IMPROVING UNIT COST ANALYSIS BY GENERATING DYNAMIC FACTORY SIMULATIONS FROM CAD GEOMETRY

S. Jinks[1], Prof J.P Scanlan[1], Prof P.A. Reed[1], Dr S. Wiseall[2]

[1] School of Engineering Sciences, University of Southampton, Southampton, SO17 1PN, United Kingdom, email: sj1103@soton.ac.uk

[2] Rolls-Royce Plc, PO Box 31, Derby, DE24 8BJ, United Kingdom

Abstract: This paper introduces the development of a framework specification that combines Computer Aided Design (CAD) and Computer Aided Process Planning (CAPP) technologies with a knowledge database, to provide data for conducting manufacturing simulations, to improve cost estimation. By combining CAD and CAPP technologies with a knowledge database and a Graphical User Interface (GUI), feedback of cost estimation results can be given directly to a design team within their design environment. The feedback should aid a design team to understand the consequences of design decisions in terms of cost and manufacturing resources. A system is being developed to implement the framework and a case study of a simple component is proposed.

Keywords: Cost estimation, Factory simulation, CAD, CAPP.

Introduction

A Product Development Process (PDP) represent the life cycle of a product from conception of the initial idea, through design, manufacture, operation and finally disposal. A PDP typically contains a number of stages that can be classified into categories. The Rolls-Royce PDP [1], illustrated in Figure 1, contains seven stages and three categories. This paper will focus on design, and more specifically stage 1, preliminary design.

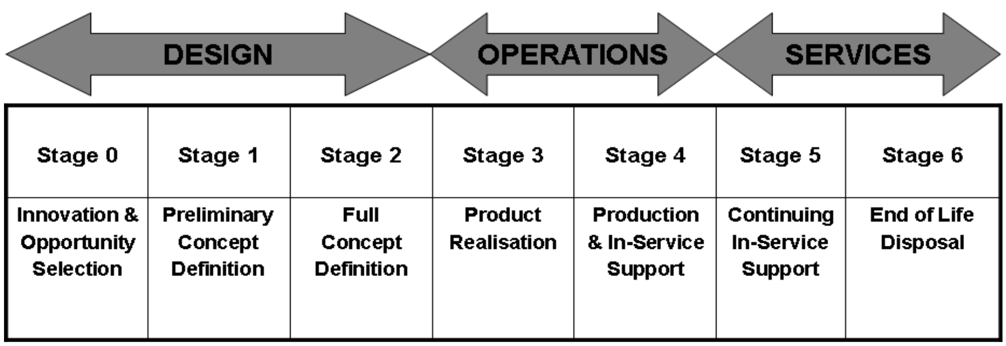

Figure 1: Rolls-Royce product development process [1]

Stage 1 is concerned with developing a preliminary design of a product that fulfils a business opportunity. To complete a stage and continue along the PDP, a series of reviews are undertake, these include [2]: manufacturing capability and capacity, supply chain capability and capacity, design and cost. Once stage 1 is complete a decision can be made whether to make a formal offer to the customer and continue the PDP to stage 2.

The product development process tries to minimise risk by reviewing and completing formal gates that must be passed before the next stage can be commenced. Reviewing a product before it is fully defined can be difficult. For instance, Figure 2 shows how the cost of a product is not fully determined until production has begun, and the majority of unit cost is unknown until the product design is frozen. Cost estimation tries to move the cost determination line towards the start of the design process, by aiding designers to analyse and understand the consequences of design decisions in terms of cost at the preliminary design stage [3].

Rush et al [4], Curran et al [5], Niazi et al [6] and Tammineni [7] have reviewed extensively cost estimation research and methods. The main cost estimating methods include: analogy, parametric, activity and feature based methods.

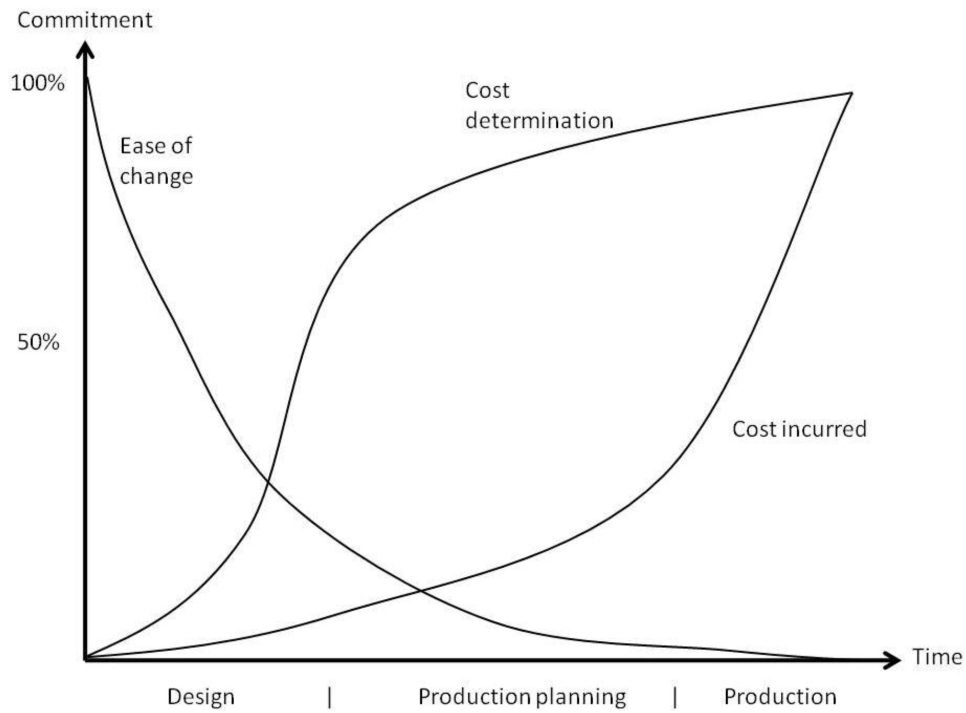

Figure 2: Product creation and costs [8, 9]

Analogy cost estimating is based on adjusting the cost of a similar product relative to the differences between the new and similar product. This method requires complete historical data of similar components, and appropriate scaling parameters to be applied [10]. There are risks associated with this method that relate to the amount and accuracy of historical data, and the scaling parameters used which require an understanding of the product.

Parametric based cost estimating uses Cost Estimating Relations (CER) and mathematical algorithms to estimate cost. A CER is developed by determining a correlation between the dependent variable cost, and independent variables such as size. A problem with this method is associated CER's having a limited range; this is because they reflect historical data.

Activity based costing focuses on estimating the cost incurred by performing activities to manufacture a component. This method can provide a detailed unit cost estimate, but requires substantial amounts of detailed data and expert knowledge to complete the estimate.

Feature based costing relates design or manufacturing based features to an associated cost. The unit cost of a component is the sum of all the associated feature costs. This method for simple components, where features are independent, allows a designer to understand the cause of cost, because when a feature is added cost is added to the total unit cost. Features are not always independent; they interact with each other, especially in machining, where many features are machined in one operation. This can result in total unit cost not reducing when a feature is removed, because the cost of an operation will be incurred regardless.

These main cost estimation methods have been successfully utilised, but have various shortcomings, including: basic visualisation in the modelling environment, little support to a user with limited programming skills, minimal ways to present manufacturing knowledge to the user and uncertainties are applied in a black box approach. These short comings have been addressed with the development of the knowledge based cost modelling system [7].

The knowledge based cost modelling system has addressed many shortcomings of the main cost estimating methods, but still contains limitations. The main limitation is an inability of static models to fully represent dynamic systems. A static model is defined as a representation of a system at a particular point in time [11]. Therefore when a static model tries to represent a dynamic system, such as a factory that machines components, each operation is represented individually, and a component flows through each operation sequentially. This is representative of a single component being manufactured, but not manufacture of multiple components. This is because queues may form if there is a bottle neck in a factory when manufacturing multiple components, such as in Figure 3.

In Figure 3, three components are being manufactured by three machines, where each machine has its own process time. When component 2 is required to be processed by machine 2, it has to wait in queue 1 for 20 time units until machine 2 is ready. A dynamic system as shown in Figure 3 is difficult to represent with static models, but not so with dynamic models. A dynamic model can be used to determine for a given production rate: required capacity (number of machines), machine utilisation and work in progress. This data can be used to improve unit cost estimates.

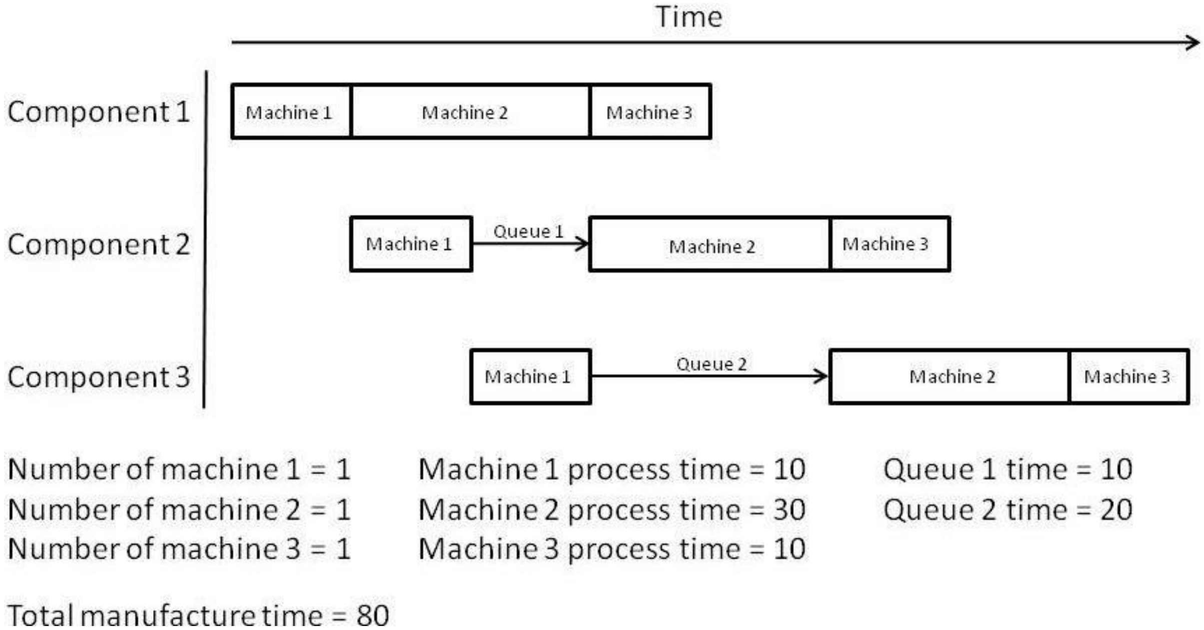

Figure 3: Holistic limitations – Manufacture of three components

Computer Aided Process Planning (CAPP) is defined as a system that can interpret a component design in terms of features, to automate process planning of component manufacture. CAPP systems are typically used to optimise the time to manufacture a component once a component has been fully defined [12, 13, 14]. This paper will utilise CAPP system methods, but at an earlier stage of design, to provide data to aid the creation of factory simulations.

However a main limitation of CAPP systems is unidirectional data flow from a Computer Aided Design (CAD) tool to a dedicated process planning system. If a system is to aid in understanding of cost and required manufacturing resources, data should flow back to the source, the CAD tool. A bidirectional data flow should allow a design team utilising a CAD tool to understand the effect of design decisions in terms of cost and required manufacturing resources.

This paper proposes a framework and system to automatically generate simulation models of manufacturing processes from CAD geometry. The system will feedback resource based data to improve cost estimation and aid design team understanding. A simple component will allow validation of the approach.

Frame work and system

To improve the current method of cost estimation for preliminary design, a new framework has been developed. The framework implements methods that aim to solve the two limitations identified earlier, these are:

- Providing a method to incorporate manufacturing simulation into unit cost estimation.

- Bidirectional data flow. This should aid designers to understand the consequences of design decisions in terms of cost.

The framework can be considered in four sections: geometry creation, component information, factory modelling and result publishing, and are linked as shown in Figure 4.

Geometry creation helps the user to create the component geometry. This section involves two stages; the first allows the user to select an initial component which is similar to the component that is required. To aid the user in selection, the user will follow a series of down selecting menus contained within the Graphical User Interface (GUI), which is linked to a database of initial component geometries. The second allows the user to modify the initial component geometry by removing or modifying features on the initial geometry, and by adding extra features that are applicable to the initial component type.

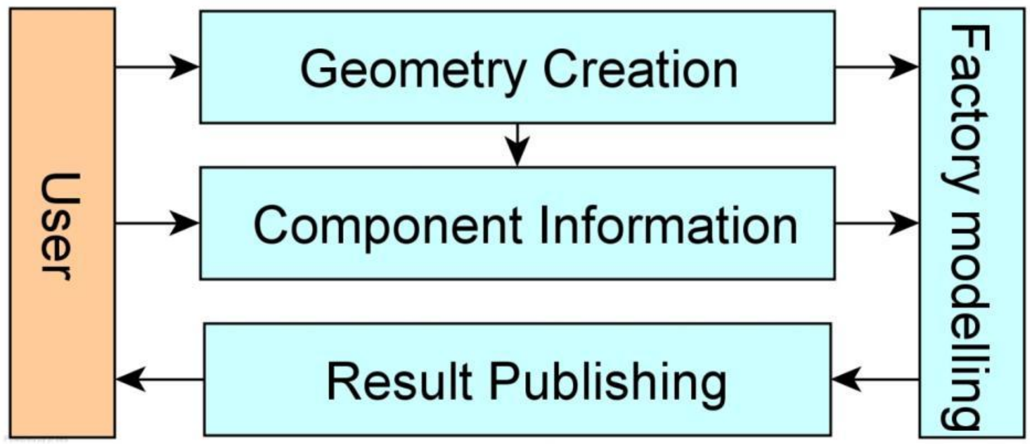

Figure 4: Framework sections

Component information collects information about the component for the factory modelling section. Two sources are utilised to complete this task, the first is a CAD tool. Specific geometry data is collected from the component, when an initial component or feature is added, initial geometry data is taken along with ID data about the geometry. When the user is finished with geometry creation each geometry data ID is checked for change, if a change has occurred the information is updated. The second source is the user. The GUI asks a series of questions, some generic and some specific to the component or manufacturing method type. This is the point at which possible manufacturing methods are identified, depending on information from the two sources and the knowledge base. The selection criteria for different manufacturing methods will check high level attributes, such as material compatibility, shape, size and historical precedence.

Factory modelling takes the information collected about the component and possible manufacturing methods, and applies it to a specific manufacturing method represented by a generic factory simulation model. A generic model is made up of processes that are always present within a specific manufacturing method. Blocks represent the individual generic processes or machines, and data is supplied to each of these, in the form of distributions from the database. Machines have a maximum utilisation, when a machine approaches this maximum utilisation another machine is brought online to fulfil the capacity requirement. This aids the determination of the number of machines required to manufacture a component at the required production rate.

The results publishing section acts as a feedback to the user. It returns a resource requirement for the specified data for each possible manufacturing method considered. If required, the user can specify a set of assumptions that allows a cost to be estimated, the user can then interrogate the cost to determine the cost drivers.

The preliminary system incorporates four tools: a CAD tool (Unigraphics NX (UGS NX)), a GUI written in C#, a database (Microsoft Access) and a simulation tool (Anylogic). Figure 5 shows which tools are used by each framework section. It can be seen that geometry creation utilises UGS NX and the GUI interface to create the geometry, and data concerning the geometry is stored in the database. Component information retrieves data concerning the geometry and manufacturing knowledge, and asks specific questions about the component and possible manufacturing methods, this data is then stored in the database. The factory modelling section retrieves data from the database, calculates process times and applies distributions, then loads the applicable generic factory model in Anylogic to perform the simulation. The results from the simulation are sent to the database, which are used by the results publishing section, that after processing are displayed to the user through the GUI.

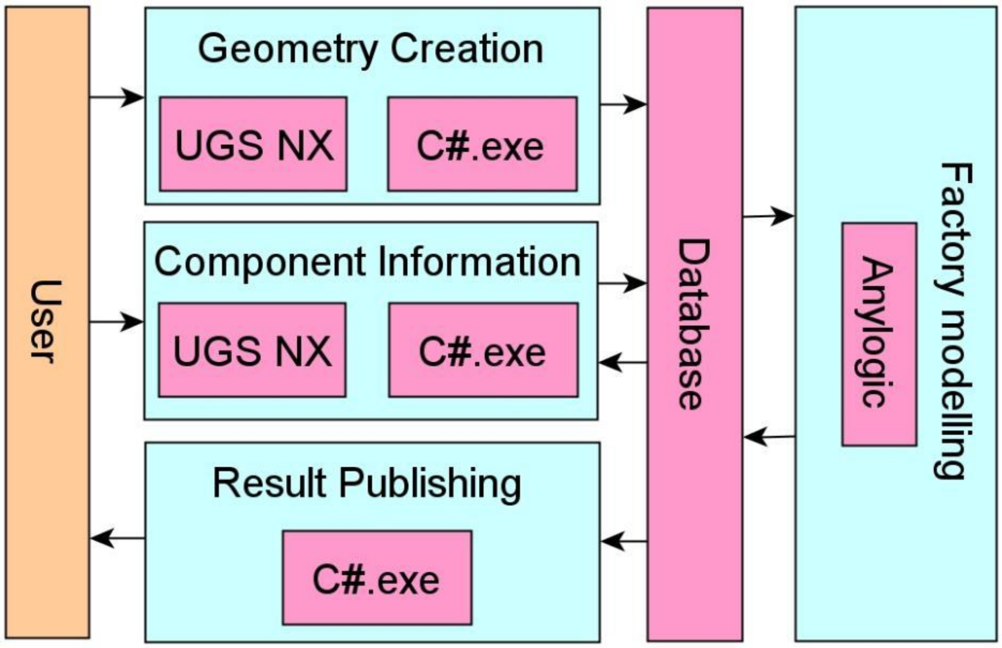

Figure 5: Framework sections with associated tools

The user interacts with the system in three locations: firstly with the geometry creation section, which allows the user to add and modify the geometry; secondly with the component information section to supply the required information for the factory modelling section, thirdly with the results publishing section. Figure 6 shows a flow chart of inputs, outputs and user interactions of the system.

Figure 6 is split into the different system sections and the grey nodes represent where the user interacts. It can be seen that the user interacts primarily with the geometry. The system should be capable of completing the factory simulations automatically once required data has been collected.

Proposed case study

The proposed case study is a simple cylinder that has three dimensions, inner and outer diameter and length (Figure 7). A cylinder was chosen because it is a simple representation of an aero engine case, which will be used for a full case study. In future work the cylinder case study will prove whether the framework and system configuration works in the correct manner, for multiple dimensional values, tolerance, materials and manufacturing processes.

The user will be required to interact with the system to supply data about manufacturing requirements and if necessary add features to a component. Figure 8 shows the add feature GUI, where the user has selected a cylinder for the initial feature, and the GUI has extracted the attributes and default values from the database. The database has 21 tables; fifteen are fixed data tables containing manufacturing specific information such as speeds and feeds for machining. Six data tables are used to store data about new components, such as the user inputted data and geometric data.

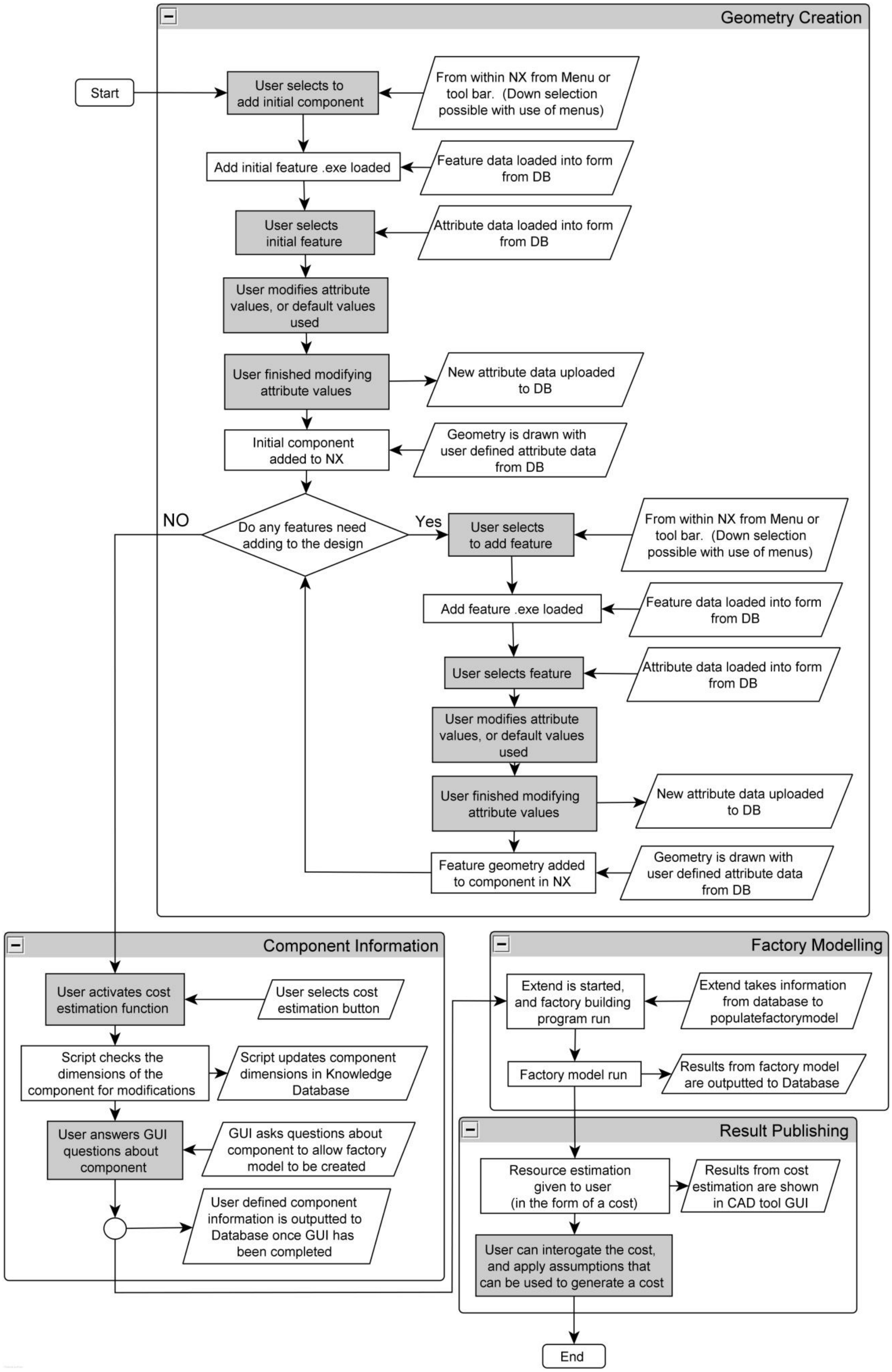

Figure 6: User flow chart of processes, inputs and outputs

The GUI asks the user questions about specific manufacturing processes depending on the geometry and previous user inputted data. Prebuilt specific manufacturing simulation models will be selected by the system, and populated with collected data. The case study will be able to simulate three manufacturing processes; machining from stock bar, machining of a ring rolled forging blank and machining of a powder Hot Iso-static Pressing (HIP) with near net shape capability. Each of these processes have a range of working, for instance powder HIP is limited by the vessel size, therefore not all the processes will be chosen for all possible designs of the component.

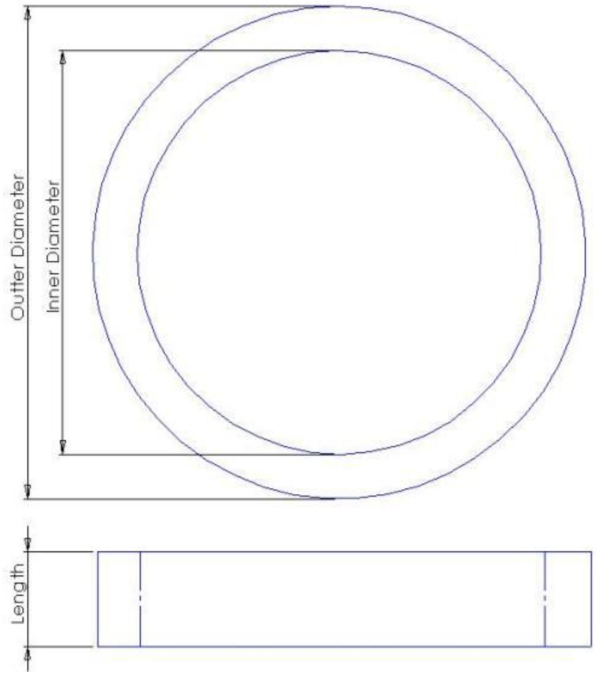

Figure 7: Cylinder case study component dimensions

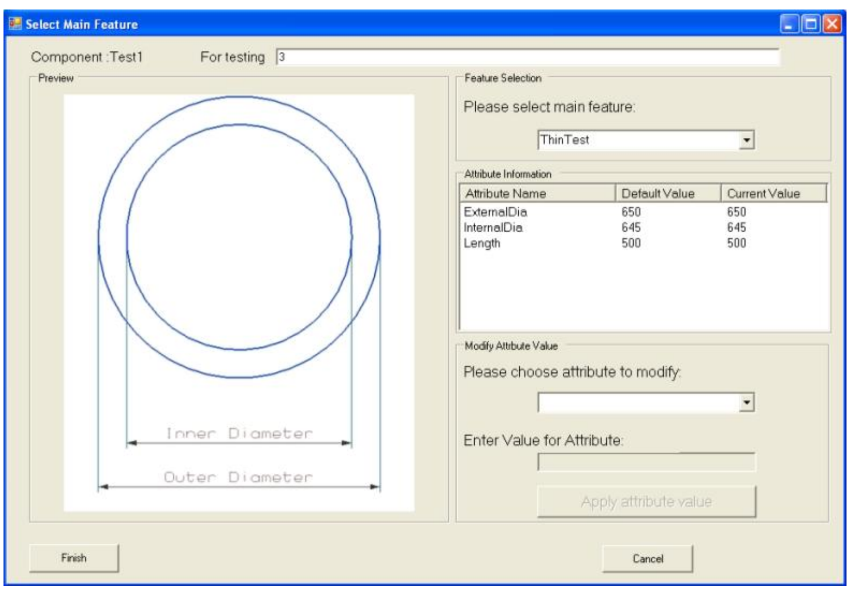

Figure 8: Add feature GUI with selected cylinder example

Summary and conclusions

A review of cost estimation methods shows that many of the methods consist of static models. Static models can be limited in their ability to properly represent dynamic systems. This limitation manifests its self in the form of gaps between manufacturing operations. It was therefore suggested that simulation should be incorporated into a costing system.

To aid in the incorporation of factory simulation into a costing system, it was decided to utilise CAPP methods. To overcome the unidirectional data flow limitation of CAPP systems, a direct coupling to a CAD tool is being implemented. This should allow data to feedback to the design environment, the CAD tool

This paper presents a framework specification that combines CAD and CAPP technologies with a knowledge database, to provide data for conducting manufacturing simulations, to improve cost estimation. By combining these technologies with a knowledge database and a GUI, feedback of cost estimation results can be given directly to a design team within their design environment. This feedback should aid a design team to understand the consequences of design decisions in terms of cost and manufacturing resources.

A system is being developed to implement the framework and a case study of a simple component (a cylinder) will be used to prove the system. A cylinder is being used because it is representative of a proposed full case study, a large civil aerospace case.

References

[1]. Lloyd M. Operational excellence. Rolls Royce Plc. 2006: 1-18.

[2]. Rolls-Royce Plc. The Jet Engine. London, 2005.

[3]. Sandberg M, Boart P, Larsson T. Functional Product Life-cycle Simulation Model for Cost Estimation in Conceptual Design of Jet Engine Components. *Journal of Concurrent engineering: research and application* 2005; **13** (4): 13.

[4]. Rush C, Roy R. Analysis of cost estimation processes used within a concurrent engineering environment throughout a product life cycle. *Seventh ISPE International Conference on Concurrent Engineering: Research and Application Conference, Location* 2000.

[5]. Curran R, Raghunathan S, Price M. Review of aerospace engineering cost modelling: The genetic causal approach. *Progress in Aerospace Sciences* 2004; **40** (8): 487-534.

[6]. Niazi A, Dia JS, Balabani S*, et al.* Product cost estimation: Technique classification and methodology review. *Journal of Manufacturing Science and Engineering* 2006; **128**: 563-575.

[7]. Tammineni SV. Designer driven cost modelling School of Engineering Sciences, University of Southampton. 2007; Doctor of Philosophy: 196.

[8]. Layer A, Brinke ET, Van Houten F*, et al.* Recent and future trends in cost estimation. *International Journal of Computer Integrated Manufacturing* 2002; **15**: 499-510.

[9]. Miles BL, Swift K. Design for Manufacture and Assembly. *Manufacturing Engineering* 1998: 221-224.

[10]. Scanlan J, Rao A, Bru C*, et al.* DATUM project: cost estimating environment for support of aerospace design decision making. *Journal of Aircraft* 2006; **43** (4): 1022-8.

[11]. Law AM, Kelton WD. Simulation Modeling and Analysis. Mc Graw Hill: 1992.

[12]. Kang M, Han J, Moon JG. An approach for interlinking design and process planning. *Journal of Materials Processing Technology* 2003; **139** (1-3): 589-595.

[13]. Marri HB, Gunasekaran A, Grieve RJ. Computer-aided process planning: A state of art. *The International Journal of Advanced Manufacturing Technology* 1998; **14** (4): 261-268.

[14]. Zhou X, Qiu Y, Hua G*, et al.* A feasible approach to the integration of CAD and CAPP. *Computer-Aided Design* 2007; **39** (4): 324-338.