A hierarchical cost estimation tool

David Koonce*, Robert Judd, Dusan Sormaz, Dale T. Masel

Department of Industrial and Manufacturing Systems Engineering, Ohio University, 268 Stocker Center, Athens, OH 45701, USA

Received 18 January 2002; accepted 17 January 2003

Abstract

The estimation of the manufacturing cost of a part in all phases of the design stage is crucial to concurrent engineering. To better estimate the cost for a product, data must be available from both engineering systems and business systems. This paper presents a cost estimation system being developed to support design time cost estimation using the Federated Intelligent Product EnviRonment (FIPER), which is being developed as part of the National Institute of Standards and Technology (NIST) Advanced Technology Program (ATP). The FIPER research team is developing an architecture that interconnects design and analysis software tools in a peer level architecture to support multidisciplinary design optimization (MDO), design for six sigma (DFSS) and robust design.

#2003 Elsevier Science B.V. All rights reserved.

Keywords: FIPER; Cost estimation

1. Introduction

One key constraint in engineering design is product cost. However, estimation of the cost of a yet to be produced part, or to be designed product, is a difficult process. A new method for design data integration, under development and being sponsored by the American National Institute of Standards and Technology (NIST) allows for a new and truly integrated cost estimation process for aiding design time cost analysis. The project, named Federated Intelligent Product EnviRonment (FIPER), unifies design tools for optimization across multiple analytical disciplines.

As implemented, FIPER now gives unprecedented access to data in design and analysis tools. For cost to be included in this optimization, a new and highly integrated cost estimation tool is now available. This tool includes the capability of developing cost estimates of a product from whatever data is available in the environment. For instance, in the early stages of a design, when data is limited, simple parametric cost relations, like weight-based relations may be used. As the design is further developed, or detailed, more information is available and cost estimates can be developed on design features, manufacturing features or even a process plan. New cost estimating relations will be used instead of the earlier parametric relations. Additional features, such as estimate error scaling and trade study capability are also included to provide a tool useful in system optimization. This paper presents the design and implementation of a new highly customisable cost integration tool to support design time optimization that considers cost as an objective function or constraint.
1.1. FIPER

FIPER [1], an acronym for Federated Intelligent Production Environment, utilizes a technology called intelligent master modeling (IMM) to allow design engineers to reduce the time for evaluating potential designs across all analytical disciplines. FIPER supports advanced design methodologies such as design for six sigma (DFSS), multidisciplinary design optimization (MDO) and robust design. While the IMM helps enable coordination between design and analysis, a supporting architecture will be built upon the concepts of federated information systems. The FIPER infrastructure is developed entirely in JAVA to support the mixed computing platforms typical in product design.

FIPER is part of a 4-year NIST ATP. In specifying the necessary capabilities and services in FIPER, five problems common to many businesses in the United States are addressed:
1. the need to reduce time to market;
2. the need to reduce design cycle time;
3. the need to reduce product costs;
4. the need to improve product performance; and
5. the need to improve product quality and reliability.

To satisfy the need to reduce production costs, FIPER must have the ability to accurately predict the cost of a potential design. The software tool that generates the cost estimates must operate in the FIPER environment and be able to, with no user interaction, generate a cost estimate as a service to a calling program.

2. Manufacturing cost estimation

It is often posited that the major portion of a product’s cost, as much as 80%, is determined early in the design process. Decisions like selection of a material can easily be seen to impact the product cost. However, decisions like a radius or blend may result in the need for a tool change, new setup or even a processing machine change, adding to the manufacturing cost of a part. As such, producibility is often included in the estimation of a part’s cost. Regardless, functional specifications usually drive the design process [2].

Despite the importance of the design details to product cost, a recent study found that the delay between the design decision and cost determination hindered the designer’s ability to learn about the process implications of design decisions [3]. In addition, the consequences of the decisions that impacted the cost were often not fed back to designers at all [3].

Numerous commercial cost estimation tools exist and many organizations have developed proprietary cost estimation systems. A detailed analysis of existing commercial software tools and major proprietary systems is found in [4]. The sophistication of these tools ranges from spreadsheets to multi-user mainframe database systems. The capability of these systems ranges from special tools with the ability to estimate costs for only highly specific parts to generic systems which can be used to estimate costs for virtually any manufactured part. These systems are developed to provide more accurate cost estimates; many companies have developed detailed cost estimating systems that are specifically designed for their products and/or processes. The extent to which companies have invested in developing these proprietary models demonstrates the importance of accurate cost estimates to industry.

A review of specific systems developed by companies involved in defense contracting is contained in [4]. The applications described include cost estimation for a contractor, systems engineering costs, and program management costs.

One of the systems described is the “cost offering method for affordable propulsion engineering acquisition and test” (COMPEAT™), which was designed to estimate the life cycle costs associated with jet engines. The model requires key engine parameters as inputs for estimating the cost. The system also considers the actual and estimated costs of existing parts and uses this information to scale the cost estimate and improve the system’s accuracy [4].

Another company-developed model was created to estimate production costs for several product lines. To provide flexibility in the situations that are considered when estimating, the system allows for a multi-year planning horizon and indefinite production quantities. The costs are estimated for the different aspects of production, including material requirements, assembly, inspection, and manufacturing support.

Lockheed also uses matching of new designs to an existing design to provide an estimate of the costs [5].
The process of identifying the airframe that is most similar to the one being designed was automated using SEER-DFM software from Galorath Inc. With the aid of the software, the time required to complete the matching was reduced by 75% over the time required when the process was performed manually.

At Parker Hannifin, a dedicated Value Added Value Engineering (VAVE) Group utilizes cost estimating software to support the purchasing department [6]. The software can consider multiple manufacturing methods, to match the conditions in a supplier’s facility, and the appropriate feeds and speeds for the material and process. They can also consider different production quantities, depending on whether it is a prototype or full-production part. The information produced by the VAVE allows the purchasing department to know what a part should cost so that they can deal with suppliers fairly.

Regardless of the sophistication or size of the system, the manufacturing cost of a part is estimated using one or more of four basic methods: intuitive, analogous, parametric and analytical [7]. The intuitive approach relies on the experience of the estimator to predict the cost. An analogical estimate is essentially a variant estimate using similar parts, often matched using a group technology code. Parametric estimates use the values of key part attributes to determine the cost. A parametric estimate may rely on very high-level parameters of the product’s performance or use detailed geometric data. Lastly, an analytical estimate relies on a summation of the steps in the production process; and as such, can only be done late in the design process.

A newer trend in cost estimation is the inclusion of manufacturing system costs in the estimation of a part’s cost. Various methods in this area include: activity-based costing (ABC), throughput accounting, target costing, life cycle costing and strategic accounting [8]. For example, traditional ABC can be further decomposed into activity-based costs such as processing cost and non-activity-based costs such as inventory holding costs [9]. Costs such as these are usually buried in an overhead factor. Overhead is also an area of cost estimation research with the notion that product complexity is the primary driver for overhead [10]. This study indicated that production volume and the number of transactions like engineering change orders strongly correlated with manufacturing overhead costs. For example, production volume may necessitate a need for increasing capacity or even process capability in the manufacturing system.

3. FIPER cost estimation tool

The architecture of the FIPER cost estimation tool (FCET) is largely influenced by the cost estimation methodologies employed. FCET can employ a combination of generative and variant costing, with designs being evaluated using either a work breakdown structure (WBS) or a parameter-based estimation from a similar part. A fully functioning prototype of the FCET has been developed using JAVA 1.3. The basic design and functioning of the tool are explained in this section.

3.1. Architecture

The FCET consists of a cost engine, graphical user environment (GUI) and an element builder. The elements are coded as JAVA classes and executed in the cost engine. Execution may be done through FIPER calls or through the GUI. Fig. 1 shows a simplified architecture of the tool. Elements are developed using the element builder. Since they are JAVA classes, detailed editing or sophisticated calculations can be manually programmed in the code.

3.2. Elements

Elements form the basis for all estimates. An element can be anything which incurs a cost or is a group of things which incur costs. For example, an element may be a part, an operation, a group of parts that make a product, or even an inspection. Stored as executable objects, elements are usable in many cost estimates and are the instantiation of an organization’s product families and processing logic. All elements are stored in a library and are available for use in many estimates. The FCET environment includes many objects that are combined in JAVA code to define an element.

To form an estimate, a WBS of cost elements is built. This WBS can be predefined in the element, or built ad hoc using the graphical user interface. Fig. 2, a simple WBS tree for a chair, shows that the chair being estimated consists of a back element, a seat element
and a front element. Similarly, a back contains a side rail, a back slat and a cross bar. Notice that while the front of the chair in the WBS has one leg element, the chair, of course, has two front legs. Since each leg is identical, with respect to the cost estimates, only one element is shown in the tree. To account for the added costs of the second leg, all elements have a quantity field, and calculations that result for the one leg element will be multiplied by the quantity. However, if each leg were unique, with respect to cost, individual elements in the WBS would be used to model each leg.

To define the WBS and produce estimates, elements have five major components: attributes, buckets, cost relationships, estimation methods and possibly, maps. The following section defines these components and describes how they relate to define an element and ultimately an estimate.

3.2.1. Attributes

Attributes are the properties of element instances. Attributes are variables used to define the element’s cost or relationship to other elements. A simple use may be to participate in a parametric cost relationship. That is, a variable, like length, may be used to determine a simple volumetric cost estimate. Attributes may also be used as an intermediate variable used in a table lookup for determining a variable in a cost relationship. For example, a table of material costs may use an attribute for material name to find the associated unit cost for that material. Attributes may even define elements in the WBS. For example, an attribute like length may be used in an equation to determine the number of reinforcing ribs necessary for a structure. Lastly, attributes may simply store an informational value, such as a part number.

The FCET defines a set of attribute types (JAVA classes) which have defined representations in the system and the GUI. In constructing an element, a developer simply specifies the necessary attributes, their default values and any variable specific proper-
ties, like digits displayed or units. Allowable attribute types are the following.

- Boolean: a Boolean attribute may contain true or false values only.
- Double: a double is a standard double precision number.
- Double array: a double array is an array of double precision values.
- Integer: an integer attribute holds integer values.
- Integer array: an integer array holds an array of integer values.
- String: a string attribute holds a string value.
- List: a list attribute holds a set of allowable string values for an attribute.
- Cascaded list: a cascaded list attribute contains a nested list which allows hierarchical values to be presented. For example, all aluminium materials would be under the aluminium branch.

Attributes obtain their values from user input, FIPER calls, predefined default values, or from their parent elements through inherited attribute values. All attributes also maintain the previously set value for quick what–if analysis using the GUI tool. Also, double attributes support an increment setting for use in the GUI.

3.2.2. Buckets

Buckets are the accumulators for estimated values. Most estimates will have a total cost bucket. Cost relationships will add their individual costs to the total cost bucket. For example, an element may calculate material cost and a machining cost which are both added to the total cost bucket. However, these individual costs, like material, or intermediate calculations like the machining time used to estimate machining costs might also be of use to a user. The FCET supports additional buckets for calculation of custom estimates like labor hours, material cost, setup hours, etc.

Exploiting the power of the WBS, the FCET sums local (element level) values as well as bucket values of identical buckets in the children of the element. So, the material costs for elements below the current element in the WBS are included in the material cost for the current element. The effect is that the material cost for the root node of the WBS is the material cost for the estimate.

3.2.3. Cost relationships

Cost relationships are the equations that produce the bucket values. Since buckets may represent accumulators other than cost, cost relationships can be used to determine other values like machining time. Coded in JAVA, cost relationships may be simple sums, sophisticated formulas, table lookups or calls to custom code.

The FCET has a powerful table manipulation facility which allows for table lookups of local and remote data. The table tool employs relational algebra for querying tables and extracting data elements necessary for calculating bucket values. Aside from developing cost relationships, tables can be used to identify allowable values for list attributes.

3.2.4. Estimation methods

All elements must have an estimation method. However, elements may have more than one estimation method. That is, there may be more than one way to estimate the cost of a part. Each method uses different cost relationships and potentially different attributes. In addition, some methods may include additional sub-elements. Methods that include additional elements are called aggregate methods, while methods without sub-elements are called base methods. Aggregate methods determine their bucket values by summing the buckets of their sub-elements. The determination of which method to use for cost estimation should be made based on the data available. For example, early in design time, little may be known about an element’s components (sub-elements) and a simple parametric estimation method would be employed. However, as more design details are developed, an aggregate method may be employed to estimate the element by summing the detailed estimates of the components. Fig. 3 shows the relationships between attributes, methods and buckets in an element.

3.2.5. Maps

Integral to methods is the ability to map attributes from a parent element to attributes in the sub-elements or map between attributes in the same element. Self-maps allow attribute values to alter or modify the values in other attributes of the same element. Between elements in a tree, maps are pull maps in which a child element pulls the values from the parent element. For example, the material attribute in an element would be
used as the material for the child element. Maps can be used directly, as described previously, or through a translation equation. For example, if a one bolt is needed for every 2 in. of circumference, the diameter of a parent element could be translated into the number of boltholes on a flange sub-element. Fig. 4 shows the simple mapping of attributes from a parent to a child.

At any time, the user may void maps by overriding an attribute’s value. Manually setting an attribute, which is normally determined by a map, changes the setting to override and the map is not enforced unless the attribute is reset to a calculated state. Fig. 5, a set of attributes from the back of the chair, shows that material is overridden from its programmed value.

The symbol to the left of the attribute indicates that the current value is not the default value or the value calculated through a map. However, height and width are calculated through the map, as indicated by the symbol to the left. Also, Fig. 5 shows that the number of slats is currently set at the default value.

3.3. Capabilities of the tool

To support the mission of iterative design optimization, including cost as a parameter, the FCET must accurately estimate costs without human intervention and be able to estimate costs with the data available during any stage of the design cycle. The two

![Diagram of attributes, methods, and buckets](image1)

Fig. 3. Attributes used in the methods and buckets.

![Diagram of maps between parent and child elements](image2)

Fig. 4. Maps between parent element and child element.
3.3.1. Attribute flow down

In a WBS tree, high-level elements will have attributes that are related to attributes in elements lower in the hierarchy. For example, a material attribute may directly link to the material of a sub-element. Or, a material attribute may link through a relationship where the selection of a parent’s material causes a different material to be used in a sub-element. Attributes that influence the values of attributes in a sub-element can be used, through FCET’s mapping capability, to support flow down. Attributes that have been overridden at a sub-element do not participate in attribute flow down. Flow down is a powerful process by which simple changes in an estimate’s attributes cause the re-calculation of the entire estimate, as the new values are evaluated in the WBS tree. Fig. 6 shows the flow down of attributes and the rollup of buckets.

A simple change in an attribute, like the length of a part, may also trigger the change of cost method. In a simple case, an attribute like length may indicate the bounds of the process capabilities and change the cost relationships. In a more complicated case, the length of a design may determine the number of stiffening ribs in the WBS tree. Lastly, an attribute’s value may signal the change of a whole design family. The change of a material attribute from a metal to a composite could dramatically change the WBS tree.

3.3.2. Zooming

The data available directly impact the ability to estimate the cost of a part. The FCET supports the use of different estimation methods for an element. Methods require specific attributes for their cost relationships. Simple cost relationships may only require a few attributes, while more sophisticated cost relationships will require added attributes. Zooming is the process of supplying added attributes to an estimate, increasing the complexity of the estimation process, to provide a higher fidelity estimate.

As with attribute flow down, zooming may change the WBS tree. As more information is known, the tree may change to support detailed estimates of the sub-elements and a roll-up of the buckets from these detailed estimates.

3.4. Enhancing an estimate

An estimate is in fact a point estimate with some associated error. The FCET environment must support techniques for identifying error, reducing error and exploring the design space with respect to cost. Three additional features of the FCET that focus on estimate

![Fig. 6. Flow down of attributes and rollup of buckets.](image-url)
variability are: cost scaling, risk analysis and trade studies.

3.4.1. Scaling

Key to cost estimation is the minimization of error in the equations estimating the cost. One method for minimizing error is to use cost estimates of existing parts. By comparing computed cost with actual cost, a measure of the error can be determined and applied to a new estimate for a similar product. Fig. 7 shows the equation for scaling a cost estimate using a closely matched part. The keys to scaling are first having a complete and accurate database of current estimated part costs with their actual costs and second determining which existing part most closely matches the planned part. The determination of a match part can be determined one of two ways. First, an estimate may begin by using an existing estimate and varying the parameters for a new product (the initial product becomes the match part). Or, the user may browse similar estimates to identify one that has similar attribute values.

While the calculation for scaling is not complicated, determining a closely matching part may be difficult. A set of matching parameters must be defined for each element. For scaling to work, the error in the equations must be duplicated with both sets of calculations. For example, if two parts with very similar parameters use significantly different materials, then the method for estimating the costs may be entirely different. In this case, scaling the new part against the old part may only add random error. Beyond scaling processing cost, labor scaling is possible if data is available. However, for purchased parts, material and labor are combined which makes decomposition for scaling more difficult.

3.4.2. Risk analysis

Risk analysis is done using Monte Carlo simulation. Given a set of inputs that may be any combination of deterministic and probabilistic values the Monte Carlo simulation generates minimum, maximum and average cost estimates. Additionally, the variance is reported as well as a graphical representation of the resulting distribution. The cost estimates are based on user-defined confidence interval levels (e.g. 90, 95%, etc.) or a user-defined run length (number of simulated trials).

3.4.3. Trade studies

Through the use of attribute flow down, the FCET directly supports a powerful what–if analysis capability. The effects of the change in a few attributes can be immediately seen in the roll ups of the buckets. Restoring the attributes to the previous values restores the estimate to the previous state. The cost estimation tool also supports the comparison of stored estimates for trade studies. An additional tool allows cost estimators to compare two or more estimate to determine the effects of different attributes on the bucket values.

4. Prototype implementation and conclusions

The FIPER cost estimation tool was prototyped in JAVA and in now a fully functional application. Both simple and sophisticated estimates have been developed to test the capabilities of attribute flow down and zooming. Additionally, extremely large and complex estimates have been developed to evaluate processing speed and memory. These estimates, with over 16,000 elements in the WBS tree and 80,000 attributes, can flow down attribute changes, like material, to all elements and roll up all buckets in under a tenth of a second on a modern microcomputer.

As part of a research and development program, 80 large, rotational jet engine parts have been modeled and estimated with the FCET methodology. The accuracy of the individual estimates improved by 22%, compared with the accuracy of existing cost estimates.

Fig. 8 shows a sample rotational part in the graphical user interface of the tool. It should be noted that the model was developed for a major aircraft engine manufacturer and that the geometry does not represent a realistic disk. The material and processes have been given neutral names. And, the hours and labor rates are not realistic. The left pane contains the current WBS tree, while the main pane contains the attributes for the
selected element’s cost method. Note that since the FCET tool is developed in JAVA, a simple 2D rendering of the design can be added to the graphical user interface. Below the main pane, is a tabbed pane region which contains summary reports, risk analysis, trade studies and calibration. Where applicable, graphics like the pie chart have been added to help users make better comparisons.

This paper has presented a new cost estimation tool capable of generating estimates at all stages of the design process. FIPER presents an excellent opportunity for the creation of a design time cost estimation tool. With FIPER providing the necessary design data and a directed search toward optimality with respect to cost and other constraints, the FIPER cost estimation tool will help designers reduce costs while meeting performance constraints.

References


David Koonce earned his BS, MS and PhD degrees from Louisiana State University. He is an Associate Professor at Ohio University, in the Industrial and Manufacturing Systems Engineering Department. His research interests include, database systems, data mining, software integration and cost estimation.

Robert Judd is the Cooper Industries Professor in the Electrical Engineering and Computer Science School at Ohio University. He has over 20 years of teaching and research experience in simulation and control of manufacturing systems and cost estimation. He has published over 100 articles and received over $3.5 million in funding in these areas.

Dusan Sormaz is an Assistant Professor of Industrial and Manufacturing Systems Engineering at Ohio University, Athens, USA. He received his PhD degree in industrial and systems engineering and MSc in computer science from University of Southern California, Los Angeles, CA, and MSc and BS degrees in industrial engineering from University of Novi Sad, Yugoslavia. His principal research interests are in process planning, CAD/CAM integration, computer integrated manufacturing (CIM), concurrent engineering, group technology and application of knowledge-based systems in integrated manufacturing.

Dale T. Masel is an Assistant Professor of Industrial and Manufacturing Systems Engineering at Ohio University in Athens, Ohio. He received his PhD from The Pennsylvania State University in Industrial Engineering. His areas of research interest are manufacturing cost estimation and the management of warehousing systems.