Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR-TP-2000-313

Engineering workflow

The process in product data technology

D.J.A. Bijwaard, J.B.R.M. Spee, P.T. de Boer

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2000-313

Engineering workflow

The process in product data technology

D.J.A. Bijwaard, J.B.R.M. Spee, P.T. de Boer

This investigation has been carried out under a contract awarded by BAE SYSTEMS, contract number SPOS0191925, ammendment nr 2.

BAE SYSTEMS has granted NLR permission to publish this report.

This report is based on a presentation held on the PDT Europe 2K, ESTEC, Noordwijk, The Netherlands on 2-4 may 2000.

Division: Information and Communication Technology

Issued: June 2000

Classification of title: unclasified

Contents

1 Introduction **3**

2 The need for Engineering Workflow **3**

3 Example engineering workflow **3**

4 Architecture of an Engineering Workflow Management System **5**
4.1 Workflow templates 6
4.2 Workflow execution 7

5 Multi-site co-operation **8**

6 Use of open standards **9**

7 Conclusions **9**

8 Acknowledgement **10**

9 References **10**

(10 pages in total)

Engineering Workflow
The Process in Product Data Technology

D.J.A. Bijwaard, J.B.R.M. Spee, P.T. de Boer
National Aerospace Laboratory NLR, P.O.Box 90502, 1006 BM AMSTERDAM, The Netherlands
Fax:+31 20 511 3210 (central fax number) *Tel:*+31 20 511 3045 (direct) Mailto:bijwaard@nlr.nl

1 Introduction

The prevailing paradigm for enterprises in the new decade is undoubtedly speed. This enterprise view is driven by the availability of e-business technology that enables new forms of collaboration between companies. The rapid developments in e-business also have an impact on the future of engineering organizations. This paper focuses on the early phases of a product's life cycle, i.e. between initial concept and release to manufacturing. New engineering workflow capabilities are presented, that have been tailored to speed up the engineering of new products.

2 The need for Engineering Workflow

Product Data Technology (PDT) plays an important role as an enabling technology for process support in engineering organizations. From its origin Product Data Management (PDM) systems offer functions for management of product data, product documentation, product structure, parts and components. Gradually this is extended with functionality for management of processes, programs and projects. In general though, the process management capabilities of PDM systems are limited to support repetitive procedures like document approval, and change orders. What is lacking in most PDM systems is support for the value-adding and often informal processes that are typical in early product engineering stages. In these processes, data type transformations are essential, and consistency management is crusial. Carrying a packet with all the different data items along with the process, like in PDM Workflow, makes it difficult to enforce this form of consistency.

So-called workflow technology (WFT) is an alternative solution that would possibly provide process management support for product engineering. Workflow technology originated from business process re-engineering initiatives, and has proven applications in office process automation. A typical workflow management system is capable of routing information on the basis of a predefined process, assigning the activities to appropriate persons in an organization, and starting the appropriate tools for activity execution. The limitation of WFT is that it offers only limited capabilities for iteration and working with candidate solutions for the same problem in the same process.

Engineering workflow is presented here as an enabler for better engineering processes. Engineering processes typically require support for many parallel activities with complex interdepencies, applying iterative design and development strategies, while maintaining configuration control and traceability. An engineering workflow management system (EWfMS) provides a shared and up-to-date process view across companies and disciplines, and integration of distributed engineering tool sets and product data repositories. The EWfMS builds on the capabilities of both PDT and WFT, and effectively integrates product data and process capabilities, with the flexibility that is needed in the early engineering phases.

3 Example engineering workflow

Engineering workflow management is concerned with the effective flow of work through activities that create and use engineering data across the product life-cycle. Engineering workflow involves not only technical activities, but also related activities like documentation, qualification, and business control. Workflows in the early engineering stages have typical properties that distinguish them from other types of workflows:

- **Transformation of data –** engineering activities add value by *transforming* input data into other types of output data, i.e. the information changes through the work performed. Transformations are essential elements of engineering workflows and make them *process centric*. The common view on workflow is information centric, and concentrates on the correct routing of data sets (or packages) between persons or roles that participate in the workflow. Information routing supports processes like document approval or change orders, and is a useful subset of engineering workflow.

- **Complex interdepencies –** changes in engineering information may have an impact on many activities that use the information, and changes are the essence of engineering workflows. The producing and consuming activities in engineering workflows drive relations like hierarchy (part-of), configuration (version-of) and variant (alternative-for). These complex interdependencies cannot be managed by data only, and must be related to the engineering processes that created them.
- **Many parallel processes –** time-to-market is a key factor in competitiveness for engineering organizations, and this has led to introduction of parallel processes like concurrent engineering and cross-functional teams. Engineering workflows should help to speed up product creation with process-driven data propagation, enterprise-wide status display and change notification, and cross-process synchronization facilities.
- **Frequent process changes –** Engineering processes are dynamic in nature, and develop along with the product that engineering teams are working on. A planned process is used at the start of a product creation process, but engineering workflows have a low degree of routine. Hence the degree of uncertainty and process adaptation is high in the early phases, leading to ad-hoc business and technical decisions. This will spawn extra work, trigger specific processes or may result in activities being abandoned altogether.
- **Iteration and alternative exploration –** Iteration is a common approach in engineering workflow, that provides targeted risk reduction of the product under development, and makes early samples available for use in concurrent teams. Exploration of alternatives is an essential element of engineering processes, followed by selection amongst alternatives based on commercial and technical criteria. Engineering workflows need to maintain the relation between data and process in iteration cycles and branches in the design space, and allow for backtracking to a previous state. Workflows must also capture the rationale for development decisions in conjunction with the then-available engineering information.
- **Multiple levels of data maturity –** Engineering information exists in multiple levels of maturity, like preliminary design, detailed design, production drawings. Engineering workflows must support the creation and dissemination of 'raw engineering data' to quickly start the process. At the same time the workflows needs to track where updates are necessary once more mature data comes available. Rules will exist for the visibility of engineering information based on the maturity level, and these must be followed in the workflow.

An example is presented in Figure 1 to illustrate the engineering workflow characteristics in a diagrammatic process view. The process shows different paths for writing different types of requirements

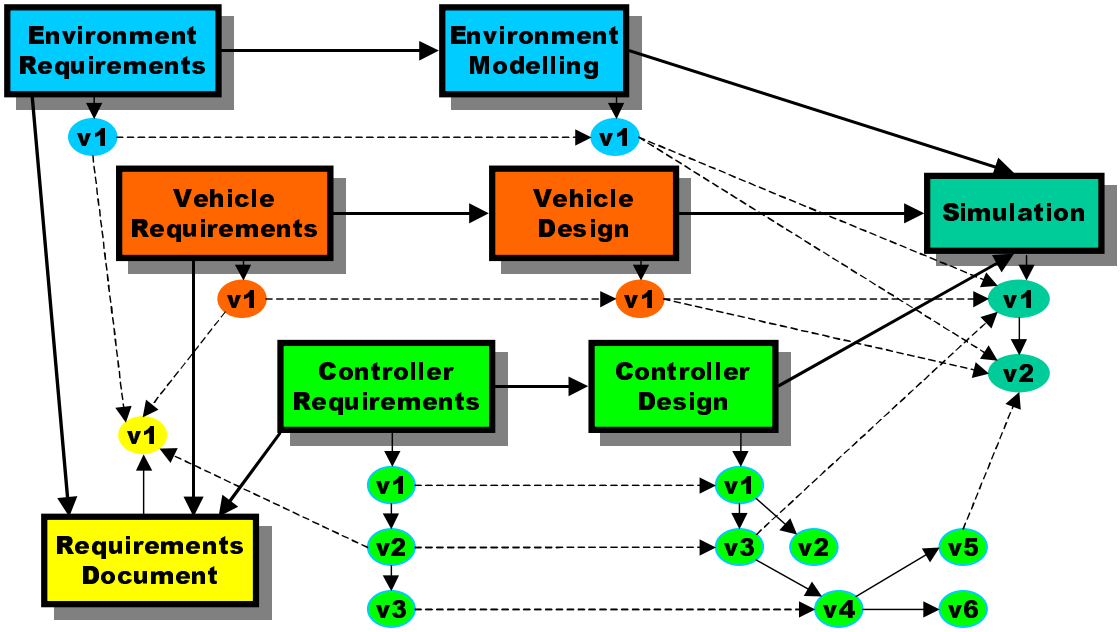

Figure 1: Engineering Workflow Example

and the subsequent modeling/design of the separate part of a simulation environment. The sets of requirements are bundled together in a requirements document task.

The example also shows that a new version of the “Controller Requirements” task is issued and that the Requirements document has not yet been updated to reflect those changes. On the other hand, the “Controller Design” is already using the new requirements and has modeled two candidates from which one (version 5) is already used in version 2 of the simulation task. A process change that might occur is the separation of the requirements and design of the vehicle in parallel engine and body tasks.

What is needed in engineering environments is computer-support for engineering workflow management, i.e. an engineering workflow management system as described in the next section.

4 Architecture of an Engineering Workflow Management System

Engineering processes are steadily increasing in complexity, but at the same time must reduce time-to-market for new products. An integration of product data technology and workflow technology is required to speed up the early phases of product engineering. A novel engineering workflow management system (EWfMS) has been developed, that extends product data management facilities with engineering-specific process functionality. The combination is a so-called computer-aided engineering environment. The EwfMS is a process-driven infrastructure that serves as an integration environment for tool sets that engineers apply, and the engineering information that is created and used in the process.

The EwfMS described here is a system composed of dedicated modules for different tasks, see Figure 2. The modules can communicate with one another using the CORBA standard (Ref. 5).

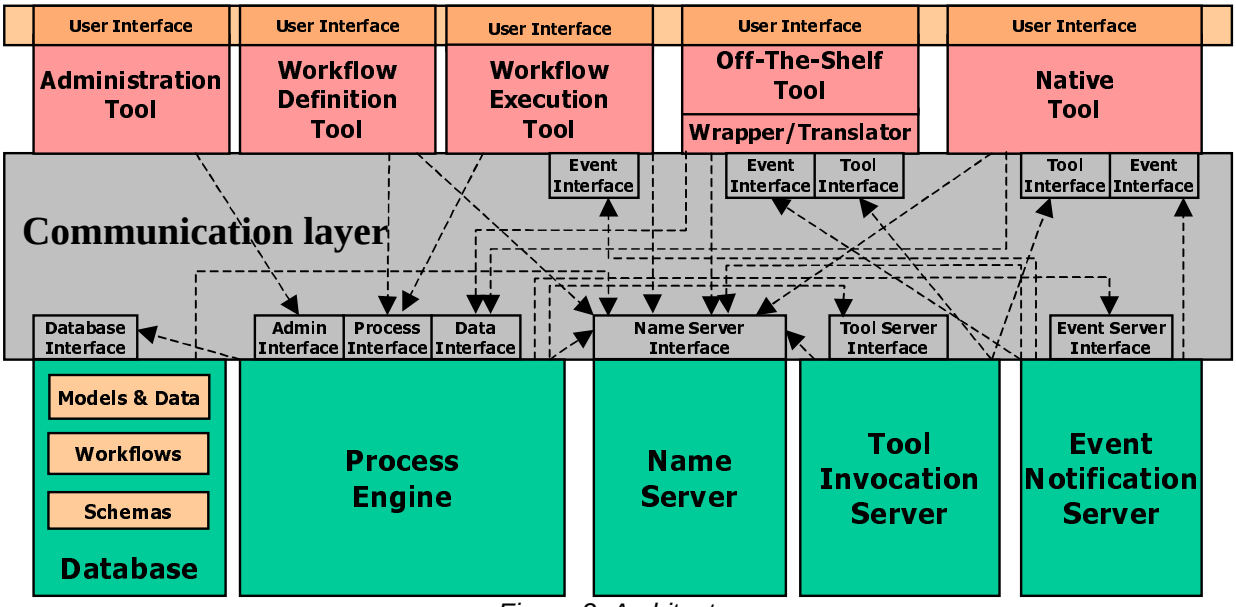

Figure 2: Architecture

The following modules are identified in the architecture:

- The **Database** stores the data models (EXPRESS schemas) and the actual data for the workflow templates, the instantiated workflows, the tool data, and the organizational data. The database is a STEP-compliant object management system, it can hold and makes accessible both data models using the EXPRESS language and the actual data using STEP (ref. 3: ISO 10303).
- The **Process Engine** is the only module that can create, modify and update workflow templates and instantiated workflows in the Database. All data storage and retrieval is done via the Process Engine, this module takes care that the data is stored at the right slot in the instantiated workflow. Apart from storing and retrieving STEP files, the Process Engine also offers a late binding CORBA interface for populating and retrieving data items in the database (the STEP SDAI interface, part 22 of ISO 10303). This binding can be used by tools to store and retrieve data in STEP format when they don’t natively read and write STEP format.

- The **Administration Tool** can be used to register users, teams, projects, roles and other organizational data as well as tools.
- The **Workflow Definition Tool** can be used to define data types, tools and workflow templates. For portability reasons this tool has been written in Java.
- The **Workflow Execution Tool** can be used to instantiate a workflow template to an executable workflow and execute it. This tool will show the status of the different activities in a workflow using color coding, and tools tied to activities can be executed from it. For portability reasons this tool has been written in Java.
- The **Name Server** makes it possible for different modules to find other modules.
- The **Tool Invocation Server** is responsible for transparently starting tools anywhere in the network. This invocation server uses SPINEware middleware services (ref. 1) for invoking tools remotely on different platforms transparently for the user.
- The **Event Notification Server** makes sure all Workflow Execution tools get notified when states change in one of their opened workflows.
- **Native tools** can directly talk to the Process Engine for storing and retrieving data.
- **Off-The-Shelf tools** can be integrated using a so-called wrapper, this wrapper talks to the Process Engine for storing and retrieving data. When the tool does not use the same data model for the input or output as present in the database, a translator will be necessary to convert it from and/or to that data model.

4.1 Workflow templates

The first step for making an Engineering Workflow is creation of workflow templates. An example of such a template is displayed in Figure 3.

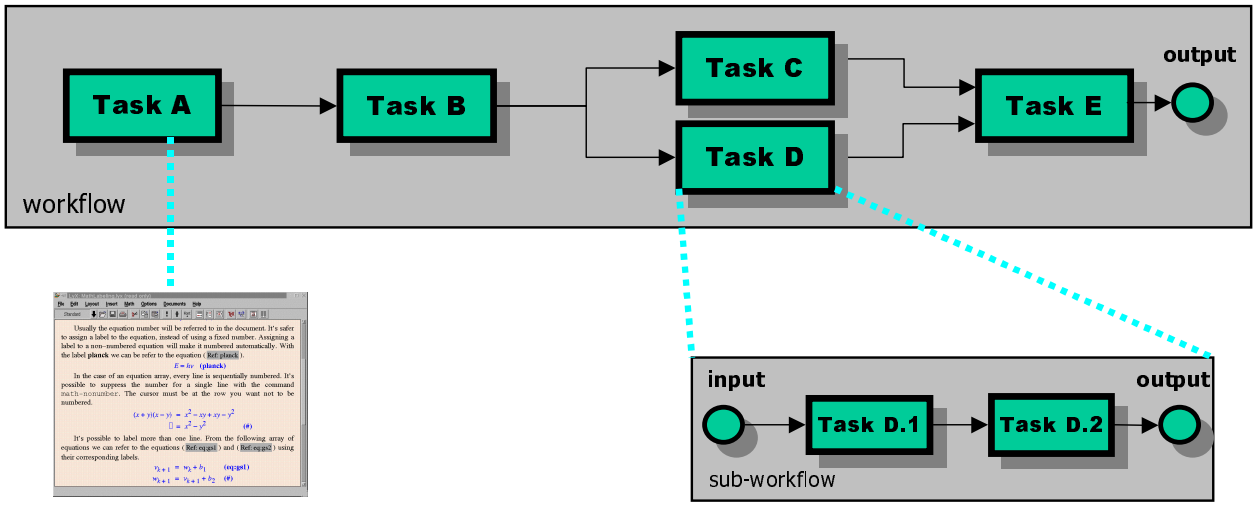

Figure 3: Workflow definition

The template shows a chain of tasks (A, B, C/D, and E) with two parallel ones (C and D). The lines between the tasks represent data flows of a defined type: activities can only be connected when their output(s) and input(s) are compatible (strong typing). There are currently two types of data types:

- A *binary data type* specifies a placeholder for blocks of binary data and optionally has a filename extension connected to it, because some tools might explicitly need that.
- A *STEP data type* has an EXPRESS schema associated with it that specifies its data model (e.g. AP233 is the EXPRESS schema for system engineering data, this schema was an inspiration for the schema that has been created for the process engine data itself).

Data types do not need to be in the database in advance, they can be created on the fly by either loading an EXPRESS schema for a STEP data type or by giving a name and optionally a filename extension for a binary data type.

To each task, either a tool or another template can be assigned. When a template is assigned to a task, this task is effectively a sub-workflow. Tools can have input(s), output(s) and a work-in-progress data type attached to it. The work-in-progress can be used for storing data items without directly making them available to subsequent tasks.

Input and output points are optional for every task in the workflow. These points can be used as hooks for reusing the flow as a task in another template. The top workflow in Figure 3 has only one output point on the last task and the bottom workflow in Figure 3 has one input point on the first task and one output point on the second task. The workflow assigned to task D shows that the output of task B is the input to task C and D1, and the output of task C and D2 are the inputs to task E.

4.2 Workflow execution

Once workflow templates are available, they can be instantiated to make them executable. An executable workflow is shown in Figure 4. The chain of tasks is the same as in the template above, extra are the states (represented as colors) and the data placeholders.

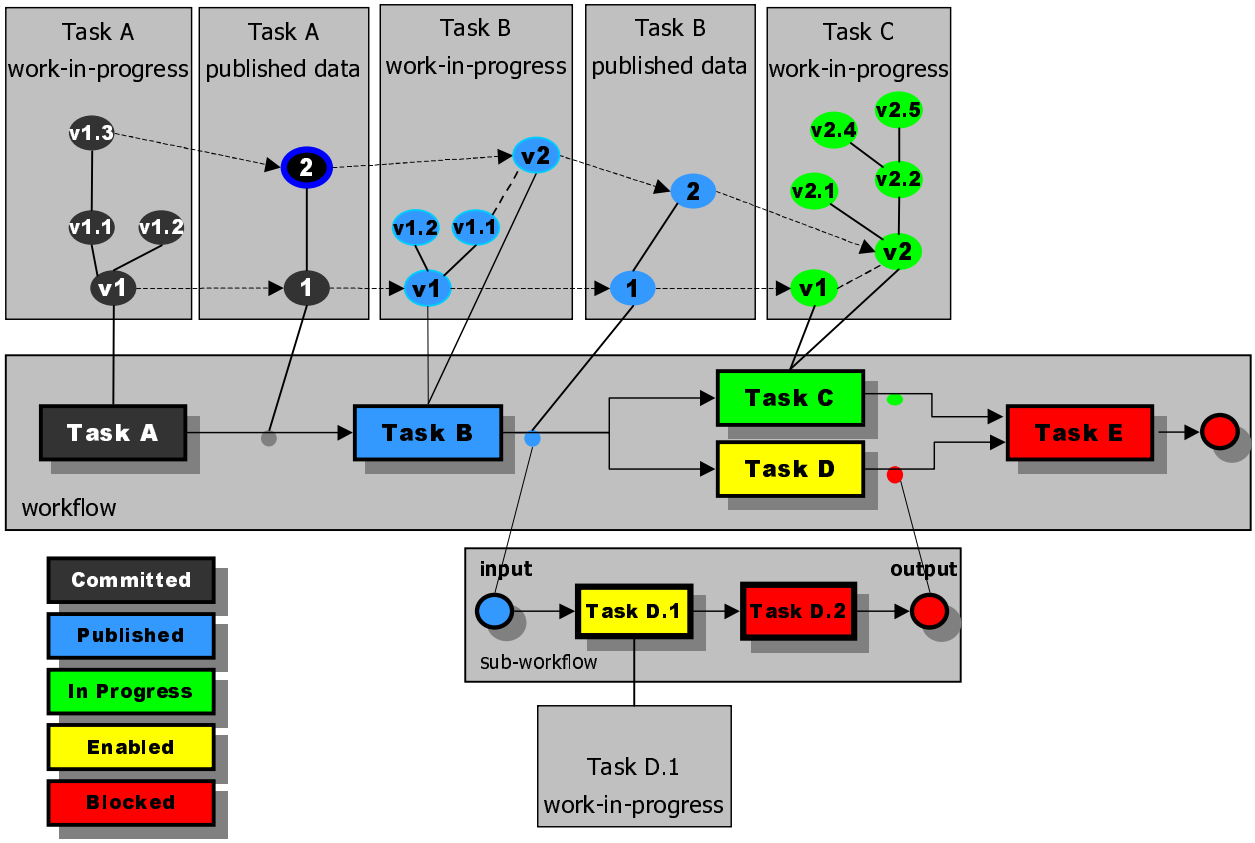

Figure 4: Workflow execution

There are two types of data placeholders:

- *Published data placeholders*: they contain the data items that are the output of a task and are accessible to subsequent tasks. When a task is completely finished, it can be committed (see Task A), one of the published results was selected (number 2) as the preferred output for subsequent use.
- *Work-in-progress placeholders*: they prove useful for tasks that take a considerable amount of time or in which different configurations/parameters are evaluated and therefore need versioning before the result is published for use in subsequent tasks.

The data type of the work-in-progress can differ from the published data, it can for instance be the native (e.g. binary) data format of the used tool.

Figure 4 also shows how the subtasks in task D retrieve their input from the task B and provide their output to task E.

Important issues in engineering workflow are on-the-fly process changes and iteration. During workflow execution, process changes can be made to the workflow and all available data items will be linked into a new version of the executing workflow. This automatically creates a history for the workflow and makes it possible to go back to the previous workflow version. Iteration can be done in different ways:

- Within the task, the work-in-progress can be used to store and retrieve different candidate intermediate results;
- The task can publish more than one result, which can be used in subsequent tasks. Once subsequent tasks start using a specific published output version, it cannot change that input for traceability reasons;
- Tasks can be versioned themselves (the major version numbers shown in the work-in-progress boxes in Figure 4). A new task version can be derived from an existing task with or without a reference to a work-in-progress item. The derived task can use different output versions than the task it was derived from.

5 Multi-site co-operation

For co-operation with other partners in the future, interoperability across companies is required. This interoperability is currently being addressed in the Innovative IT task of the ENHANCE project, this project is a cooperation between 52 European aerospace companies.

There are workflow engines available that can communicate and share workflows with multiple instances of themselves. However, forcing a specific Workflow engine upon partners is not feasible, most of them have different requirements and/or have already adopted a certain product company-wide. Fortunately, the Workflow Management Coalition (WfMC, ref. 2) provides a standard interface (interface 4) for communicating with other workflow systems as shown in Figure 5.

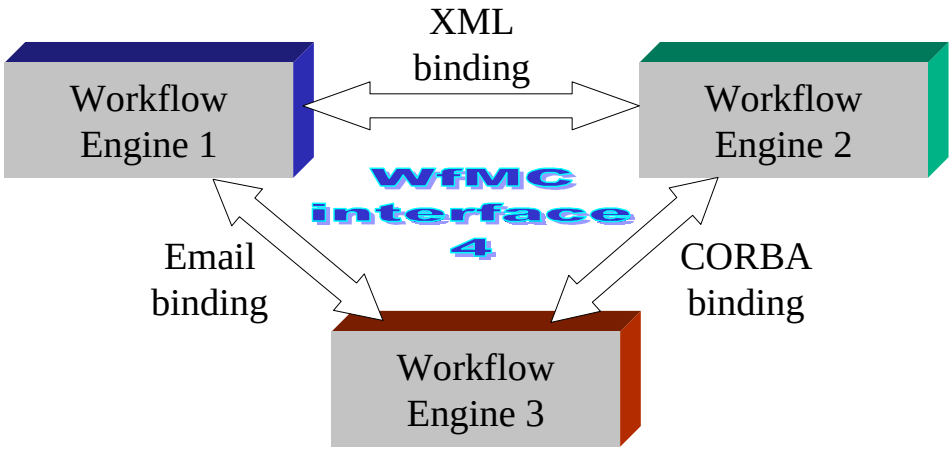

Figure 5: Multi-site Interoperability

The WfMC currently has an abstract definition of interface 4 and specific ones for XML and Email. As both of these implementations involve parsing text at each site, CORBA is proposed as a third binding, and the WfMC submission (ref. 4) to the OMG should do just that. Through the binding, workflow engines can inquiry each other about workflow templates, executing workflows and their state. It is up to the different sites to determine which parts of the workflow are to be available to other Workflow engines.

6 Use of open standards

The engineering workflow architecture as well as the data storage utilizes open standards where available and applicable. The use of open standards is very important, because it offers the following advantages:

- It offers the possibility to use the best modules for each function;
- Most technology for the open standards is already available, and can therefore quickly be integrated using Commercial Off-The-Shelf software;
- The modularized and open architecture makes cooperation with suppliers easier (virtual project teams).

The architecture uses the CORBA standard for all interfaces between modules. The Workflow Definition and Execution tools are both written in Java to ensure portability on different platforms. The server parts have all been written in C(++). The Administration Tool is currently a simple text interface written in C++ and will probably be changed to a graphical tool in Java.

The architecture has a lot in common with the Workflow Reference Model from the WfMC, the main difference is that it is more modularized and that the interfaces are not completely adhering to the WfMC interfaces yet. The WfMC components map to the following modules in the architecture described here:

- *WfMC Workflow Enactment Service*: The combination of the Database Engine, the Process Engine, the Event Notification Server and Tool Invocation Server;
- WfMC Workflow Client Application: The Workflow Execution Tool;
- *WfMC Administration and Monitoring*: Administration Tool, Workflow Execution Tool;
- *WfMC Invoked Applications*: Native Tool, Off-The-Shelf Tool;
- *WfMC Process Definition Tool*: The Workflow Definition Tool.

A difference in the operation of the architecture is that it uses availability of published data as start condition for subsequent tasks (data driven) opposed to the more general start condition for a task described by the WfMC. This has proven to increase the ease of definition and use of the workflow.

The biggest difference to the WfMC and PDM systems with respect to data management is the use of EXPRESS for data modeling and STEP for the operational data of both the Process Engine and the tools. This makes the system very open for all different types of data and can reduce the amount of conversions considerably, especially now more and more Application Protocols become available in ISO-10303. The late binding of SDAI calls using CORBA (ISO-10303 part 26) ensures that not every tool needs to read and/or write STEP part 22 (STEP physical file) natively, and thereby reduces the time required to write translators. In the future, also the early binding SDAI CORBA interface could be made available to reduce the effort even more.

7 Conclusions

The Engineering Workflow architecture described in this paper combines the best of both PDM Workflow, Workflow Technology (WFT) and open standards, to create a highly modularized and open system that is flexible enough to support engineering processes. The main advantages of this combination are:

- Explicit modeling and consistent management of data transformation through the process;
- Capabilities for working with candidate solutions for the same problem in the same process;
- Use of neutral formats for data exchange between tasks.

Flexibility is offered both in definition and execution of workflows. In workflow definition, hierarchical workflows can easily be created by reusing existing templates and new EXPRESS data models can be downloaded into the system on the fly to create new data types. In execution, flexibility is offered by making both iteration and on-the-fly workflow changes possible and by providing both binary and STEP

(part 22 as well as late binding CORBA) support for populating and retrieving data items. To support the engineer even more, using network middleware for tool invocation (e.g. SPINEware) hides all network peculiarities associated with tools running on different platforms.

The system is open for general use and is being introduced in the design of integrated vehicle systems in airplanes. It is also being applied in the ENHANCE project to enable co-operation with workflow engines at other companies that adhere to interface 4 of the WfMC.

8 Acknowledgement

The engineering workflow system described here follows on from a program NLR ran with BAE SYSTEMS.

9 References

1. E.H. Baalbergen, H. van der Ven, SPINEware – a framework for user-oriented and tailorable metacomputers, Future Generation Computer Systems 15 (1999), p. 549-558.
2. David Hollingsworth, Workflow Management Coalition, The Workflow Reference Model, TC00-1003, Issue 1.1, January 1995.
3. International Standardisation Organisation, Industrial automation systems and integration – Product data representation and exchange -- Part 1: Overview and fundamental principles, ISO 10303-1, 1994 (formerly STEP: Standard for Exchange of Product data).
4. OMG BODTF RFP #2 Submission, Workflow Management Facility, Revised Submission, July 4, 1998.
5. Object Management Group, The Common Object Request Broker: Architecture and Specification, revision 2.1, September 1997.