to remember is not to be lured into the romance of using artificial intelligence over human abilities. That is, sometimes it is better to use human expertise. [42]

1.1 Representation of information

One distinguishing feature of expert systems is the *ontology* — the system of abstraction from the specific vocabulary of the application to more general terms. The use of information about the information is also known as "meta-data." For example, in this project, the term "resources" describes properties as diverse as braiding machine feed rate, injection machine maximum pressure, and resin cure rate. (See Chapter 6 and Chapter 7 for more detail.) Also, the choice language used to describe the information in the model should be made carefully. Different representations could be functionally equivalent, but one is more difficult to implement, e.g., `1997` v. `MCMXCVII`.

One important decision in this project was whether to describe the manufacturing steps in a "part-of" or "kind-of" hierarchy. There are two types of hierarchy trees: part-of and kind-of hierarchies. Their function is identical in terms of inheriting characteristics downward throughout the tree, but the order of the hierarchy is important for the convenience of the inheritance. A graphical comparison of the two types of hierarchy trees is shown in Figure 1 and Figure 2.

I chose a kind-of hierarchy because this allowed more flexibility in terms of comparisons across dissimilar manufacturing processes. For example, all processes have steps such as inspection, setup, and transportation which are pretty much uniform in form. In more practical terms, the part-of hierarchy corresponds to a time-based listing of process steps, while the kind-of hierarchy corresponds to a category-based listing. Note that all of the elements are accounted for in both trees.

FIGURE 1.

A “part-of hierarchy” corresponds to a time-based listing.

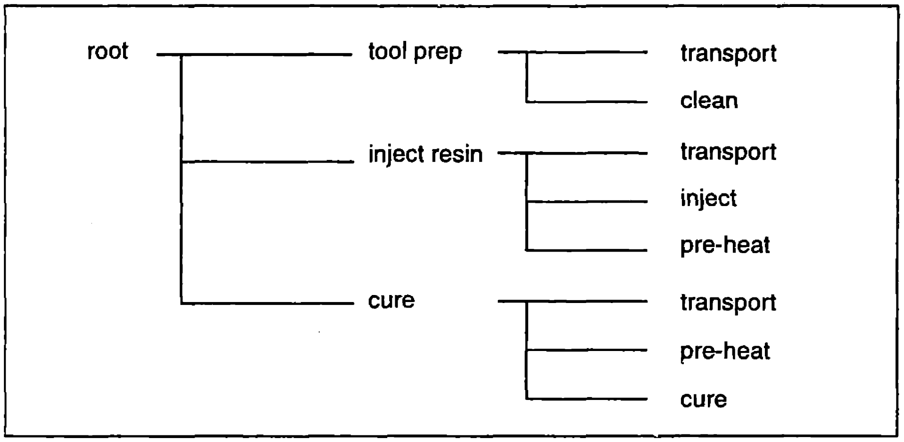
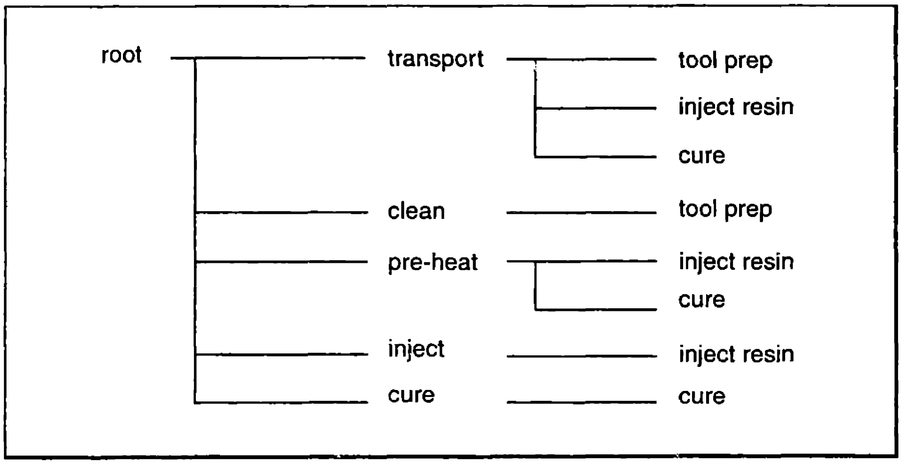


FIGURE 2.

A “kind-of hierarchy” corresponds to a category-based listing.



1.2 Other knowledge-based applications

Our project uses a rule-based system for cost modeling. For further background and examples of applications in other fields, please see the following references:

- Packaging [45]
- Mineral exploration [38]
- Autoclave arrangement [39]
- Medical diagnosis [44]
- Determination of British nationality [41]

Although these applications span very different fields, the implementations have similarities in terminology and also in the separation of domains between the user and the inference engine.

For comparison, Figure 3 shows a simplified version of the architecture we developed in this project (see Chapter 6 for details). Like some of the other software architectures, there is a separation between the model user and the model builder. However, in our system, we want to design the model such that the user can understand and has access to things inherent in the program (e.g., equations, rules).

FIGURE 3. Simplified flowchart for this project

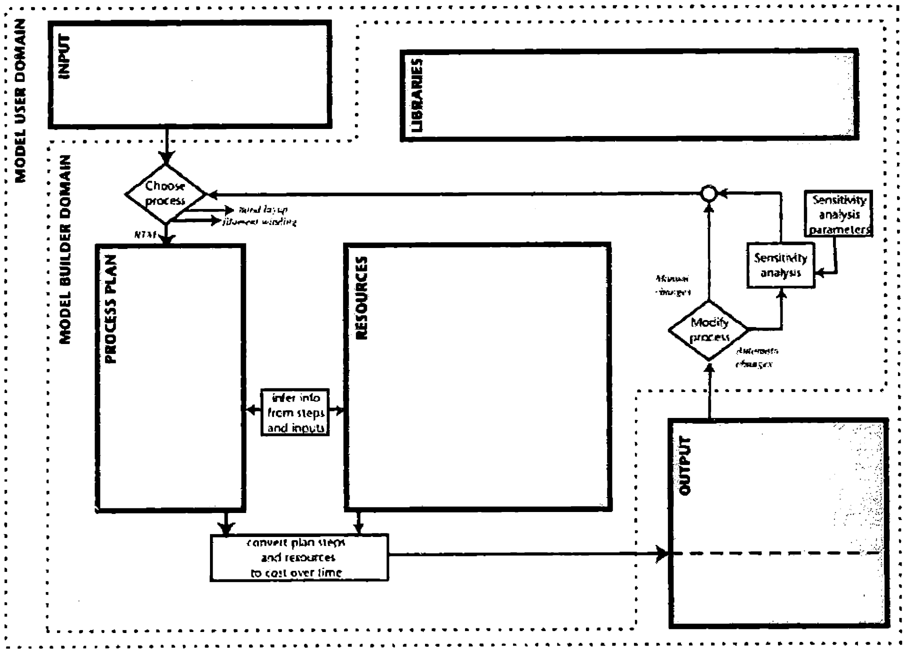
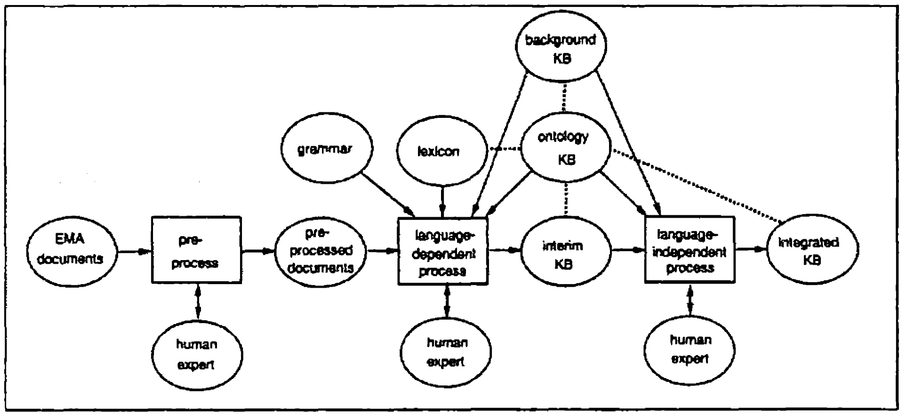


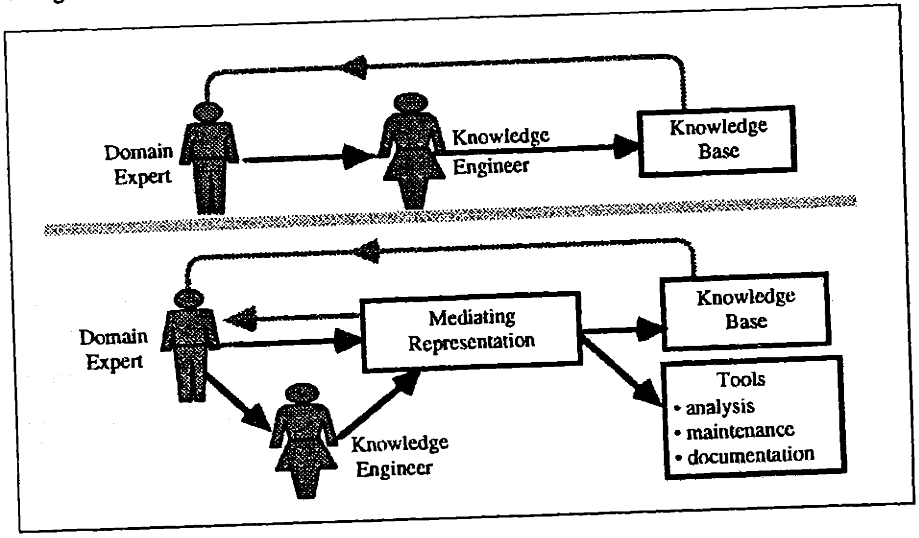
Figure 4 is a flowchart which shows a more general flow of information, specifically used in the project to determine whether a person is a British citizen. In this model, there is more of a separation between the language-dependent and language-independent knowledge bases (KB). In this case, there are clear pre- and post- processing steps to "translate" a specific application (e.g., RTM) to the generic logic engine.

FIGURE 4. Concept Systems as an Aid for Sharing and Reuse of Knowledge Bases in Material Science." [40]



In Figure 5, the separation of the experts and the computer implementation is further emphasized by the introduction of the "mediating representation." One of the benefits could be clearer communication between the parties, but this further separates the users from the developers, making it difficult for the users make direct changes to the system in the future.

FIGURE 5. Using Knowledge-Based Systems to Define Materials Technology in the Aircraft Design/Build Process." [40]



2.0 The ideal expert

An expert can be defined as [35]

$$(Expert - AveragePerson) = Knowledge \qquad \textbf{(EQ 1)}$$

However, while technical competence is important in an ideal expert, there are also many other factors to be considered. In each subspecialty — design, manufacturing, and cost estimation, in this project — the potential experts would also ideally:

- Be cooperative
- Know where to find the information
- Simplify rules as much as possible
- Look at the bigger picture, leaving details for later
- Consistently use of jargon and terminology
- Not have a lot to lose by divulging his information
- Be familiar with expert systems (or willing to learn)
- Give their time on a regular basis (e.g., 3 hours per month)
- Be geographically nearby

A key issue is how to motivate the expert to cooperate. He should feel that he is an important part of the process and "buy into" the project. Some suggestions would be to provide:

- A copy of the model
- Results and documentation
- Constant feedback and involvement (i.e., equal partner)
- Good use of his time at meetings
- Direct use of one of his applications

In preparation for the expert interviews, we prepared a list of questions, a baseline process plan from which to proceed, and geometry families for reference.

3.0 Guidelines for creating rules

When creating rules for the rule-based system, recognize that the relationships between variables is more important than the actual values. E.g., the heating rate is described as "very fast" instead of "10 degrees per minute." The exact value could then be customized for different applications. There are two types of information to formalize:

1. Rules

   e.g., `If A and not B, then C, else D`

2. Object-Value-Attribute

   e.g., `HeatedPress-10deg/min-HeatingRate`

In writing rules, it is important to make only one step in logic at a time. This makes the rules easier for future developers to understand. Also, when running the program, the user can then trace through the entire string of logic, one step at a time.

Another guideline is for the knowledge engineer to try to understand and use the same vocabulary which the expert uses. This will also facilitate future use of the program, as well as provide more credibility to other users.

4.0 Resolution of conflicts

It is likely that the rules for a parameter are in conflict with one another. One method of dealing with this is to use "certainty factors," which specify the degree to which the condition dictates the output. For this discussion, we will use the following rules:

1. `If A, then D. (0.6)`
2. `If A, then E. (0.3)`
3. `If B, then not D. (0.7)`
4. `If C, then not E (0.4), else D (0.1)`
5. `If C, then F. (1.0)`
6. `If E, then G. (1.0)`

All certainties (shown in parentheses) are between `0.0` and `1.0`, with `1.0` being "absolutely certain." Table 1 shows an example set of forward chaining inputs and outcomes. Any uncertainty about the input is also reflected in uncertainty about the outcome. Note the difference between a false statement and a statement which produces a false value.

TABLE 1. Example of rule firings.

Input	Rule	Outcome	Total +	Total -
A = true (1.0)	1 2	D = true (0.6) E = true (0.3)	D = true (0.6x1.0) E = true (0.3x1.0)	
[E = true (0.3)]	6	G = true (1.0)	G = true (1.0 x 0.3)	
C = false (1.0)	4 5	D = true(0.1) (none)	D = true (0.1x1.0)	
B = true (0.6)	3	D = false (0.7)		D = false (0.7x0.6)

When combining the outcomes, certainty factors are not combined in the manner of probabilities. For example, *probabilities* of 0.3 and 0.6 combine to $0.3 \times 0.6 = 0.18$, while *certainty factors* combine to $[0.3 + (1 - 0.3) \times 0.6] = 0.72$. Intuitively, this means that if you have two positive evidences, then you should be more sure of the outcome. However, you cannot simply add the numbers because it is impossible to exceed 100 percent certainty. So, starting with 30 percent certainty, we then add another 60 percent *of the remaining amount* to the 30. The order of the rule firings does not matter, since the math works out the same -- $[0.6 + (1 - 0.6) \times 0.3] = 0.72$.

Combine the positive and negative weighting separately for each outcome, then add them together. The final outcomes for the above example are in Table 2.

TABLE 2. Calculation of total certainty factors.

Outcome	Math explanation	Value
D	[0.6 + (1 - 0.6) x 0.1] + [-0.42] = 0.64 - 0.42	0.22
E	0.3	0.3
F		unknown
G	0.3	0.3

Hard restrictions can also be used, meaning that even if there is some uncertainty, one input may be strict enough to mandate a certain outcome.

Chapter 4

Cost modeling

There are many reasons for and against the modeling of costs of a manufacturing process before it goes into production. It is important to formalize these costs and weigh the tradeoffs in order to understand the process. We will summarize several methods used to predict cost, both based on theoretical principles and experimental data.

This chapter includes:

- Reasons to use cost modeling
- Cost estimation

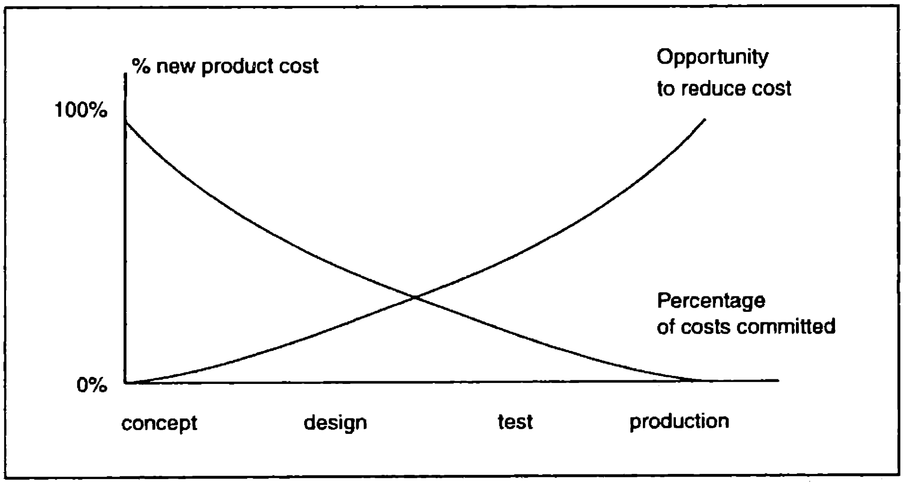
1.0 Reasons to use cost modeling

Cost is only one of several important manufacturing system attributes that need to be estimated. Others include quality, production rate, and flexibility. There are a variety of decision situations when a cost model is needed:

- Predict cost before a proposal is made
- Guide the design process
- Guide process selection
- Guide make-or-buy decisions and vendor negotiations
- Validate costs (e.g., for government contracts)

Often, a large percent of product cost is determined in the early design stage while design activities often contribute only a small percent of that cost (see Figure 1). Therefore, it is crucial that the cost impact of assembly considerations, material selection, and product structure be examined at the design stage. [1]

FIGURE 1. Commitment of design and production costs over time (qualitative representation).



Cost modeling can be performed at all stages of the manufacturing process -- concept development, product development, and production. This project concentrates on the "concept development" phase, which means that the design and process details are not well known. In practice, this translates to more reliance on theory, application unspecific data, and rules-of-thumb than actual production data. With an early cost model (in the design or pre-production states) the relative ranking of several designs or manufacturing technologies (e.g., hand layup vs. resin transfer molding) is as much or more important than an absolute bottom line cost.

The stages of cost modeling include:

- Back-of-envelope
- Automated/database inputs
- Automated calculation
- Sensitivity analysis
- Process plan logic
- System simulation
- Optimization
- Documentation

The biggest problem in developing an expert system-based cost model is the cooperation between stakeholders: designers, process engineers, and cost analysts. They are each experts in their own domain, and there is no one person who would be able to bridge that gap. By definition, these experts are working to opposing goals; the designer may specify certain features that are difficult or costly to manufacture. In reality, one stakeholder may have more say over the others. For example, in aviation, the structures team ensures that an aircraft will hold together during flight, while the weight team ensures that it is light enough to get off the ground. In more competitive industries, cost may be the driving factor. The key to the success of any program is that the user can consistently access information he needs without attending another meeting.

Also, there are two scenarios where the cost model would be useful:

- In large manufacturing environment
- In examining one particular domain (design, manufacturing, etc.)

In the first case, domain experts work together within the same organization to improve the overall product. The challenge is to build and then maintain a common platform for all of the areas. In the second case, it would be harder to obtain expert cooperation. A company's competitive edge is in being able to beat the rivals in his domain (e.g., braiding manufacturers), not in making the overall product (e.g., RTM parts). He has more to lose by volunteering his expert knowledge and would need strong incentives to cooperate.

Design for manufacturing doesn't have universal support, however, even if the numbers do turn up favorable. Some reasons for resisting the use of these methods are:

- Lack of time
- Resistance against other methods
- Confusion about which experts to use
- Low production volume does not justify analysis
- Information is proprietary

The social and political hurdles can be as difficult to overcome as the technical problems of estimating cost.

2.0 Calculating the manufacturing cost

We divided the manufacturing costs into labor costs and resource costs. The time of each operation affects the allocation of resources, including laborers and equipment. Details about the specific implementation used in this project are in the Implementation (Chapter 7). Below, we list several of the options used by other cost modeling projects.

First-order time estimation. Many electro-mechanical systems have first-order response, and also human actions [24]. The first-order equation is shown in Eq 1,

$$v = v_0(1 - e^{(-t/\tau)}) \tag{EQ 1}$$

where v is the velocity, t is time, and τ is the time constant. The velocity can be in terms of any physical units. For example, a fluid measure could be `gallons/minute` and a linear measure could be `area/minute`.

Comparison with history. When a part has been put into production, we can use data from the actual process to refine the model [].

Information content. The complexity of a part can be calculated from its geometry, quantified using information theory [31].

Estimation from theory. Reference [9] shows an example of time and cost estimation from basic principles. The braiding theory includes variables such as braid angle, feed rate, and number of layers.

Chapter 5

Software selection

In selecting the software package on which to develop the cost model, we also implicitly decided upon the amount of computer development versus model development in the project. Many software programs can be configured to do similar tasks, but the difference is in the amount of effort on the part of the model developer. We started out with a bias toward higher-level programming environments.

This chapter describes the software selection process based on our system requirements. The information provided in this section is meant as guidelines for future projects. However, the choices of software may differ based on the functionality required of the program. Based on a survey of available software features and the decision criteria, we chose *Cost Advantage* from Cognition Corporation.

We defer a discussion of future software development to the Conclusion Chapter.

This chapter includes:

- Decision criteria
- Problems with existing software
- Decision matrix
- Cognition's Cost Advantage
- Other software choices considered

1.0 Decision criteria

From looking at the features available in a variety of commercial software packages, the following lists of requirements was compiled (see Table 1 through Table 7). The categories roughly correspond with the sections in the Conceptual Model (Chapter 5). Here, "commercial software" refers to an off-the-shelf software package. "Program" refers to the specific cost model we would build using this software package.

TABLE 1. Requirements for inputs.

Input	**(dynamic)**
Default values	Allow the model to have default values, so the user can proceed with reasonable data even if that is not his area of expertise.
Reset-able defaults	The user can change back to the default values, even after having made some changes.
Changing equations	The user enters coefficients for equations expressed in standard format. The equations are not embedded in the model, but accessible from outside the program run.
Various input types	The program accepts a variety of inputs, including numbers, variables, text, one-of choices, many-of choices, and Boolean values.
CAD drawings	Import geometry information from a feature-based CAD drawing program such as Pro Engineer or Ideas.
List of manuf. steps	Import a pre-determined process plan.
Levels of detailed model	Depending on the design stage, time available, and accuracy of model required, the user chooses whether to use a preliminary, detail, and final production model.

TABLE 2. Requirements for libraries.

Libraries	**(quasi-static)**
Tables	Import information from existing company tables such as Excel or tab-delineated files
Charts	Import information from graphical representations of data. Interpolate between known points
Databases	Import from databases such as Access or Paradox.
Standard equations	Set up equations in standard format, either mathematically (e.g., 1st order exponential) or by function (e.g., `rate of change * difference in values`)

TABLE 3. Requirements for outputs.

Outputs	
Cost drivers	Display the major sources of cost in rank order in a Pareto chart.
Cost sensitivity	Determine affect of changing variables on the total cost.
Resource usage	Cost of material, equipment, overhead, tooling, etc.
Real time updates	Update the cost results as data is input, so the user can make some decisions on the fly.
Conflict alerts	Alert the user if he has violated rules.
Information alerts	Alert the user if there is a potential for confusion.
Factory simulation	Export to a factory simulation program such as Witness.

TABLE 4. Requirements for analysis.

Analysis	
Cost comparison	Save and compare costs between different scenarios.
Suggest variables to change	Make suggestions for how to reduce cost based on cost drivers and sensitivity.
Optimize system	Based on various cost reductions, find best solution.
Merge projects	Merge several projects into one project. For example, several parts compose an assembly.

TABLE 5. Requirements for aesthetics.

Aesthetics	
Graphical user interface	The model builder creates a graphical user interface in a simple way (i.e., not using XMotif or Tcl/Tk)
Intuitive user interaction	Questions are asked in plain English. The user understands the flow of questioning either by overall help pages or by meaningful prompts along the way.
Units for everything	All variables have clearly displayed units.
Leave notes for yourself	Model builder leaves notes visible only to builder.
Leave notes for user	Model builder leaves notes for the user.
Help pages	Able to access other documents (HTML, Acrobat, XV, etc.) for information

TABLE 6. Requirements for architecture.

Architecture	
Process plan hierarchy	The various process plan steps are grouped in a hierarchy tree. There are two kinds of trees — "kind-of" and "part-of" — as described in Chapter 3.
Underlying database	The data required to run the model is stored in a relational database.
Sensitivity analysis	The program has to allow an outside script to perform sensitivity analysis on the model. i.e., after changing a variable, trigger the program to do another run.
Run model at different levels of detail	Program allows levels of detailed modeling. The interaction between variables should matches level of detail.
Save/compare cases	Minimum pre- and post-processing.

TABLE 7. Requirements for development platform.

Development platform	
Cost	The cost of the commercial software is reasonable both for MIT (as an educational institution) and for Draper Laboratories.
Cross-platform transferability	The commercial software is usable over a variety of computer platforms without major changes (e.g., Unix, WinNT, MacOS)
Administrative upkeep	The commercial software (and corresponding hardware) is easy to maintain.
Location of company	The company is located near the MIT campus.
Customer service	The commercial software company has readily available customer service.
Position in industry	The commercial software has a good customer base in industry, and they will be around for future support.

2.0 Problems with existing cost estimation software

Based on the above features that we wanted in a model, the problems with existing cost estimation software fell into two categories: error in data entry and difficulty of use. The human user may be asked to answer questions for which he is not qualified or he may accidentally omit data which are major cost drivers. In terms of difficulty of use, problems fall into several categories:

- Assumed knowledge base of user inappropriate
- Lack of complete model logic and data
- Difficult to adapt model for other applications / manufacturing technologies
- Poor documentation

Perhaps a suitable metaphor for the software would be that of a printed circuit board versus an electronic breadboard. In the PC board, all connections are "hard-wired" into place and unchangeable. True, the engineer may exchange different chips (corresponding to exchanging the "resin transfer molding chip" for a "hand layup chip"), but the configuration is restricted. A bread board, on the other hand, allows any wiring options and any types of chips. This may be disadvantageous because it allows too much freedom and might be too difficult for a non-expert.

What we tried to create was a mid-point between these two extremes. Our model can be compared to a bread board which has certain modules provided — default values and default logic — but also allows change by an expert user. Further discussions about the software architecture are continued in the Conceptual Model (Chapter 5) and Software Implementation (Chapter 6).

3.0 Decision matrix

We understand that a lot of the functionality we want can be implemented in any program, but the question is, "How easy will it be?" We want to optimize the amount of useful contribution instead of spending a whole year on just configuring a database or language within which to work.

We only compared four options in depth, as shown in Table 8.

TABLE 8. Comparison of software packages.

	S/sheet	LabView	GA SEER	Cognition
Spreadsheet compatible	x	x	x	x
Graphical output	x	x	xx	x
User interface included	x	x	x	x
CAD links				xx
User alerts/warnings		x	x	x
Notes fields				x
Underlying database		x	x	x
Producability Rules	x	x	x	xx
Run model at levels of detail			x	xx
Default values for unknowns			x	xx
Transparent data structures	x	x	x	x
Industry acceptance			xx	xx
Cost model documentation	x			xx
Fast development time			xx	x
Program size capability			xx	x
Hardware requirements	PC/Mac	PC/Mac	PC/Unix	Unix /PC
Rough cost	$100s	$1,000s	$10,000+	$10,000+

One drawback to our selection process was the cost limitation and also that we needed to make a decision right away. For example, the BDI injection molding version was released in July 1997. Even at the writing of this thesis, the Cognition program was about a year late from its announced NT-version release date.

4.0 Cognition's Cost Advantage

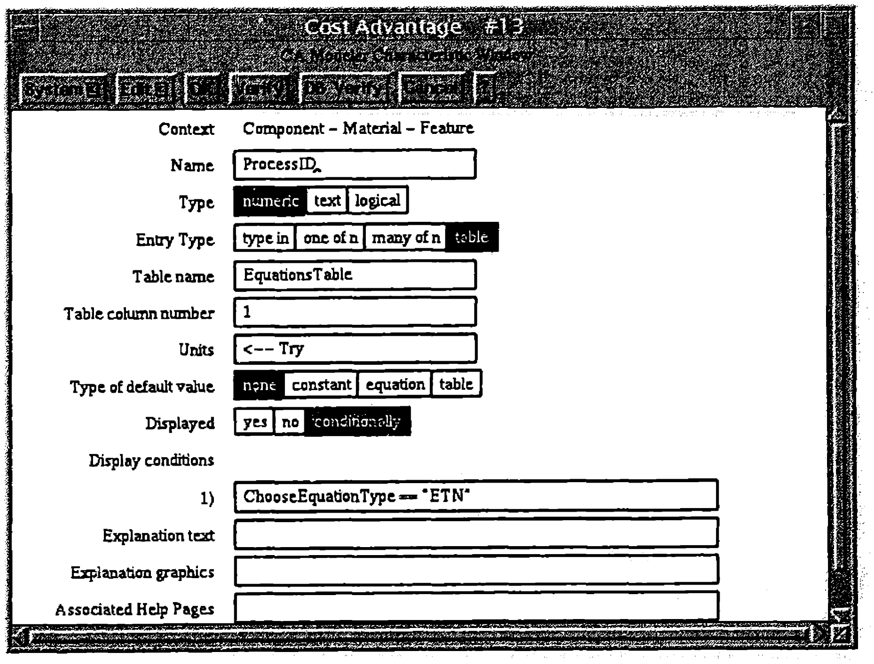
`http://www.ci.com`

We chose Cost Advantage (developed by Cognition Corporation in Bedford, Mass.) based on the decision matrix above. One of the driving factors was the possibility of linking with other programs. Below is a brief description of the program.

Expert system. CA is an expert system written in Lisp. This means that it employs backward chaining and the ability to make inferences from logical statements. See chapter 2.3 for more details.

Easy-to-program user interface. The model builder and model user both use similar interfaces to enter information, i.e. form-based. This makes it easy for a first-time user. The forms may be a bit clumsy for large scale data entry, but there are other mechanisms for that. English-language programming with mostly graphical interface, so you don't actually have to know lisp, though you can program in Lisp. Figure 1 shows several model-building choices.

FIGURE 1. Example window with easy-to-program model-building options.



Context	Component – Material – Feature
Name	ProcessID
Type	numeric text logical
Entry Type	type in one of n many of n table
Table name	EquationsTable
Table column number	1
Units	<-- Try
Type of default value	none constant equation table
Displayed	yes no conditionally
Display conditions	
1)	ChooseEquationType == "ETN"
Explanation text	
Explanation graphics	
Associated Help Pages	

Alerts. The program alerts the user, either automatically if a value falls out of range or violates a rule; or to provide information. - indicators if you forgot to fill in a field

Hierarchy tree. The process steps can be arranged in a hierarchy tree, which is governed by rules and relationships between variables and equations. The inheritance of properties down the hierarchy simplifies the definition of new nodes in the tree.

Automatic process plan generation. Based on input values and built-in rules, CA can automatically generate a default process plan. i.e., both values, variables, and hierarchy steps can be controlled by rules.

Traceable values, both rules and equations. Cost Advantage provides a function by which the user can view the specific equation which was used to calculate a variable. In addition, the program tracks variables which that variable affects. Unfortunately, the tracing only happens one step at a time, i.e., the program does not keep backtracking until only input values or library values are found.

Defaults. The program tracks default values, and the user can reset to the default value after changing it.

CAD compatibility. Cognition provides links with many CAD packages such as Pro/Engineer, Ideas, etc. Cost Advantages reads in the feature-based description of the parts and can link them directly with feature steps in the cost model.

WinNT compatibility. The Windows NT version is scheduled for fall 1997.

Geographical location. Cognition Corporation is located in Bedford, Mass.

User support. There is a user group and a responsive customer support department.

Major users in industry. Cognition is well established in industry, with customers such as those listed in Table 9.

TABLE 9. Major users in industry of Cognition's *Cost Advantage*.

AMP Corporation	Boeing	Eaton Corporation
Fiat	Ford	Fuji Xerox
General Electric	Goodyear	Lockheed Martin
McDonnell-Douglas	Mitsubishi	Pitney Bowes
Plasma Quest	Pratt & Whitney	Rolex
Samsung	Sanyo	Seagate Technology, Inc
Seiko Epson	Teledyne Brown Engineering	Texas Instruments
Toyota	United Technologies Research Ctr.	Xerox

Educational discount. MIT qualifies for the educational price of $500 per module. We have purchased the model builder license, one model user license, and the composites template. We plan to purchase the link to Pro/Engineer link in the future, which would allow the use of Pro/Engineer files directly as input. The total would be approximately $20,000 at full price.

5.0 Other software choices considered

5.1 Existing Draper resources

Draper spreadsheet model. Draper's spreadsheet model attempted to use separate files for quick access to important variables, but too many of the equations were hard-wired into the model. The user would have to know exactly where to find a value or equation within the spreadsheet in order to change it. The program included a very comprehensive list of parameters, but the interface was difficult to use or change.

Draper LabView model. Draper had a model which was developed using Matlab and LabView by National Instruments. This program was not specifically a cost model, but showed some of the possible features of LabView.

Draper already owned LabView, and engineers were already familiar with it. They also used Matlab as an intermediate processing tool, which they were also familiar with. LabView is good for empirical data, displaying graphics, and it can be converted to HTML format. We probably would not need the data acquisition function, just the graphical user interface builder. LabView is usually known as a data processing software used to read signals from physical hardware. Although there is a built-in logic engine and graphics builder, the mechanism is not very easy to use. The programming is graphical, and the user manipulates various blocks, such as adders, multipliers, etc. The resulting "circuit" looks like a rat's nest and would be very difficult to figure out or debug. You also can't save different scenarios for comparisons; each one will be its own file.

5.2 GA SEER-Design for Manufacturing

`http://www.gaseer.com/`

GA SEER's Design for Manufacturability/Assembly is a software tool for new product development and re-engineering, used during the concept phase of new product development. The DFM module is very comprehensive, including composites processes such as hand layup, filament winding, pultrusion, and composite spray. In terms of non-composite manufacturing processes, SEER-DFM also includes machining, fabrication, assembly, electrical assembly, molding, casting, forging, finishing. In addition, other SEER products cover the fields of software project control, software sizing, hardware development and production, hardware operations and support, and integrated circuit development and production.

SEER-DFM has an underlying spreadsheet engine, with a standard-windows interface. The program also allows for uncertainties in the model. The process plan is ordered by "work elements" and allows the user to view different levels of model complexity. It was felt that incorporating data from Draper's spreadsheet into this architecture would have been a good first pass at the problem. Because there are so many different types of models, GA SEER does not allow for much flexibility in the structure of the models. In addition, the user is limited to the few cost categories defined in the model. Although this would have been a good choice of commercial software, they did not offer an educational discount so their price was out of range for this project.

5.3 PCAD

PCAD (Process Cost Analysis Database) and COSTADE (Cost Optimization Software for Transport Aircraft Design Evaluation), both developed by NASA and Boeing, work together to find an optimal structural design of a composite airplane body. Variables include cost, weight, design, stress, and manufacturing plants. Database engine is Microsoft Access.

5.4 PRICE

PRICE (Provides Life Cycle Cost Estimating and Asset Valuation Solutions), developed by Lockheed Martin, has modules for electronics, electro-mechanical systems, structural assemblies, cost, cost effectiveness, and operational availability.

5.5 Boothroyd/Dewhurst Inc. DFMA

`http://www.dfma.com`

Boothroyd Dewhurst, Inc., is a company founded out of the University of Delaware. Their Design for Manufacturing and Assembly software includes: design for assembly, design for environment, and design for service. Specific modules are: machining, injection molding, sheet metal working, and die casting.

Chapter 6

Conceptual model

In the development of the conceptual model, we took the program requirements and tried to abstract away from the specific implementation. Recognizing the importance of re-using the model for different processes, we developed the cost model architecture independent of manufacturing process and specific software implementation. We wanted to try to account for all factors in a cost model categorically to make it easier to compare dissimilar technologies.

One important feature is the division between the model user and model builder domain. The main distinction is that users can only change values in their local copy of the model, whereas builders are responsible for developing the process step choices as well as any rules which affect them. This is useful especially in a large organization or over a large cross-section of program users. Still, for a small number of users, it is useful for maintaining continuity and rules between different cost models.

This chapter only explains the overall concept of each module. Details about the actual implementation using *Cost Advantage* are in Implementation (Chapter 7). We reserve discussions about future implementation suggestions to the Conclusion (Chapter 9).

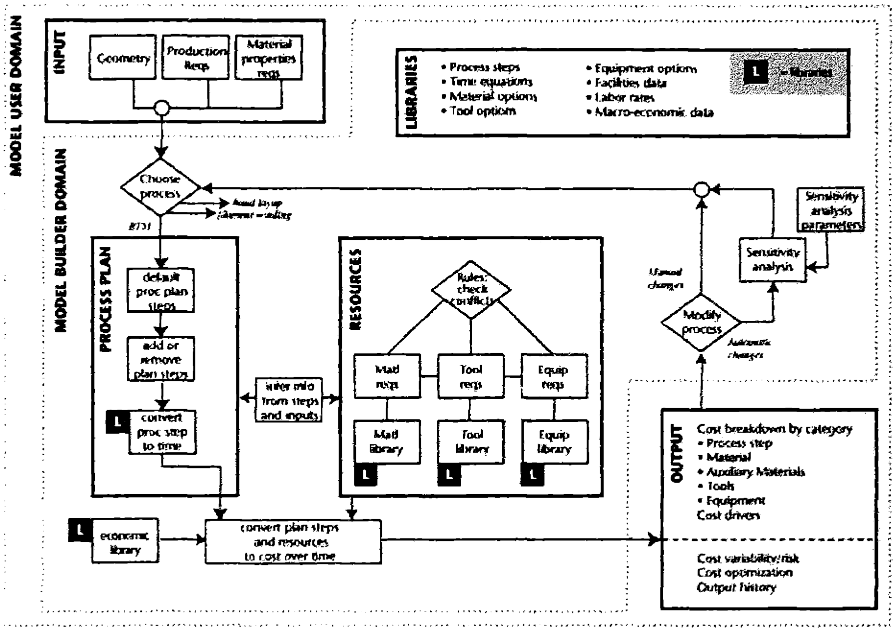
This chapter includes:

- Overall flowchart
- Components of flowchart
- Example of applicability across technologies
- Ideal user interface
- Future work

1.0 Overall flowchart

Figure 1 is a chart of the main portions of the cost architecture. It is also an index of the sections in this chapter.

FIGURE 1. Overall flowchart



The Model Builder Domain includes the process plan, resources, cost conversion, analysis, and libraries. However, each distinct manufacturing process would have an architecture as given in Figure 1. The model, as currently implemented is specialized for resin transfer molding.

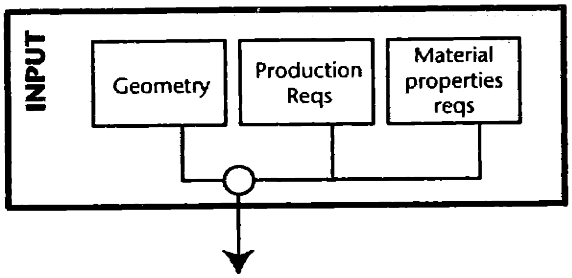
The Model User Domain includes the input and output, as well as access to change things in the Model Builder Domain, but he is not responsible for providing and organizing that information in the first place. This can be viewed as a sort of model default; certain values and options are suggested by the program in the Model Builder Domain, the user can change them if desired.

2.0 Components of overall flowchart

2.1 Inputs

The model user specifies the geometry, production, and material properties requirements for the design at hand. The current model assumes that geometric design is an input; in the future, it should be considered as a variable which can be changed after performing a sensitivity analysis. We also hope that future models will be able to directly use information from CAD drawings.

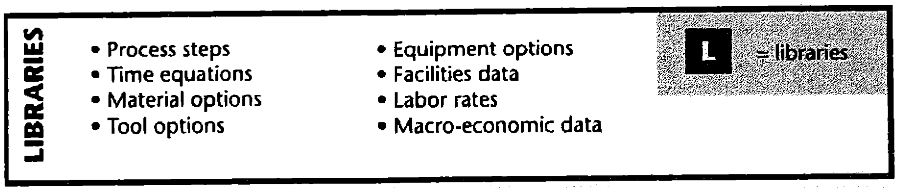
FIGURE 2. Inputs.



2.2 Libraries

Libraries contain centralized tables or databases which are changeable by the model builder and accessible by the model user. Any information which is a physical constant (such as modulus of elasticity of a material) or which can be represented in table format should be put in a library.

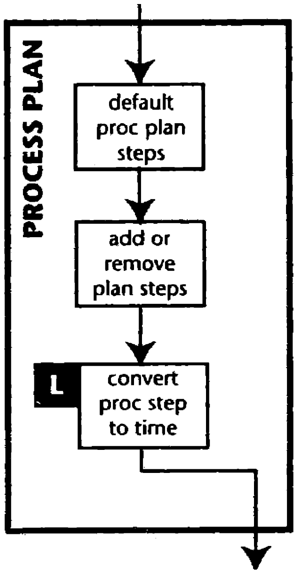
FIGURE 3. Libraries



2.3 Process plan

The process plan categorically accounts for all types of process steps: preparation, setup, material handling, operation, transportation, inspection, and finishing steps. This structure is independent of the manufacturing process, though the individual process steps (stored in the libraries) are customized for each manufacturing process.

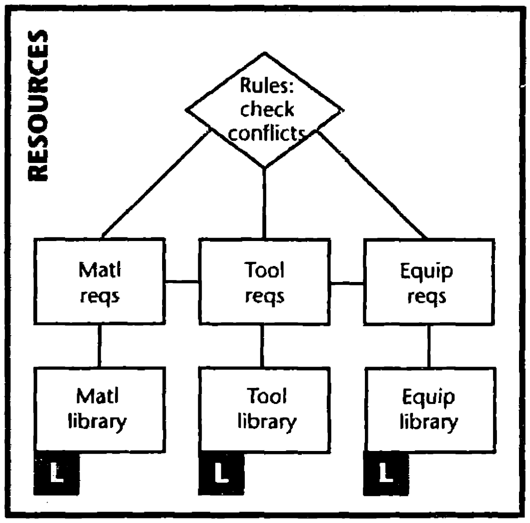
FIGURE 4. Process plan



2.4 Resources

Resources are a company-wide database of materials, tools, and equipment. These values change throughout the manufacturing process, depending on the inputs.

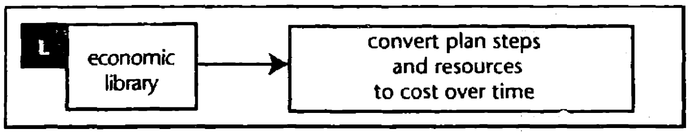
FIGURE 5. Resources



2.5 Conversion to cost

This part of the cost model converts process plan times and resource usage to cost over time, based on theoretical equations and empirical data. Costs are separated into several categories: labor, material, tool, equipment.We also make the differentiation between fixed (non-recurring) and variable (recurring) costs for each category. Fixed costs can either be accounted for in one lump-sum or they can be distributed over all of the parts in that production job.

FIGURE 6. Conversion to cost



2.6 Output

The output features cost breakdown (Pareto charts, material usage), cost drivers, calculation of cost variability, and cost optimization. These categorizations can also vary over time, based on factors such as capital depreciation, economic changes over time, and faster operations due to familiarity with the operation.

FIGURE 7. Output

OUTPUT

Cost breakdown by category
- Process step
- Material
- Auxiliary Materials
- Tools
- Equipment

Cost drivers

Cost variability/risk
Cost optimization
Output history

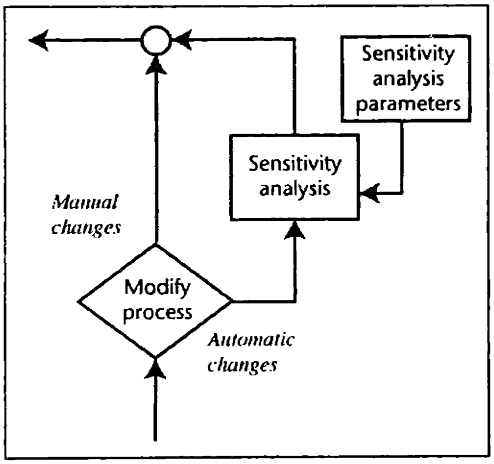
2.7 Analysis

We would like to take a three-part approach to solving the problem:

- Model user chooses variables and the range over which to analyze
- Program suggests choices of variables, based on cost drivers
- Program automatically performs sensitivity analysis

However, since the analysis was not implemented in this phase of the project, we reserve the discussion for the Conclusion (Chapter 9). Briefly, we considered using complexity factors, decision theory, Monte Carlo simulation, Taguchi box analysis (design of experiments), data verification, and a full-factory simulation. We stayed away from options which would require brute force search or optimization, because of the size of the search field.

FIGURE 8. Sensitivity analysis



2.8 Interactions between modules

The modules explained in the previous sections are dependent upon one another. When planning the structure of the model, we needed to set up a methodical way to cross-reference these interactions and make sure that we did not overlook any important parameters. We also kept in mind that some of the parameters (especially the physical description of the part) would be used in multiple process steps. In other words, we wanted to allow the user flexibility to think of the parameters either associated with each process step (process driven) or associated with each physical area of the shop floor (resource driven).

Table 1 shows an example of how a process plan might be mapped out. The right-side columns of the table represent the "resources" part of the flowchart. The full details for the model we implemented are in Appendix A.

TABLE 1. Sample planning chart for interaction between modules

Process plan steps	Time	Material	Tooling	Equipment
Clean surface	area * (time/area)	aux matls	tool area	...
Clean injection ports	10 min / ports	...	num ports	...
Transport	Eq 1, v=5	...	...	crane
Inspect	table lookup	...	...	...

3.0 Example of applicability across technologies

The following example (see Table 2) shows a very brief subset of the information needed to perform a cost estimate. Instead, the example illustrates how the architecture does not depend on the technology. RTM and hand lay-up are somewhat similar, while milling is not.

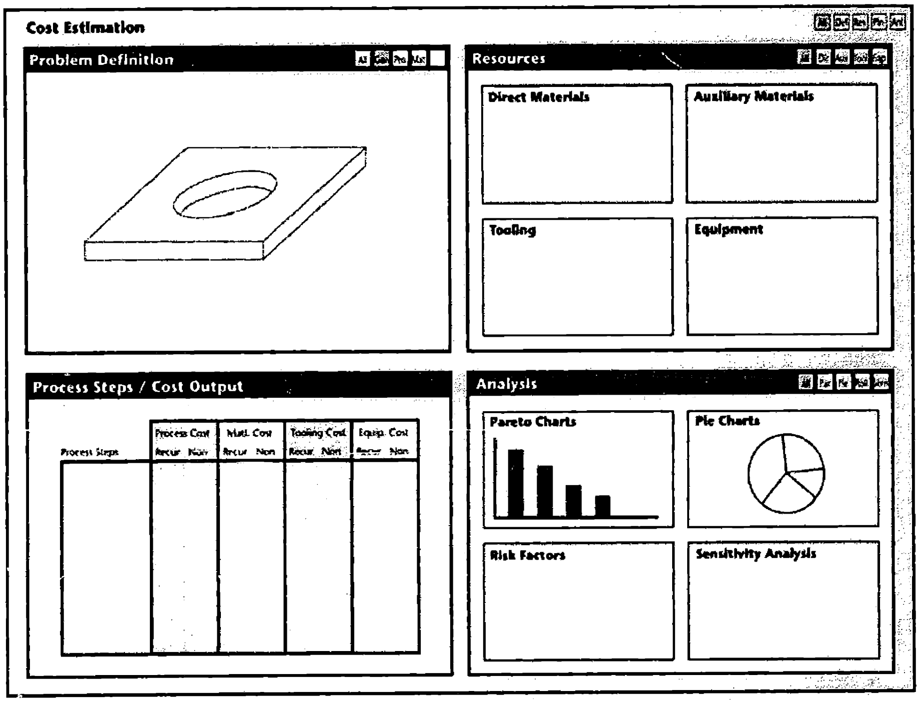
TABLE 2. Example of applicability across technologies.

Module	RTM	Hand lay-up	Extrusion/milling
Inputs	graphite-epoxy box beam	graphite-epoxy box beam	Al6061 box beam
Libraries	composites matls composites labor	composites matls composites labor	metals matls machining labor
Process plan	braid preform inject resin cure	lay down layers compact cure	extrude fixture part mill to finish
Resources	braiding machine injection machine oven	vacuum oven	extruder milling machine
Output	cost estimate cost drivers	cost estimate cost drivers	cost estimate cost drivers
Analysis	sensitivity	sensitivity	sensitivity

4.0 Ideal user interface

Figure 9 shows an ideal graphical user interface, which was conceived separately from the choice of software. It emphasizes that a lot of information is simultaneously considered in producing a cost model. Although this layout is graphically different from the actual implementation (see Chapter 7), the elements are mostly the same.

FIGURE 9. Ideal user interface.



The upper left window defines the problem, including geometry, production requirements, and material requirements. It currently shows the geometry of the part.

The upper right window keeps track of the various resources -- direct and auxiliary materials, tooling, and equipment -- required to produce the part. Entries into these lists are triggered by the problem definition, process steps, and interdependencies between resources.

The lower left window lists the manufacturing steps used to create the part, as well as their related costs in tabular format. Whereas the resources window groups consumption by tangible measures, the process/output window expresses this consumption in chronological order and is useful for understanding the added progressively through the different stages of production.

The lower right window offers various output formats for interpreting the cost breakdown and understanding how to reduce major cost drivers.

Chapter 7

Implementation

This chapter is the software implementation of the concepts of the Conceptual Model (Chapter 6). See Appendix B for a more technical discussion of how to use *Cost Advantage*. Also refer to the CA User's Manual. In each figure caption, it is noted whether the screen shot is in the model builder or model user domain.

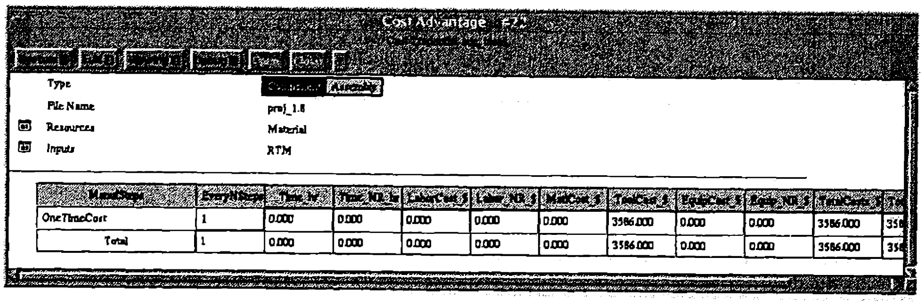
The sections in this chapter are:

- Model builder and model user
- Main *Cost Advantage* window
- Resources and inputs
- Conversion to cost
- Libraries
- Default process plan and cost summary
- Manufacturing steps
- Analysis
- Help pages

1.0 Main *Cost Advantage* window

The main cost window, shown in Figure 1, consists of two parts. The part "above the line" can be thought of as the inputs to the system. They are defined as "resources" and "inputs" in this model. "Below the line" is the process plan and the summary table, which show the output of the model.

FIGURE 1. The main *Cost Advantage* "Cost Note" window. The manufacturing steps have not been filled in. [Model user]



Type
File Name: proj_1.8
Resources: Material
Inputs: RTM

ManufSteps	EveryNSteps	Time_hr	Time_NR_hr	LaborCost_$	Labor_NR_$	MatCost_$	ToolCost_$	EquipCost_$	Equip_NR_$	TotalCosts_$	To
OneTimeCost	1	0.000	0.000	0.000	0.000	0.000	3586.000	0.000	0.000	3586.000	35
Total	1	0.000	0.000	0.000	0.000	0.000	3586.000	0.000	0.000	3586.000	35

2.0 Resources and inputs

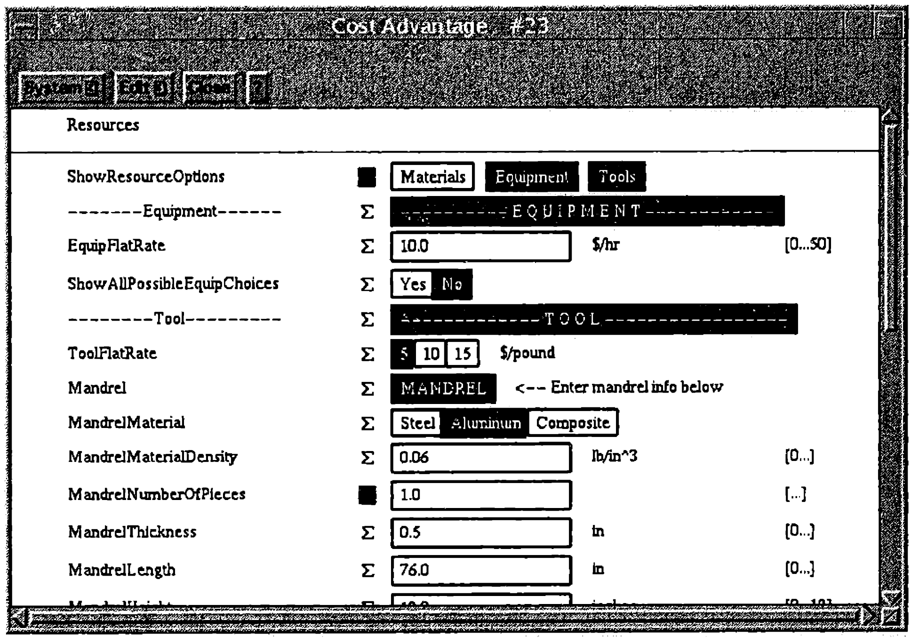
Most of the information in these windows is used for reference (e.g., modulus of elasticity) or for variables which appear in many subsequent equations (e.g., labor cost per hour). Resources include information about materials, equipment, and tooling. Inputs include information about the geometry, production requirements, material requirements, and economic data.

The resources include materials, equipment, and tools. The inputs are divided into four categories: geometry, production, microstructure, and economics. The toggle property of the `ShowInputOptions` variable allows the user to show or hide input fields to make the screen less crowded. In Figure 2, the production and microstructure categories are hidden.

The variables (or "characteristics") on the left column are not static. For example, depending on `ShapeClass`, different physical parameters are displayed. If `ShapeClass` is `cyl_large` then the characteristics will be `PartThickness`, `PartLength`, and `PartDiameter`.

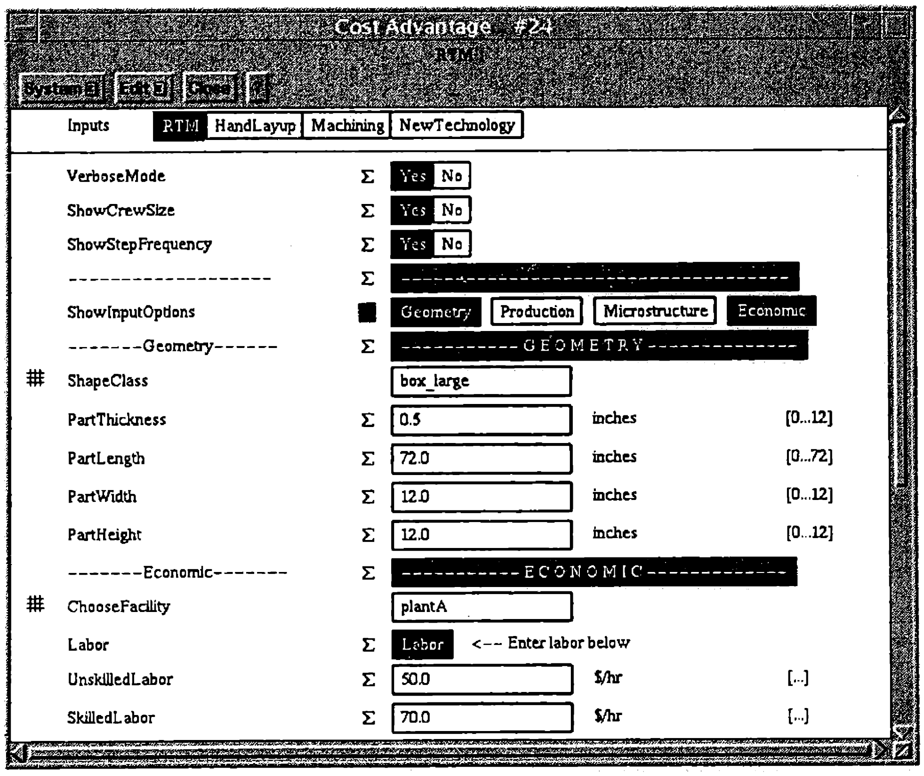
For simplicity, the model presumes that the design is an input and the process is the variable for change. In reality, these would both be interdependent.

FIGURE 2. The resources and input windows. [Model user]



Resources

ShowResourceOptions ■ Materials [✓ Equipment] [✓ Tools]
-------Equipment------ Σ --------EQUIPMENT----------
EquipFlatRate Σ 10.0 $/hr [0...50]
ShowAllPossibleEquipChoices Σ Yes [✓ No]
--------Tool--------- Σ ------------TOOL----------------
ToolFlatRate Σ [✓ 5] 10 15 $/pound
Mandrel Σ MANDREL <-- Enter mandrel info below
MandrelMaterial Σ Steel [✓ Aluminum] Composite
MandrelMaterialDensity Σ 0.06 lb/in^3 [0...]
MandrelNumberOfPieces ■ 1.0 [...]
MandrelThickness Σ 0.5 in [0...]
MandrelLength Σ 76.0 in [0...]



Inputs [✓ RTM] HandLayup Machining NewTechnology

VerboseMode Σ [✓ Yes] No
ShowCrewSize Σ [✓ Yes] No
ShowStepFrequency Σ [✓ Yes] No
-------------------- Σ ------------------------------------
ShowInputOptions ■ [✓ Geometry] Production Microstructure [✓ Economic]
-------Geometry------ Σ -----------GEOMETRY-------------
ShapeClass box_large
PartThickness Σ 0.5 inches [0...12]
PartLength Σ 72.0 inches [0...72]
PartWidth Σ 12.0 inches [0...12]
PartHeight Σ 12.0 inches [0...12]
-------Economic------- Σ -----------ECONOMIC------------
ChooseFacility plantA
Labor Σ Labor <-- Enter labor below
UnskilledLabor Σ 50.0 $/hr [...]
SkilledLabor Σ 70.0 $/hr [...]

2.1 Assumptions

There are a few assumptions of our RTM model that greatly simplified the work required and are also probably non-trivial to implement.

- Inject resin when tool is horizontal
- All sample parts require a one-part mandrel and a two-part tool.
- Constant pressure / mass flow rate
- Standard draft angle
- Planar parting line
- No curvature along length
- Constant thickness along length
- Constant cross-sectional area, chosen from fixed list.

2.2 Geometry

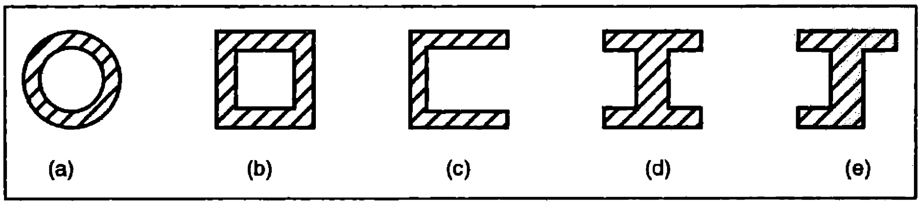
The process of specifying geometric requirements could become very tedious for all but the most simple shapes. To allow more flexibility in the geometry specification, the input files are phrased in a very generic way, and the equations for calculating derived variables use these generic names (see Table 1).

TABLE 1. Example of direct and derived variables.

Direct variables	Derived variables
`Var1 = Length`	`Area = ?`
`Var2 = Width`	`((Var1 x Var2) WHEN Var1 == "Length";` `(Var1 x Var1 x PI) WHEN Var1 == "Radius")`

Other direct variables include: shape-specific dimensions, shape-specific tolerances, fiber volume fraction, fiber characteristics, orientation of fibers, and permeability; cure kinetics and resin characteristics; pressure or mass flow rate; surface finish; and tool material. Other derived variables include: surface area, part weight, and tool weight. Figure 3 shows the part families which have been included in the model.

FIGURE 3. Cross-sectional profiles of part families in model: (a) cylinder, (b) box beam, (c) C-channel, (d) I-beam, (e) J-frame.



The example below shows the data input file used to create the selection chart for the geometry. In the future, using functions built into Cost Advantage, the geometry can be imported directly from feature-based CAD programs such as Pro/Engineer.

```
"Shape_class" "GeomVar1" "GeomVal1" "GeomVar2" "GeomVal2"
    "GeomVar3" "GeomVal3" "GeomVar4" "GeomVal4"

"box_small" "thickness" 0.25 "length" 72 "width" 6 "height" 6

"box_large" "thickness" 0.5 "length" 72 "width" 12 "height" 12

"cyl_small" "thickness" 0.25 "length" 72 "diameter" 6 blank blank

"cyl_large" "thickness" 0.5 "length" 72 "diameter" 12 blank blank
```

Note that all text must be enclosed in double quotes and contain no spaces. For fields which should have no value (as opposed to zero) use the word `blank` without quotation marks.

3.0 Conversion to cost

Conversion to cost may seem like a straight-forward calculation, but there are many different approaches depending on the company or the data available. As long as the data accurately represents the actual manufacturing operation, the choice of cost calculation method does not really matter. However, often there are absences in data, which makes the whole proposition much more difficult.

We divided the manufacturing costs into labor costs (based on the process plan) and resource costs. All of the elements depend on the time of each operation, both in terms of direct wage allocation and in terms of equipment requirements to complete the job. Time estimation can be based on empirical data (e.g., first-order system response) or theoretical calculations (e.g., braiding time model [9]).

Costs are also categorized as either fixed (non-recurring) or variable (recurring). Fixed costs occur once per job, such as capital equipment and machine setup costs. Variable costs occur periodically — once per part or every few parts. In generic terms, the cost of a job includes:

$$\begin{aligned}&\sum[(Time \cdot LaborRate)\\&+(Material \cdot ScrapRate) \cdot MaterialCost\\&+EquipmentCost\\&+ToolingCost]\end{aligned} \tag{EQ 1}$$

In this model, labor, materials, and equipment costs are considered variable costs; equipment and tooling are considered fixed costs. Equipment usage may fall under both categories because of the charge of acquiring the machine, as well as the hourly costs.

For the time estimation, we took advantage of the fact that many electro-mechanical systems have first-order response, and also human actions [24]. The first-order equation is shown in Eq 2,

$$v = v_0(1 - e^{(-t/\tau)}) \tag{EQ 2}$$

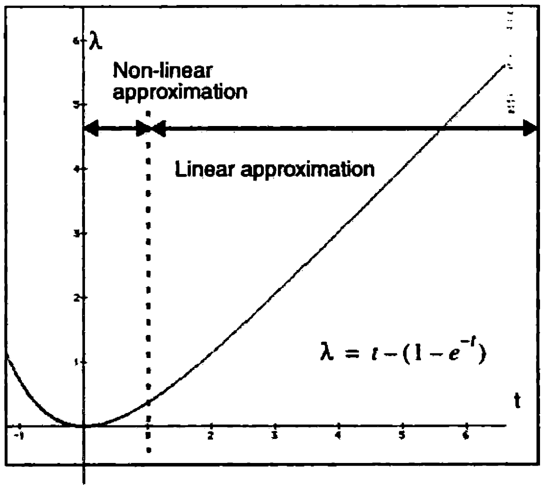
where v is the velocity, t is time, and τ is the time constant. The velocity can be in terms of any physical units. For example, a fluid measure could be `gallons/minute` and a linear measure could be `area/minute`. Integrating, we can solve for λ.

$$\lambda = v_0[t - \tau(1 - e^{(-t/\tau)})] \tag{EQ 3}$$

where λ is the physical unit of measure corresponding to $v=\lambda/t$.

Figure 4 shows a graph of the first-order equation in Eq 3 with $v_0 = 1$ and $\tau = 1$ we can see that there is a non-linear start-up region and a linear long-term region.

FIGURE 4. Graph of first-order equation.



For $t < \tau$, time is approximated as

$$t \approx (((2\tau)/v_0)\lambda)^{1/2} \qquad \textbf{(EQ 4)}$$

For $t > \tau$, time is approximated as

$$t \approx \tau + (\lambda/v_0) \qquad \textbf{(EQ 5)}$$

In practice, Eq 4 and Eq 5 are reinterpreted as [5]:

$$Time = \left[\left(\frac{setup}{run}\right)+\left(\frac{delay}{operation}\right)\left(\frac{operations}{run}\right)\right]\cdot\frac{(parts/shipset)}{(parts/lot)(lots/run)} \qquad \textbf{(EQ 6)}$$

$$Time = \left[\left(\frac{setup}{run}\right)+\left[\left(\frac{delay}{operation}\right)+\sqrt{\left(\frac{v_1}{v_{01}}\right)^2+\frac{2\tau_1 v_1}{v_{01}}}\right]\left(\frac{operations}{run}\right)\right] \qquad \textbf{(EQ 7)}$$
$$\cdot\frac{(parts/shipset)}{(parts/lot)(lots/run)}$$

where the variables are defined as follows:

- **setup** — The time to set up equipment and prepare to do the operation.
- **delay** — The time between successive parts, after the equipment has been set up.
- v_{01} — The extensive velocity.
- v_1— The variable that is input to the cost equation.
- τ_1— The dynamic time constant of the system (63% of terminal velocity).

Table 2 and Table 3 show the lookup table for the first-order equations.

TABLE 2. File for first-order equations. (Part I).

"ID"	"Name"	"Eq"	"Setup_min"
0	"Blank_ETN"	blank	blank
1	"Time_only"	blank	blank
2	"Generic"	blank	blank
50	"Apply_parting_agent"	2	5
80	"Apply_vacuum"	1	5
90	"Attach_lines_resin_injection_RTM_mold"	2	1.67
140	"Attach_lines_vacuum_RTM_mold"	2	1.67
160	"Attach_overhead_equipment"	2	4
170	"Braiding_2D_triaxial"	2	1
260	"Clean_tool_mandrel_braiding"	2	3
290	"Clean_tool_resin_transfer_mold"	2	3
360	"Cure_liquid_shim"	1	480
380	"Cure_RTM_cure_cycle_shell_1895"	1	285
390	"Cure_RTM_postcure"	1	285
550	"Identify_required_items"	1	8
890	"Load_dry_fiber_spool_braiding_equipment"	2	0
930	"Load_two_part_resin_injection_equipment"	2	0
960	"Position_hoist_assisted_braiding_mandrel"	2	1
1000	"Position_hoist_assisted_part_NC_trim"	2	1
1170	"Position_manual_heat_lamps"	6	5
1240	"Position_manual_radius_filler"	2	2
1250	"Position_manual_RTM_mold_lid"	2	5
1260	"Position_manual_RTM_mold_seal"	2	4.762
1460	"Remove_hoist_assisted_braiding_mandrel"	2	1
1470	"Remove_hoist_assisted_part_from_NC_trim"	2	1
1500	"Remove_lines_resin_injection_RTM_mold"	2	0.668
1550	"Remove_lines_vacuum_RTM_mold"	2	1.67
1690	"Remove_manual_heat_lamps"	6	5
1770	"Remove_manual_RTM_mold_lid"	2	2.5
1780	"Remove_manual_RTM_mold_seal"	2	4.76
1950	"Setup_machine_braiding"	1	10
2040	"Setup_machine_NC_trimming"	1	15
2080	"Setup_machine_resin_injection"	1	5
2100	"Setup_tool_assembly_jig"	1	10
2340	"Trim_manual_gr/ep_deburr"	2	4
2350	"Trim_manual_gr/ep_deflash"	2	4

TABLE 3. File for first-order equations (Part II).

"Delay_min"	"Vo1"	"V1"	"Vo1a"	"V1a"	"Tau1"	"Crew"
blank	blank	blank	blank	blank	blank	1
blank	blank	blank	blank	blank	blank	1
blank	blank	blank	blank	blank	blank	1
0.1	612	"tool_area"	blank	blank	2.25	1
10	blank	blank	blank	blank	blank	1
0.25	289	"injection_mold_length"	4	"lines"	blank	1
0.25	289.2	"injection_mold_length"	4	"lines"	blank	1
11.1	1751	"part_area"	0.09	"attachments"	blank	2
blank	0.065	"fiber_weight"	blank	blank	blank	1
blank	493.3	"tool_area"	blank	blank	7.104	1
blank	167.6	"tool_area"	blank	blank	0.046	1
blank	blank	blank	blank	blank	blank	0.25
blank	blank	blank	blank	blank	blank	1
blank	blank	blank	blank	blank	blank	1
blank	blank	blank	blank	blank	blank	1
blank	2.5	"preform_fiber_weight"	2	"time_per_spool"	blank	1
blank	22.2	"resin_weight"	0.037	"#_resin_drum_sets"	blank	1
0.5	18	"tool_length"	1.35	blank	blank	2
1	14.38	"part_length"	blank	blank	blank	2
blank	20	"interface_length"	1	"#_heat_lamps"	blank	1
0.5	60	"filler_length"	blank	blank	blank	1
blank	8	"tool_lid_perimeter"	2	"#_bolts"	blank	1
blank	50.4	"seal_length"	blank	blank	blank	1
0.5	22.5	"tool_length"	blank	blank	0.54	2
1	36	"part_length"	blank	blank	blank	2
blank	723	"injection_mold_length"	10	"#_lines"	blank	1
0.13	578.4	"injection_mold_length"	8	"#_lines"	blank	1
0.5	50	"interface_length"	2.5	"#_heat_lamps"	blank	1
blank	8	"tool_lid_perimeter"	4	"tool_perimeter"	blank	1
blank	126	"seal_length"	blank	blank	blank	1
blank	blank	blank	blank	blank	blank	1
blank	blank	blank	blank	blank	blank	1
blank	blank	blank	blank	blank	blank	1
blank	blank	blank	blank	blank	blank	1
0.25	112	"trim_length"	blank	blank	0.16	1
0.25	112	"trim_length"	blank	blank	0.16	1

In addition to the time required per manufacturing step, we need to know the labor requirements. The crew size was divided into four categories:

- Unskilled labor — the number listed in Table 2 and Table 3.
- Skilled labor — default 0.0 persons
- Supervisory labor — default 0.1 persons
- Engineering labor — default 0.1 persons

The cost is then determined by the simple calculation in Eq 8.

$$LaborCost = \sum(CrewSize \cdot LaborRate) \qquad \textbf{(EQ 8)}$$

Table 4 is the labor conversion chart used in the model; all units are dollars per hour.

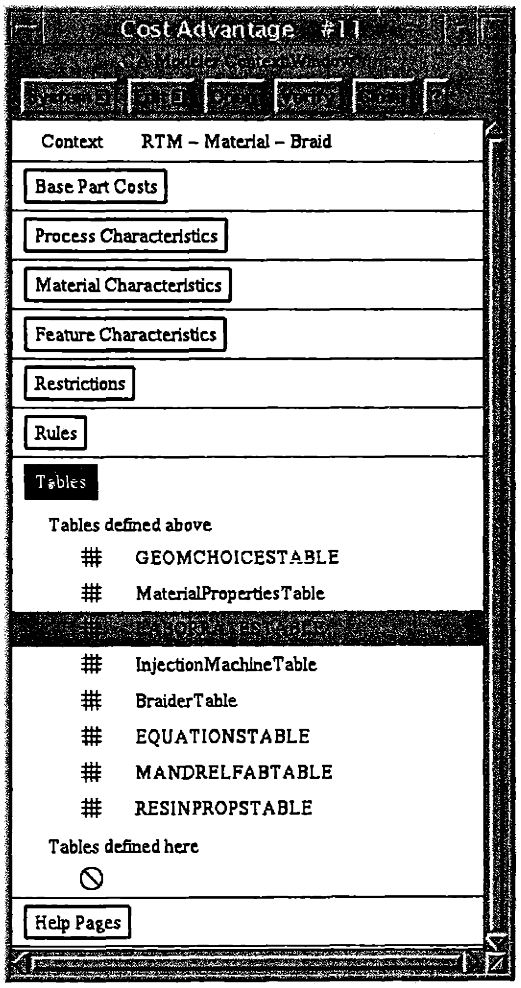
TABLE 4. Labor conversion table

"Facility"	"Unskilled"	"Skilled"	"Supervisory"	"Engineering"
"plantA"	50	70	90	110
"plantB"	55	75	95	115
"plantC"	60	80	100	120
"plantD"	65	85	105	125

4.0 Libraries

The libraries window corresponds to the library box on the overall flowchart.

FIGURE 5. A list of all of the (quasi-static) libraries in the model. [Model builder]



Context RTM – Material – Braid

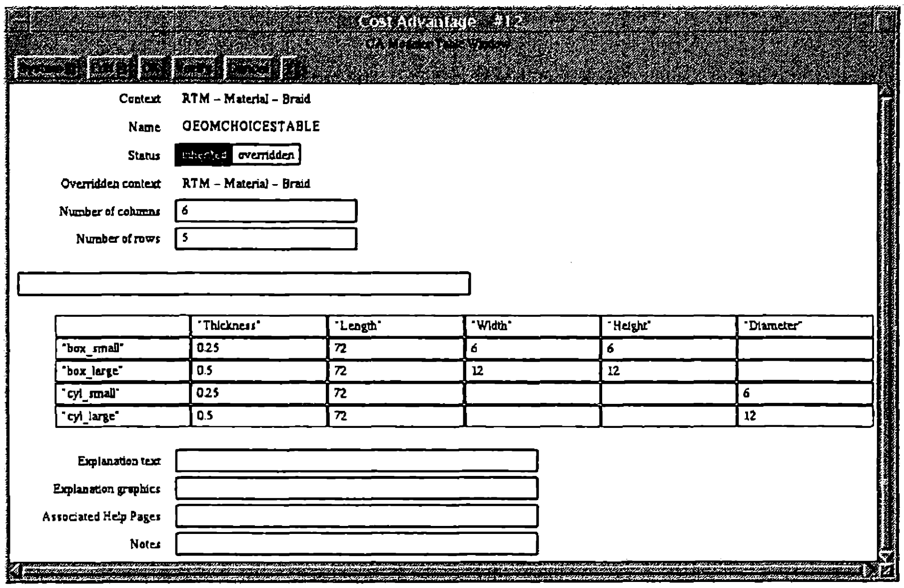
Tables defined above

- ## GEOMCHOICESTABLE
- ## MaterialPropertiesTable
- ## LABORRATESTABLE
- ## InjectionMachineTable
- ## BraiderTable
- ## EQUATIONSTABLE
- ## MANDRELFABTABLE
- ## RESINPROPSTABLE

Tables defined here

⦸

FIGURE 6. A sample library window, showing the choices of geometry. [Model builder]



Context RTM – Material – Braid

Name GEOMCHOICESTABLE

Status [illegible] overridden

Overridden context RTM – Material – Braid

Number of columns 6

Number of rows 5

	"Thickness"	"Length"	"Width"	"Height"	"Diameter"
"box_small"	0.25	72	6	6	
"box_large"	0.5	72	12	12	
"cyl_small"	0.25	72			6
"cyl_large"	0.5	72			12

Explanation text

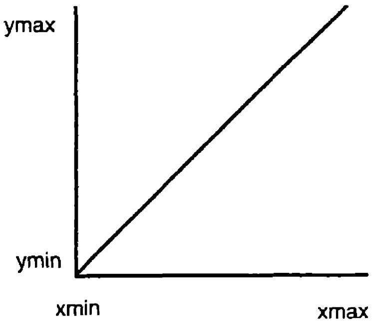
Explanation graphics

Associated Help Pages

Notes

Another use of the libraries would be to define several common types of equations for easy use. For example, a bell curve could be normalized and saved as a lookup table. When the program is run, a value could then be calculated based on the x-range, y-range, standard deviation, etc. This simplifies the amount of data which must be stored and standardizes any references to the variables describing the shape. Figure 7 shows how a value would be calculated for a linear graph.

FIGURE 7. Calculation of a value based on normalized graph.



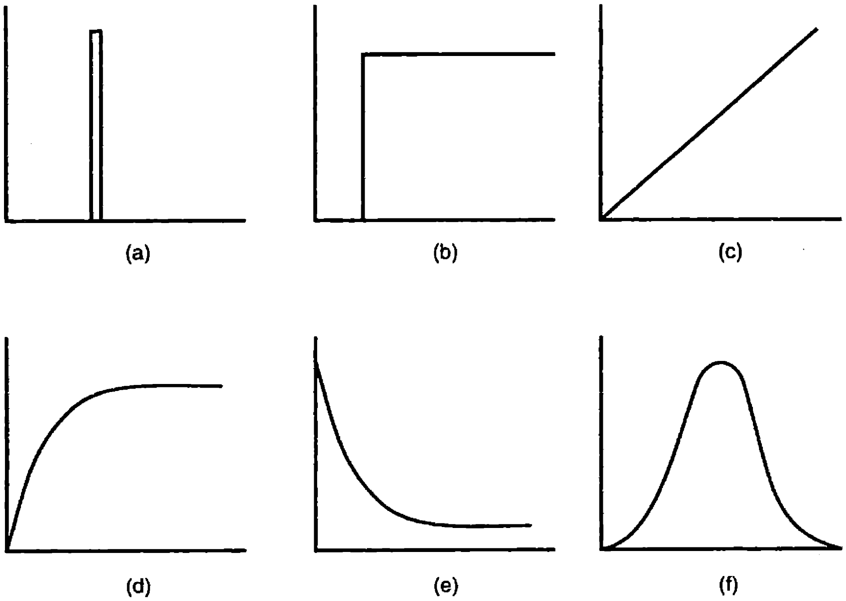
Generic formula — $y = mx + b$

Specific formula

$$(y - ymin) = \frac{(ymax - ymin)}{(xmax - xmin)} \cdot (x - xmin)$$

Figure 8 shows several basic shapes which would be very useful.

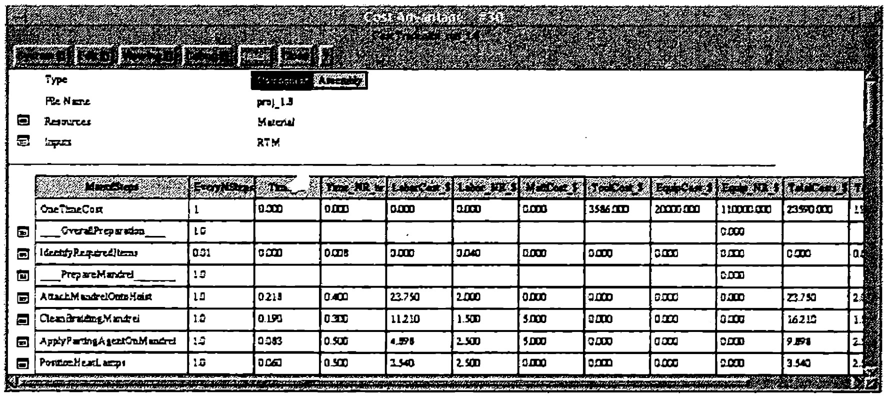
FIGURE 8. Several graph types which could be stored as lookup tables: (a) spike, (b) step, (c) linear, (d) asymptotic, (e) exponential, (f) bell curve.



5.0 Default process plan and cost summary

The first column shows all of the process steps in chronological order. The "default" plan is based on basic input information. Other plan steps may also appear based on selections in a previous process. For example, if you decided to drill a hole, you may have to center drill it first.

FIGURE 9. The process plan and summary table window. [Model user]



Type
File Name proj_1.3
Resources Material
Inputs RTM

ManufSteps	EveryNSteps	Tim...	Time_NR_hr	LaborCost_$	Labor_NR_$	MatlCost_$	ToolCost_$	EquipCost_$	Equip_NR_$	TotalCost_$
OneTimeCost	1	0.000	0.000	0.000	0.000	0.000	3586.000	20000.000	110000.000	23590.000
____OverallPreparation____	1.0								0.000	
IdentifyRequiredItems	0.01	0.000	0.008	0.000	0.040	0.000	0.000	0.000	0.000	0.400
____PrepareMandrel________	1.0								0.000	
AttachMandrelOntoHoist	1.0	0.218	0.400	23.750	2.000	0.000	0.000	0.000	0.000	23.750
CleanBraidingMandrel	1.0	0.190	0.300	11.210	1.500	5.000	0.000	0.000	0.000	16.210
ApplyPartingAgentOnMandrel	1.0	0.083	0.500	4.898	2.500	5.000	0.000	0.000	0.000	9.898
PositionHeatLamps	1.0	0.060	0.500	3.540	2.500	0.000	0.000	0.000	0.000	3.540

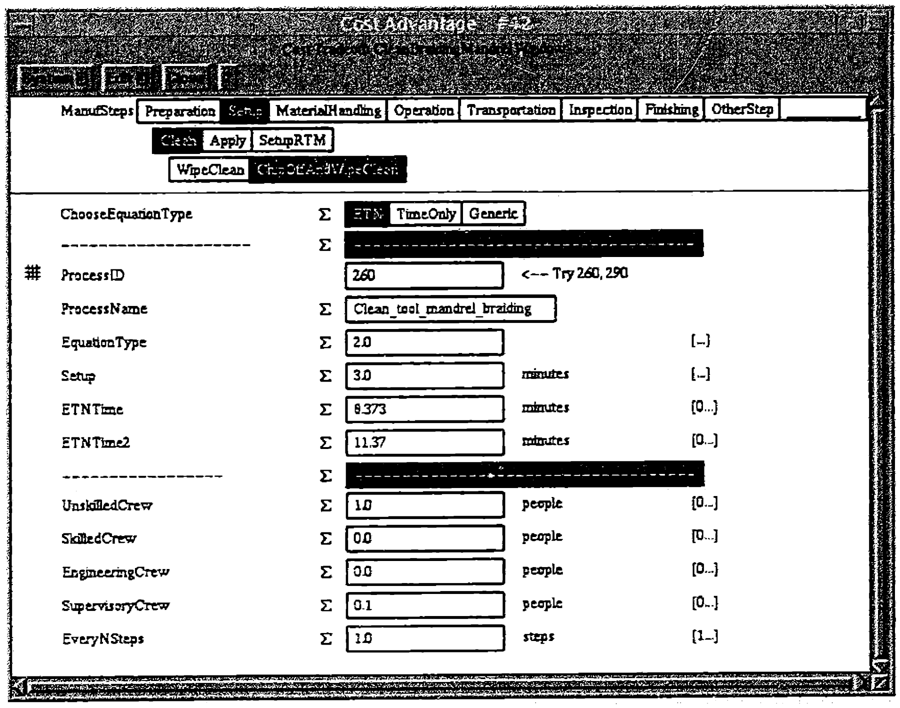
The other columns in Figure 9 comprise the summary table. Here, we have specified process time, process cost, material cost, equipment cost, and tooling cost.

The import file for the default process plan is listed in Appendix A.

6.0 Manufacturing steps

Click on one of the manufacturing steps in Figure 9 to the specific process window, shown in Figure 10.

FIGURE 10. Example “clean braiding mandrel” process step window. [Model user]

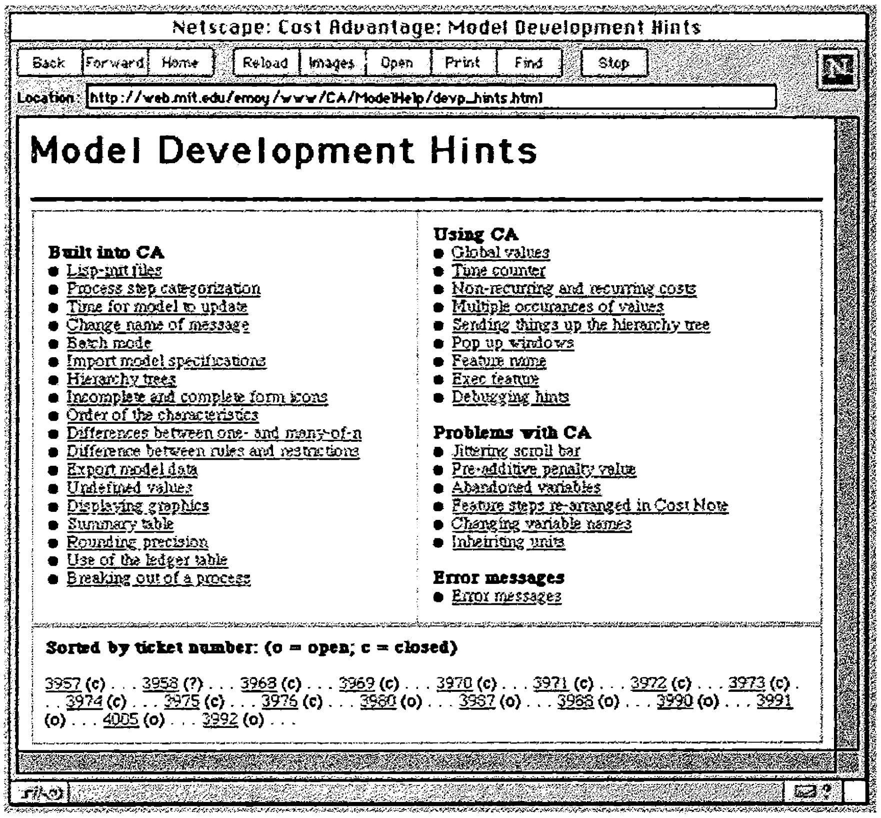


ChooseEquationType	Σ	ETN / TimeOnly / Generic		
ProcessID		260	<-- Try 260, 290	
ProcessName	Σ	Clean_tool_mandrel_braiding		
EquationType	Σ	2.0		[--]
Setup	Σ	3.0	minutes	[--]
ETNTime	Σ	8.373	minutes	[0...]
ETNTime2	Σ	11.37	minutes	[0...]
UnskilledCrew	Σ	1.0	people	[0...]
SkilledCrew	Σ	0.0	people	[0...]
EngineeringCrew	Σ	0.0	people	[0...]
SupervisoryCrew	Σ	0.1	people	[0...]
EveryNSteps	Σ	1.0	steps	[1...]

7.0 Help pages

The help pages in this example linked to a web page. This is convenient because after launching the web browser, any format of information accessible by the browser — text pictures, videos, animations, etc. — can be accessed. The information is also included in Appendix C.

FIGURE 11. Menu part of the help page. [Model user]



http://web.mit.edu/emoy/www/CA/ModelHelp/devp_hints.html

Model Development Hints

Built into CA
- Lisp-init files
- Process step categorization
- Time for model to update
- Change name of message
- Batch mode
- Import model specifications
- Hierarchy trees
- Incomplete and complete form icons
- Order of the characteristics
- Differences between one- and many-of-n
- Difference between rules and restrictions
- Export model data
- Undefined values
- Displaying graphics
- Summary table
- Rounding precision
- Use of the ledger table
- Breaking out of a process

Using CA
- Global values
- Time counter
- Non-recurring and recurring costs
- Multiple occurances of values
- Sending things up the hierarchy tree
- Pop up windows
- Feature name
- Exec feature
- Debugging hints

Problems with CA
- Jittering scroll bar
- Pre-additive penalty value
- Abandoned variables
- Feature steps re-arranged in Cost Note
- Changing variable names
- Inheiriting units

Error messages
- Error messages

Sorted by ticket number: (o = open; c = closed)

3957 (c) . . . 3958 (?) . . . 3968 (c) . . . 3969 (c) . . . 3970 (c) . . . 3971 (c) . . . 3972 (c) . . . 3973 (c) . . . 3974 (c) . . . 3975 (c) . . . 3976 (c) . . . 3980 (o) . . . 3987 (o) . . . 3988 (o) . . . 3990 (o) . . . 3991 (o) . . . 4005 (o) . . . 3992 (o) . . .

Chapter 8

Results

The results for Year 1 of the project includes the model architecture and preliminary cost estimates for our sample part, a large box beam.

This chapter includes:

- Inputs and resources
- Cost Advantage output
- Simple analysis

1.0 Inputs and resources

We implemented a sample for the cost model for resin transfer molding (RTM) using a large box beam. Briefly, we are making a total of 10 parts according to the geometry in Figure 1.

FIGURE 1. Schematic drawing of large box beam used to test cost model.

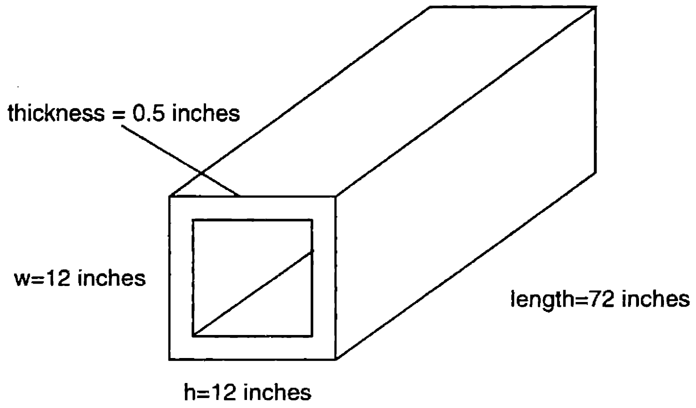


Table 1 and Table 2 show the detailed inputs and resources we used in our example cost model. The information grouped under "inputs" would fall to the designer or high-level planner. The information grouped under "resources" would probably fall to the manufacturing expert.

Note that in both tables, most of the information is left to the default values (second column is "yes"). Many of these defaults are calculated by equations or preset logic within the model.

TABLE 1. Inputs to the model. Not the actual input file. (D = default values; S = show variable)

Name	D	Value	Units	Range	S
VerboseMode	y	TRUE		[]	y
ShowCrewSize	y	TRUE		[]	y
ShowStepFrequency	y	TRUE		[]	y
------------------	y	-----------------------------------		[]	y
ShowInputOptions		Microstructure, Production, Economic, Geometry		[]	y
-------Geometry------	y	-----------GEOMETRY------------		[]	y
ShapeClass		box_large		[]	y
PartThickness	y	0.500	inches	[0...12]	y
PartLength	y	72.000	inches	[0...72]	y
PartWidth	y	12.000	inches	[0...12]	y
PartHeight	y	12.000	inches	[0...12]	y
-------Production------	y	-----------PRODUCTION----------		[]	y
Parts		10.000	parts	[1...500]	y
Run	y	1.000	runs	[1...10]	y
Operation	y	1.000	operations	[1...10]	y
Shipset	y	1.000	shipsets	[1...500]	y
Lot	y	1.000	lots	[1...10]	y
JobDuration	y	50.000	days	[20...200]	y
-----Microstructure-----	y	------MICROSTRUCTURE-------		[]	y
FiberVolFraction	y	50.000	percent	[40...65]	y
LongitudinalFibers	y	FALSE		[]	y
DesiredBraidAngle	y	60.000	degrees	[15...80]	y
FiberCoverage		0.000	percent	[...]	y
DesiredStiffness		0.000	?	[...]	y
DesiredStrength		0.000	?	[...]	y
-------Economic-------	y	-----------ECONOMIC------------		[]	y
ChooseFacility		plantA		[]	y
Labor	y	Labor	<-- Enter labor below	[]	y
UnskilledLabor	y	50.000	$/hr	[...]	y
SkilledLabor	y	70.000	$/hr	[...]	y
EngineeringLabor	y	110.000	$/hr	[...]	y
SupervisoryLabor	y	90.000	$/hr	[...]	y
Factory	y	Factory	<-- Enter factory below	[]	y
ProdLossPercentage		10.000	percent	[0...100]	y
ScrapPercentage		10.000	percentage	[0...100]	y
ReworkPercentage		10.000	percent	[0...100]	y
LearningCurve		fast		[]	y
OverheadRate		1.000	percent	[0...100]	y
Depreciation	y	10.000	percent	[0...100]	y
Inflation	y	5.000	percent	[0...100]	y

TABLE 2. Resources in model. Not the actual input file. (D = default values; S = show variable)

Name	D	Value	Units	Range	S
ShowResourceOptions		Tools,Equipment,Materials		[]	y
------Material-------	y	----------MATERIAL---------		[]	y
Resin	y	RESIN	<--Enter resin reqs below	[]	y
ResinWeight		10.000		[...]	y
ResinCostPerPound	y	50.000	dollars/pound	[0...]	y
Fiber	y	FIBER	<-- Enter fiber reqs below	[]	y
FiberWeight		0.000		[...]	y
FiberCostPerPound	y	10.000	dollars/ pound	[0...]	y
AuxiliaryMaterials	y	AUXILIARY_MATERIALS	<-- Enter aux matl reqs below	[]	y
AuxMatlFlatRate	y	50.000	$/job	[]	y
------Equipment-----	y	---------EQUIPMENT----------		[]	y
EquipFlatRate	y	10.000	$/hr	[0...50]	y
ShowAllPossibleEquipChoices	y	FALSE		[]	y
BraidingMachine	y	BRAIDING MACHINE	<-- Enter braider info below	[]	y
ChooseBraidingMachine		braiderA		[]	y
BraidNumOfSpools	y	72.000		[6...144]	y
BraidSpoolWindSpeed	y	60.000	rev/min	[0...]	y
BraidSpoolDiameter	y	3.000	inc	[0...]	y
BraidMaxLength	y	72.000	in	[0...]	y
BraidMinLength	y	12.000	in	[0...]	y
BraidMaxDiameter	y	20.000	in	[0...]	y
BraidMinDiameter	y	4.000	in	[0...]	y
BraidAngle	y	60.000	degrees	[0...90]	y
BraidFeedRate	y	10.000	in/min	[0...]	y
InjectionMachine	y	INJECTION MACHINE	<-- Enter injection info below	[]	y
ChooseInjectionMachine		injA		[]	y
InjectionHeatRate	y	5.000	deg/min	[0...]	y
InjectionCoolRate	y	5.000	deg/min	[0...]	y
InjectionTempMax	y	350.000	deg	[...]	y
InjectionTempMin	y	150.000	deg	[...]	y
InjectionPressureMax	y	5.000	psi	[...]	y
InjectionPressureMin	y	1.000	psi	[...]	y
InjectionNumOfPorts	y	2.000		[1...]	y

2.0 Cost Advantage output

A summary of the costs for our large box beam are shown in Table 3. The non-recurring costs are mostly capital expenses, as indicated in the first row ("OneTimeCost") in the full summary table (see Appendix A). A decision could be made to account for these costs separate from the per-part cost.

TABLE 3. Summary of total times and costs, based on their recurring or non-recurring nature.

Category	Total value	Notes
Time_hr	9.44	Labor associated with each step
Time_NR_hr	72.22	Non-recurring labor (e.g., setup time)
LaborCost_$	646.70	Labor associated with each step
Labor_NR_$	361.10	Non-recurring labor, e.g., setup time
MatlCost_$	35.00	Cost of materials to make part, as well as auxiliary materials (e.g., gloves)
ToolCost_$	3,586.00	Cost of the RTM mold
EquipCost_$	30.00	Per-hour charge for equipment usage
Equip_NR_$	110,000.00	Capital equipment costs (e.g., braiding machine)
TotalCosts_$	4,290.00	Sum of recurring costs
Total_NR_$	113,900.00	Sum of non-recurring costs

To estimate the cost per unit, we need to consider both the recurring and non-recurring costs:

$$\frac{Cost}{Unit} = Recur + \frac{NonRecur}{Units} \qquad \text{(EQ 1)}$$

In this example, the cost per unit (in dollars) would be

$$\frac{Cost}{Unit} = 4290 + \frac{113900}{10} = 15680 \qquad \text{(EQ 2)}$$

If we ignored the capital equipment costs, (that is, to assume that the equipment is already available in the factory), the cost per unit (in dollars) would be

$$\frac{Cost}{Unit} = 4290 + \frac{3900}{10} = 4680 \qquad \text{(EQ 3)}$$

which seems more reasonable, given that a person would not make capital equipment investments for a job of 10 parts.

3.0 Simple analysis

The analysis of our box beam was very limited, since we did not have enough range of data and logic to see the effects of changes in parameters. We set up simple Pareto charts based on the major steps corresponding to the output tables:

A. Overall preparation
B. Prepare mandrel
C. Prepare braider
D. Braid preform
E. Prepare mold
F. Capture preform in mold
G. Prepare injection machine
H. Heat and vacuum mold
I. Prepare resin
J. Inject resin and cure
K. Disassemble manifold
L. Remove part from mold
M. Post cure
N. Finishing

These were sorted by the amount of time per operation (Figure 2), then plotted according to the labor cost (Figure 3). At a glance, we note that processes M and J contribute more to the overall time than the others. However, we cannot tell whether significant savings can be made in these areas without going further into the details. For example, the time to cure (process M) is dependent on the resin type and temperature characteristics and not necessarily the processing technique.

The Pareto chart is also useful for identifying major trends. One question that comes out of this simple analysis is: Although the cure time -- steps M and J -- are very long, the labor costs should be lower since the curing process requires little attention or supervision.

A full analysis would require performing cost sensitivity tests of the major and minor steps of the process plan. This was not addressed at the current stage of the project and should be a high priority in the future.

FIGURE 2. Major steps in creating the box beam, sorted by time.

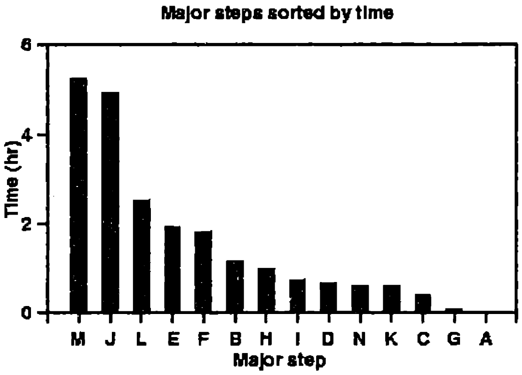
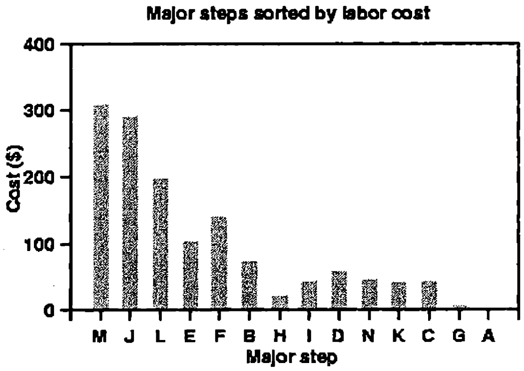


FIGURE 3. Major steps in creating the box beam, sorted by labor cost.



Chapter 9

Conclusion

Chapters 2, 3, and 4 briefly listed some of the background information about RTM, expert systems, and cost estimation. Chapters 5, 6, and 7 explained the implementation details which were executed in Year 1 of the project, and Chapter 8 wrapped up with the resulting data.

In this chapter, we review the work done so far, as well as list remaining ideas which have not been implemented but should be added to increase the accuracy or integrity of the model.

This chapter includes

- Milestone schedule
- Analysis
- Future considerations

1.0 Milestone schedule

The original milestones, as set for by Draper Laboratory, are listed in Table 1. We interpreted the milestones, and adjusted the scheduling based on the software that we chose. For example, the choice of software package dictated the amount of work which needed to be done with the graphical user interface.

TABLE 1. Original milestone schedule

#	Milestone	Start	End
1	Decide on format, computer system, software. Determine format of database and process plan.	7/96	10/96
2	Visit RTM factory and familiarize with RTM process.	9/96	12/96
3	Learn software & hardware. Begin software development.	10/96	1/97
4	Develop preliminary working process cost examples.	10/96	6/97
5	Define graphical user interface.	1/97	3/97
6	Further development software tool. Add data for preform, RTM, VaRTM, and hand layup.	3/97	10/97
7	Further refine software tool. Enhance part description, material database, and data display.	10/97	6/98
8	Generalize process plan and time estimation procedures for other processes.	1/98	6/98

The following subsections offer advice for future phases of the project.

1.1 Defining the goals

There are a few suggestions about the timing of the project. Although we did accomplish the Year 1 milestones and began on some of the Year 2 milestones, part of the difficulty of the project was that we, in conjunction with Draper Laboratory, did not really define the goals of the project in enough detail until after the first six months. The first few months were spent getting consensus from the different parties involved at MIT and Draper. We were trying to shape the software without a concrete idea of what functions it should have. While there may be a tendency to wait on the implementation until the goals are set, we found that there was not enough time in the project for that luxury.

1.2 Choice of software

The choice of software would have benefited from more comprehensive attention. We could not survey many of the choices popular in industry because of time restrictions or lack of an evaluation copy of the software. Also, part of the decision was fueled by a personal bias toward a high-level, expert system rather than low-level coding.

In retrospect, even after using *Cost Advantage* for half a year and touching on *PCAD*, I am not sure whether our choice of *CA* was the best for this project. The ability for *CA* to interface with CAD programs and databases was a definite draw. However, choosing *CA* requires a commitment towards the use of a knowledge-based system which goes beyond the manufacturing expertise needed for cost tradeoffs. Although the model user does have access to the values embedded in the software, there would need to be a designated model builder -- somebody who would efficiently make structural or semantic changes to the program -- even after the initial building of the model.

Perhaps what hurt us most in trying to build a rule-based system was the lack of rules. We had access to only a few experts in this particular project, but this is a problem which will be present in any "advanced manufacturing technology" where the rules may not have even been developed yet. Future discussions should consider whether a rule-based system is best suited for this type of application, compared to the alternatives (listed in Chapter 3).

1.3 On-site visit

The on-site visit to Emerson & Cuming was useful as a sanity check, to verify that our assumptions were reasonable. However, it was not as helpful as it could have been because Emerson & Cuming did not make the same type of parts as we were trying to model. Also, the trip came late during the project year. One important thing to consider not to go into the interview without having "done your homework." In this project, that meant that we had a rough process plan and a list of questions prepared ahead of time. There should be at least one other plant visit in the next phase of the project. Ideally, the company would have a personal interest in helping out and become a long-term partner in the development of the cost model.

1.4 Graphical user interface

A user-friendly graphical user interface (GUI) can make or break a software package. If the user cannot easily figure out what he is supposed to do, he will be more reluctant to use the software.

In this project, we had schedule GUI-development for the middle of Year 1. This is important, especially for the end-of-year demonstration. However, the final GUI implementation would be more useful toward the end of a project, after we have a better feel of how the user may want to navigate through the program. Also, most of the commercial software packages came with some sort of minimum user interface, but these need to be further added upon because of aesthetic or functional reasons. One important reason for expanding on the GUI is that the user may not want to see the results in raw data format, but only after some post-processing steps have been performed.

1.5 Additional students

The project is currently very narrowly defined, as we felt it was more important to close the (analysis) loop than fully flesh out the model. If we could have another student on the project, the work could be broken down as follows:

TABLE 2. Distribution of work for two students. Student 2 is coming onto the project in year 2.

Task	Student 1	Student 2
Overall design of system	x	
Resin transfer molding	x	
Hand layup or other manuf technology		x
Design user interface	x	
Database management; data pre-processing	x	x
Data post-processing	x	x
Feedback analysis automation		x
CAD/Witness interface		x
Documentation	x	x

Student 2 would need considerable time to get up to speed on Cost Advantage, expert systems (6.871 Knowledge Based Systems is recommended), and the second manufacturing technology. Also, an undergraduate student more familiar with building user interfaces (e.g., using Tcl/Tk or XMotif) could implement the look-and-feel of the program.

2.0 Analysis

In the first year of the project, we concentrated on accounting for many of the contributors toward cost and setting up the software architecture to facilitate this. However, we did not perform extensive analysis on the model, i.e., try to find the optimal combination of design inputs and resources to make the part. Indeed, this optimization would probably not be satisfactorily addressed until the end of Year 2 or the beginning of Year 3. In all practicality, there is no one "right" way to perform a cost optimization, compounded by difficulties inherent in the problem:

- The solution space is very large
- Many of the variables are interdependent (e.g., the relationship between braid angle, feed rate, and fiber volume fraction)
- Some of the variables have discrete values (e.g., the jump in cost between overhead cranes of different lifting capacities)
- The problem definition may change over time (e.g., workers learn to perform their tasks more efficiently over time or may in fact adjust the process plan in an *ad hoc* manner)

The remainder of this section summarizes several methods which have been used for analyzing other models, though not necessarily in the area of cost estimation.

Complexity factor. Use the geometry and layout of the fibers to estimate the "complexity" of the part, which then correlates to cost. The techniques here are borrowed from the field of information technology. [24] [31]

Decision theory. Each if-then rule is assigned a "certainty factor" which represents the strength of the relationship between the cause and the effect. The following example is given with 90 percent certainty.

```
if weight > 30 pounds
then need two people (0.9)
```

The rule weights are then combined similar to multiplying values in probability. However, the numerical values of the final "certainties" should not taken literally, but only used to differentiate the relative strengths of the statements they describe. [35]

Monte Carlo simulation. Using statistics of random events, choose a set of "experiments," or cost scenarios, to run using the baseline model. If the set is well chosen, then there is a good probability that a near-optimal solution is found.

Taguchi box analysis. Taguchi box analysis is a design-of-experiments technique which is used to narrow down the solution space. In the simplest conceptualization, we have three variables, each of which can have a `HIGH` and `LOW` value (sometimes also a `MEDIUM` value, which results in more experiments to run).

Data verification. If the model is used in parallel with a fully operational manufacturing operation, then the real-time data can be used to verify model data. This is especially important for quickly changing processes or data values.

Full factory simulation. One very important thing which we have deferred in this model is the role of the full factory simulation in identifying cost drivers and calculating bottom-line costs. One of the implicit assumptions we relied on was that we had an infinite factory: unlimited labor, unlimited equipment to meet production requirements, and a no-bottleneck production line. Of course, this factory does not exist in reality, and our model is limited without taking these factors into consideration.

3.0 Future considerations

In addition to "beefing up" the analysis engine, there are several other issues which should be addressed in the model, both practical and philosophical.

Inputs to the model. In the current model, the mechanical design is treated as a fixed input, which the process plan is changed to compensate. However, in a more complete model, all of the design and process variables should be interdependent.

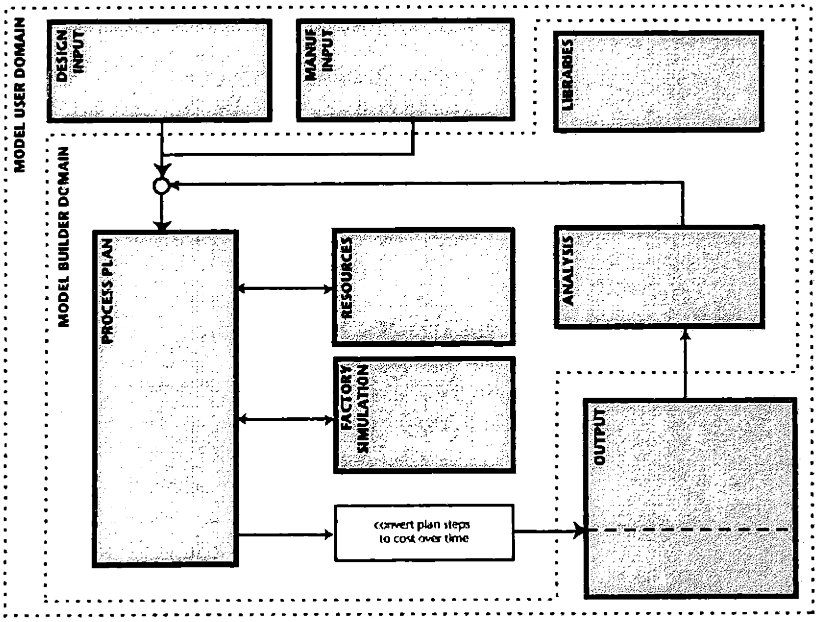
Instant model updates. Many cost models do not offer the option of seeing the effects of your inputs right away. That is, the analysis engine is separate from the input platform. On a basic level, *Cost Advantage* does automatically update the database after each new value is entered. However, for post-processing functions, where an outside script performs *CA* commands in a batch mode, automatic model updates may be more difficult.

Besides the technical aspects of updating the model data, why would you choose to have or not to have instant model updates? The advantage is that the effects of changes could be seen right away. The disadvantage is that constant messages and alerts which the model detects could be distracting to the user.

Automated process plan generation. Currently, our "automatic process plan" is created by loading a generic pre-saved skeleton of a model which the user then builds upon. Plan steps are then added, deleted, or otherwise adjusted based on rules and user inputs. Perhaps a future model could have process steps appear only when the rules concerning it are met, or the model could ask the user to choose between different process options.

Ability to run models at different levels of detail. Sometimes the user may be willing to trade a faster turnaround time for accuracy, or he may not know the answers to all of the questions asked. The model should be able to run at different levels of detail to accommodate this. The "unasked" questions may either have default values, or a certain variable could be estimate using a simpler method.

Taking into account all of the above suggestions, the flowchart might be redrawn as such:



4.0 Final comments

In the first year of this project, we have established a software architecture using *Cost Advantage* by Cognition Corporation. This framework, although first implemented using resin transfer molding, is flexible enough to accommodate other manufacturing processes. We faced many challenges stemming from the immense scope of engineering and computer knowledge required, and the project certainly could have benefited from a strong collaboration with an outside manufacturing company. Nevertheless, we feel that with the work done in this first year, the ground work is set for further implementation in the upcoming years of the project.

Appendix A

Oversized pages

- Process plan
- Results summary table

Process/Question	Assumptions	Src	Range	Time / Intermediate Variables	Input Variables	Material Variables	Aux Matl Variables	Tool Variables	Equipment Vars
Preparation									
Identify required items	Necessary items available	ETN 550		Eq 1 setup 8					
Mandrel preparation									
Load mandrel piece on overhead hoist	Overhead "equipment" == overhead hoist	ETN 160	0 to 500 min	Eq 2 setup 4, delay 11.1 vo1 1751 in^2/min, v1 part area vo1a 0.09 attachments, v1a attachments vo1b 19460 in^2/attach, v1b attach/in^2				Mandrel surface area	Overhead hoist • number attachments • attachments / in^2
Clean braiding mandrel	Chip away resin; clean w/ solvent	ETN 260	0 to 70 min	Eq 2 setup 3 vo1 493 in^2/min, v1 tool area tau 7.1			$100/job (BF)	Mandrel number pieces Mandrel surface area Mandrel weight Mandrel CG	Overhead hoist
Apply parting agent on mandrel		ETN 50	0 to 175 min	Eq 2 setup 5, delay 0.1 vo1 612 in^2/min, v1 tool area tau 2.25			$100/job (BF)	Mandrel number pieces Mandrel surface area Mandrel weight Mandrel CG	Overhead hoist
Position heat lamps		ETN 1170	0 to 130 min	Eq 6 setup 5 vo1 20 in/min, v1 interface length vo1a 1 lamp, v1a qty lamps vo1b 20 in/lamp, v1b lamp spacing				Mandrel length	Overhead hoist Heat lamps • number • spacing
Cure parting agent on mandrel	Using heat lamps; crew size 0.25	1) ETN 360 2) BF		1) Eq 1 setup 480 2) 10-15 min					Overhead hoist Heat lamps
Remove heat lamps		ETN 1690	0 to 500 min	Eq 6 setup 5, delay 0.5 vo1 50 in/min, v1 interface length vo1a 2.5 lamps, v1a qty lamps vo1b 20 in/lamp, v1b lamp spacing v2 % of length				Mandrel length	Overhead hoist Heat lamps • number • spacing
Assemble mandrel		BD							Overhead hoist
Prepare braider									
Setup braiding machine	Not sure what's assumed	ETN 1950		Eq 1 setup 10					Braider
Wind tow onto spools		FK		Time to wind spools		Preform fiber weight			Braider • number spools • max lbs/spool • winding speed
Replace spools	Knot tow and keep going	FK		Time to wind spools Time to knot tow		Preform fiber weight			Braider • number of spools
Load spools	Quick connect	ETN 890	0 to 20 min	Eq 2 vo1 2.5 lb/min, v1 preform fiber weight vo1a 2 spools/min, v1a time/spool vo1b 1.25 lbs/spool, v1b lbs/spool		Preform fiber weight			Braider • number spools • time/spool • lbs/spool
Thread tow into braider		FK							Braider • number of spools
Load mandrel into braider	crew size 2	ETN 960	0 to 150 min	Eq 2 setup 1, delay 0.5 vo1 18 in/min, v1 mandrel length tau 1.35				Mandrel weight Mandrel CG Mandrel length	Overhead hoist
Unload mandrel piece from overhead hoist	Assume same as loading	ETN 160	0 to 500 min	Eq 2 setup 4, delay 11.1 vo1 1751 in^2/min, v1 part area vo1a 0.09 attachments, v1a attachments vo1b 19460 in^2/attach, v1b attach/in^2				Mandrel surface area	Overhead hoist • number attachments • attachments / in^2

BD = Boothroyd & Dewhurst • BF = Bob Faiz • EC = Emerson Cuming • ETN = Ein Teck Neoh • FK = Frank Ko • PB = Paul Boyajian • TG = Tim Gutowski

Process/Question	Assumptions	Src	Range	Time / Intermediate Variables	Input Variables	Material Variables	Aux Matl Variables	Tool Variables	Equipment Vars
Braid preform onto mandrel									
Braid preform over mandrel	216 carrier, tri-axial braider	1) ETN 170 2) PB	0 to 800 min	1) Eq 2 setup 1 vo1 0.065 lb/min, v1 fiber weight 2) EQ(feed rate, v l, theta, type of braid, number of passes, total thickness)		Fiber weight			Braider
Load mandrel piece on overhead hoist	Overhead "equipment" == overhead hoist	ETN 160	0 to 500 min	Eq 2 setup 4, delay 11.1 vo1 1751 in^2/min, v1 part area vo1a 0.09 attachments, v1a attachments vo1b 19460 in^2/attach, v1b attach/in^2				Mandrel surface area	Braider Overhead hoist • number attachments • attachments / in^2
Remove mandrel from braider	For E, F, J, Z, C, L sections; crew size 2	ETN 1460	0 to 110 min	Eq 2 setup 1, delay 0.5 vo1 22.5 in/min, v1 tool length tau 0.54				Mandrel length	Overhead hoist Braider
Mold preparation									
Load tool onto overhead hoist	Overhead "equipment" == overhead hoist	ETN 160	0 to 500 min	Eq 2 setup 4, delay 11.1 vo1 1751 in^2/min, v1 mold area vo1a 0.09 attachments, v1a attachments vo1b 19460 in^2/attach, v1b attach/in^2				Mold surface area	Overhead hoist • numbeer attachments • attachments / in^2
Clean resin transfer mold	Chip away resin; clean w/ solvent	ETN 290	0 to 150 min	Eq 2 setup 3 vo1 168 in^2/min, v1 mold area tau 0.046			$100/job (BF)	Mold area	Overhead hoist
Clean o-ring groove	0.5 in o-ring diameter, divide above by groove width and divide rate by two	TG	0 to 300 min	Eq 2 setup 3 vo1 168 in/min, v1 groove perimeter tau 0.046			$100/job (BF)	Mold groove perimeter	Overhead hoist
Apply mold release to RTM mold		ETN 50	0 to 175 min	Eq 2 setup 5, delay 0.1 vo1 612 in^2/min, v1 mold area tau 2.25			$100/job (BF)	Mold area	Overhead hoist
Position heat lamps		ETN 1170	0 to 130 min	Eq 6 setup 5 vo1 20 in/min, v1 interface length vo1a 1 lamp, v1a qty lamps vo1b 20 in/lamp, v1b lamp spacing				Mold length	Overhead hoist Heat lamps • number • spacing
Cure parting agent on mold	Using heat lamps; crew size 0.25	1) ETN 360 2) BF		1) Eq 1 setup 480 2) 10-15 min					Overhead hoist Heat lamps
Remove heat lamps		ETN 1690	0 to 500 min	Eq 6 setup 5, delay 0.5 vo1 50 in/min, v1 interface length vo1a 2.5 lamps, v1a qty lamps vo1b 20 in/lamp, v1b lamp spacing v2 % of length				Mold length	Overhead hoist Heat lamps • number • spacing
Unload tool from overhead hoist	Overhead "equipment" == overhead hoist	ETN 160	0 to 500 min	Eq 2 setup 4, delay 11.1 vo1 1751 in^2/min, v1 tool area vo1a 0.09 attachments, v1a attachments vo1b 19460 in^2/attach, v1b attach/in^2				Tool surface area	Overhead hoist • numbeer attachments • attachments / in^2
Setup tool assembly jig		ETN 2100		Eq1 setup 10					Tool assembly jig
Assemble RTM mold		BD							

BD = Boothroyd & Dewhurst • BF = Bob Faiz • EC = Emerson Cuming • ETN = Ein Teck Neoh • FK = Frank Ko • PB = Paul Boyajian • TG = Tim Gutowski

Process/Question	Assumptions	Src	Range	Time / Intermediate Variables	Input Variables	Material Variables	Aux Matl Variables	Tool Variables	Equipment Vars
Capture preform in mold									
Set preform into mold	crew size 2	ETN 960	0 to 150 min	Eq 2 setup 1, delay 0.5 vo1 18 in/min, v1 mandrel length tau 1.35				Mandrel weight Mandrel CG Mandrel length	Overhead hoist
Unload mandrel piece from overhead hoist	Assume same as loading	ETN 160	0 to 500 min	Eq 2 setup 4, delay 11.1 vo1 1751 in^2/min, v1 part area vo1a 0.09 attachments, v1a attachments vo1b 19460 in^2/attach, v1b attach/in^2				Mandrel surface area	Overhead hoist • number attachments • attachments / in^2
Secondary preform placement	Place noodle	ETN 1240	0 to 45 min	Eq 2 setup 2, delay 0.5 vo1 60 in/min, v1 filler length		Filler length			
Position o-ring mold seal		ETN 1260	0 to 100 min	Eq 2 setup 4.76 vo1 50.4 in/min, v1 seal length				O-ring length	
Load top plate onto overhead hoist	Overhead "equipment" == overhead hoist	ETN 160	0 to 500 min	Eq 2 setup 4, delay 11.1 vo1 1751 in^2/min, v1 mold area vo1a 0.09 attachments, v1a attachments vo1b 19460 in^2/attach, v1b attach/in^2				Top plate surface area	Overhead hoist • numbeer attachments • attachments / in^2
Attach top plate to press									Overhead hoist Heated press, 300 ton ($100k — BF)
Unload tool from overhead hoist	Overhead "equipment" == overhead hoist	ETN 160	0 to 500 min	Eq 2 setup 4, delay 11.1 vo1 1751 in^2/min, v1 tool area vo1a 0.09 attachments, v1a attachments vo1b 19460 in^2/attach, v1b attach/in^2				Tool surface area	Overhead hoist • numbeer attachments • attachments / in^2 Heated press
Match mold with top plate on press	Alignment is part of the tool	ETN 1250	0 to 500 min	Eq 2 setup 5 vo1 8 in/min, v1 tool lid perimeter vo1a 2 bolts/min, v1a qty bolts vo1b 4 in/bolt, v1b bolt spacing				Tool lid perimeter Bolt number Bolt spacing	Heated press, 300 ton ($100k — BF)
Apply clamping force	Only force to close o-ring needed	BF		constant?					Heated press
Injection machine preparaton									
Setup resin injection machine		ETN 2080		Eq 1 setup 5					Heated press Resin injection machine
Attach vacuum line	Quick connect; manual hookup	ETN 140	0 to 4 min	Eq 2 setup 1.67, delay 0.25 vo1 289 in/min, v1 injection mold length vo1a 4 lines/min, v1a lines vo1b 72.3 in/line, v1b line spacing			Vacuum line • number • spacing	Tool length	Heated press Resin injection machine
Attach RTM line	Quick connect; manual hookup	ETN 90	0 to 4 min	Eq 2 setup 1.67, delay 0.25 vo1 289 in/min, v1 injection mold length vo1a 4 lines/min, v1a lines vo1b 72.3 in/linee spacing			Vacuum line • number • spacing	Tool length	Heated press Resin injection machine
Mold preparation									
Preheat platens (and tool)	Partial crew size				Start temp End temp				Heated press
Plug heater into mold	Integrated tool heater depends on size and resin temp req's								Heated press
Heat up mold	Partial crew size 5 to10°F/min			f(mass, thickness, rate)	Start temp End temp			Tool mass Tool thickness	Heated press
Evacuate mold	Partial crew size	ETN 80		Eq 1 setup 5, delay 10					Heated press Vacuum pump

BD = Boothroyd & Dewhurst • BF = Bob Faiz • EC = Emerson Cuming • ETN = Ein Teck Neoh • FK = Frank Ko • PB = Paul Boyajian • TG = Tim Gutowski

Process/Question	Assumptions	Src	Range	Time / Intermediate Variables	Input Variables	Material Variables	Aux Matl Variables	Tool Variables	Equipment Vars
Resin preparation									
Preheat resin (to lower viscosity)	partial crew size; 10°F/min			constant?	Start temp End temp				Heated press Resin injection machine
Mix two parts of resin	partial crew size			constant?					Heated press Resin injection machine Positive displ. pump
Vacuum degas resin	Worker performs other tasks	ETN 80		Eq 1 setup 5, delay 10					Heated press Resin injection machine Positive displ. pump Vacuum pump
Load mixed resin onto injection machine		ETN 930	0 to 2.5 min	Eq 2 vo1=22 2 lb/min, v1 resin weight vo1a 0 037 drums/min, v1a qty of drums vo1b 600 lbs/drum, v1b weight/drum		Resin drums quantity Resin drum weight Resin weight			Heated press Resin injection machine Positive displ pump
Inject resin and cure									
Inject resin	~1lb/min; upper limit = gel time					Resin x2			Heated press Resin injection machine Pneumatic plunger with heated lines
Cure in RTM mold	3501-6 resin	ETN 380		Eq 1 setup 285	Start temp End temp				Heated press
Disassemble manifold									
Detach vacuum line		ETN 1550	0 to 6 min	Eq 2 setup 1.67, delay 0 13 vo1 578 in/min, v1 injection mold length vo1a 1 line/min, v1a qty lines vo1b 72.3 in/line, v1b line spacing			Line quantity Line spacing	Tool length	Heated press
Detach RTM line		ETN 1500	0 to 4 min	Eq 2 setup 0.67 vo1 723 in/min, v1 injection mold length vo1a 1 line/min, v1a qty lines vo1b 72.3 in/line, v1b line spacing			Line quantity Line spacing	Tool length	Heated press
Unplug heater from tool	(handling)								Heated press
Remove RTM mold lid	Alignment is part of the tool	ETN 1250	0 to 500 min	Eq 2 setup 5 vo1 8 in/min, v1 tool lid perimeter vo1a 2 bolts/min, v1a qty bolts vo1b 4 in/bolt, v1b bolt spacing				Tool lid perimeter Bolt number Bolt spacing	Heated press
Remove tool from heating press	(handling)								Heated press
Remove part from tool									
Move tool into cooling press									Cooling press
Close top plate	Generic plate assumed								Cooling press
Cool down part	Cool down until 150°F -5°F/min (forced cooling) -1°F/min (natural convection)								Cooling press
Open top plate									Cooling press
Remove tool from cooling press									Cooling press

BD = Boothroyd & Dewhurst • BF = Bob Faiz • EC = Emerson Cuming • ETN = Ein Teck Neoh • FK = Frank Ko • PB = Paul Boyajian • TG = Tim Gutowski

Process/Question	Assumptions	Src	Range	Time / Intermediate Variables	Input Variables	Material Variables	Aux Matl Variables	Tool Variables	Equipment Vars
Load mandrel piece on overhead hoist	Overhead "equipment" == overhead hoist	ETN 160	0 to 500 min	Eq 2 setup 4, delay 11.1 vo1 1751 in^2/min, v1 part area vo1a 0.09 attachments, v1a attachments vo1b 19460 in^2/attach, v1b attach/in^2				Mandrel surface area	Overhead hoist • number attachments • attachments / in^2
Remove part from mold	For E, F, J, Z, C, L sections; crew size 2; tradeoff in ease of effort w/ "Remove mandrels"	ETN 1460	0 to 110 min	Eq 2 setup 1, delay 0.5 vo1 22.5 in/min, v1 tool length tau 0.54				Tool length	Overhead hoist Hydraulic mandrel puller
Remove mandrels	Based on surface area and relative shrinkage	FK							Overhead hoist Hydraulic mandrel puller
Unload mandrel piece from overhead hoist	Overhead "equipment" == overhead hoist	ETN 160	0 to 500 min	Eq 2 setup 4, delay 11.1 vo1 1751 in^2/min, v1 part area vo1a 0.09 attachments, v1a attachments vo1b 19460 in^2/attach, v1b attach/in^2				Mandrel surface area	Overhead hoist • number attachments • attachments / in^2
Remove o-ring		ETN 1780	0 to 40 min	Eq 2 setup 4.76 vo1 126 in/min, v1 seal length					
Post cure									
Put part in fixture									
Post cure		ETN 390		Eq 1 setup 285					Heated press (?)
Remove part from fixture									
Finishing steps									
Load part into overhead crane								Part length Part weight	Overhead crane
Deflash		ETN 2350	0 to 450 min	Eq 2 setup 4, delay 0.25 vo1 112 in/min, v1 trim length tau 0.16	Part length				
Manual debur		ETN 2340	0 to 450 min	Eq 2 setup 4, delay 0.25 vo1 112 in/min, v1 trim length tau 0.16	Part length				
Setup NC trim		ETN 2040		Eq 1 setup 15					NC machine
Load part into NC	crew size 2	ETN 1000		Eq 2 setup 1, delay 1 vo1 14.4 in/min, v1 part length	Part length				
NC trim									
Remove part from NC		ETN 1470		Eq 2 setup 1, delay 1 vo1 36 in/min, v1 part length	Part length				

BD = Boothroyd & Dewhurst • BF = Bob Faiz • EC = Emerson Cuming • ETN = Ein Teck Neoh • FK = Frank Ko • PB = Paul Boyajian • TG = Tim Gutowski

Summary

ManufSteps	EveryNSteps	Time_hr	Time_NR_hr	LaborCost_$	Labor_NR_$	MatlCost_$	ToolCost_$	EquipCost_$	Equip_NR_$
OneTimeCost	1	0	0	0	0	0	3586	0	110000
____OverallPreparation____	1	0	0	0	0	0	0	0	0
IdentifyRequiredItems	1.00E-10	0	0	0	0	0	0	0	0
____PrepareMandrel________	1	0	0	0	0	0	0	0	0
AttachMandrelOntoHoist	1	0.218	0.4	23.75	2	0	0	0	0
CleanBraidingMandrel	1	0.19	0.3	11.21	1.5	5	0	0	0
ApplyPartingAgentOnMandrel	1	0.083	0.5	4.898	2.5	5	0	0	0
PositionHeatLamps	1	0.06	0.5	3.54	2.5	0	0	0	0
CurePartingAgentOnMandrel	1	0	1.5	0	7.5	0	0	0	0
RemoveHeatLamps	1	0.116	0.5	6.844	2.5	0	0	0	0
____PrepareBraider________	1	0	0	0	0	0	0	0	0
SetUpBraidingMachine	1.00E-10	0	0	0	0	0	0	0	0
WindTowOntoSpools	0.01	0.002	0	0.147	0	0	0	0	0
LoadSpoolsOntoBraider	0.01	0	0	0.004	0	0	0	0	0
ThreadTowIntoBraider	0.01	0.002	0	0.147	0	0	0	0	0
PositionMandrelIntoBraider	1	0.079	0.1	8.629	0.5	0	0	0	0
DetachMandrelFromHoist	1	0.218	0.4	23.75	2	0	0	0	0
____BraidPreform_________	1	0	0	0	0	0	0	0	0
BraidPreformOverMandrel	1	0	0.1	0	0.5	0	0	0	0
AttachMandrelOntoHoist	1	0.218	0.4	23.75	2	0	0	0	0
RemoveMandrelFromBraider	1	0.073	0.1	7.998	0.5	0	0	0	0
____PrepareMold__________	1	0	0	0	0	0	0	0	0
CleanResinTransferMold	1	0.191	0.3	11.27	1.5	5	0	0	0
CleanORingGroove	1	0.949	0.3	56.02	1.5	5	0	0	0
ApplyPartingAgentOnMold	1	0.083	0.5	4.898	2.5	5	0	0	0
PositionHeatLamps	1	0.06	0.5	3.54	2.5	0	0	0	0
CurePartingAgentOnMold	1	0	1.5	0	7.5	0	0	0	0
RemoveHeatLamps	1	0.032	0.5	1.908	2.5	0	0	0	0
____CapturePreformInMold__	1	0	0	0	0	0	0	0	0
SetPreformInMold	1	0.079	0.1	8.629	0.5	0	0	0	0
DetachMandrelFromHoist	1	0.218	0.4	23.75	2	0	0	0	0
SecondaryPreformPlacement	1	0.028	0.2	1.672	1	0	0	0	0
PositionORingMoldSeal	1	0.059	0.476	3.473	2.381	0	0	0	0
AttachTopPlateOntoHoist	1	0.218	0.4	23.75	2	0	0	0	0
AttachTopPlateOntoPress	1	0.25	0	14.75	0	0	0	0	0
AlignMoldAndTopPlate	1	0.371	0.5	21.88	2.5	0	0	0	0
ApplyClampingForce	1	0.25	0	14.75	0	5	0	0	0
____PrepareInjMachine_____	1	0	0	0	0	0	0	0	0
SetUpResinInjMachine	1.00E-10	0	0	0	0	0	0	0	0
AttachVacuumLine	1	0.009	0.167	0.506	0.835	0	0	0	0
AttachRTMLine	1	0.009	0.167	0.506	0.835	0	0	0	0
____HeatAndVacuumMold_____	1	0	0	0	0	0	0	0	0
PreheatPlatensAndMold	1	0.233	0	5.017	0	0	0	2.333	0
HeatUpMold	1	0.5	0	10.75	0	0	0	5	0
EvacuateMold	1	0.167	0.5	3.583	2.5	0	0	1.667	0
____PrepareResin__________	1	0	0	0	0	0	0	0	0
PreheatResin	1	0.233	0	5.017	0	0	0	2.333	0
MixTwoPartsResin	1	0.25	0	32.25	0	0	0	2.5	0
VacuumDegasResin	1	0.167	0.5	3.583	2.5	0	0	1.667	0
LoadResinIntoInjMachine	0.04	0	0	0.018	0	0	0	0	0
____InjResinAndCure_______	1	0	0	0	0	0	0	0	0
InjectResin	1	0.167	0	9.833	0	0	0	1.667	0
CureInMold	1	0	28.5	0	142.5	0	0	0	0
____DisassembleManifold___	1	0	0	0	0	0	0	0	0
DetachVacuumLine	1	0.004	0.167	0.258	0.835	0	0	0	0
DetachRTMLine	1	0.002	0.067	0.104	0.334	0	0	0	0
RemoveMoldLid	1	0.371	0.5	21.88	2.5	0	0	0	0
RemoveMoldFromPress	1	0.079	0.1	8.629	0.5	0	0	0	0
____RemovePartFromMold____	1	0	0	0	0	0	0	0	0
MoveMoldIntoCoolingPress	1	0.25	0	14.75	0	0	0	0	0
AlignMoldAndTopPlate	1	0.371	0.5	21.88	2.5	0	0	0	0
ApplyClampingForce	1	0.25	0	14.75	0	5	0	0	0
CoolPart	1	0.5	0	29.5	0	0	0	5	0
RemoveMoldFromPress	1	0.079	0.1	8.629	0.5	0	0	0	0
AttachMandrelOntoHoist	1	0.218	0.4	23.75	2	0	0	0	0
RemovePartFromMold	1	0.073	0.1	7.998	0.5	0	0	0	0
RemoveMandrel	1	0.5	0	54.5	0	0	0	0	0
RemoveORing	1	0.024	0.476	1.389	2.38	0	0	0	0
____PostCure______________	1	0	0	0	0	0	0	0	0
PositionPartInFixture	1	0.25	0	14.75	0	0	0	0	0
PostCure	1	0	28.5	0	142.5	0	0	0	0
RemovePartFromFixture	1	0.25	0	14.75	0	0	0	0	0
____Finishing_____________	1	0	0	0	0	0	0	0	0
Deflash	1	0.017	0.4	1.019	2	0	0	0.173	0
Deburr	1	0.017	0.4	1.019	2	0	0	0.173	0
SetUpNCTrim	1.00E-10	0	0	0	0	0	0	0	0
PositionPartInNC	1	0.1	0.1	10.91	0.5	0	0	0	0
NCTrim	1	0.25	0	14.75	0	0	0	2.5	0
RemovePartFromNC	1	0.05	0.1	5.45	0.5	0	0	0	0
Total	1	9.439	72.22	646.7	361.1	35	3586	30	110000

Appendix B

Model builder's guide

This chapter is a user's manual for the model builder. It is meant to supplement Cognition's User Manual, Chapter 4. One important assumptions in the later sections is that the reader understands Cost Advantage's concept of the Process-Material-Feature "tuple" and inheritance of the tuple.

In addition to a discussion on tuples, Table 1 shows a summary of the other topics in this Appendix.

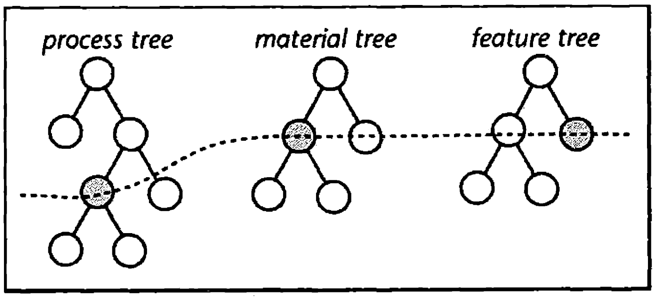
TABLE 1. Summary of topics in this Appendix.

Category	Specific questions
Basic functionality	One-of n and many-of-n options
	Rules and restrictions
	Displaying graphics
	Rounding precision
	Pre-additive penalty value
	Summary table
	Ledger table
Advanced functionality	Global constants and values
	Undefined values
Accessing external files	Lisp-init file
	Import model
	Export model
	Batch mode
	Messages
The way things look	Change label names
	Characteristics order
	Pop up windows
	Display verbosity switches
Designing the model	Recurring and non-recurring costs
	Form icons
	Hierarchy tree design
	Passing values up the hierarchy tree
Implementation problems	Time counter
	Multiple occurrences of variables
Known Cost Advantage bugs	Inheriting units
	Jittering scroll bar
	Features rearranged
	Ghost variables
	Accidental erasure of feature costs
If Cost Advantage should crash	Breaking out of a process
	Time for model to update
	Error messages at the Unix prompt

1.0 Explanation of CA tuples

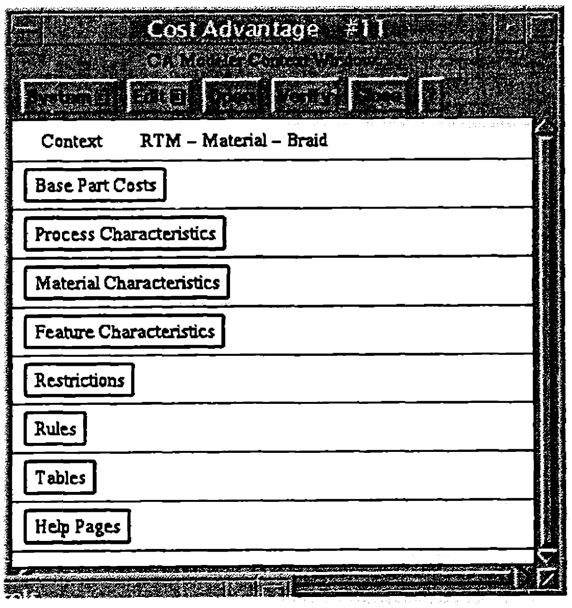
Rules and variables are defined by a "tuple" of parameters — process, material, and feature — as shown in Figure 1.

FIGURE 1. Graphical representation of a tuple.



Inheritance is determined by hierarchy trees, i.e., at any P-M-F level, variables defined at or above that tuple of nodes are valid. Each tuple corresponds to a "context window," shown at right. Under each "context category," Variables are either "defined above" or "defined here." A rule or variable can override its previous definition. (All of the tuple's children will then inherit the new definition.) Figure 2 shows a sample context window.

FIGURE 2. Sample context window showing the Component-Material-Braid_preform tuple.



The context categories include:

- Base part costs: The columns of the output table, categories for cost.
- Process, Material, Feature characteristics: These are categories for variables. They are understood by all other variables at that level.
- Restrictions: These limitations on values or combinations of variables are either hard (user must undo) or soft (user chooses to override).
- Rules: Rules test for situations that affect costs, characteristics, and manufacturability. Rules specify specific outcomes.
- Tables: These are used to store data in a compact form. All columns and rows are accessible as variable values.
- Help pages: These may contain text, graphics, or links to other programs.

2.0 Basic functionality

This section lists straight-forward functions of Cost Advantage which are explained in the CA User's Manual. They are listed here for emphasis, to point out the limits of the functionality, or to include a non-standard use of the functionality.

2.1 One-of n and many-of-n options

? When would you choose to use the `one-of-n` or `many-of-n` options?

? Does the ManyOfN function treat its options as text strings or elements of a set? I.e., would I use something like `A = "foo"` or `"foo" Member A`?

The function treats its options as elements of a set; i.e., you would use

```
"foo" Member A
```

2.2 Rules and restrictions

? What is the difference between a rule and a restriction?

A *rule* adjusts the value of a cost or characteristic based on one or more conditions.

A *restriction* is a process limitation.

2.3 Displaying graphics

? Problems displaying bitmap images

? Calls to external graphics programs

Graphics cannot be displayed in bitmap format from within CA, as described in Cognition's User's Manual. Normally they would be called at the prompt using:

```
displaybm <filename>
```

A better solution, in terms of resolution and flexibility of display, is to save the files in a form which is readable by xv or a web browser. This greatly increases the range of media to jpg, gif, movies, hyperlinked text — whatever is readable by the viewer.

For aesthetic purposes, use a border on all images, since XV uses the smallest enclosing rectangle otherwise.

2.4 Rounding precision

- ? Controlling the round-off precision
- ? Problems with round-off precision in the Quantity field

The round-off precision depends on the value set in "Significant Digits" field of the `Viewing->SetupWindow` dialog box (model user). The user may specify both the total digits and the number of digits after the decimal.

However, this functionality does not seem to affect the Quantity row. For fractional quantities (e.g., an operation which occurs once every two parts would be assigned a Quantity of 0.5), the number rounds off to eight significant digits. This is more an aesthetic question, though numbers such as (1.0000199 e-10) wouldn't have a great impact on the final cost anyway.

2.5 Pre-additive penalty value

- ? What is the "pre-additive value?"

The "pre-additive value" is the value of a cost (in the Summary Table) that would have been used if no rules were applied on that cell. The Summary Table cells are defaulted at "?" which can create a problem when trying to add value to it. E.g.,

```
Rule: TestAdd
Variables: P = Some(Feature)
If: TRUE
Then: Add(ProcessCost(P),100)
```

If the Process Cost of a feature was not previously defined, then the result of TestAdd would not be a valid statement for Cost Advantage (? + 100 = ?). The workaround is to default all Costs to 0. However, this clutters up the SummaryTable and confuses whether a value is zero because it was zero or just undefined.

2.6 Summary table

- ? Show a different number of columns than defined under BasePart Cost
- ? Add together selected columns
- ? Keep track of partial model costs which can be accessed by equations

The summary table appropriates costs to each feature. In this model, we track Labor_hr, LaborCost_$, MatlCost_$, ToolCost_$, and EquipCost_$ for each manufacturing process step. The columns in the table are the same in number and order as in the BasePart Costs. This cannot be changed.

Cost Advantage provides a default "Total" column (on the far-right of the SummaryTable) will automatically sum all of the values in one row given that the values have the same units. However, since our model uses "Cost" for both time and dollar units, the "Total" column remains blank. The solution was to create a new cost named "TotalCost" which would sum the relevant columns. The "Total" column remains unused.

The motivation for wanting to keep track of partial model costs is to calculate the added value of the part at different points in the manufacturing process. This can be used to make decisions about inspection steps or whether it would be cost effective to rework defective parts.

2.7 Ledger table

- ? Function of the ledger table
- ? Problems with changing number of columns

The Ledger Table is used to hold results from the Summary Table — either from individual rows or from a summation of selected rows — for comparison. Note that CA copies the values, and not a reference to the variables, in a given row of the Summary Table into the Ledger Table. Thus, the individual values cannot be accessed by the model. Also, if the number, order, or names of the columns should change, this will not be reflected in the Ledger Table.

Unfortunately, Cognition does not plan to include spreadsheet-like capabilities in Cost Advantage. The user should export the data to a spreadsheet for further analysis.

3.0 Advanced functionality

The items in this section are features which may be familiar to programmers of other programming languages or software. They require a little more knowledge about how Cost Advantage works, and may also require that the model builder can write or recognize simple Lisp code.

3.1 Global constants and values

- ? Define global constants
- ? Define global variables

The lisp-init file, located in the home directory, is loaded by Cost Advantage when the program starts up. Constants can be defined in this file using standard Lisp code.

```
(defconstant lightspeed 300000000)
```

Some constants are already pre-defined in CA, such as pi. Variables can also be defined in the lisp-init file in a similar fashion.

```
(defvar heating_rate)
```

The variable can also be initialized. The example below states that the heating rate is 10 degrees per minute.

```
(defvar heating_rate 10)
```

Alternatively, global variables can be placed at a relatively high level of the hierarchy tree. As a variable's function or scope is more limited, the characteristic should be placed lower on the tree to speed up the operation of the program.

3.2 Undefined values

? Defined a variable, but leave its value as NULL

Undefined variables are useful when a variable might or might not exist in all cases. Another variable could be triggered off of whether the first exists. The undefined value is noted as "?" as shown in the example below. In this case, if the weight of a part is greater than 50 pounds, the crane is used for 10 minutes; if greater than 100 pounds, 20 minutes. Otherwise a crane is not needed at all.

```
weight = 100
crane = {10 WHEN weight > 50;
     20 WHEN weight > 100;
     ? OTHERWISE}
```

4.0 Accessing external files

After you get a good feeling of how Cost Advantage works, the process of entering data or even parts of the model may turn into a repetitive series of steps. There are several data import methods to help in this.

4.1 Lisp-init files

? Enter additional lisp code

? Is there a difference between these two commands?

The lisp-init file was previously mentioned in the "Global constants and values" section. The file, located in the same directory as the Cost Advantage executable, contains snippets of Lisp code that augment the source code. Example of code to include or modify are:

- Global constants and variables
- Constants governing the graphical layout of CA user interface
- "Patches," or bug fixes, provided by Cognition

The model builder should be very careful in defining new constants or procedures, since the new definition will override any defaults and may adversely affect the operation of the program.

The second question asks if there is a difference between:

```
(in-package "CMG")
(in-package :cmg)
```

Both of these Lisp statements are mentioned in the technical tips. The only difference was a change in Cognition's Lisp compiler. Both statements should work.

4.2 Import model

? Save import files separately for ease of editing

? Blank spaces are not allowed in field values

? Access to the Process and Material Characteristics while not at the root level

One of the most convenient features of Cost Advantage is that after the model is designed, all of the information can be entered via import files. This separates the software architecture from the user interaction, and allows more flexibility in the program. For example, data files include:

- Time/cost conversion table
- Common geometry choices
- Labor rates

The import file for labor rates is shown in figure x. One way to easily update this data is to make separate files for each table, and then use the Unix command:

```
cat A B C > all
```

and import "all" into the model. More information is included in Technical Tip number 951010

FIGURE 3. Labor rates table import file

```
"Facility" "Unskilled" "Skilled" "Supervisory" "Engineering"
"plantA"   50 70 90 110
"plantB"   55 75 95 115
"plantC"   60 80 100 120
"plantD"   65 85 105 125
```

One point of caution concerns the field values. Blank fields are not allowed in tables (i.e., the table must be rectangular). Also text strings cannot include blank spaces (i.e., "plantA" not "plant A").

One must also be careful in specifying the tuple into which the data is imported. Specifically, in the normal operation of the modeler, the builder is not allowed to create characteristics at the non-root level of the Process and Material hierarchies. However, this is possible when importing data from a file. Be careful that all tuples are in the form:

```
Process-Material-<some feature>
```

Otherwise, the results are unpredictable.

4.3 Export model

- ? Not possible to export model data

The motivation behind this option is to be able to move data from one Cost Advantage model to another, or from a CA model to an external program such as a spreadsheet. We are assuming the data was found within the model, but was not available in a text file in the first place.

Cost Advantage is very limited in reading in model summaries to create new models. The options currently available are to print or save-as-file the model summaries. The user would then need to post-process using external files or SQL databases.

4.4 Batch mode

- ? Use of batch mode
- ? Specify a partial cost report using batch mode

The motivation for using batch mode is to automatically create CA runtime data, once the model is fully built. This pre-processing shell would automatically run a set of data or provide other data input/output menus. For example, in a sensitivity analysis, certain variables would be stepped through a range of values.

The CA Batch/Design Command Language provides a facility for directing input to the Cost Advantage runtime system. Design Files contain a variety of inputs and directions for saving to an output file. You might want to even have the creation of the Design File automated via a script. The Design Command Language is described in the appendix of the CA User's Guide.

Question: Can the model builder specify a partial cost report while using batch mode? i.e., not all values from the output file are needed for post-processing

5.0 The way things look

The items in this section answer questions about changing the look-and-feel of Cost Advantage.

5.1 Change label names

- ? Change the names of labels in the cost note

The motivation behind this option is to be able to customize the Cost Note for a specific application. In this project, we used the following substitutions:

- Process —> Inputs
- Material —> Resources
- Feature —> ManufSteps

The names of various labels in the cost note are controlled by a data file, set at the default of:

```
<install-dir>/data/messages/
```

specifically in the file:

```
<install-dir>/data/messages/database.msg.1
```

The model builder can either change the original "message" file or copy it to a local directory and redirect Cost Advantage accordingly. For example, in order for CA to recognize message files located at:

```
~/Messages
```

the following command should be typed at the Unix prompt (case sensitive):

```
setenv CMGMESSAGEDIR ~/Messages
```

Changes are valid after restarting the program. Note that the messages files are not only changing the name cosmetically, but also in terms of access to those fields using design commands.

5.2 Characteristics Order

? Display characteristics in the cost note in a different order than they are defined in the modeler.

The motivation behind this option is that at times it may be easier to define characteristics based on one categorization, but that it would make more sense to present the questions to the users in another categorization.

Unfortunately, the order of characteristics in the model determines the order they are displayed. The only exceptions are:

- Conditional displays can make a characteristic not appear.
- A lower level context can have a different display order than a higher level, just by moving characteristics around in the model.

5.3 Pop up windows

? Add new pop-up windows with Cost Advantage as part of the program

The motivation behind this question is that the various inputs and resources in the Overall Flowchart are used throughout the user's design-manufacturing specification process. The user may want to see these characteristics grouped differently depending on the task at hand. For example,

- The user may want to see a list of the materials used in the entire model, or he may want to see the materials associated with their particular manufacturing step. However, since CA does not allow a variable to be defined in two different places, the next thing to try is to create an outside display which can access the variable's *value* more than once.
- The user may want to keep a running list of all of the variables which have already been defined.

Unfortunately, pop-up windows within Cost Advantage are not possible. The CA does not provide a general programming API to allow you to gather, process, query, or process variables outside the program.

Of course, it is possible to add functionality or new user interfaces as a pre- or post-processing step. Cost Advantage data can be displayed through programs such as Excel or Matlab, by building a user interface through programs such as Tcl/Tk or XMotif, or processed using querying languages such as SQL or Perl.

5.4 Display verbosity switches

? Control the verbosity of inputs or outputs

Depending on the expertise of the user, he may or may not need as much explanation for the inputs and outputs to the model. To control whether certain characteristics (or calculations or alerts) are displayed, add an extra global variable with boolean value called "verbose."

The global variables may also be useful in debugging during the model building stage. After the model is being built, the global variable "debugging" could be permanently set to FALSE.

6.0 Designing the model

This section describes the reasoning behind certain aspects of the model design. Some of these limitations result in a difference between the conceptual architecture and the implementation.

6.1 Recurring and non-recurring costs

? Options for both recurring and non-recurring costs in the summary table

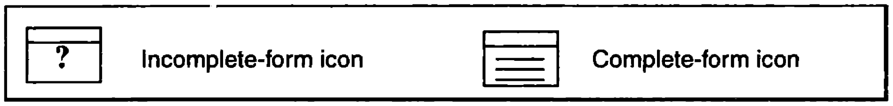
One option might be to duplicate every cost category into recurring and non-recurring. The advantage is that costs breakdowns are easy to visually summarize. The disadvantage is that the model user would need a higher general knowledge about the process in order to explicitly differentiate between recurring and non-recurring costs. Our option was to use the Quantity field to designate the frequency of operation.

6.2 Form icons

? Danger of using incomplete- and complete-form icons

The form icon, shown in Figure x, is located at the left-most side of the Cost Note. The incomplete-form icon means that there is (at least one) user-visible characteristic that does not have a data value assigned to it. The filled-form icon means that all of the user visible characteristics have values assigned.

FIGURE 4. Incomplete- and complete-form icons



The danger comes when certain fields may be hidden at run time because of menuing or because of a poor model design. It's the responsibility of the model builder to determine when the characteristic should be displayed. Cost Advantage does not try to determine whether the displayed values associated with a summary line have all been assigned enough information to calculate to completion. One measure of prevention is to assign as many fields to reasonable default values as possible.

7.0 Unsolved implementation problems

This section discusses features or functionality that I have not found an acceptable solution or workaround for. This would be good for future work or suggested improvements to Cost Advantage.

7.1 Time counter

? Use a time counter variable to simulate a factory over time.

This is difficult because of internal CA checks against cycles, even if they are supposed to "time out" eventually.

7.2 Multiple occurrences of values

? The user can enter a value in one of two places. Neither has precedence.

I don't think this can be done, except perhaps with an external script which contains the one variable but displays it twice separately.

8.0 Known Cost Advantage bugs

This section discusses known bugs in the Cost Advantage software. While the solution is to write a bug report and wait for the next patch, they are important to know about ahead of time.

8.1 Inheriting units

? A child's units may not override a parent's units

A child's units should not be able to override a parent's units, but this is not always the case. Sometimes the units do override. To avoid confusion, do not design a model which requires units to be overridden.

Nonetheless, there are two motivations behind wanting to override a parent's units. In one case, the child's units are more specific than the parent's. For example, the parent's units for a particular characteristic may be

```
generic rate of change (change/min)
```

while the child specifies

```
rate of heating the plate ( F/min)
```

In the second case, the units field was not used for recording physical units of measure, but for extra comments without cluttering the screen with extra alert windows. For example, the parent's units may be

```
<-- Try using eq 90, 100, 150
```

while the child specifies

```
<-- Try using eq
```

8.2 Jittering scroll bar

? CA sometimes crashes when clicking on the left scrollbar in a window

In some windows there is a secondary scrollbar besides the standard one on the right side of the window. If you click on the rectangular bar instead of the two up and down triangles, this will sometimes cause Cost Advantage to crash. The error message is as follows:

```
Error: Received signal number 10 (Bus error)
[condition type: SYNCHRONOUS-OPERATING-SYSTEM-SIGNAL]
[changing package from "COMMON-LISP-USER" to "CMG"]
```

The problem has been found in older versions of CA and when using a PC to remotely log into a workstation. The newest version contains a bug fix, but otherwise the only course of action is to kill the process and restart Cost Advantage.

8.3 Features rearranged

? Uniquely name Features in Cost Note to prevent internal CA confusion.

The problem of Features randomly rearranging themselves within a cost note is a bug within CA which is potentially very harmful. The problem is that, although the internal characteristics of each Feature may be defined separately, CA cannot distinguish between them if they have the same name. The order of the steps will then be randomly rearranged when you reload the model. To amplify the problem, after the features rearrange themselves, it's pretty impossible to tell which steps are which, since they have the same name.

This bug was discovered when I created many features and neglected to change the generic names of "Clean," "Attach," etc. with the intention of changing them later. In actual use, the model builder would have a clear idea of all of the feature steps and probably read them in from a file, with the names already uniquely defined.

The problem is easily reproducible:

- Create a simple model with three Features: F1, F2, and F3
- Create a cost note and add features in a noticeable pattern, making sure to repeat some of the steps. (e.g., F1, F2, F3, F3, F2, F1)
- Save both model and note
- Re-open the note, and the steps should be out of order.

8.4 Ghost variables

? Variables which linger after being erased may upset hierarchy tree

In Version CA1-9-1XS4, there is a bug which loses track of characteristics (variables) within the hierarchy tree. This is especially a problem for cases where the child characteristic had overridden the parent. When the parent is erased, the child erroneously remains in place. Sometimes the ghost characteristic cannot be erased, or reports that it inherits from one of its children. The problem may have been fixed, as described in the *CA 1.9.0 Patch Notes*.

8.5 Accidental erasure of Feature Costs

? Improper erasure of Feature Costs will irrecoverably crash model

The BasePart Costs dictate both the Summary Table column headings and the Feature Costs. In order to make a change to either of these, the change must be made in the BasePart Cost.

The bug is that Cost Advantage does not restrict the model builder from erasing Costs from the Feature list. Immediate side effects include a mismatch in the number and order of costs between the BasePart and Features. More importantly, when the user runs a Cost Note, CA will crash on the AddFeature menu command. This bug is very reproducible, and the only solution is to revert to a previous saved model. The corrupted model cannot be salvaged.

To reproduce on a fresh model, just delete some of the Feature Costs (not Feature Characteristics), try to RunCostModel, and AddFeature.

9.0 If Cost Advantage should crash

Cost Advantage does have its occasional unexplainable errors. While they are not always recoverable, you can usually take a snapshot of the model at the time of failure to help with further debugging.

9.1 Breaking out of a process

? Is there a function similar to `<cntl>-c` which will break out of a command?

According to Cognition, sometimes pressing `<cntl>-c` in the CA window for a while will break out of the command.

Pressing `<cntl>-c` in the Unix window will break into Lisp mode.

9.2 Time for model to update

? Did Cost Advantage crash, or is it still processing?

Also, how long should it take for updates, especially if you're changing things high in the hierarchy. sometimes it seems up to 1/2 or 1 minute. maybe it's because I'm running on a Sparc2 (110MHz).

9.3 Error messages at the Unix prompt

? Is this an important error message?

? Collect information about the crash to send to Cognition

Sometimes when Cost Advantage stalls, a message may appear at the Unix prompt. Two examples are listed below:

```
Error: Attempt to take the cdr of #
which is not listp. [condition type: SIMPLE-ERROR]
[changing package from "COMMON-LISP-USER" to "CMG"]
[1]

5019880 bytes have been tenured, next gc will be global.
See the documentation for variable
EXCL:*GLOBAL-GC-BEHAVIOR* for more information.
```

The first message is a real error message, while the second is for information only. The Cost Advantage documentation does not generally explain error messages. If there is a recurring message, collect information about the crashed run and send it to Cognition. The options are:

- Create a "circlefile"
  According to Cognition, a circlefile is a circular — i.e. fixed-size, first-in-first-out (FIFO) — log file. It starts relatively small, builds to a maximum, then starts "eating its tail." The circlefile program (called by the "calog" command) decodes such a file, formats it, and prints it out. Note that CA version 1.9.2 is required for this feature to function properly. If CA crashes, type one of the following commands immediately:

  ```
  $CABASE/utilities/calog -t 99
  $COGBASE/ca/utilities/calog -t 99
  /Cognition/ca/utilities/calog -t 99
  ```

- Type `:zo` at the Unix prompt
- Compress (tar) the model directory and send it to Cognition
- Revert to a previous uncorrupted version of the model

Appendix C

References

The references are divided by topic in terms of: (1) Design for manufacturing, (2) Manufacturing processes, (3) Cost modeling, and (4) Knowledge-based systems.

Design for manufacturing

[1] Boothroyd, Geoffrey, Dewhurst, Peter, and Knight, Winston. *Product Design for Manufacture and Assembly.* Marcel Dekker, Inc., 1994

[2] Ciriscioli, Peter R., Mantell, Susan C. (eds). *Composites: Design and Manufacture for Cost Effectiveness.* presented at 1994 International Mechanical Engineering Congress and Exposition, Chicago, Illinois, November 6-11, 1994.

[3] Gonzalez-Zugasti, Javier. *Computer Modelling of Advanced Composites Forming.* SM, Massachusetts Institute of Technology, 1991.

[4] Hahn, H. Thomas. "Design for Manufacturability of Composites" in *Composites Design, Manufacture, and Application.* Proceedings of the Eighth Internation Conference on Composite Materials, Society for the Advancement of Material and Process Engineering, 1991.

[5] Neoh, Ein Tech. *Adaptive Framework for Estimating Fabrication Time.* SM, Massachusetts Institute of Technology, 1996.

[6] Srivatsan,T.S., Ramani, K., Ramulu, M. (eds). *Processing, Design, and Performance of Composite Materials.* International Mechanical Engineering Congress and Exposition, Chicago, Illinois, Nov. 6-11,1994, sponsored by the Materials Division, ASME.

[7] Rosato, Donald V. *Designing with Plastics and Composites: A Handbook.* Van Nostrand Reinhold, 1991.

Manufacturing processes

[8] Boccard, Alexis, Lee, Woo Il, and Springer, George S. "Model for Determining the Vent Locations and the Fill Time of Resin Transfer Molds." *Journal of Composite Materials*. Vol 29, No. 3, 1995.

[9] Boyajian, Paul G. *Manufacturing and Design of Textile Preforms for Composite Materials*. SM, Massachusetts Institute of Technology, 1993.

[10] Elber, Gail. "Resin Transfer Molding: Tips for Successful Application." *Advanced Composites*, Sept/Oct 1993.

[11] Lindner, E. and Unger, P. (Eds) *Gastrow Injection Molds, 108 Proven Designs*. Hanser Publishers, 1993.

[12] MacDermott, Charles P. and Shenoy, Aroon V. *Selecting Thermoplastics for Engineering Applications*. Marcel Dekker, Inc., 1984.

[13] Masi, Barbara A. *Fabrication Methods and Costs for Thermoset and Thermoplastic Composite Processing for Aerospace Applications*. SM, Massachusetts Institute of Technology, 1986.

[14] *Materials Research Agenda for the Automotive and Aircraft Industries*. National Materials Advisory Board, Commission on Engineering and Technical Systems, National Research Council. Washington, D.C.: National Academy Press, 1993.

[15] Pilato, Louis A. and Michno, Michael J. *Advanced Composite Materials*. Springer-Verlag, 1994.

[16] Polgar, Karch C. *Simplified Time Estimation for Basic Machining Operations*. SM, Massachusetts Institute of Technology, 1996.

[17] Stover, Debbie. "Braiding and RTM Succeed in Aircraft Primary Structures." *High-Performance Composites*. Jan/Feb 1994.

[18] Strasser, P. and Woite, B. "Low-Pressure Processes for Decorative Interior Fittings" in *Plastics in Automotive Engineering*.

[19] Tam, Albert S. *A Deformation Model for the Forming of Aligned Fiber Composites*. SM, Massachusetts Institute of Technology, 1990.

Cost modeling

[20] Allan, David Charles. *Process Sensitive Steady-State Part Cost Computer Modeling of Job Shops.* SM, Massachusetts Institute of Technology, 1990.

[21] Busch, John V. *Technical Cost Modeling of Plastics Fabrication Processes.* PhD, Massachusetts Institute of Technology, 1987.

[22] Chen, Andrew C. *Economic Aspects of Materials Substitution in Horizontal Automotive Body Panels: The Issue of SMC Surface Finish.* SM, Massachusetts Institute of Technology, 1991.

[23] Chien, Tze-Thong and Serben, Saul and Taylor, William. "A Computer-Aided Minimum Cost Transfer Machine Layout Design." Society of Manufacturing Engineers MR75-198. 1975.

[24] Gutowski, Timothy and Hoult, David and Dillon, Greg and Neoh, Ein-Teck and Muter, Stuart and Kim, Eric and Tse, Mawuli. "Development of a Theoretical Cost Model for Advanced Composite Fabrication" in *Composites Manufacturing*. 1994.

[25] Hendrichs, Nicohas Jorge. *Steel-Plastic Substitutions in Automotive Structural Applications.* SM, Massachusetts Institute of Technology, 1989.

[26] Ilcewicz, L.B., and Mabson, G.E. and Metschan, S.L. and Swanson, G.D. and Proctor, M.R. and Tervo, D.K. and Fredrikson, H.G. and Gutowski, T.G. and Neoh, E.T. and Polgar, K.C. "Cost Optimization Software for Transport Aircraft Design Evaluation — Design Cost Methods." NASA CF 4737. Aug 1996.

[27] Kumar, Jayanthi and Kendall, Elizabeth and Satyawadi, Amit and Malu, Chand Kumar. "Cost Estimates of Composite Aerospace Structures Based on Complexity." *draft*

[28] Li, Mengchen and Kendall, Elizabeth and Kumar, Jayanthi. "A Computer System for Lifecycle Cost Estimation and Manufacturability Assessment of Composites." *draft*

[29] Mabson, G.E. and Graesser, D.L. "Cost Optimization Software for Transport Aircraft Design Evaluation -- User's Manual." NASA CF 4738. Aug 1996.

[30] Mabson, G.E. and Ilcewicz, L.B., and Graesser, D.L. and Metschan, S.L. and Proctor, M.R. and Tervo, D.K. and Tuttle, M.E. and Zabinsky, Z.B. "Cost Optimization Software for Transport Aircraft Design Evaluation -- Overview." NASA CF 4736. Aug 1996.

[31] Muter, Stuart. *Cost Comparisons of Alternate Designs: An Information Based Model.* SM, Massachusetts Institute of Technology, 1993.

[32] Ostwald, Phillip F. *AM Cost Estimator.* Penton Publishing, Inc., 1988.

[33] Proctor, M.R. and Metschan, S.L. and Klein, H.S. "Cost Optimization Software for Transport Aircraft Design Evaluation -- Process Cost Analysis Database." NASA CF 4739. Aug 1996.

[34] Tse, Mawuli. *Design Cost Model for Advanced Composite Structures.* SM, Massachusetts Institute of Technology, 1992.

Knowledge-based systems

[35] 6.871 Knowledge-Based Systems class notes, Spring 1997.

[36] *Composites Design, Manufacture, and Application.* Proceedings of the Eighth Internation Conference on Composite Materials, Society for the Advancement of Material and Process Engineering, 1991.

[37] Demeri, M.Y. (ed) *Expert System Applications in Materials Processing and Manufacturing.* Proceedings of a symposium sponsored by the TMS Shaping & Forming Committee, the TMS Synthesis & Analysis in Materials Processing Committee, and the ASM-MSD Computer Simulation Activity, held at the TMS annual meeting in Anaheim, California, February 18-22, 1990

[38] Duda, R., Gaschnig, J., and Hart, P., "Model Design in the Prospector Consultant System for Mineral Exploration." Expert Systems in the Microelectronic Age. (ed.) Edigurg University Press, 1979.

[39] Hinkle, David and Toomey, Christopher. "Applying Case-Based Reasoning to Manufacturing" in *Artificial Intelligence Magazine.* Vol. 6 No. 1, 1995.

[40] McDowell, James K. and Meltsner, Kenneth J. (eds.) *Knowledge-Based Applications in Material Science and Engineering.* Proceedings of a symposium sponsored by the ASM-MSD Computer Simulation Committee, held at the TMS Annual Meeting in San Francisco, CA, February 28, 1994.

[41] Sergot, et al. "The British Nationality Act as a Logic Program." Communications of the ACM, May 1986.

[42] Sharpe, John. (ed.) *AI System Support for Conceptual Design.* proceedings of the 1995 Lancaster International Workshop on Engineering Design, March 27-29,1995

[43] Swift, K.G. *Knowledge-Based Design for Manufacture.* Kogan Page, 1987.

[44] Szolovits, P. "Uncertainty and Decisions in Medical Informatics." Methods of Information in Medicine. vol 34, 1995.

[45] Topolski, Alvin S. and Reece, Douglas K. "Packaging Advisor: An Expert System for Rigid Plastic Food Design" in *Proceedings of Innovative Applications of AI.* 1989.