WHAT IS COMPUTER-AIDED ESTIMATING ?

"Jack, that new customer needs this critical estimate in an hour—
try to make a really accurate guess this time."

Prepared by
Micro Estimating Systems

An Opinion?

A Quote?

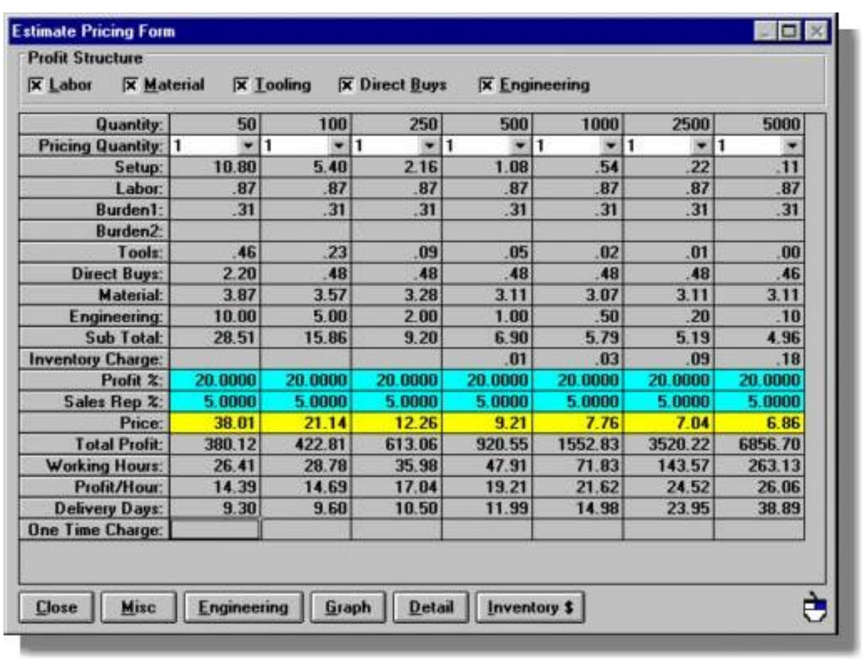
Or As Close to Correct as Possible?

With computer-aided estimating systems today, the user has a wide choice -- capability wise. The first choice to be made is between accuracy or economy. Many of the 'economy' choices are made without concern for accuracy issues. For the 'accuracy' issues, however, there is no source where you can discover what those choices involve. The important fact to remember, when looking at computer aided estimating systems, is that each time you send a quote to a customer, you are betting your business on the accuracy of the times in the estimate.

Would you believe that many shop estimators are capable of producing more accurate estimates than the software they use to do so?

Surprised? That scenario is entirely possible. Unless you understand the strategies used to 'build' a computer-aided estimate, that statement probably will surprise you. And as with any software, there's also the ever present, added admonition of "garbage in, garbage out."

Nonetheless, during the past ten years, the value of computer aided estimating has grown as a manufacturing tool, as manufacturers and job shop owners realized the varied benefits this software brings. Published studies in trade magazines often discuss user's perceptions of direct benefits they've seen. Still, this is clear: among both prospective - and existing-users of estimating software, most do not understand how the software prepares an estimate, what makes a computer estimate accurate, or the different emphases played by the machine tool, the process, or the material in preparing the computer estimate.



Estimate Pricing Form

Profit Structure

[X] Labor [X] Material [X] Tooling [X] Direct Buys [X] Engineering

Quantity:	50	100	250	500	1000	2500	5000
Pricing Quantity:	1	1	1	1	1	1	1
Setup:	10.80	5.40	2.16	1.08	.54	.22	.11
Labor:	.87	.87	.87	.87	.87	.87	.87
Burden1:	.31	.31	.31	.31	.31	.31	.31
Burden2:							
Tools:	.46	.23	.09	.05	.02	.01	.00
Direct Buys:	2.20	.48	.48	.48	.48	.48	.46
Material:	3.87	3.57	3.28	3.11	3.07	3.11	3.11
Engineering:	10.00	5.00	2.00	1.00	.50	.20	.10
Sub Total:	28.51	15.86	9.20	6.90	5.79	5.19	4.96
Inventory Charge:				.01	.03	.09	.18
Profit %:	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000
Sales Rep %:	5.0000	5.0000	5.0000	5.0000	5.0000	5.0000	5.0000
Price:	38.01	21.14	12.26	9.21	7.76	7.04	6.86
Total Profit:	380.12	422.81	613.06	920.55	1552.83	3520.22	6856.70
Working Hours:	26.41	28.78	35.98	47.91	71.83	143.57	263.13
Profit/Hour:	14.39	14.69	17.04	19.21	21.62	24.52	26.06
Delivery Days:	9.30	9.60	10.50	11.99	14.98	23.95	38.89
One Time Charge:							

Close | Misc | Engineering | Graph | Detail | Inventory $

The estimate pricing form summarizes all job costs. Here the estimator assigns multiple efficiency ratings for setup and production, based upon quantities. He may consider per piece costs and adjust for multiple setups. When the estimator has chosen multiple methods, this screen identifies the ideal method by quantity.

One wonders how anyone selects the proper package for their needs from among the available types and variations of software featuring "estimating" capabilities today. In fact, it is likely that many perspective users may not understand that "estimating" software may range from products that simply replace a pencil and pad, to software called "standards-based," to systems that might go so far as to simulate the entire machining or fabricating process.

What issues must a software estimating program consider?

To give you a quick idea of the complexities that can be involved, here's a simple example for drilling a half-inch hole, three inches deep. Will the software automatically give you a different cycle time for a CNC Swiss machine than it does for a turret lathe or an engine lathe? Moreover, will that be by machine vendor's model and type? Will it give a different time for a different model? When cycle time is for a specific model, the program times can be quite close.

Those times can be closer yet using machine tool emulation. Emulation is difficult, for it involves more than simply altering rapid travel rates.

Finally, will the software arrive at a different time for the same metal whether or not it has been heat treated? If the tool is a carbide or high-speed drill? Or when requiring a different micro-finish on your turn?

Even more difficult is the process of accounting for the length-to-diameter ratio on an end mill. The feed/speed books do not address this issue. For example, perhaps the plunging depths are based upon 1-½ X diameter. What if your depth is ½ X or 3 X the diameter into the stock? The feed rate will change. Is it necessary for the estimating program have a provision to deal with that situation?

Four quoting and estimating categories

Let's look again at the proposition that a manual estimate might be more accurate than one a computer generates. That implies neither a case of sloppy users nor of poor input data. Rather, as we will demonstrate, the same part specifications will result in considerably different results from the four different types of estimating software. That's correct: there are four different types of computer-aided estimating software. Yet, if someone surveyed a broad demographic sample of estimators, few would know that. That's because -- unlike CAD/CAM software -- there is no effort in the manufacturing industry to teach, or to learn, the underlying strategies that the different estimating software systems use to produce estimates. By now most of us realize that just because someone sells a CAD/CAM product, that doesn't mean it is as precise or as sophisticated as a manual drawing, or as another CAD/CAM product. By now, everyone understands what a parametric CAD/CAM product offers. How many, however, understand what 'intelligent simulation' means in relation to estimating software? Or what 'standards' implies?

The industry neither has a standard definition for an estimate, nor universally understands what an “estimating” program could or should provide. That is a key point and deserves reiteration:

> **The industry neither has a standard definition for an estimate, nor universally understands what an “estimating” program could or should provide.**

Because of the lack of any single definition for estimating as it applies to software, many who use such software may be getting unexpected results — and often they are unaware of the situation.

With ‘estimating’ software costs ranging from $199 through $10,000, common sense dictates that differences exist. For the shop that chooses to purchase the $199 package, rather than the $3000 or $10,000 package, often it is an economic decision. Anyone smart enough to own a shop or manage a department must understand that the capabilities of those packages are different. Some will argue, however, that if economics really dictated that decision, then one of the more expensive products might have been a better choice. The lack of understanding -- the lack of definitions -- is unfortunate and becomes a disservice to estimators and their companies -- and even to their customers. With this article, we’d like to suggest a foundation for an industry-wide understanding of the computer-aided estimating process.

In the quoting/estimating arena, today we find *quoting* programs and several classes of *estimating* software. Quoting programs quickly provide prices for quoting. (See Addendum I.) Within the estimating classification, some software systems are standards-based, some are engineering-based, and some exhibit genuine intelligence. Today, some programs also are evolving from one form toward another, as even software developers begin to better understand the actual estimating process.

Quoting & Estimating **need delineation**

Vendors often use “estimating” when describing the quoting function. In fact, quoting software has no computer-aided estimating capability. In our effort to define what to expect from estimating software, let’s begin by identifying the range of functions performed. Here we find two groups of software:

> Software which simply assists the shop owner to provide a price to the customer. If the estimator manually calculates or guesses the time it will take, let’s call this a computer prepared quote.
>
> Other software provides, in addition to that same quoting function, estimates of actual times the job will require in the shop. When the computer estimates the times, let’s call this a computer-generated estimate.

Time is the differentiator. Using time makes it rather simple to differentiate between quoting and estimating software.

Estimating needs definition and explanation

Having thus removed quoting from the discussion, here is where defining estimating becomes involved. It is not only because of the diversities between basic program types, such as simple spreadsheets, report generators, and so forth, but also the many shades of grey within the estimating (software) category.

Distinguishing between the several current types of estimating programs requires a look at their basic design. Estimating programs are based on one or more of the following design strategies:

1. **Standards-based**
2. **Engineering-based**
3. **Intelligent simulation**

When we talk about specific, historic system-designs used for estimating software, we can classify them in this manner. However, since the mid-nineties an evolutionary process has been underway, and the lines between standards-based estimating and engineering-based estimating are blurring. Similarly at the upper end, where we find “non-intelligent” and “intelligent” estimating systems.

Remember that pure quoting does not calculate time, while estimating bases its quote on calculated time. ‘Standards-based’ estimating programs use a table with standard times assigned to similar operations. Systems which rely on libraries of machine speeds and feeds are ‘engineering-based.’ Here is where the blurring begins. Clarifying this point requires only the knowledge to ask the right question.

Differentiating between the engineering-based and the “intelligent” packages is harder. Both systems come up with an answer — for the portion(s) of the process that they compute. Determining — rather, knowing how to determine — how thorough the estimating program can be, is far more difficult.

Standards-based toward engineering-based evolution

Standards-based estimating software, while similar to quoting software, can take more time for initial setup after it is purchased. It may require the user either to enter the standards or to update those provided to reflect the shop’s actual operations. Often this software already has the standards included; however, companies must realize that if they use someone else’s standards they may only be accepting someone else’s guesses. In estimating, there really are no industry-wide standards -- there are only company standards. If you purchase a standards-based system, you will want, at the least, the ability to quickly and easily change any standard.

Standards-based knowledge offered the earliest computer-aided estimating assistance and programs based on this alone may provide the least accurate estimating method. (See Addendum II.) The ongoing evolution is from the standards-based programs toward the engineering-based programs. Many vendors are revising their systems to become more engineering-based. While some still don't deliver a precise conclusion, they can be more accurate than the standards-based packages.

These evolving, pseudo engineering-based systems may already include one or more sophisticated modules for hole making or milling (vs. individual computations for complex movements and operations). They employ simple formulae to compute the time required for a procedure — mill a pocket, or drill a hole. What they may lack in any formula is a concept of machine movement and operations. Typically, the software in this evolving class also omits consideration for many idle time elements or specific machine capabilities. For example, a 600 IPM machining center does not make 600 IPM for a one-inch move. In addition, while some systems include basic indexing times, they neglect the more precise indexing time components, such as acceleration/deceleration, spindle index, or rapid ramp-up.

Machining mathematics — by itself — is quite simple:

(CUT LENGTH)/FEED = REVOLUTIONS; REVOLUTIONS/RPM = TIME
Figuring cut length is fairly simple:
To ream a hole one inch deep, you will ream one inch deep *plus* an approach distance.
To mill a pocket, you will have to take multiple passes of a particular length.

The difficulty lies in knowing the time the operation takes on a particular machine — for example an engine lathe. Does the operator have to manually turn the crank to remove the drill and the chips? And then turn the crank back in to resume drilling? That process knowledge is more important than simple mathematical results, and it adds a level of complexity you should expect in the better estimating systems. This is the level of program depth that is beginning to be called 'intelligent simulation' estimating.

Engineering-based estimating

The engineering-based estimating programs use the same formulae used when manual estimators need a precise cost. They are unchangeable, based on the laws of mechanics. Given a drill diameter and style, the depth of hole, the material, and the machine doing the drilling, there can be only one correct time. The formula to derive that time never changes. Feed rates and surface footage-rates may change; once those rates are set, the math is constant. The same is true of idle times. For different machines, the rates may differ, but the movements are the same.

While a standards-based system having a standard for every half inch of hole depth gives ½ -inch accuracy (See Addendum II.), in an engineering-based system, setting up the core data covers any hole, at any depth, and is accurate in every situation. Period. There's no stated or implied disclaimer of being "within a half an inch." The engineering-based system may seem to demand more setup than a standards-based system. However, you could never set up enough standards to reach the same level of accuracy. There is a difference in the time needed to drill a one-inch hole and a hole 1.021 inch deep, and the engineering approach can be 100% correct -- but the standards-based approach can only be approximate.

Being engineering-based, however, doesn't guarantee perfection. Speeds and feeds data differ for every material, every tool, every machine type. If an engineering-based estimating program doesn't provide speeds and feeds data for every material, but rather has assigned them on the basis of material *groups*, then accuracy is sacrificed.

While any engineer can look up the feed rate for 12L14 steel, that data is irrelevant without knowing the machine type. Would not use the same feed rate on a CNC lathe as on a screw machine? The speeds and feeds data is only the starting point. If every material is given its own speed and feed and multiple classes, you can change 12L14 to represent your shop's experience, without affecting 12L50. That would not be possible in any engineering-based system that grouped speeds and feeds by material type.

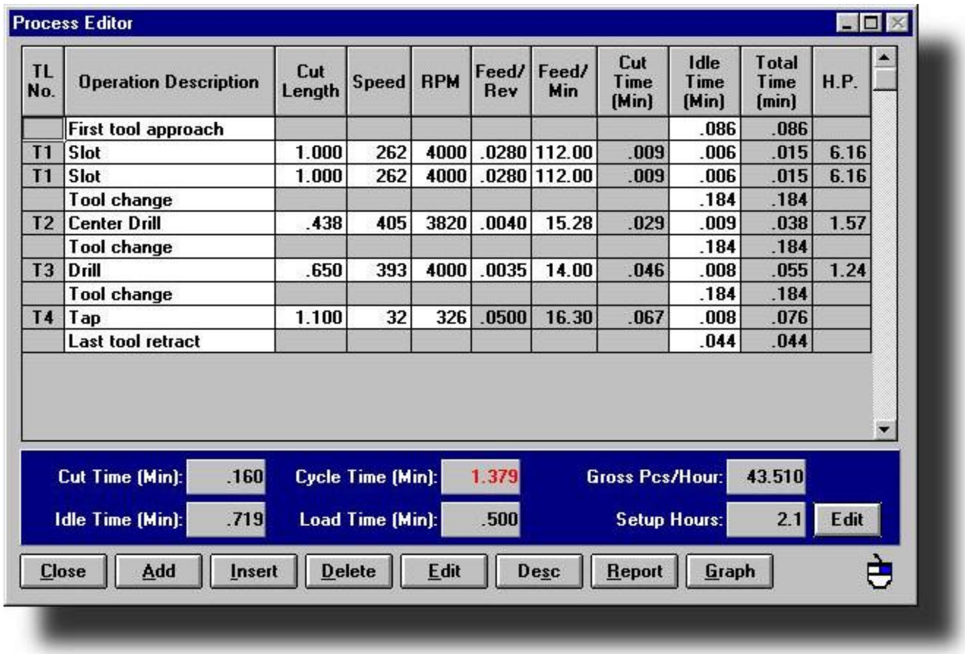
The engineering-based computer-aided estimating system mimics the correct manual estimating method, using the same equations. The added benefit of computer aided estimating is its ability to take more into account, such as idle time movements or rapid travel rates, to retain the correct information, and to always use the correct data. These examples are simply some of the more difficult to account for times, since there are so many.

Intelligent emulation

Having the capability to account for issues like rapid travel and complex idle times defines the difference between the engineering-based and the estimating systems that emulate the machines -- we call this **intelligent emulation.**

Compared to standards-based estimating, engineering-based is more comprehensive, and more accurate. An approach based upon intelligent simulation will improve upon the engineering-based results, but requires the same amount of data from the user, the same effort for initial setup. It's claim for being closer to the actual time is that it is a closer representation to the real world. In its estimate, it attempts to include everything significant that will happen, as opposed to what should happen. When it comes to estimating, you really don't care that it *should* take one minute -- you care about how long it *will* take.

If you rely on unchangeable standards, you're letting the computer tell you how long something should take, rather than the software *determining* how long it does take. Once again, it comes down to the estimator knowing the time the machinist requires for each step. That’s where intelligent simulation attempts to determine how long it really will take as opposed to how long it should take.



TL No.	Operation Description	Cut Length	Speed	RPM	Feed/ Rev	Feed/ Min	Cut Time (Min)	Idle Time (Min)	Total Time (min)	H.P.
	First tool approach							.086	.086	
T1	Slot	1.000	262	4000	.0280	112.00	.009	.006	.015	6.16
T1	Slot	1.000	262	4000	.0280	112.00	.009	.006	.015	6.16
	Tool change							.184	.184	
T2	Center Drill	.438	405	3820	.0040	15.28	.029	.009	.038	1.57
	Tool change							.184	.184	
T3	Drill	.650	393	4000	.0035	14.00	.046	.008	.055	1.24
	Tool change							.184	.184	
T4	Tap	1.100	32	326	.0500	16.30	.067	.008	.076	
	Last tool retract							.044	.044	

A CNC Processing editor screen lets an estimator review the process and to try what-if scenarios. Changing any speeds or feeds here, for example, will instantly change the Cycle Time and Gross Production figures at the bottom of the screen – the estimator knows immediately the results of his changes

Intelligence in humans relates both to ability and to capability. By *ability*, let’s distinguish between the genius, a person of average intelligence, and someone with a low IQ. Then, *capability* could relate to the difference between an adult and a child. Computer software is no different; some software is both smarter and has greater capability than others.

Returning to our human analogy, compare the step between engineering-based and intelligent simulation to a high school math whiz. The whiz can compute any math problem; an engineer, however, knows how to apply that knowledge to building a car. In addition to the in-depth machine knowledge required to add such intelligent simulation to an estimating program, it requires tedious and time consuming programming effort, as well. How many types of machines are used today by shops? How many different styles? How many different vendors? All this must be factored into the simulation process. It is costly for a software vendor to develop and maintain such software. An estimating system that provides generic standards — such as for turning or drilling — cannot be compared to one that recognizes 60 independent machine types for those processes.

Does the program handle every job the same way? Some do. They assume that manual lathes, CNC lathes, turret lathes, automatic screw machines and others require the same time. In that situation, your estimate is the same for each machine type! Can we call that an estimate?

Perhaps it is a more educated guesstimate than many would make without software — but probably not. At least in the manual situation a person knows enough to allow for the idle movements, indexing, and other factors specific to the machine in question.
A thorough estimate provides allowances for clearances between tools and makes use of actual spindle speeds and feeds from the gear charts, when appropriate. It will also consider rapid travel rates based on distance and spindle speed ramp-up/ramp-down. Estimating software that intelligently simulates machine operations allows for all these items, but to do so, it must be written for each specific machine. This is the key difference between engineering-based and intelligent simulation estimating software.

Given the choice between a generic estimating program with no machine specific knowledge, a standard, and a human guess, you should take the person over the computer every time. Similarly, a bad guess is worse than no guess. Is a pseudo-engineering system any better — or worse — than a quoting system? At least the quoting system forces YOU to come up with a time, while the former may assign an inaccurate time, with the implication that it is correct.

The determinations that should be made:

- How the software calculates cycle time
- Does it calculate cycle time for specific types of machines (CNC mill, Bridgeport)
- Does it calculate a cycle time for a specific model within a type (model A or B

A genuinely intelligent system accounts for:

- Set up time
- Idle time
- Complete machine time for the correct type and model of machine
- Available horsepower
- Minimum and maximum RPM and feed
- Fixed or variable RPMs and feed rates
- Special lever/gear/control settings or rules

Such software literally emulates the machine’s movements, stepping through them one at a time. For example, to estimate a facing operation, the program doesn’t resort to a single predefined equation; rather it simulates what will actually happen by calculating every revolution and recording the time.

To determine the correct, total time, for example, the estimating program literally models the operation for each process involved and the software loops through every turn the machine will make, making adjustments at every step to account for RPM, feed, minimums, maximums and so forth. To do constant surface footage correctly, the program computes a new RPM for every revolution and considers minimum and maximum RPM at each step. The program must know things like how long it takes to reach top speed, as well as what the top speed is for the type of tooling being used; how long to remain at top speed; and how long it will take to reach center. In other words, the SFM formula at exact center calculates an infinite RPM (which is not possible), and it steps down from there towards the outside diameter of the cut. The machine maximum RPM must be considered. The estimating loop goes through every movement, whether for lathe or pocket milling. Machine tool emulation is the basis of the estimate's accuracy, and it is based upon the combination of all pertinent parameters at every moment in the operation.

Similar, stand-alone software called "tool path optimization" software, is considered fairly new today, when applied to NC output from CAD/CAM programs. However, this feature also should be the basis of any comprehensive computer-aided estimating program — and there it is referred to as machine tool emulation. That's what an intelligent, engineering-based estimating program does. Some have been doing so for a long time.

In the end, a choice must be made between accuracy and economy. Remember, however, that each time you send out a quote, you bet your business on the accuracy of the times in the estimate.

Addendum – What is ‘quoting’?

Precisely, what is “quoting” software? To begin with, quoting software can be very simplistic -- as simple as a spreadsheet. It is quite fast, in fact quoting software is the fastest method for obtaining a customer price, next to guessing. It is not standards-based, but can provide a consistent template to follow. Still, the quotes can be only as good as the guesses that the person enters. Not being standards-based indicates that your guess today could be different from your guess tomorrow. Quoting software is not expected to provide consistently accurate estimates, only a speedy means of obtaining a quote for a customer.

Why would you use quoting software? Primarily for simplicity, often for the software’s low cost, and perhaps to help employees gain familiarity with the quoting process. Quoting software might only be the front end of a shop management software system -- a speedy means of entering the basic part information required to produce a quote, and which also is needed by other modules of the shop management software. Since there are no databases to set up, little of your time goes toward setup. The results typically will be no more complex than what you can do on a spreadsheet.

Addendum II – Estimating a 1-inch hole with standards

Consider a simple standard for a one-inch hole, drilled one inch deep into a steel part. To be truly accurate, there would need to be standards for every drill diameter, every drill depth, every drill-type, and every material. You can see the impossible number of standards suggested by this one example. To avoid this data entry nightmare, not to mention a need for gigabytes of disk storage, software developers will interpolate “nonstandard” situations. Interpolation may be quite appropriate in operations like assembly, where there are no speeds, no feed’s, and no factors that affect time like the increasing effect of resistance from chips in deep holes, but interpolation is inappropriate for machining.

A standard that *can* be described is the time it takes to turn one revolution. Using such a standard, an estimator could apply the number of revolutions required to drill a hole to arrive at the time. However, this standard is the basis for an engineering-based system, and is not found in a standards-based system. So you can see that while interpolation is appropriate for manual elements, it does not apply equally well to machining, because interpolation cannot account for their variations.

If a company chooses to use a standards-based estimating system, then it should do so on the basis that any standards included with the package are not *its* standards and will need to be modified. Still, the standards *are* just guesses -- if not someone else’s, then they are yours. Your guesses could be different next week. The hackneyed computer phrase could apply: Garbage in, garbage out.

Time standards can be proven or fine tuned. The times provided are corrected based on reality or time studies, real things that are done. However, if you correct a time standard based on how you feel today, then you are doing nothing better than a quoting program might, and you really are not correcting anything. In other words, if every time you look at that "time" and say 'that time looks a little wrong,' and you change it, you've defeated the purpose of the standards-based software package.

A downside of standards-based estimating is this. To deal with a broad range of situations, the number of standards can be held to an acceptable level, but your groupings will become quite large. To demonstrate the impractical lengths to which this could be taken, consider this example of a standard: A standard for drilling holes having diameters from 1-5 inches, to depths varying from 1-5 inches. But what's the point? This standard is too broad, so why not begin adding more and more precise standards? That means there will be more to be entered, more to be maintained, and users might easily end up with a database containing 10's of thousands of standards. And that's quite likely, considering the fact that there are thousands of material types, each requiring a standard.

That would be a pure standards-based approach to estimating, and that is impractical. So, standards-based programs may use interpolation to determine the time for the 1.5" hole from the 1.0" standard. Realistically, you can't take pure ratio's for they do not consider the travel times, nor chip removal times. Such a method would be a combination between a pure standards-based approach, where there is no interpolation of the data, and a standards-based with interpolation. Standards-based without interpolation uses the standards as provided. Standards-based with interpolation indicates that some manipulation has occurred using the standard, based upon the specific circumstances of the job. Now, here's the danger with interpolation: it can invalidate a standard, completely, for a standard is a standard only if it retains its value. Any modification, whether by guessing or by interpolation, destroys the standard and there no longer is reliability.

For a standards-based estimating system, its accuracy is based on how much time you put into it. In reality, the time it will take a user to correctly set up a standards-based package will rule out your setting it up correctly. For example, all it takes is a change in cutting fluid to invalidate your standards, for suddenly everything is different. You finally got your half a million standards set up and purchasing changes cutting fluids because of price or performance. And don't forget: the cut time is different for different machines -- say a 5 a 10 HP lathe -- and the idle times are different. The possibilities are endless.

Addendum III -- Estimating's (near) future

Feature recognition

The success the solid modelling markets have seen to CAD products, apart from the steep price decline in the past few years, is obvious: even someone in accounting can look at the image and recognize the part. They do not need an engineering background to be able to "see" the part. And from that solid, it is possible to make G-code, since you can recognize all of the features: the holes, flat surfaces, radii, and so forth. As a result, most CAM systems now either integrate with a solids package, or have their own, internal solids drawing capability.

Now, feature recognition promises to take part design in a new direction. Solids packages enable feature recognition by other software. No more visualization, no more isometric views. Clusters of software packages that use solids will revolve around a technical database. This supports a consistent method of programming and manufacturing. In that technical database will be relationships of features to manufacturing methods. For example, a database entry for a ¼ - 20 tapped hole, ¾ deep 6061-T6 aluminum may specify that a .201 diameter drill be used and a .125 center drill spot that hole prior to drilling. When you machine a part out of 316 SS with a certain finish, the technical database can suggest that you machine using the specified speeds and feeds and tools you have determined are appropriate. Consistency. This is the way that hole will be made, because this is the rule you set up. Though this is the way it should be done today, on this machine, it doesn't mean that you can't change it. However, it helps you to quickly and efficiently identify how a part should be made.

Did you wonder why I mentioned a material in the previous example? Why is the material important in a solids program? Because knowing it there can provide the part weight, center of gravity, and possibly highlight other potential design or manufacturing challenges. Solids are already at the next level, today, where finite element analyses can help designers make parts faster, stronger and more efficiently. A solids program gives you the power of information.

Automatic feature recognition and estimating

Soon an estimator, knowing a part was designed in a solids program, will click on a pull down menu and choose "recognize features." When the software recognizes a "shaft adapter," and the estimator selects the machining center to run the part, the software will suggest a center drill, drill, face milling operation, contour end milling operation, and a burnishing operation. That's the set of work instructions based on the features it saw. The estimator will be able to drag and drop and rearrange any tool or process, based upon his knowledge of the facility. The point here is that this is a technology tool and will still require experience to know what can be done.

Testing of feature recognition in computer aided estimating has already begun. Consider a simple, flat part, linked from a solids package to estimating and then to a CAD/CAM system. The identified features from the solids technical database are matched with the estimating program's technical database. With this data, the estimating program recognizes the part and knows which tools to use. Perhaps there are several machines on which the job might run, so the estimator selects the machine. By automatically reviewing the machine parameters and the part, he may find that the initially selected machine is unavailable, or otherwise not optimal for the job; perhaps the first choice has insufficient tonnage, tolerances, or size limitations. Therefore, another machine could be selected by the estimator.

Let's take this to the next step, by linking with a tool management database. Perhaps it is found that one of the appropriate tools is unavailable and it is necessary to modify some tool selections. The estimating program helps make this choice, based on machine parameters/capabilities. Finally, the software calculates the cycle time, applies it against the shop labor rates, efficiencies, and other considerations and provides you the part cost.

Addendum IV -- Estimating example

Pocket Mill Example

4.25" long x 3.25" wide x 2.25" deep pocket
.750 diameter end mill
50% step over
1" axial doc (depth of cut)

Standards based
31.078 cubic inches / standard time to mill 1 cubic inch = time
Engineering based - dumb
31.078 cubic inches / removal rate per cubic inch = time
Engineering based - smart
7 radial passes x 4.25" long x 3 axial passes = 89.251 inches
89.251 inches / feed rate = time
Simulation
Note: Rapid travel consists of acceleration time + rapid travel rate based on rate increasing with distance + deceleration time
Note: All feed rates consider maximum cutting feed rate and horsepower of machine

1. Rapid approach part from tool home position
2. Plunge cut - .750 diameter x 1.00 deep (plus approach distance) using plunge feed rate
3. First pass - .750 wide slot x 3.5 long 100% buried using slot feed rate adjusted for axial depth
4. Step over - .750 wide step over .375 deep (50% of cutter) using slot feed rate
5. Peripheral pass - 3.50 long pass at .375 radial depth using peripheral feed rate
6. Repeat steps 4 and 5 for 6 peripheral passes
7. Last peripheral pass - Only a partial pass remains 3.50 long at .250 deep using new peripheral feed rate
8. Rapid move away from part .020 for clearance
9. Rapid travel back to original plunge location to begin second axial pass
10. Repeat steps 2 - 9 for second axial pass
11. Only .250 remains for semi finish pass, so repeat steps 2 - 9 using .250 axial doc and adjust feed rates accordingly
12. Rapid out of pocket 2.25 plus approach for clearance
13. Rapid back to home position for tool change if required

Rough Turning Example

3.500” stock size
3.075” diameter x 1.00 long
2.600” diameter x 1.00 long
2.150” diameter x .750 long
.125” radial depth of cut/pass
Constant SFM
.005 radial depth left for grinding

Standards based

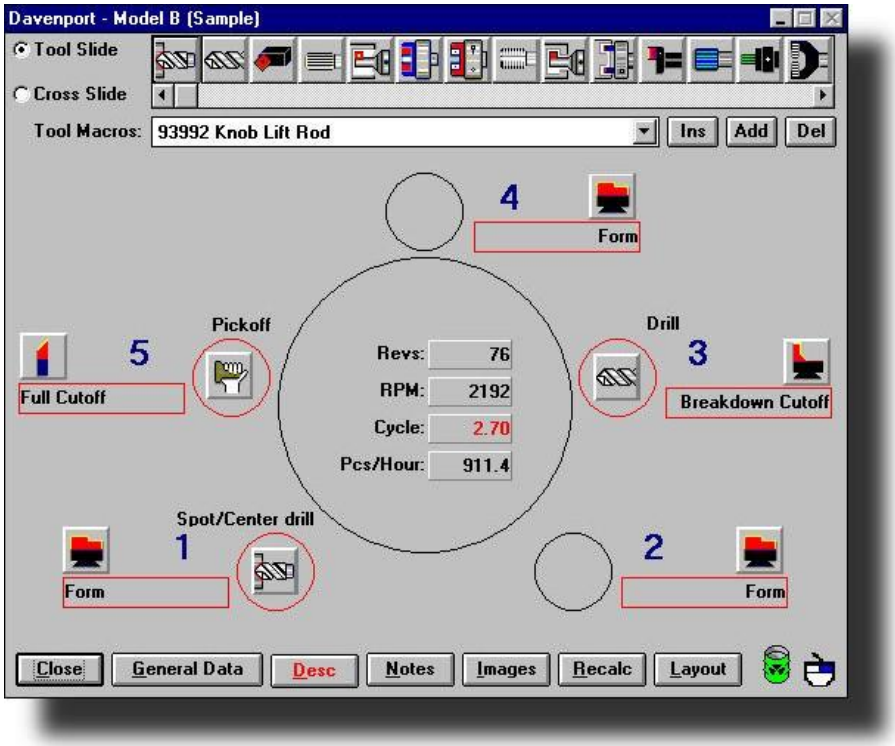
1. Standard time to turn 3.5” diameter .5 deep factored for actual 3.075 diameter and .2125 depth
2. Standard time to turn 3.0” diameter .5 deep factored for actual 2.600 diameter and .2375 depth
3. Standard time to turn 2.5” diameter .5 deep factored for actual 2.150 diameter and .2250 depth

Engineering based - dumb

1. 2 passes at 3.500 diameter x 2.750 long/feed rate +
2. 3 passes at 3.075 diameter x 1.750 long/feed rate +
3. 2 passes at 2.600 diameter/feed rate x .750 long = time

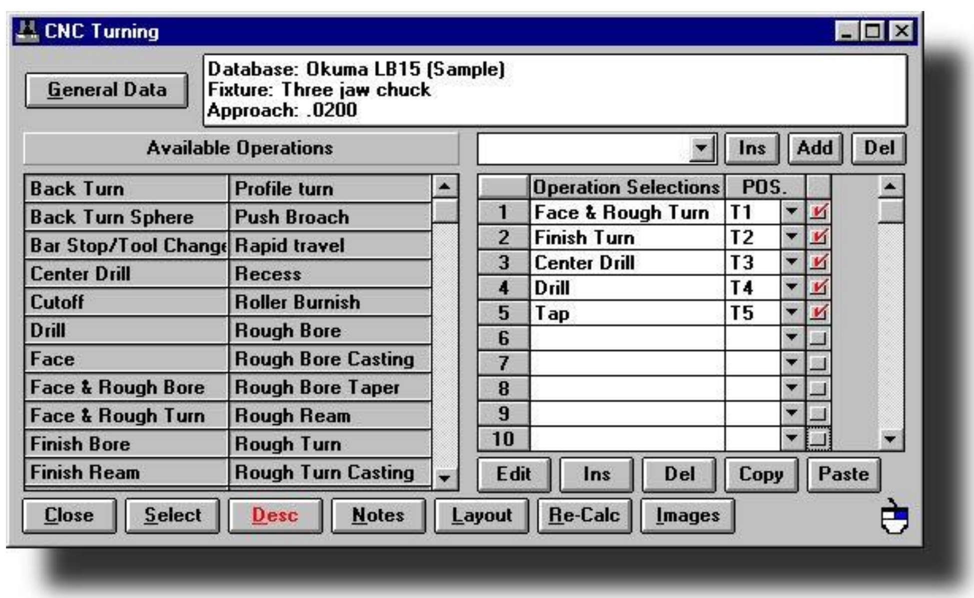
Simulation

Note: Rapid travel consists of acceleration time + rapid travel rate based on rate increasing with distance + deceleration time
Note: All feed rates consider maximum cutting feed rate and horsepower of machine
Note: RPM is recalculated at each depth of cut to allow for constant SFM
Note: RPM is constantly compared to machine maximum and minimum for conflict
Note: RPM is calculated at cutting point of tool (diameter - doc. *2)
Note: Approach distance (.100) is added to each pass
1. Rapid travel from home position to .100 away from face of part
2. Turn 3.250 diameter x 2.850 long at .125 doc
3. Machine up face .125 + .010 to clean up the face and retract the tool
4. Rapid 2.850 back to face of part and down to next cutting diameter
5. Only .0775 remains on first diameter after leaving .005 grinding stock so look ahead to next diameter. Turn 3.00 diameter 1.85 long at .125 doc, face from 3.00 to 3.080, turn 1 inch at .0775 doc and face up .0775 + .010 to clean up face and retract tool. (Feed rate varies with doc and RPM is calculated appropriately including facing cuts)
6. Rapid 1.850 back to face of part and down to next cutting diameter
7. Turn 2.750 diameter 1.850 long at .125 doc and face up .125 + .010
8. Only .070 remains on the second diameter, so look ahead to next diameter. Turn 2.50 diameter .850 long at .125 doc, face from 2.50 to 2.605, turn 1 inch at .070 doc and face up .070 + .010
9. Rapid 1.850 back to face of part and down to next cutting diameter
10. Turn 2.250 diameter .850 long at .125 doc and face up .125 + .010
11. Only .045 remains on the third diameter, so turn 2.160 diameter .850 long and face up .045 + .010 to clean the face and retract the tool
12. Rapid back to home position for next tool change if required



This screen represents a Davenport screw machine to the estimator. Across the top of the screen are varieties of tools that can be assigned to this machine's five stations. After choosing tools for each of the five stations, the estimator assigns an operation (form, Cutoff) to each cross slide. With minimum input, the estimating does the work and also provides the graphical analysis tools that help the estimator select the best machine for the ideal results.

For the CNC Turning operation, the estimator selected five operations from the Available Operations list at the left: Face & Rough Turn, Finish Turn, Center Drill, and Tap. These accomplish the same result as the Davenport method above. When given the diameter and length of the turn, the estimating system returns the maximum depth-of-cut based upon the machine horsepower and machine type. Complete standard work center descriptions include predefined load, unload, and setup time standards.



SOFTWARE FUNCTION CAPABILITY CHECKLIST

To Assist in the Software Evaluation Process
Developed by and reprinted with permission of the PMPA's Software User's Group

ESTIMATING

A. General

1. Quote Log

2. Quotes Due Report

B. Header Information

  1. Integrated to customer file
  2. Quote # internally assigned
  3. Salesman's commission file

C. Material

  1. Material weight calculated
  2. Scrap factor estimated
  3. Scrap factor calculated
  4. Scrap value calculated
  5. Number of quantity fields

D. Operations

  1. Sequence numbers in multiples of 10
  2. Work Center numbers
  3. Operation numbers
  4. Operation description field
  5. Setup hours and rate override ability
  6. Cycle time in seconds calculation
  7. Cycle time in seconds estimated
  8. Production rate estimated
  9. Production rate calculated
  10. Production rate in Pcs/Hr
  11. Efficiency factor for each work center
  12. Override default efficiency factor
  13. Efficiency factor based on bar loading/tool life
  14. Factor for # machines attended by operator

E. Tooling Costs

1. Special tooling amortization option
2. Special tooling charged to customer option
3. Perishable tooling amortization option
4. Description field

F. Outside operations

1. Minimum charges
2. Price per unit
3. Surcharge option
4. Freight charge
5. Certification required (Y or N) field
6. Final cost calculation

G. Optional Information

1. Estimated annual usage
2. Inventory carrying charge (%/yr) constant
3. Calculate Economic Production Qty (EPQ)
4. Calculate Economic Order Qty (EOQ)

H. Reports

1. Quote Log or Status Report
2. Estimate with details
3. Quote from or letter for customer
4. Monthly Quote Summary