Provision of a Web Based Decision Support system for Wing Box Tooling

P Hale[1], J Scanlan[2], T Hill[3], M Nour[4], C Bru[5], A Jocelyn[6], M Round[7], M Dunkley[8]

[1-6] *University of the West of England, Faculty of Engineering, Frenchay Campus, Coldharbour Lane Bristol BS16 1QY*

[7-8] *Airbus UK, Filton, Bristol*

Abstract

The ACES (Aerospace Cost Engineering Systems) team at UWE has been engaged in developing cost models to the aerospace industry for a number of years. These models range from large spreadsheet based tools to knowledge based systems.

The work described in this paper attempts to address some of the issues that have emerged in creating costing tools in the past, namely: -

1) Limited reusability of code/data
2) Lack of sophisticated data structures in spreadsheets
3) Provision of wide access to cost data/knowledge
4) Integration of a wide variety of code/data sources.

The new system is intended for network use. The project uses the open architecture of the Internet to create a system of interacting objects. These objects communicate with systems such as CAD and Databases. This openness ensures that the software can be improved in an evolutionary way.

This paper describes the status of the project.

Keywords

Aerospace, Tooling, Design, Manufacture, Decision Support, Knowledge Base, XML (Extensible Markup Language)

1 Introduction

Work completed recently has involved the use of Visual Basic and development of a spreadsheet. [Buehlmann, Ragsdale, Gfeller 2000] also used this approach. A problem with our spreadsheet approach is that it encouraged the creation of islands of information. An individual engineer could customise the spreadsheet and change the information. Development of isolated software makes communication between engineers difficult. This implies a need for network based software which uses open communication standards and a centralised modular approach to development and maintenance. The Coiner system [Marsh 2000] is more sophisticated and models materials, parts, and processes in a more structured way than the spreadsheet but it is difficult to integrate this software into a network.

Our approach is to create new software and integrate this with existing software by means of middleware. Middleware such as Common Object Request Broker Architecture (CORBA) can be used to integrate Enterprise Resource Planning (ERP) Systems and enables communication independent of the hardware or software platform. It is important that information created using one system is able to communicate its data to another system. If design information is exported from a CAD system into a costing program the results can then be shown in a spreadsheet. The designer and cost engineers could view these results.

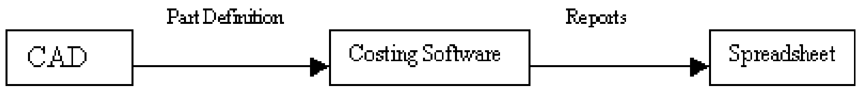

Figure 1 Software information flow

This implies a need for feature recognition in order to create software classes to represent design features.

The long-term goal would be to automate the steps from design to manufacture. This implies the use of intelligent automated reasoning at all stages of the value chain not just an automated CAD/CAM link. This is a difficult task, [Duverlie, Castelain 1999], [Feng, Kusiak, Huang 1996], and [Qu-Yang, Lin 1997] explain the theory behind this. It is important to have an intermediate target attainable over a shorter term. This is to automate some decision making, but this decision making would be driven by the software user. The user can then make some high level decisions which allow the software to perform relevant cost calculations and make more detailed decisions. The system can then be used at the start of a project, late on in development, and after production has begun (to evaluate possible design changes). Some of the software might be re-applied for other product types. Accurate cost estimation requires a high level of part definition but the best opportunities for decisions, which reduce product cost, are early in the product life cycle when part definition is still ill defined. So the level of detail which users can provide varies. The system must be able to provide an estimate even if a user provides only high level information, and report on what further information is required.

2 Approach used

A software system using open standards is flexible to changes in requirements and technology because it is possible to use whatever software is most appropriate to solve a particular problem. Thus it is possible to create an integrated system. This could facilitate design and manufacture using CAD/CAM, cataloguing of parts in a library using a Database and Structured Query Language, costing using spreadsheet 'what-if' analysis and modelling, capacity planning and scheduling applications. This can be achieved by linking diverse software together.

This approach is appropriate to concurrent engineering because definition of parts, processes and tooling in software is difficult. This means the software has to be well structured, and able to deal with complexity. Also a network-centric approach is important to prevent a proliferation of single computer software.

The current project involves research into how to apply the work of Tim Berners-Lee and others in the World Wide Web Consortium (W3C) [W3C 2001], [Berners-Lee 1999]. In order to define a part, material or a process it is necessary to use Meta-data. This takes the form of Meta–tags to define words as having a particular meaning. A Meta-data language XML (eXtensible Markup Language) is used for structuring information. This is a data definition language and W3C ensures that the language is standardised so different software systems can interpret it. This language can be used to structure web pages so that they can represent and link to programming objects. Knowledge sharing within an Internet expert system and between it and other systems can be facilitated using XML [Dennis 2001]. This Meta language is useful because it fits well with both an Object Oriented and a Rule-Based approach to problem solving and creating software. Using Meta-tags defined with XML it is possible to create documents which define their own structure. Using scripting languages the information is then passed to a program which can take the value and process it or pass it to other objects within a system to facilitate decision support. XML also helps to provide common definitions to make it easier to use computer middleware such as CORBA and DCOM (Distributed Component Object Model). The XML code is linked to XSL (eXtensible Stylesheet Language) code to define appropriate formatting and so output the information in a standardised and comprehensive way. This makes it possible to provide a consistent and understandable user interface.

A system like this must hide its complexity behind an easy to use front-end in order to ensure its usability. This does not necessarily imply that the actual data or algorithms are simplified but that the data is presented only when necessary and at the appropriate level of detail. It is also important to ensure that a user's knowledge is entered into the system in the best way possible. Each user's knowledge will be different. Thus it is necessary to adapt the user interface to accept different information from each type of user but in a standardised interface. The system can then output the results calculated with the input from that user type. This means there is a need for polymorphic software that selects the appropriate calculation method given the information received. The programming language objects can find the relevant XML tag within a web page and check whether the user has provided the information related to this tag (otherwise it would need to use a default value). The object can also check whether the user could subsequently improve the accuracy of the estimate by adding further information. Intelligent agent objects can be used to achieve this. An intelligent agent is an object that can react and adapt autonomously to changes [Grove 2000], [Sycara 1998].

3 Development and implementation

The software is intended to be both an aerospace tooling knowledge base and a costing/decision support system. The knowledge base has a dual purpose of knowledge representation and providing the inputs to and outputs from the costing software. It provides an easy to use interface and modular structure for creation of new pages as required. It is possible to provide standard templates for web page creation, which allow pages to be added by those who are not web designers.

The tooling is used to construct wing box components so the software needs to represent these components. In order to define the components and the tooling, information is required on the materials from which the parts are constructed, and also the manufacturing processes for the tooling. This made it necessary to construct three class hierarchies to represent parts, materials, and processes. Objects of these classes contain methods, which can perform calculations and pass messages to related objects. For example a part object could pass a message to a material object to request a material name. Additionally a flow relationship is required for the process objects, as manufacturing processes must be carried out in a particular order.

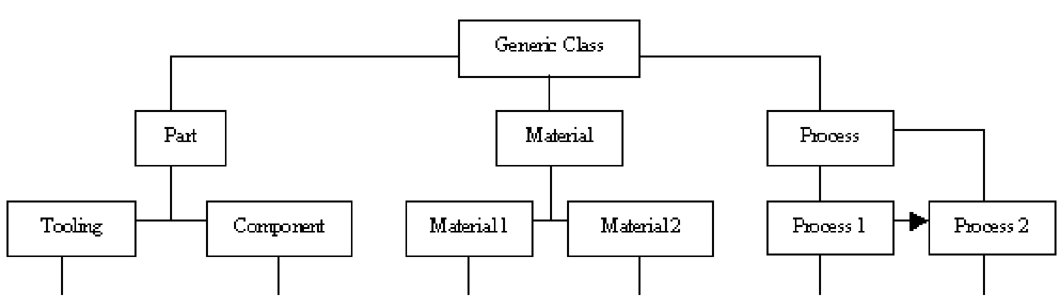

Figure 2 Class Structure

It is intended that the system should accept inputs from and send outputs to a variety of other programs and systems. The architecture shown in Figure 3 enables this and is explained in more detail later.

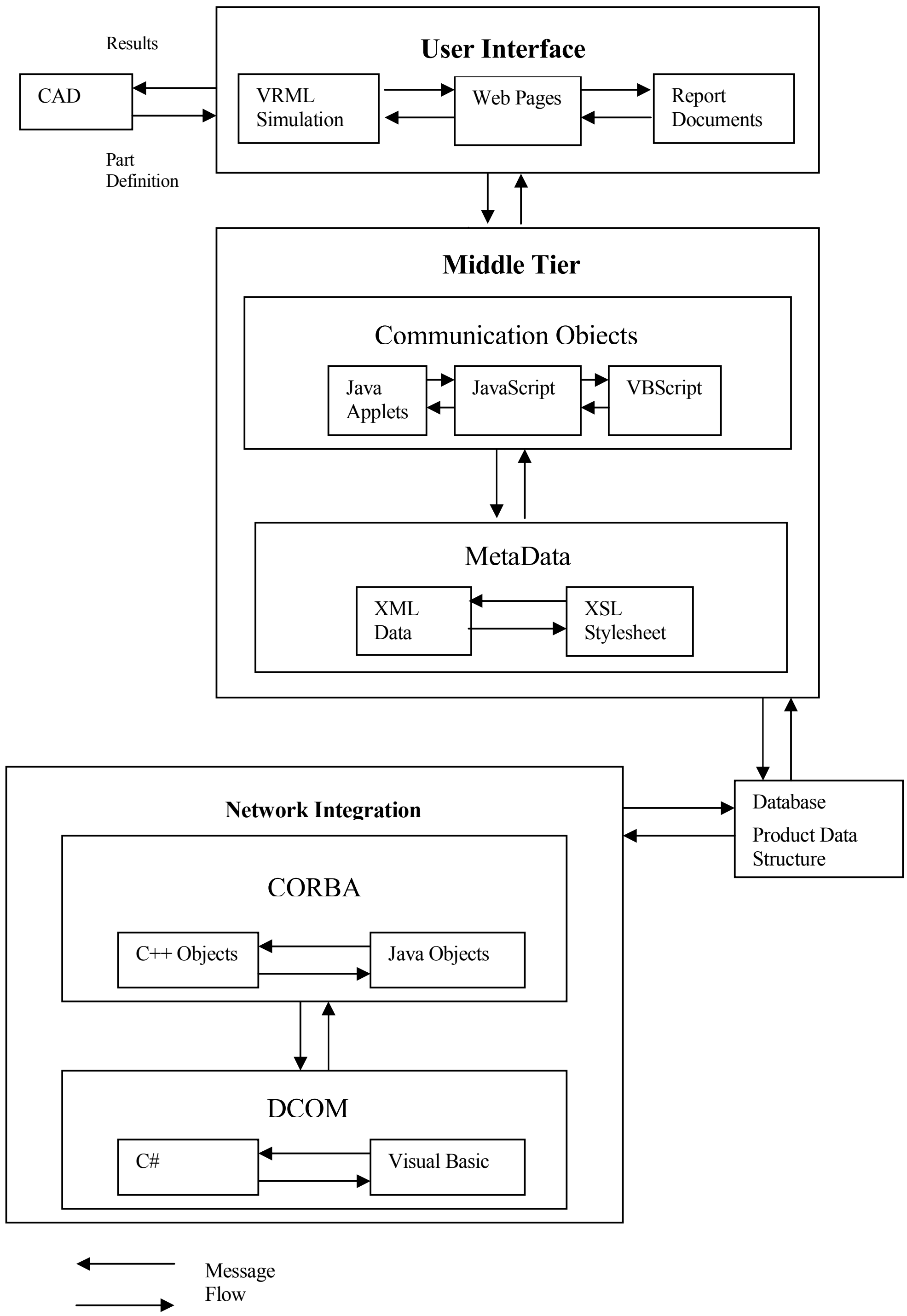

Figure 3 Software architecture

3.1 User Interface

It is possible for a user to access any web pages using the main menu, defined with XML based on Channel Definition format. Figure 4 shows the menu after the user has accessed a web page in the Component channel. A file metaphor is used for a sub menu within the Materials channel (Figure 7).

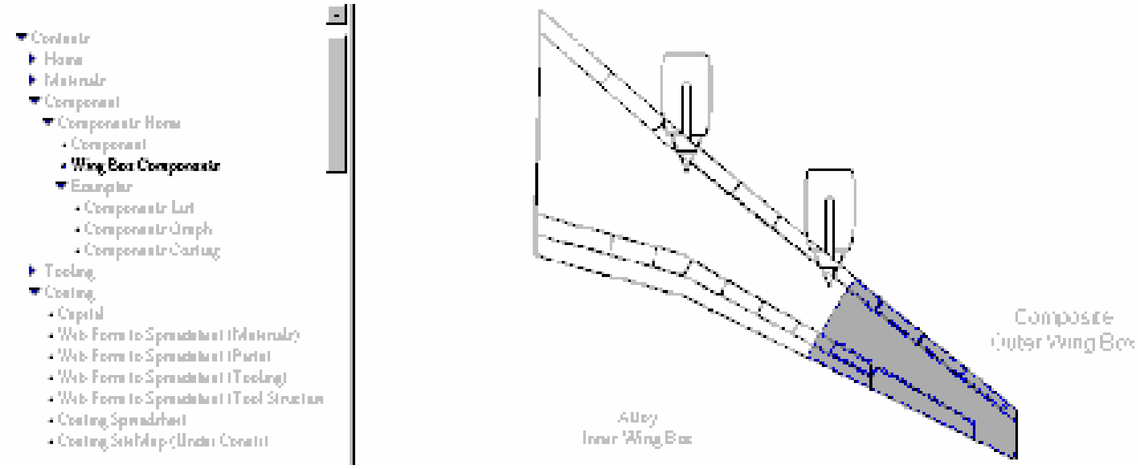

Figure 3.1

Figure 4 Contents Menu

Another type of site menu is an image map and this is more suitable for component costing. A user can click on a particular sub-component of the main component and see the costing information related to that sub-component. This facilitates drill down to specific data from generic data. A similar concept is used for drilling down from graphs to access detailed information on how figures are calculated. A search menu has also been implemented using an applet provided by Sun, which allows a user to find information by means of a word-based search or by pressing a labelled button (Figure 5). The applet shows an abstract of the web page contents, and links to the full page.

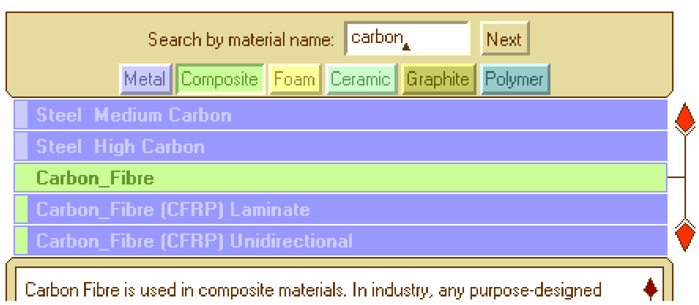

Figure 5 Search/Glossary Menu

The User Interface has been created using the sophisticated facilities now available for web page design. The web pages are a means to send information to and receive information from the middle tier (Figure 3).

Web pages can be converted to different formats such as Word Processed documents or Spreadsheets. Also CAD data can be represented using VRML (Virtual Reality Modelling language). VRML enables CAD diagrams to be represented as 3D objects, which can be manipulated using the interactive facilities of a VRML player. A significant advantage of VRML simulations over other media is that they contain a subset of CAD information that can be communicated into the system. Figure 6 shows a VRML representation of a CAD model [Bristol University - McMahon, Croxson 2001]. [Kim, Choi, Bong Yoo 2001] explain how VRML can be used for collaborative viewing and amending of mechanical parts. [Nidamarthi, Allen, Shiram 2001] made use of VRML to allow scientists and engineers to collaborate on the design of Nanotechnology parts.

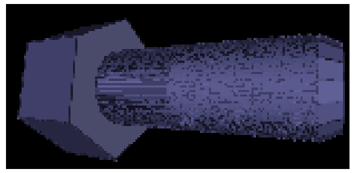

Figure 6 VRML from CAD, University of Bristol

2D CAD data and diagrams can also be represented on web pages using another language VML (Vector Markup Language), an XML Language. Figure 3 was created in a word processor and translated into VML to display on the Intranet. Again it is possible to communicate this structured data to the system. VRML and VML diagrams load more quickly into a Web page than pictures because they are defined using text.

3.2 Middle Tier

The middle tier is a communication layer, which enables translation between human readable documents and a software readable object hierarchy, each object contains attribute values. There are three possible ways for a user to enter data into this middle tier.

- Enter values in the web pages
- Create a CAD model with appropriate properties
- Enter data into database tables

This makes it possible to integrate the system with other software.

The middle tier consists of data defined using XML. This data is sent for output to an HTML (HyperText Markup Language) page via an XSL stylesheet. Communicating objects take XML data and send it to the Database. The information taken from the database can be arranged into a hierarchical structure, a flow structure or a combination structure as necessary. Figure 7 created with a Java applet shows the hierarchical structure imposed on data retrieved from a relational database Materials table. Each folder represents a class and each file represents an instance of a class. This is also used as a navigation map to allow any user who is familiar with navigation of a visual file system to use this metaphor as a way of navigating through the materials web pages.

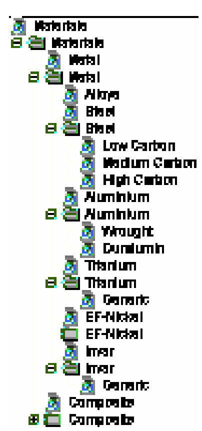

Figure 7 Materials Object Hierarchy and site map

Two common techniques are available for middle tier communication. The first is the use of ActiveX controls combined with JavaScript or VBScript. The second is the use of Java Applets.

3.3 Network Integration

In past and current projects, software objects have been written in C++, Java, and Visual Basic. Because of communication incompatibilities between different languages and platforms it was essential to devise a network integration strategy early in this project to ensure these objects can communicate. This involves the use of middleware. Middleware is an all-embracing term meaning software to bind together applications written in different languages or installed on different hardware. There are two competing software integration techniques under development by major companies.

The first technique is Microsoft's .Net strategy. Their strategy is centred on C# (pronounced C Sharp) and other Microsoft languages, it will be possible to convert Java programs to C#. It also involves their DCOM framework. DCOM facilitates communication between software and languages running on Microsoft Windows. When ActiveX controls provide the communication from the middle tier, DCOM is the most useful middleware. ActiveX controls support several computer languages but are currently used for PCs only. This makes it possible to link PC software together. The second technique is Sun's Service-driven Network strategy. Their strategy is centred on Java. CORBA can be used to link PC and Unix computers, and can be used for communication with several languages. When applets are used to communicate from the middle tier with Java, CORBA is the most useful middleware. Applets are computer architecture independent and communicate with Java programs. CORBA makes it possible to link PC software to Unix CAD/CAM software and Fortran programs (many Fortran engineering programs exist).

[IONA 2001] have developed a CORBA-DCOM bridge. It is possible for objects integrated using DCOM to communicate with objects integrated using CORBA. [Huang, Mak 2001] explains the issues behind software for web applications relating to product design and manufacture, giving an overview of the most useful Internet related languages and techniques.

4 Summary

An Intranet based aerospace tooling decision support system has been developed. This uses Object Oriented techniques and languages to provide an infrastructure. This enables the flexibility essential for future development, and makes it possible to link the system to other software. The structure can be updated with new information or can be used as a framework for developing further software systems. A communication strategy was used to allow the Intranet system to link to other systems created using different software or hardware.

The system provides a knowledge base for aerospace design and manufacturing. It achieves this by means of a library of Parts, Materials and Processes. This library is also implemented as software objects. Information can be entered via web pages, CAD diagrams, or database tables and is communicated to interacting software objects. Results can be output to, database tables, web pages, or office documents.

Acknowledgement

This work has been partly funded by Airbus UK through the European Commission TANGO project. The authors wish to thank Airbus UK and the European Commission for their support.

References

Berners-Lee Tim, (1999), "*The future of the Web*", http://www.w3.org/Talks/1999/0414-LCS35-tbl/slide1-1.html

Buehlmann Urs, Ragsdale Cliff T, Gfeller B, (2000), "*A spreadsheet-based decision support system for wood panel manufacturing*", Decision Support System, *Vol 29*

Duverlie P, Castelain J M, (1999), "*Cost Estimation During Design Step: Parametric Method versus Case Based Reasoning Method*", The International Journal of Advanced Manufacturing Technology, *Vol 15: pp 895-906*

Dennis G B, (2001), "*Koala Electronic Publishing*", http://www.koalapub.co.uk/

Feng C, Kusiak A, Huang C, (1996), "*Cost evaluation in design form features*", Vol 28 Computer Aided Design *No 11, pp 879-885*

Grove R, 2000, "*Internet-based expert systems",* Expert Systems, *Vol. 17 No. 3*

Huang G Q, Mak K L, 2001*, "Issues in the development and implementation of web applications for product design and manufacture*", Computer Integrated Manufacturing, *Vol 14 No 1. Pp 125-135*

IONA, 2001, "*Deploying an OrbixCOMet Application*", http://www.iona.com/docs/manuals/orbix/33/html/orbixcomet33_pguide/deploy.html

Kim Y, Choi Y, Bong Yoo S, 2001, "*Brokering and 3D collaborative viewing of mechanical part models on the Web*", Computer Integrated Manufacturing, *Vol 14 No 1. pp 28-41*

Kusiak, Andrew, 1993, "*Concurrent Engineering Automation, Tools, and Techniques*", Mc Gill, New York

Marsh R, 2000, *ACES KappaPC Costing Application Volume 1*, Report and User Manual

McMahon C, Croxson K , 2001, *"VREL Virtual Reality Engineering Library Home Page",* http://www.dig.bris.ac.uk/virtual_lib/

Nidamarthi S, Allen R H, Ram D S, 2001, "*Observations from supplementing the traditional design process via Internet-based collaboration tools",* Computer Integrated Manufacturing, *Vol 14 No 1. pp 95-107*

Qu-Yang C, Lin T S, 1997*, "Developing an Integrated Framework for Feature-Based Early Manufacturing Cost Estimation"*, The International Journal of Advanced Manufacturing Technology, Vol 13 *pp 618-629*

Sycara K. P, 1998, "*The many faces of agents",* AI Magazine, Journal Paper

Weber, Frithjof, 1998, "*CE Projects Information*", CE-NET, Concurrent Engineering Network of Excellence, *ESPRIT project 25946, Deliverable D11a*

World Wide Web Consortium (W3C), "*Leading the Web to it's Full Potential*", 2001, http://www.w3.org/