Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-98600

Collaborative engineering environments

Two examples of process improvement

J.B.R.M. Spee, D.J.A. Bijwaard and D.J. Laan*

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-98600

Collaborative engineering environments
Two examples of process improvement

J.B.R.M. Spee, D.J.A. Bijwaard and D.J. Laan*

* M.I.S. Organisatie-ingenieurs B.V.

This report is based on an article published in the Proceedings of the 8th Annual International Council on Systems Engineering 1998 "People, Teams and Systems", July 26-30, 1998, Vancouver (BC), Canada.

Division: Flight
Issued: January 2001
Classification of title: Unclassified

Collaborative Engineering Environments – Two Examples of Process Improvement

J.B.R.M. Spee
National Aerospace Laboratory NLR
Amsterdam, The Netherlands
spee@nlr.nl

D.J. Laan
M.I.S. Organisatie-ingenieurs B.V.
Maarssen, The Netherlands
dick_laan@compuserve.com

D.J.A. Bijwaard
National Aerospace Laboratory NLR
Amsterdam, The Netherlands
bijwaard@nlr.nl

Abstract. Companies are recognising that innovative processes are determining factors in competitiveness. Two examples from projects in aircraft development describe the introduction of collaborative engineering environments as a way to improve engineering processes.

A multi-disciplinary simulation environment integrates models from all disciplines involved in a common functional structure. Quick configuration for specific design problems and powerful feedback / visualisation capabilities enable engineering teams to concentrate on the integrated behaviour of the design.

An engineering process management system allows engineering teams to work concurrently in tasks, following a defined flow of activities, applying tools on a shared database. Automated management of workspaces including data consistency enables engineering teams to concentrate on the design activities.

The huge amount of experience in companies must be transformed for effective application in engineering processes. Compatible concepts, notations and implementation platforms make tangible knowledge like models and algorithms accessible. Computer-based design management makes knowledge on engineering processes and methods explicit.

INTRODUCTION

Observations of current industry initiatives show a focus on business process improvement. Innovative approaches to processes are recognised as determining factors in competitiveness. Improvements are aimed at efficiency (faster, cheaper, better) and effectiveness (doing the right things, at the right time) of processes. Process improvement is also necessary to handle the increasing level of complexity (the combined effect of multitude, diversity, and dynamics).

Process improvements are mostly the result of a combination of organisational and technological measures. Organisational solutions have been introduced like cross-functional teams and supply chain integration. These organisational structures have focused attention on collaborative aspects. One of the expected process improvements is a better team integration, i.e. improved communication and co-ordination in collaborative processes.

In the technology domain an array of solutions is proposed. Computer-based systems like Product Data Management and Enterprise Resource Planning play a dominant role in this area. Other possibilities are offered by existing packages for Computer-Aided Design with extensions for multi-user support and data management.

The main question addressed in this paper is - what can current technology contribute to process improvement in collaborative engineering? This paper describes two examples where so-called collaborative engineering environments have been introduced as a way to improve engineering processes. A collaborative engineering environment is a computer-based infrastructure, serving as a platform for application of engineering methods and tools, with specific support for team operations.

Both examples have been taken from projects in aircraft development. They start with analysis of the process, pointing to the origin of the problem. Then the solution is described, followed by the way it was introduced to engineers. The result in terms of process improvement closes each example. The final section of the paper relates these engineering process improvements to available knowledge in companies.

EXAMPLE 1: A MULTI-DISCIPLINARY SIMULATION ENVIRONMENT FOR AIRCRAFT DESIGN

Process Analysis. This example is concerned with the design of a yaw damper control law for a civil aircraft (Laan 1996). The yaw damper should provide proper damping characteristics of some aspects of the aircraft flight dynamics while inducing a minimum structural penalty. The structural penalty stems from additional control system generated loads especially on the aircraft fin. The design involves disciplines like aerodynamics, flight handling and aircraft loads.

Before the integrated process approach was introduced the various disciplines used to have their own simulation-tool for aircraft manoeuvres, for instance for stability & control, control law development and loads manoeuvres. This considerably obstructed an effective co-operation.

When for instance a control law developed by the systems specialist had to be analysed by the loads specialist, the control law first had to be implemented in the loads simulation-tool. This extra implementation work was the smallest problem however, worse were the discussions that might arise from the change in simulation environment. Control law issues were mixed up with discussions about differences in modelling for instance of aerodynamics or mass.

Integrating Disciplines in the Process. An integrated approach was defined in which each discipline provides sub model data concerning his expertise to a central simulation tool, thus using the sub models (and expertise!) of other disciplines (figure 1). The Matlab / Simulink™ software was chosen as modelling / simulation / analysis environment. A generic aircraft simulation model was defined. The model has standard six degrees of freedom (rigid body motion) but can be extended with elastic degrees of freedom if necessary for certain applications (the corresponding mode shapes are computed using a NASTRAN Finite Element Model). The simulation environment is supported with appropriate tools for configuration control, which are essential in a multi-user environment.

Procedures and tools were developed to be able to quickly provide the generic aircraft simulation model with disciplinary data for a specific aircraft, i.e. mass data, landing gear data, engine data, aerodynamic data, and systems data. For instance mass data was fully harmonised between the various disciplines. Various methods are available to generate mass data depending on the stage of the project (conceptual, feasibility, full-scale development). Mass distribution and totalled mass data, i.e. cg, moments of inertia are provided to the central simulation model.

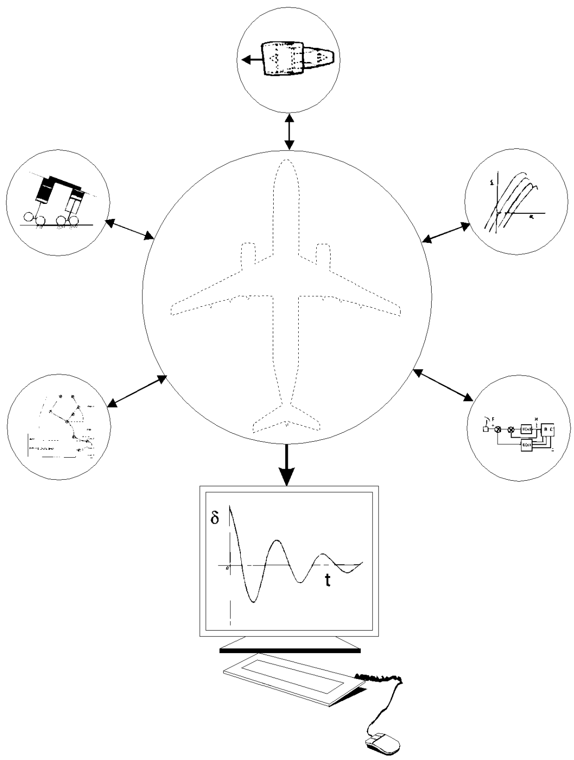

Figure 1: A Symbolic Picture of the Multi-disciplinary Simulation Environment, showing Modules for Mass, Landing Gear, Propulsion, Aerodynamics and Control

Another important input for the aircraft simulation environment is the aerodynamic database, describing the aerodynamic forces acting on the aircraft as a function of various state variables. Again, the aerodynamic coefficients had to be harmonised between the conceptual design group, the aerodynamics group and other specialists using aerodynamic data. A semi-empirical tool is used during conceptual design to provide a first estimate of aerodynamic coefficients and aerodynamic lift and moment distributions. The aerodynamic database is to be updated by an aerodynamics specialist in a later phase, on the basis of previous experience using scaling rules or using wind tunnel data combined with Computational Fluid Dynamics calculations.

Design exercise. A conceptual design of a 130 seat civil aircraft was used to exercise the new flight dynamics process. Specialists from stability & control, systems and loads demonstrated the integrated use of the simulation environment. A stability & control specialist analysed the Dutch roll behaviour, using a rudder deflection to trigger the aircraft. The simulation model produced the resulting aircraft response. From the results of the Dutch-roll investigation, it was concluded that for passenger comfort a yaw damper was necessary to add extra artificial damping. The stability & control specialist in co-operation with the systems specialist developed a yaw damper control law.

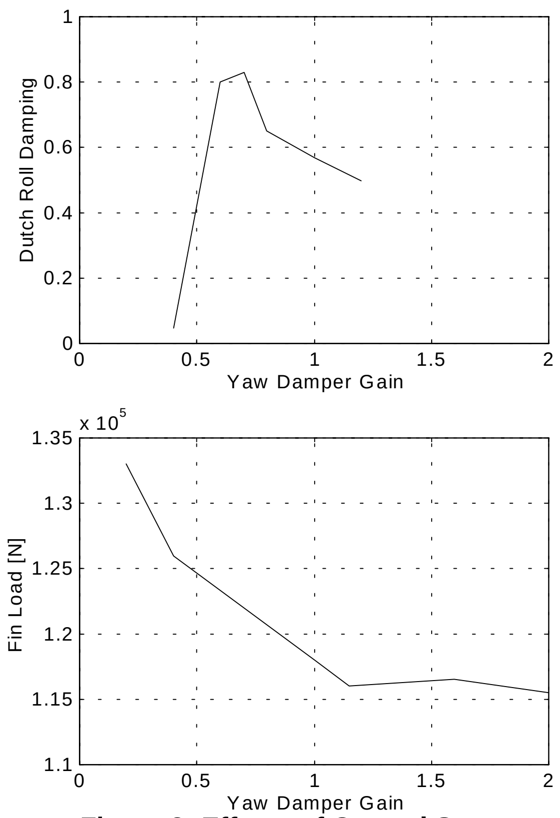

Figure 2: Effects of Control System on Aircraft Flight Dynamics and Structure Loading

As a further step a number of load cases were investigated by the aircraft loads specialist to analyse the effect of the yaw damper on the fin structure loads. The influence of the yaw damper on the fin loads strongly depends on the yaw damper gain (figure 2).

For this exercise it was concluded that for flight handling a gain of .7 would be optimal while a gain of 1.2 would give minimum fin loads. The discussions directly focused on this multidisciplinary trade-off that has to be made, because both specialists work together in the same simulation environment.

Process Improvement Result. The integrated approach allows engineers from different disciplines to effectively evaluate the design using an iterative dialogue between design problem and design solution. This has been realised by:

- Defining an adequate functional description of the system;
- Enabling disciplines to quickly transfer their detailed knowledge to the simulation environment;
- Using a collaborative simulation environment with appropriate feedback and visualisation capabilities that allows engineers to gain a thorough insight in the integrated behaviour of the design.

EXAMPLE 2: AN ENGINEERING PROCESS MANAGEMENT SYSTEM FOR FLIGHT CONTROL SYSTEM DEVELOPMENT

Process Analysis. This example concerns the development of a flight control system (FCS) for a military aircraft (Spee et al. 1997). The primary function of the flight control system is to control the aircraft along a pilot-commanded trajectory. At the same time the control system has to maximise the achievable attitudes and speeds of manoeuvring by exploiting the aerodynamics of the aircraft.

The flight control system has to fulfil these functions over a wide range of operational conditions. The aircraft dynamics vary with speed and altitude of flight, and with the aircraft configuration (e.g. external stores, weight). Failures and damage of airframe and systems modules have to be handled with minimal reductions in performance.

A typical approach in flight control systems development is to apply a grid of points over the envelope of operations. Points in the operations grid have multiple configurations and failure/damage conditions tied to them. Analysis, design, test and qualification activities are performed for all applicable combinations in every point in the grid, and for operational scenarios involving trajectories through multiple grid points.

From the above view on flight control system development it is clear that the process suffers from combinatorial explosion. During the detailed design phases of the FCS life cycle it is not uncommon for design teams to spend between 30-60% of their time on data management!

Computer-Based Engineering Frameworks. Recent developments in computer-based engineering often involve the notion of a framework. Frameworks are the result of the need for tool inter-operability in computer-aided design environments. Tool integration evolved from file-exchange in compatible formats, via common data base and design data management, to the current state-of-the-art: engineering process management (Wolf 1994).

A computer-based engineering framework offers common services for end-users in engineering processes, and serves as an integration environment for tools. The framework presented here is a product called SiFrame™. The available framework services in SiFrame are (figure 3):

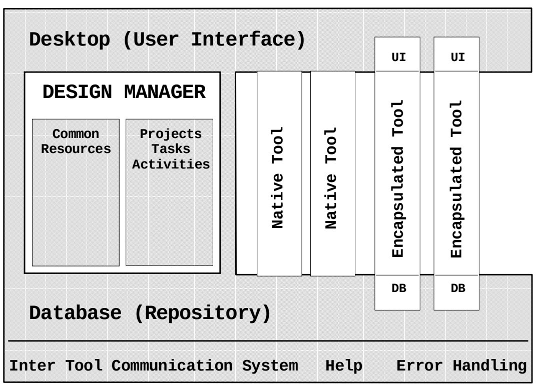

Figure 3: Engineering Framework Architecture

Desktop services - a graphical user interface to access all objects in the framework. Engineering process objects are presented in trees (hierarchy) or networks (a-cyclic graph). Colours and texts indicate the state of process objects. Changes in process structure or state are reflected immediately on the desktop of all connected clients.

Database services - management of all engineering process information in the framework and links to the design results. Each logical database contains information on common resources and multiple engineering projects. This includes the states of all tasks in running projects. Databases can be physically distributed, to support multi-site development.

Common basic services - general support functions for framework tools. Examples are inter-tool communication services, a help system, and error handling.

Design management services - the core functionality and rules for engineering processes in the framework. One part of design management is concerned with common resources, i.e. the registration, definition and control of resources available to all projects in a particular logical database. The other part of design management involves actual engineering projects, i.e. the definition of process structure and control of task execution in the framework. The next section describes some of the concepts for engineering process management.

Concepts for Engineering Process Management. Engineering process management in SiFrame is centred on concepts for process definition and process execution. Process definition is based on the concepts: projects; tasks; teams; and activities in flows (see figure 4):

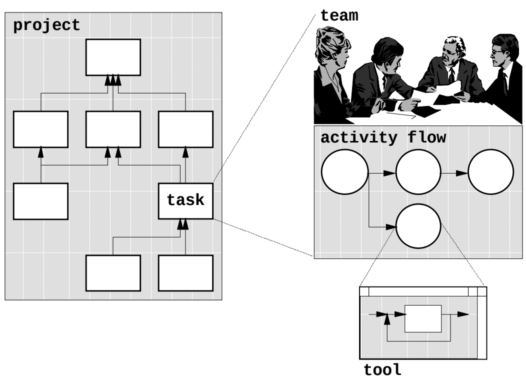

Figure 4: SiFrame Concepts for Engineering Process Management

Projects are the top-level in the engineering process organisation. The available resources in the project are selected from the common resources. Only members from teams assigned to the project have access to its contents.

Tasks are the building blocks from which an engineering project is constructed. The task structure defines the top-level engineering process flow, displaying both time and data dependencies between tasks. Results flow from bottom to top (!).

Teams are assigned to each task in the project. Each task has one team attached. Team members can execute engineering activities, if the task is reserved into a team or user workspace.

Activity flows are an essential element in this framework. The flow defines in which order activities should be executed. As such an activity flow defines the time dependency between activities in one task.

Activities are the lowest level at which process execution is defined and controlled. The definition comprises - the tool to be used for the activity; the data type(s) it can use as input(s); and the data type(s) it can produce as output(s).

Data types define data dependencies between activities, both within and across tasks.

Engineering process definition is based on the above concepts. Execution of engineering workflow is based on the concepts: workspaces, states, publication, task and activity versions and consistency.

Workspaces are in this framework to enable concurrency in engineering processes. Teams and users have to reserve tasks into their workspace prior to performing any activities in them. Reservation of an object locks it from access by others.

Publication is the mechanism to make results available to users outside the workspace in which they were produced. Published data is marked as read-only in the repository, effectively 'freezing' the design result.

States describe the dynamics of tasks and activities in the engineering process. Task states can be: empty (no previous execution); in work (one/more activities have been executed); partly published (result of activities have been issued); or complete (entire task result has been published). An activity can be: not executed; executed (tool has run with valid result); invalid (input has changed by predecessor activity); or finished (result has been published).

Several **task and activity versions** can be created. This allows for engineering iterations, with or without storage of engineering history. Execution of an activity version starts the assigned tool and provides the correct version of data from the database. Task versions can be linked to create different alternatives. The framework can show a graph of dependencies between task versions, as a powerful and intuitive means to traceability.

Consistency is the central and most powerful concept in SiFrame. The framework guarantees a consistent relation between all the engineering activities, by monitoring the state of activities and the data they produce. Consistency is controlled by application of rules in the framework: before, during and after activity execution; upon publication of results; and upon change of the process structure (task version network). SiFrame provides consistent data and time relations in engineering processes, which is an important contribution to efficiency improvement.

Review through Prototyping. The possibilities and limitations of the engineering management system were assessed in a prototyping project with engineers from several European aircraft manufacturers.

All participants received a 3-day training to get acquainted with the engineering environment. A part of a flight control system development process was defined, based on analysis of the working practices from one of the project partners. The process and supporting tools were implemented in the engineering environment. Subsequently evaluation scenarios were used to assess various aspects of the engineering environment when working in teams. Questionnaires were filled out afterwards to capture new user requirements and comments on working with the prototype.

A snapshot of the development process during one of the evaluation sessions is shown in figure 5. The flight control system development process is implemented in the engineering environment as a project with tasks. Tasks can exist in several versions, as indicated by automatically assigned version numbers. The state of the tasks reveals how far the development project has progressed. Most progress has been made in the flight control laws design. One design path has been completed, i.e. results have been published: the system architecture (ARCH1) was input to control law architecture (FCL1), followed by parallel tasks for lateral and longitudinal control law design (LAT1, LON1). These parts of the design were evaluated together (EVAL1). An iteration of the lateral control law design is in progress (LAT2), and will be combined with the original longitudinal design in a second evaluation (EVAL2).

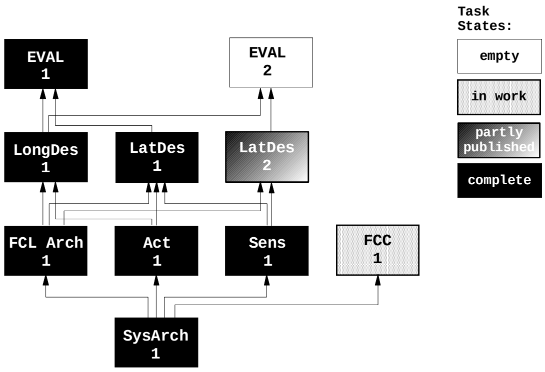

Figure 5: Snapshot of Flight Control System Development Process in the Engineering Process Management System

A snapshot of the process is called a **decomposition** in SiFrame. The decomposition shows the actual state of the process, and the relation between the task versions. This view on the process and data configuration is always up-to-date and consistent, since the information is retrieved on-line from the database of the engineering environment.

Process Improvement Result. Support for data management was the most important requirement for the flight control system development process. The engineering environment uses the ‘road map model’ for data storage (Hamer et al. 1996): the task and activity structure in the process is a placeholder for engineering data in the database and the dependencies between versions of data. The responsibility of the users of the engineering environment is to build a task network that leads to the desired result. The Design Manager of the engineering environment performs all remaining data management functions. Engineers rated the integrated and transparent data management as very beneficial.

Most discussions concerned process issues. Projects, tasks and activities with associated tools were regarded as an intuitive view on industrial engineering processes. Several suggestions for alternative rules and authorisations in the design management module of the engineering environment were received. Company-specific process control would be preferred.

DISCUSSION & CONCLUSIONS

In many companies a huge amount of experience is gathered from research work or from previous projects. Attention must be given to transform this knowledge in such a way that it can be applied effectively in engineering processes.

The most tangible engineering knowledge is available as models and algorithms. Accessibility is often low, because of incompatibility of concepts, differences in notations, and separated implementation languages and computing platforms.

The processes and methods applied in engineering are different types of knowledge. Most of this knowledge is implicit though; i.e. kept in the heads of experienced engineers, and fragmented over several disciplines.

One possible approach to process improvement was shown in the example 1. Detailed knowledge of previous designs was transformed in quick and reliable estimation methods (e.g. for mass, aerodynamic characteristics). These methods were closely coupled to the multi-disciplinary simulation environment. Efficiency was improved because disciplines contributed models from their expertise area only. Effectiveness was improved, because the integrated simulation environment focussed discussion on the true issue: the behaviour of the design concept.

A different approach to process improvement was shown in example 2. The engineering process is made explicit in the engineering management system. The process definition is executed on-line in the system, and the process state is maintained in the central database alongside the results. This is the basis for transparent data management, reducing the workload on designers and the number of errors made, especially in large projects where large numbers of data items with complex relations exist. It is also the basis for process traceability and design history, effectively capturing current engineering practices (‘as is process’), and offering possibilities for introduction of new engineering methods. The design management concepts provide a common terminology, which is essential in process improvement efforts.

REFERENCES

Hamer, P.van den and Lepoeter, K., "Managing Design Data: The Five Dimensions of CAD Frameworks, Configuration Management, and Product Data Management." *Proceedings of the IEEE, Vol. 84, No. 1, January 1996.*

Laan, D.J., "A Multi-Disciplinary Approach in Computer-Aided Engineering." *AGARD Structures and Materials Panel, 82nd Meeting, AGARD Report 814, 1996.*

Spee, J.B.R.M. and Bijwaard, D.J.A. "Computational Aircraft Control Engineering Environment Overview Document." *GARTEUR TP-088-10, 1997.*

Wolf, P. van der, *CAD Frameworks, Principles and Architecture.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

BIOGRAPHIES

Ir. J.B.R.M. (Jacques) Spee, obtained his degree in aerospace engineering from Delft University of Technology. Since 1992 he has been working at the National Aerospace Laboratory NLR of The Netherlands in the areas of real-time simulation, flight mechanics and systems research. His is currently active in flight-testing of future air traffic management systems, and in engineering process improvement.

Ir. D.J. (Dick) Laan, consultant at M.I.S. Organisatie-ingenieurs B.V., has over 15 years of experience in applying Information Technology in Engineering processes. He graduated at the Delft University of Technology in Applied Mathematics.

During his career at Fokker Aircraft he has gained a broad overview of the processes within and surrounding engineering. He started as a design engineer working on Computational Fluid Dynamics within Aerodynamics. In 1994 Dick became project manager Computer Aided Engineering and MDO.

Ir. D.J.A. (Dennis) Bijwaard, obtained his degree in computer science from the University of Twente in 1994. After his study he developed a prototype for automated face recognition at Sentient Machine Research. Since 1996, he has been working in the Data and Knowledge Systems department of the National Aerospace Laboratory NLR. He is currently active in network middleware, process management and computer working environments.