
This version is available at https://strathprints.strath.ac.uk/62044/

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (https://strathprints.strath.ac.uk/) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to the Strathprints administrator: strathprints@strath.ac.uk

The Strathprints institutional repository (https://strathprints.strath.ac.uk) is a digital archive of University of Strathclyde research outputs. It has been developed to disseminate open access research outputs, expose data about those outputs, and enable the management and persistent access to Strathclyde's intellectual output.
Using Systems Engineering to develop a novel manufacturing cost model

N. Kuppusamy, R. I. Whitfield, A. Hird, S. Brown, C. McCubbin, P. McCubbin

Department of Design, Manufacture and Engineering Management, University of Strathclyde, 75 Montrose Street, Glasgow, G11XJ, UK.

Spirit AeroSystems (Europe) Ltd, Advanced Technology Centre, Building 38a, Glasgow Prestwick Airport, Prestwick, KA9 2RW.

Abstract. This paper describes the established process of cost model development and highlights opportunities for further improvement. Combinations of Systems Engineering tools were used within the modelling strategy in order to overcome the limitations of individual techniques. A functional modelling language was used to analyse, develop and integrate the cost model functions. The relationships between the manufacturing systems were modelled using Unified Modelling Language (UML). Furthermore, the benefits obtained by introducing changes to the cost model development process were evaluated by drawing comparisons between the two model development approaches.

Keywords. Cost Model, Estimation, Systems Engineering

1. Introduction

Organisations are continually trying to improve processes, increase productivity and save money. Often, the process of improvement involves generating cost forecasting models to study the contributing factors that influence high cost and plan for the future. Additionally, cost forecasting models are utilised to evaluate the impact and opportunity of emerging new technology. The development of cost model enables the business to gain competitiveness in emerging markets and tender submission for new industrialisation proposals to its customers. Essentially, industry requires the ability to win new business and retaining existing customer contracts.

The future of cost modelling in manufacturing is evolving rapidly in parallel with advancement in both software and hardware. However current concern in respect to many Commercial Of The Shelf (COTS) cost modelling tools rely heavily on populating large sets of historical data, with intelligent algorithms that evaluate data trends in order to generate cost estimation values. Many internal and external factors contribute to fluctuations in manufacturing costs and becomes complex to evaluate with historical data sets. Furthermore, data trends with little or no historical data will likely result in highly erroneous cost evaluation. To overcome this drawback, a bespoke costing tool will be developed for estimating costs and simulating real event variables (direct cost) developed with systems engineering principles.

A novel approach with Systems Engineering techniques applied with coding regimes efficiently delivers functionality and enables an accurate mapping of the user’s design requirements for development of a bespoke costing tool. A designed and developed

1 Corresponding Author, nithian.kuppusamy@strath.ac.uk
tool that adapts to the user’s environment is more preferable than a tool designed for the user’s that is required to adapt to the environment. This study will start by looking at the current procedures utilised in the development and functional use of COTS cost model. The paper will then be choosing a cost forecasting method.

2. Cost forecasting

Cost forecasting is a method for predicting future cost, based on past and present data. Generating a simple cost forecasting solution for a new project is highly complex and often errors increase with complexity, unless evidence exists that complexity contributes to accuracy [1].

Research in the field of forecasting evolved as a way of examining the predictability and identifying the most probable occurrences of any given trend. The developments in forecasting are to help decision making and planning in the present and enable the ability to modify variables know to alter (or be prepared for) the future. Extensive work has been conducted to categorise estimating methods, understand application methods and promote the application of forecasting methodologies by J.S. Armstrong [2].

Figure 1 illustrates a selection to support decision making and to improve model selection criteria and improve accuracy of forecasting. Often a combination of two methods is used to improve these forecasting of estimations.

Extrapolation is a method used to examine cycles and trends of historical data with applying mathematical solution to identify forecasts in the future. Many of the forecasting methods work under the assumption that the information responsible for creating the past, will continue to have similar trend pattern into the future. Often these conditions can be acceptable assumptions when forecasting for short term, but when evaluating medium and long term forecasting, the margin error increases. This is due to probability of changes in the trend, increasing with time and as a result variations occur that are different to a historical pattern/trend. Therefore, the greater the time to forecast, the smaller the accuracy becomes.

Single or multiple regressions analysis is a common mathematical technique used to model complex systems which involves using relationships between two or more variables and has become a primary focus for many forecasting solutions at present. When compared to trend extrapolation models, which uses only the historical variable to be forecasted, single or multiple regression models uses the relationship between the variables to be forecasted and with two or more different variables. The aim of regression analysis is to be able to understand, how a group of variables interact with each other and affect other variables. There are many other forecasting models based on trends and cycles. When identifying an appropriate model for a particular forecasting application, it depends on the historical data and most often involves using various forms of weighted smoothing function to improve the accuracy of forecasting. The similarities between mathematical models are the historical data, which is the essential element for generating a forecast. However two models that use the same historical data are not the same. The mathematical model varies from modeller to modeller and the applied smoothing coefficients, constants and other parameters will be different. This results in two different forecasts for the same models. Therefore, the selection of such key parameters, constants or coefficients determines the outcome of any forecast. Although, the use of regression analysis is a simple solution for
estimating, a change in the level of complexity introduces greater levels of uncertainty to any of the values obtained.

Figure 1. Illustration of different forecasting models strategy adapted from J.S. Armstrong’s [2].

When the level of complexity increases, any individual model alone is not adequate in providing accurate forecasts and where possible, a combination forecasting can be used to improve the accuracy. Evidence indicates that by combining individual forecasts, improves accuracy and by adding qualitative forecasts to quantitative forecasts reduces accuracy [1]. It is unclear which condition or complementary method yields the optimal combination for accurate forecasting and further research is required to identify a method. Research conducted by Armstrong et al. [2] indicates that by applying their conservative check list approach when selecting, can improve the chances of increasing the accuracy in selection of an optimum mathematical model [3, 4]. It is important to note that a trend based forecast can be not one hundred percent accurate and still be useful.

Research conducted by Hird et al. highlight existing methods that provide solution for application in New Product Development (NPD), that is available for Software Development (SD), but the NPD lacks historical data and presents potential challenges [5]. Although, this could be mitigated with a little information to achieve a working
forecasting model, the forecasting model will be largely prone errors and as a result provide inadequate results. Often forecasting models are not able to translate small changes that could have a high impact on associated forecasts [5]. It is apparent that the choice of forecasting methods will not always yield accurate estimations every time, hence it is better to develop an alternate solution.

Manufacturing industries continually seek to reduce cost and often utilise trend based estimation for forecasting. It is more accurate to use direct cost forecasting strategies, considering any small changes that affect the overall cost. Therefore, it is better to avoid estimated trend forecast cost model and develop or employ a direct cost forecasting model.

Developing and employing bespoke forecasting model in a large complex business application with many users, will require efficient strategy in planning and clear, concise communication with the team members during the progress of the project.

Developing a visual representation of the problem, user interaction, mapping processes and structure will enable efficient capturing of user requirements; reduce ambiguity and maintain clear definition of the model. Applying Systems Engineering principles during the development stages for both core and interface code is highly valuable.

3. Systems Engineering Technique

Systems Engineering is a method for generating and executing multidisciplinary process that enables requirements and developments to be captured qualitatively throughout the model’s life cycle. The method as shown in figure 2 consists of seven steps: Stating the problem, Investigating alternatives, Modelling the system, Integrating the model, Launch the system, Assess performance, and Re-evaluate. Furthermore, the Systems Engineering method is not performed sequential and instead the steps are performed iteratively in parallel.

![The Systems Engineering Process](image)


Applying Systems Engineering principle to coding regime requires the employment of mapping of the whole problem for understanding before developing a solution, translating measurable requirements, examining all feasible alternatives before developing the ideal solution, considering the entire system life cycle from start to the end of concept extending to maintenance, replacement and decommission and
testing individual components functionality. It is essential to employ several mapping techniques that will provide support for conceptual software development.

Many mapping techniques exist that can be applied in parallel with Systems Engineering principles for model development. Identifying and selecting an ideal combination of mapping techniques for the developing models is complex. A select number of key mapping techniques are applied for the development of the cost model. This paper proposes an approach to generate clear, efficient and compact model mapping for development combined with Systems Engineering principles. Dependency diagram was created to model the complex relationships between different sections and sub-sections of the model. The dependency diagram identifies constituent parts that depends on other part, and/or have parts that depend on them. N-squared ($N^2$) diagram used for identify, define, design, constrain the physical and functional interfaces between the model elements. The $N^2$ diagram is valuable for identifying the confluence of models interaction between systems and categorising functional systems. IDEF0 is a technique designed to model the function of the system, such as inputs, controls, outputs and mechanisms. IDEF0 was applied to develop, analyse and communicate the function and scope of the model with team members. The communication enhances consensus of decision making and deeper involvement of expert users within the team.

Visualising the structure and behaviour of the model in one large detailed map containing all the relationships is very difficult. Alternatively breaking down the system within the model into several diagrams with each one focussing on the individual subsections can provide detailed structural or behavioural diagrams of the relationships. Representing each function of the model as an object is essential in aiding code development. The structural and behavioural representation of the model functions as objects or methods is known as Unified Modelling Language (UML). UML is a comprehensive software modelling language used for developing code and is commonly used in both structural and behavioural mapping for code development. There are many different structural and behavioural diagrams to describe static structures and dynamic behaviours.

![Figure 3. The novel System Engineering approach developed for complex systems and models.](image-url)
The structural diagrams shown in figure 3 depict static structure and substructure of the model in different levels of implementation and how they are related to each other. The components in the dependency and IDEF0 diagram represent the overall model concept, and include a breakdown of the model subsections. Class diagram are static structural diagram depicting the relationships between classes, objects, attributes, and operations. A structural diagram does not represent model concepts with respect to time. Although the structural diagram is a valuable diagram, it is also important to visualise dynamic behaviour of the concept model for ensuring a functional analysis of relationships between components in the model. Use Case is a valuable dynamic or behaviour diagram that is used for model development. The Use Case diagram depicts the model’s functionality of the system using actors and Use Cases defining interaction of internal and external influences between roles to achieve the concept model goals.

Applying Systems Engineering principles to the mapping techniques enables accurate capturing of user requirements, consensus of model functionality and reduces time during development. It is difficult for many developers to describe and define the entire system or model effectively and often developers describe the system in one perspective only either structurally or behaviourally. In general it is difficult for most developers to completely describe, define and achieve full consensus with end users using traditional strategies. However, this can be achieved by using the novel process described in this paper, which implements systems engineering principles to key select mapping techniques sequentially.

4. Conclusion

Costing exercises are critical in the manufacturing industry, often required to be accurate and within a short “window” of time. Greater attention is given to accurately estimating and the solution to estimate direct cost. As a result developing a direct costing model requires detailed description of the complex system. The novel approach developed using systems engineering principles enables efficient use of mapping techniques for creating a highly complex bespoke cost model. The novel approach described in this paper will enable other developers to describe and define the functional model or system efficiently and simplify translation of methods or objects required for constructing the model source code.

References