Strathprints Institutional Repository

Whitfield, R.I. and Coates, G. and Duffy, A.H.B. and Hills, W. (2001) *A system for co-ordinating concurrent engineering.* In: 13th International Conference on Engineering Design (ICED 01), 2001-08-21 - 2001-08-23, Glasgow.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21–23, 2001

A SYSTEM FOR CO-ORDINATING CONCURRENT ENGINEERING

Robert Ian Whitfield, Graham Coates, Alex H.B. Duffy and Bill Hills

Keywords: Tactical design co-ordination, distributed design, concurrent engineering.

1 Introduction

Design of large made-to-order products invariably involves design activities which are increasingly being distributed globally in order to reduce costs, gain competitive advantage and utilise external expertise and resources. Designers specialise within their domain producing solutions to design problems using the tools and techniques with which they are familiar. They possess a relatively local perception of where their expertise and actions are consumed within the design process. This is further compounded when design activities are geographically distributed, resulting with the increased disassociation between an individual designer's activities and the overall design process. The tools and techniques used by designers rarely facilitate concurrency, producing solutions within a particular discipline without using or sharing information from other disciplines, and seldom considering stages within the product's life-cycle other than conceptual, embodiment or detail [1, 2]. Conventional management and maintenance of consistency throughout the product model can subsequently become difficult to achieve since there are many factors that need to be simultaneously considered whilst making a change to the product model.

Concurrent Engineering (CE) is often regarded as a means of significantly reducing design time to enhance competitive advantage by maximising the amount of parallel design activities. Winner et al. [3] described concurrent engineering as:

"a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life-cycle from conception through disposal, including quality, cost, schedule, and user requirements."

This definition is representative of many of those encountered in that the main emphasis is placed on its systematic approach to the design process as it reduces the design time and hence total design cost, which is one of the main considerations in the entire design process. CE however cannot be fully realised without co-ordination [4] and given a co-ordinated approach without de-coupling dependencies, concurrency will only be achieved where the design process permits.

The DMS is intended to tackle some of the issues arising from the above statements by providing a formalism that would enable the design process to be co-ordinated in a tactical manner. This involves determining the actions required to be undertaken to perform a specific task for the right reasons, to meet the right requirements and to give the right results [5].

Design co-ordination is a relatively new concept within the engineering design community and is aimed at improving the performance of the engineering design process. One of the most prominent frameworks associated within design co-ordination is the Design Co-ordination Framework (DCF) [5]. The DCF presents a number of frames, each of which is aimed at representing different aspects of design co-ordination. The DCF also describes the management of the links between the frames. The research presented in this paper identified that tactical co-ordination may be represented by: the goal/result model; the discipline/technology model, and, the task model frames within the DCF. These models are described within section 3. The focus of this tactical design co-ordination system is the management of design tasks, information, goals and rationale within the product development process such that the process may be performed in a timely and appropriate manner. The Design Management System (DMS) was used to manage and co-ordinate these tasks, and evaluate the constraints, maximising concurrent design activity, whilst maintaining inter-task dependencies.

Close contacts with industry were maintained during the research programme which focused on the development of the DMS system towards tackling design issues that were considered pertinent to the companies such as reducing design time, ensuring informational consistency, and facilitating, not automating, the decision making process. An application of the DMS within an industrial case study is demonstrated within section 4. The case study represents a number of inter-dependent tasks that need to be performed and constraints that need to be satisfied for the design of a stage of blades within a steam turbine. Conclusions and recommendations are made within section 5.

2 DMS architecture

The DMS was constructed to co-ordinate the design activity of an agent-oriented design process with respect to managing design tasks, information, goals and rationale, and facilitating the decision making process. A basic representation of a software-based design agent was produced to: manage the functionality of existing design tools; enable communication with the DMS and other design agents, and, demonstrate how the distributed design problem may be managed and co-ordinated. The focus of this work was, however, not towards the development of autonomous agents. Malone et al. [6] proposed two design principles for the design of agent-based systems which were used during the development of both the DMS and the associated design agents:

“Don’t build computational agents that try to solve complex problems all by themselves. Instead, build systems where the boundary between what the agents do and what the humans do is a flexible one.”

“Don’t build agents that try to figure out for themselves things that humans could easily tell them. Instead, try to build systems that make it as easy as possible for humans to see and modify the same information and reasoning processes that the agents are using.”

2.1 The discipline/technology model

The discipline/technology model represents the disciplines which are present within the product as a whole and also within its subsystems [5]. The model also indicates which technologies are required for the design development, as well as displaying how the disciplines and technologies are distributed over the artefact.

Within the context of an agent-oriented distributed design architecture, the discipline/technology model represents the structure of the agents and the technologies that they utilise. That is, the agents and associated design tools often reflect the discipline knowledge and technologies within the domain. The model also contains information pertaining to the tasks that the design agents are capable of undertaking. Through the production of a design process displaying the inter-dependencies between the tasks, it is possible to display the relationships between the design agents as well as the technologies.

The design agents register their information with the DMS to enable communication, as well as providing information regarding the tasks that the agents are capable of undertaking. Currently this process is confined to software agents, where the tasks that may be performed are defined by the functions that the associated design software may achieve. However, developments are underway to define and model tasks and their associated disciplines and technologies within the DMS to enable the management and co-ordination of human agents. The DMS may manage human agent task enactment by sending an email to an engineer containing a description of the task that needs to be performed as well as the information required to perform the task and the information that the task will produce. When the task is completed the engineer would submit the produced information back to the DMS.

2.2 The design process builder and task model

A graphical user interface was developed which would enable the designer to define the design process at any level of abstraction using information obtained from the discipline/ technology model. A number of different events were defined such as: perform task; branching operations (to facilitate concurrency); process pause; and, decision events, which could be used to define the process as well as co-ordinate the activities of the design agents.

The Design Structure Matrix has been adopted as a mechanism for managing the tasks and inter-task relationships within the design process [7, 8]. This concept has been extended such that iteration and decision based branching may be represented within the design process with each process having its own matrix representing the tasks that need to be undertaken to achieve a particular goal. Providing individual representations in this way has enabled the overall goals of a process to be identified based upon the tasks contained within it, as well as enabling the DMS to manage more than one design process simultaneously. Allowing simultaneous management of processes further increases the functionality of the DMS since different aspects of the same design problem may be modelled and evaluated concurrently, for example the determination of the performance of a design artefact within different life phases. The DMS not only manages the dependencies within each individual process model, but also manages the dependencies between process models that are required to co-ordinate concurrent design activity.

The new Design Structure Matrices also include decision based dependencies to provide an exit condition from an iterative loop. A number of partitioning and concurrency based criteria were included to evaluate the performance of the design process.

2.3 Process information model

The DMS contains a formalism for the management of design information based upon an object-oriented approach. Currently, four different types of information have been developed based upon this formalism: Parameter, Option, File and File Group. The formalism requires all design information used within the DMS to inherit from a base class which provides mechanisms for comparing different pieces of information, serialising information such that it may be communicated across a network or stored persistently on a disk, and enabling information to be displayed graphically within the DMS. Extending the base class enables new types of information to be developed to suit the intended application. For example, if the design problem is to develop an internal combustion engine, the base class could be extended to produce a new information type which provides all of the information required to represent a piston. In addition, this information may be displayed three dimensionally within the DMS, enabling the designer to visualise the information, as well as modify it. Knowledge may also be included within the new information type to constrain how the information may be changed.

Each design process has an associated information model, containing information that has been either provided or produced, which is stored centrally within the DMS. When a new piece of information has been generated as a result of performing a task it is placed into the produced area of the information repository, replacing existing or outdated information as necessary. The DMS allows concurrent access to the information model by multiple agents.

2.4 The goal/result model

The DMS does not attempt to take the decision making process away from the designer, instead, it facilitates the decision making by providing the designer with information which would enable decisions to be made more appropriately.

Using parametric information made available from within the process information model, it is possible to construct and evaluate mathematical expressions which represent certain requirements or constraint-based goals of the design process, such as, for example, checking that the calculated efficiency is greater than a minimum efficiency. The design goals will be evaluated automatically when the design process has been enacted and the design solution for a particular design concept produced. If a goal has not been satisfied, then the user is notified of the problem, such that decisions may be made with respect to the design concept in order to satisfy the goal.

The DMS currently lacks optimisation-based goals, i.e. there are no mechanisms for expressing the goal, “maximise efficiency”. Current developments within the DMS include a suite of optimisation software for both the optimisation of the design process and the design product. The optimisation software includes: a robust design tool which may be used to either increase the robustness of a product, or give a good “ballpark” design which is near the optimum, a multi-criteria genetic algorithm which either may be applied to optimising the design process or product, and a multi-criteria simulated annealing algorithm which may also be applied to optimising the design process. Both the multi-criteria genetic algorithm and simulated annealing algorithm have been implemented within the design structure matrix and have successfully demonstrated that the design process may be improved with respect to both concurrency and partitioning.

3 Implementation and case study

The industrial case study used to validate the DMS software consisted of the co-ordination of a number of disparate design agents that were used to evaluate particular aspects of the performance of a stage within a turbo-generator.

The associated tasks are represented within the design process seen in Figure 1, which describes the activity that needs to be performed to determine aerodynamic, stress, and vibration characteristics of the turbine, as well as providing information regarding the dependencies between the tasks. The design process area also contains a list of the information that is provided as input to the design process which describes the design concept to be evaluated, a list of output information which describes the corresponding design solution, as well as a list of design goals.

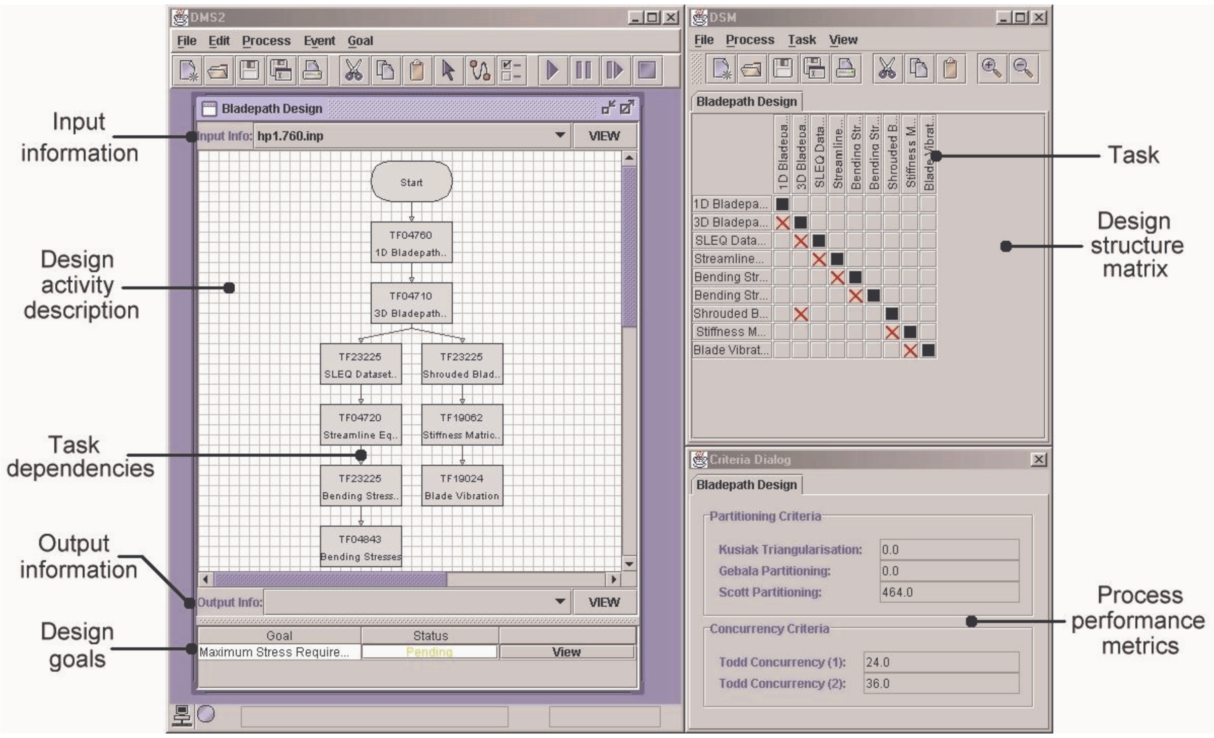

Figure 1. DMS Showing Bladepath Design Process.

A Design Structure Matrix is also used to represent the tasks and dependencies of the design process and provide evaluations of design process based performance criteria. A number of optimisation algorithms may be used to re-sequence the tasks within the design structure matrix with the objectives of improving the process performance with respect to the partitioning and concurrency criteria. Different weightings may be applied to the dependencies to represent strong and weak relationships. In this particular example however, all of the dependencies are defined as being strong since it is not possible to start a tool prior to the completion of a preceding tool.

3.1 Previous implementation

A typical evaluation of a design concept started with the production of a preliminary concept which would provide the necessary information for the process. The concept would then be evaluated through the enactment of the design process shown within Figure 1 requiring the manual execution of each of the tools in the order depicted by the design structure matrix. The flow of execution through the process is from top to bottom, although it may be modelled in any direction within the DMS. It can be seen from Figure 1 that certain tasks may be undertaken concurrently, however the process was generally managed by a single designer, and as such proved to be difficult to manage more than one tool simultaneously. Upon completion of each task, the information generated required moving to the correct location for the following tasks to maintain informational dependencies. This process was undertaken manually, and found to take approximately 11 minutes. The design solution, spread across a large number of files and a number of different disciplines, would then be examined to extract certain information which would be used within additional manual calculations to determine whether the design concept satisfied the design constraints. Rules would then be applied based upon the designer's experience to vary the parameters, changing the design concept to satisfy these design criteria. A number of iterations are generally required to satisfy the constraints.

3.2 DMS implementation

Design work commences with the design agents registering with the DMS. Agent details are registered within the discipline/technology model of the DMS. This initial communication informs the DMS that the specified agent is capable of undertaking some design activity. The DMS then makes a request that the design agent provides information regarding the nature of the design activity that it can undertake. This information will take the form of a list of tasks, details of files or parameters that the task may require, constraints that need to be satisfied prior to task enactment, and files and criterion that result from the enactment of the task. These task details may be either low-level or high-level terms and may describe an individual atomic task or a group of tasks depending upon the level of abstraction of the implementation of the design agent. The design process may then be designed by decomposing it into the relevant tasks available. Informational dependencies must be maintained during the design of the process. The DMS will notify the user if a task has been included within the process that requires information that has not yet been produced. The task would then disable itself until this informational dependency has been satisfied. The consequence of ensuring informational dependency is that tasks may not be decoupled, and concurrency may only occur as defined by the process.

The design concept to be evaluated may be selected and modified by viewing the input files within the design process. Once a suitable design concept has been defined, the process is started with the DMS communicating to the design agents in turn providing them with the necessary information to undertake the current task. Once the design agent has processed the information and executed the associated tool, the design agent will notify the DMS of the information that has been produced from the tool. The DMS then determines from the design structure matrix which tasks are now capable of being enacted and communicates with those agents in a similar manner. Once the process has been completed, the DMS will evaluate any goals that have been provided and notify the user of the outcome.

Experiments using the DMS to co-ordinate the activity of those agents operating on the same workstation have produced the same results as that produced using the applications manually, but within 75% of the time due to the removal of the overhead associated with manual enactment. The experiment was repeated with the agents distributed across a number of workstations, resulting with a 46% reduction in the time taken to evaluate the concept whilst guaranteeing the consistency of the information within the process. This greater reduction was due to concurrent design activity across a number of computational resources. The DMS also ensures that if an application failed to produce a solution, a warning would be issued to the user with a description of the nature of the failure, and the remaining tasks would be suspended.

4 Conclusions

A new approach to tactically co-ordinating distributed, agent-based design activities has been presented. The approach enables concurrent engineering to be realised through the management of dependencies. From a CE perspective the focus is more towards enabling concurrency, rather than maximising concurrency by de-coupling activities, thus allowing the design process to progress naturally.

The approach was evaluated using an industrial case study that had a well-established design process. The activities within the process as well as the management of the information resulting from the activities were previously performed manually using a number of different computer-based design tools. Upon completion of the process, the results from the design activities would be used within calculations to check that the design concept satisfied the constraints. Alterations would subsequently be made to the parameters that were thought to be preventing the concept from satisfying the constraints, and the process would be repeated until the design concept fulfils these requirements. The time taken to evaluate a design concept through the manual enactment of the process was approximately 11 minutes.

Each design tool was subsequently managed by a design agent within the DMS. The design process was then constructed using information obtained from these agents. Through the management of the dependencies within the process, two threads of concurrent activities were made apparent. The evaluation of a design concept was managed using the DMS using the same platform as the manual test with the agents distributed across three Ultrasparc workstations and produced a reduction in process time of approximately 46%. This time obviously depends upon the amount of the resources available to the agents, however, the same principle applies for the manual case, and in both the manual and DMS experiments, the resources had 100% of the cpu available to the user. This time may be considerably reduced further with the inclusion of a generic “operational co-ordination” module [4] through the dynamic co-ordination of resources.

The DMS was also responsible for managing the constraints such that the stresses within the blades and the vibration characteristics met certain pre-determined requirements. This was achieved by altering various geometrical characteristics of the blades using specific guidelines.

5 Acknowledgements

The authors gratefully acknowledge the support given by the Engineering and Physical Science Research Council who provided the grant RES/4741/0929 that enabled this work to be undertaken. Additionally, the authors acknowledge Siemens Power Generation Systems, for their advice, expertise and for providing the case study.

References

[1] Coates G, Whitfield RI, Duffy AHB & Hills W, "A Review of Co-ordination Approaches and Systems - Part II: An Operational Perspective", *Research in Engineering Design*, Vol. 12, No. 2, 2000, pp. 73-89.

[2] Whitfield, RI, Coates G, Duffy AHB & Hills W, "A Review of Coordination Approaches and Systems - Part I: Strategic Perspective", *Research in Engineering Design*, Vol. 12 No. 1, 2000, pp. 48-60.

[3] Winner RI, Pennel JP, Bertrand HE & Slusarczuk MMG, "The Role of Concurrent Engineering in Weapons System Acquisition", *IDA Report R-338*, Institute of Defence Analyses, Alexandria VA. 1988.

[4] Coates G, Whitfield RI, Duffy AHB & Hills W, "Enabling Concurrent Engineering Through Design Co-ordination", *6th ISPE International Conference on Concurrent Engineering*, Bath, United Kingdom, September 1-3 1999.

[5] Andreasen MM, Duffy AHB, MacCallum KJ, Bowen J & Storm T, "The Design Co-ordination Framework: key elements for effective product development", *International engineering design debate: The design productivity debate, Meeting*; 1st Sept. 1996, Glasgow, pp. 151-174.

[6] Malone TW, Lai K & Grant KR, "Agent for Information Sharing and Co-ordination: A History and Some Reflections", *Software Agents*, AAAI Press, 1997, pp. 109-143.

[7] Eppinger SD, Whitney DE, Smith RP & Gebala DA, "A Model-Based Method for Organizing Tasks in Product Development", *Research in Engineering Design*, 6, pp.1-13, 1994.

[8] Kusiak A & Wang J, "Decomposition of the Design Process", ASME Transactions: *Journal of Mechanical Design*, Vol. 115, No. 4, 1993, pp. 687-695.

Dr. R.I. Whitfield
CAD Centre, Department of Manufacturing and Engineering Management, University of Strathclyde, 75 Montrose Street, Glasgow, G1 1XJ, Scotland. Tel.: +44 (0)141 548 3020 Fax: +44 (0)141 552 7896 E-mail: ianw@cad.strath.ac.uk
Formerly:
Newcastle Engineering Design Centre, Armstrong Building, University of Newcastle upon Tyne, Tyne and Wear, NE1 7RU, England. Tel.: +44 (0)191 222 8556 Fax: +44 (0)191 261 6059 E-mail: r.i.whitfield@ncl.ac.uk