AN APPLICATION OF THE IPROD SOFTWARE FRAMEWORK TO SUPPORT THE PRODUCT DEVELOPMENT PROCESS IN THE AUTOMOTIVE AND AEROSPACE DOMAIN

Maurice F.M. Hoogreef  
Fac. of Aerospace Engineering, Dep. of Flight Performance and Propulsion  
Delft University of Technology  
m.f.m.hoogreef@tudelft.nl  

Reiner E.C. van Dijk  
Gianfranco La Rocca  
Fac. of Aerospace Engineering, Dep. of Flight Performance and Propulsion  
Delft University of Technology  
{r.e.c.vandijk, g.larocca}@tudelft.nl  

Robberto d’Ippolito  
R&D Manager  
Noesis Solutions N.V  
roberto.dippolito@noesissolutions.com

ABSTRACT

The Product Development Process (PDP) of manufacturing companies requires the efficient management of huge amounts of data from different sources and their integration in the sub-processes that compose the product development chain. This is a very challenging endeavor for which an integrated approach does not yet exist. The EC FP7 Project iProd aims at filling this void, by developing a flexible and service-oriented software framework, supported by a knowledge base that is structured by means of ontologies, to improve the efficiency and the quality of the PDP in the preliminary design phase. This paper discusses the first prototype of this software framework. The logic, overall software architecture and some of the implementations details of the framework are described. The functionalities of the software framework are demonstrated by means of two use-cases from two different domains, i.e. the automotive domain, represented by the development process of a car door, and the aerospace domain, represented by the development process of a rudder for a business jet. Preliminary testing, using the first prototype, indicates that the application of the framework to the two use-cases can yield benefits in terms of a reduction in development time in the preliminary design phase and results in product quality improvements, by having additional time for more design iterations to increase the maturity level during this phase. However, the framework is to be improved in terms of reliability, efficiency and maintainability. These improvements will be done during the development of two more software prototypes.

KEYWORDS

Capturing product information, Knowledge engineering, Knowledge management, Industrial workflow enhancement, Ontologies, Knowledge-based technologies to support the product development process

1. INTRODUCTION

This section provides a brief introduction to the subject and the objective and structure of the paper.

1.1. Knowledge and information management challenges in the PDP

Current engineering processes within modern-day organizations are becoming more and more knowledge-intensive as also the products themselves become more complex. Large amounts of data, information and knowledge have to be handled, stored, retrieved and reused with care and precision. The information model, the design knowledge and expertise are residing in the heads of the experts, as well as being partially formalized across a large set of heterogeneous and disconnected software systems. This knowledge and expertise is combined along the product development process. Shortening time-scales and geographic distribution have made effective management of this information very important. The Concurrent Engineering (CE) approach was proposed as a collaborative approach to support the development of products (e.g. Clarkson and Eckert [1]) to achieve “faster, better, cheaper” products. CE emphasizes the importance of frequent iterative interaction between traditionally separate functional teams. However, this increases managerial chal-
allenges and introduces additional pressure into the engineering process.

Industrial collaboration today is achieved through employing e.g. Product Data Management (PDM) and Enterprise Resource Planning (ERP) systems; industrial collaboration along the whole product life-cycle essentially depends on this ICT support. However, these systems use heterogeneous data models. As a result, there are many gaps between information systems. The iProd project aims at closing the gaps within the product life cycle, by means of dedicated knowledge management tools and methods, and improving the collaboration between stakeholders. The capability to manage semi-structured (e.g. requirements, specification documents) as well as structured (e.g. Computer Aided Design/Engineering (CAD/CAE), Bill of Materials (BOM), etc.) engineering information, both individually and in combination is a great challenge. Not addressing this challenge will result in an abundance of data, distributed over different systems, without any interaction.

1.2. Objective and structure of the paper

This paper discusses the first prototype of the iProd software framework. A short introduction to the iProd project and the use-cases presented in this paper is given in Section 2. The logic, overall software architecture and some of the implementation details of the framework are described in Section 3, to explain how the different framework modules connect, cooperate and reason on a structured knowledge base in the form of a graph database. Section 4 describes the flow through the iProd PDP, the architecture of the iProd software framework and the software architecture of the framework. The functionalities of the first prototype of the software framework are described in Section 5 by means of two use-cases from two different domains, i.e. the car door and the rudder for a business jet. Section 6 presents the conclusions and future work.

2. THE IPROD PROJECT

This section provides an introduction to the iProd project and presents the use-cases that will be used in the paper, to illustrate the first prototype of the software framework.

2.1. The iProd project

The iProd (Integrated Management of product heterogeneous data) general aim is to improve the efficiency and quality of the PDP of innovative products by developing a flexible and service oriented software framework that, reasoning and operating on a well-structured knowledge base, will be the backbone of the computer systems associated with current and new product development processes. A graphical representation of the iProd software framework is shown in Figure 1. iProd aims to integrate complex, large, heterogeneous and potentially conflicting data along the PDP to efficiently handle product development and tests. To this end, iProd committed to the development of both domain-independent and domain-specific ontologies to semantically structure engineering design, business and process knowledge, into one extensive, reusable and extendable knowledge base. Methods are being developed to fill the knowledge base with heterogeneous data coming from different tools and sources (including commercial and in-house systems), making them available for reasoning and support the operations of some knowledge based applications, developed in iProd, to automate some domain specific engineering processes. In this way, the iProd framework is able to support the decision makers and automate several parts of the PDP.

2.2. Industry use-cases

To validate and justify the development within iProd, several industry use-case are included in the project. These use-cases span three engineering domains, i.e. aerospace, automotive and home appliances. All three domains are represented by two use-cases each. These use-cases provide end-user feedback on the developments in the project and can also be used to test and demonstrate the software framework in an industrial, engineering application. In order to present the first prototype of the software framework that is described in this paper, two use-cases are used. From the aerospace domain, the Fokker Aerostructures use-case and from the automotive domain, the Pininfarina use-case. This section presents a general description of the two use-cases.

Pininfarina use-case

The Pininfarina Group, located in Cambiano, Italy, is a leading Italian design house and supplier of engineering solutions and parts in the automotive domain. The Pininfarina use-case is focused on the capture and reuse of the knowledge during the product development process. The objective is the design of the hinge of the NIDO vehicle door such that the door does not sag when opened, under its own weight and an additional point load.

Pininfarina’s expectations of the framework include
the correlation between performance and product sub-systems, to link tests and requirements to product structure elements. The framework is also expected to support task leaders in the decision making, by offering insight into the previous design decisions and the flow of information, thereby offering less risky decisions, to achieve a more optimal planning of the development phase. Moreover, the iProd framework should provide an a methodology for working in a collaborative environment. Ultimately, Pininfarina hopes to speed up the PDP for managing projects, especially those projects related to different versions/variants within the same family of vehicles, reducing the concept phase to weeks instead of months, through the introduction of standards and templates. Also, the goal is to integrate heterogeneous tools, to save time and costs and in the meantime trace the entire design process, from Computer Aided Design (CAD) to Computer Aided Engineering (CAE) and virtual testing.

Fokker Aerostructures use-case

Fokker Aerostructures (FAE), located in Papendrecht, The Netherlands, is a business unit of Fokker Technologies, the descendant of the former aircraft manufacturer. FAE is a specialist in the design, development and manufacturing of lightweight structures, modules and landing gears for the aerospace and defence industry. FAE is interested in having a software framework that can automatically find the optimal conceptual design (as the output of the proposal phase) for an aircraft box structure. The use-case introduces the business-jet rudder as example. The design is defined “optimal” when the solution results in maximum customer satisfaction. Most important characteristics in this respect are weight and cost.

The ultimate goal is to provide an on-line tool to aircraft integrators which can calculate in real time the price and weight of a structural component based on FAE technology resulting in a proposal. The expectations from the iProd framework are to capture design knowledge and make it available for (automated) re-use in future project, speed up the development and increase the number of design iteration, to achieve optimization through automation and the support of Multi-disciplinary Design Optimization (MDO). Similar to the Pininfarina use-case, the goal is to integrate heterogeneous tools and data, to save time and cost.
3. IPROD FRAMEWORK LOGIC

This section discusses the logic behind the iProd framework, describing the PDP in iProd and the tools that were developed to support the PDP. Additionally, the way the design knowledge is structured is described.

3.1. The PDP in iProd

The PDP that is considered in iProd is depicted in Figure 2. This generalized PDP was composed from ISO/IEC 15288 [4], which establishes a common framework for describing the life cycle of “systems created by humans”, and it defines a set of processes and associated terminology.

In particular, iProd addresses the technical process and the planning phase of the product development process, illustrated in Figure 2 (blue). The circular process in Figure 2 represents the fact that product development is an iterative process. The specific tools and solutions developed within iProd to support and improve the various stages of the PDP are indicated in the picture (orange) and will be described in some details in the next sections. Indicated in the innermost circle is the PDP process that is considered. The labels in the green, outermost circle represent the “Knowledge”, i.e. design rationale, product knowledge, process knowledge, algorithms and data from other (legacy) software sources, contained in a knowledge base in the iProd framework implementation. The knowledge base allows data from different stages of the PDP to be connected, allowing e.g. design data to influence requirements in a next iteration. In those cases, it could happen that an iteration does not pass all stages, but a new iteration is started directly once new information is discovered. More information on the knowledge base is provided in sub-section 3.2. The orange colored middle circle represents the tools and solutions that have are being used to implement the PDP process. This section describes the PDP process in the innermost circle and highlights the tools being used for each of the individual steps.

Requirements Definition

The first step in the PDP cycle in Figure 2 considers the definition of requirements for the new project and the investigation into product alternatives and competitor products. This information can be fed into the knowledge base through the iProd Project Configuration Tool, which also handles the project set-up, and the Correlation Matrix Tool. The tools will be illustrated in Section 5.

Requirements Analysis

The requirements analysis concerns the quantification of requirements into detailed specifications that must
be met by the product. Additionally, technical specifications determine which tests, both physical and virtual, must be performed during the design of the product, to verify these specifications. The quantification of requirements is done through the Correlation Matrix Tool. This tool also allows specification of the product structure and it allows relating all requirements to product structure elements and tests, which determines the actual Correlation Matrix. Standard product structures, tests and requirements related to a certain product are stored as templates in the knowledge base and are retrieved by the Correlation Matrix Tool. Any changes will be stored in the knowledge base for the particular project, such that the knowledge can be reused later.

Work Breakdown

The Work Breakdown Tool represents the intermediate step between the correlation matrix and task plan management in Figure 2, that retrieves the tests to be performed from the correlation matrix and combines those with the additional knowledge of the business process. All information is retrieved from the knowledge base and represented in the form of a Work Breakdown Structure (WBS). “The WBS is an overview that divides all activities into work packages, or a hierarchy of task packages. These tasks can in turn be subdivided, ultimately down to the level of tasks for individuals.”[5] The WBS can be edited to change the process and the resulting process is stored in the knowledge base again. The Work Breakdown Tool can also be used to inspect the process of the optimized task plan and also the results of a simulation workflow optimization.

Task Plan Management

The task plan management in Figure 2 involves the Task Planning Optimization, a tool which retrieves the non-optimized task plan from the work breakdown. This is the task plan that is constructed from the project set-up in the requirements definition and requirements analysis and the process that is stored in the knowledge base. The task planning optimization module retrieves the tasks with associated duration, required resources, available resources, etc. This information is passed to an algorithm that minimizes the total duration of the task plan and delivers this optimized task plan.

The Simulation Workflow Optimization tool deals with the whole process of virtual testing, e.g. using CAE tools. It optimizes the individual parameters and configurations of all applications combined in the simulation workflow, based on the requirements (e.g. accuracy, optimization time or number of iterations) for the simulation tasks, to minimize the combined duration of all simulations, without harming the design. Once an optimized simulation workflow is determined, or selected from several possible ones, the task planning optimization can be executed again, to come up with a new optimized task plan, including the, now optimized, simulation workflow. Thereby potentially reducing the total duration even more.

Design

The next step is the actual design of the product, assembling the product structure according to the design norms, rules and requirements, to come up with a bill of materials for the actual product structure. The design also involves the simulation workflow that was optimized by the Simulation Workflow Optimization. The design is supported by Knowledge Based Engineering (KBE) applications. The KBE applications are programs to solve specific problems within the design process, generally related to geometry modeling according to design rules, allowing Multi-disciplinary Design Optimization (MDO). The applications use so-called “High-Level Primitives” (HLP’s) [6] for the geometry modeling, parametrized models that can be automatically triggered and modified. This automation can take over the repetitive and non-creative tasks from the engineer. Details and examples on KBE can be found in articles by La Rocca [6] and Van Dijk et al. [7].

Verification and Execution

The verification and execution contains the verification of the product, through physical and virtual testing. This involves the execution of the simulation workflow and the corresponding KBE applications. MDO is supported by these applications in iProd, to increase the number of design iterations and improve product quality and design space exploration.

System Elements Validation & Validation

Subsequently the system elements and the overall product that has been designed have to be validated by the user. The results of the Verification and Execution will be presented through the user interface of the iProd framework and all data related to the design is consolidated in the knowledge base, for easy re-use and knowledge retention. The iProd PDP process has to be iterated until a satisfactory product is designed, before the project is delivered.
3.2. Knowledge Base and Knowledge Structuring

The knowledge base is a graph database that stores all information in the form of triples [8]. These triples are structured using the iProd ontology architecture that represents the backbone of the software framework, as shown in Figure 3. This follows the hybrid ontology import methodology by Woll et al. [9], that provides the possibility to have links between the domain independent ontologies at two levels. Relating high-level concepts of two domain independent ontologies, e.g. ReqIF and STEP233 can be done in the upper ontology, but also attributes of these concepts from the individual ontologies can be related in the common ontology.

Some of these ontologies have been specifically developed for the iProd project, such as the Correlation Matrix ontology or the Human Workflow ontology, yet others have been based on industry standards, such as the ReqIF ontology to describe requirements and a light version of the STEP233 ontology to describe product structures. These domain independent ontologies form the core of the iProd framework, they describe concepts common to any domain. Hence, they can be considered independent of a domain. The individual ontologies that form the domain independent ontologies import the so-called upper ontology, which provides the links between the high-level concepts defined in these ontologies. These domain independent ontologies contain concepts that are considered common standards, such as the STEP233 product definitions, and common concepts to different domains.

The use of different vocabularies in different domains or different companies is covered in the domain and use-case specific ontologies. The domain specific ontologies form a common ontology for a particular domain. An additional common ontology is placed in between the domain independent ontologies and the domain specific ontologies. This allows for the hybrid importing scheme, as presented by Woll et al. [9], to relate parameters and attributes of different domain independent ontologies. The common ontology is created through importing the domain independent ontologies, providing the possibility to link attributes of concepts from one domain independent ontologies to another, as explained in Woll et al. [9]. The domain specific on-
tologies provide the means to introduce company-specific vocabularies and concepts to the iProd framework for the considered domains, i.e. aerospace, automotive and home appliances. Concepts or terminologies or even whole processes that are common across different companies within one domain, could be collected in a domain common ontology.

Data is entered in the graph database in the form of triples. The information is either retrieved through interviews with product experts and engineers working on the product. Additionally, it is possible to enter data from other tools through the legacy data connector (LDC) [10]. This allows heterogeneous data from different sources, that otherwise would not have been accessible in the same environment, to be retrieved from a single source of truth. The LDC maps this data to the iProd ontologies. This way, product data, requirements and process data is accessible from one single, iProd, environment. The LDC can connect to e.g. spreadsheets, requirements management systems or PLM systems, extracting the data and combining it in one knowledge base.

Every project consists of its own graph, to distinguish between data belonging to different projects. A graph consists of triples belonging to a project, that can also have references to other graphs. This is used for triples that relate to domain independent ontologies. These ontologies are stored in the form of triples in one, domain independent, graph, which is used to provide the knowledge base structure.

4. PROCESS FLOW AND ARCHITECTURE

This section describes the iProd process flow through the PDP, as illustrated in Figure 2, the iProd process architecture and related software architecture.

4.1. Process Flow

Given the iProd PDP process in Figure 2, a flow through the software modules that compose the iProd framework can be constructed. These modules, as shown in Figure 4, are different to the steps in the PDP, as some modules are wrapped around several steps. This flow is the implementation of Figure 2 in the software environment. The project configuration, correlation matrix and work breakdown (partly) can be considered data acquisition steps in the flow. Here, the data necessary for the execution of the project is collected, through user interaction. The project configuration covers the requirements definition and requirements analysis steps in the PDP, see Figure 2.

The work breakdown, task planning optimization and simulation workflow optimization reason on the collected data and provide feedback loops. The task planning optimization follows the approach that was discussed earlier. The simulation workflow optimization deals with the whole process of virtual execution in Figure 2, it optimizes the individual parameters and configurations of all applications combined in the simulation workflow, based on the requirements for the simulation tasks (e.g. accuracy, optimization time or number of iterations), to minimize the combined duration of all simulations. Once an optimized simulation workflow is determined, or selected from several possible ones, the task planning optimization can be executed again, to come up with a new optimized task plan, including the, now optimized, simulation workflow, potentially reducing the total duration even more.

The execution module, as considered in the framework is not so much a separate module as it is a part of the PDP for iProd. Execution relates to the executonal part of the framework, i.e. the optimization of the actual product in a KBE application, controlled by the simulation workflow manager and coupled to legacy CAE tools. During the execution, the product is optimized according to the requirements and specification of the product structure in the project set-up and the correlation of requirement and product structure elements with required (virtual) tests in the correlation matrix. The virtual test execution is controlled by the simulation workflow management software.

The verification and validation is the final phase of the iProd PDP. Testing and optimization results should be analyzed and compared to the imposed requirements. In case of a test-failure or requirement nonconformity, the user should be directed back to the project configuration, to make the necessary modifications and restart the process. This phase requires the most interaction and design decisions from the user, as not all decisions for new products can be captured in design rules. During this phase, the user may introduce creative or innovative solutions to solve non-conformities. Introduction of these solutions may however be captured and reflected in the knowledge base.

4.2. Process Architecture

The iProd flow in Figure 4 provides data management capabilities in a single environment, federating and integrating heterogeneous data. Additionally part of the process can be automated, including optimization of
the products through KBE applications and automatically executed simulation workflows. In addition to this, the framework offers collaboration and decision support. The general set-up of the architecture is visualized in Figure 5.

As illustrated in Figure 5, the different framework modules, such as the correlation matrix or the project configuration, operate on a structured knowledge base. The user interacts with these modules through a user interface, where the user is provided with data templates that can be extended to contain project specific information. This information will in turn be stored in the knowledge base and will be used by some of the modules, such as the task planning optimization and simulation workflow optimization, in the background. Reasoning on the provided information and the rules and information already stored in the knowledge base, these modules will be able to construct simulation workflows for automated execution, which support MDO. These simulation workflows perform optimizations supported by specific KBE applications, using parametrized geometry, FEM analyses and other commercial or in-house tools. The simulation workflows of the KBE application, possibly connected to the CAE tools, can automatically optimize a product according to the provided user inputs. In return, the user is provided with the results, which will again be stored in the knowledge base and visualized in the user interface.

Software Architecture

The iProd software framework is role-based and has a graphical interface that is installed on a virtual machine, that controls access through an OpenVPN client. This user interface connects to different machines in different geographical locations, to retrieve data or to execute services. This set-up also allows for the framework to be completely located in a single location, within the network of one company. The knowledge base includes a web-service that accepts SPARQL queries and returns the requested data. The knowledge base is located on a physical server that connects to the virtual machine. Similarly the work breakdown module is an application that is running on a different machine and exposing a web-service with a user interface to control this application. A similar set-up is used for some of the other modules. Yet, it is also possible to have a framework module, exposing a web-service, running on the virtual machine, such as the interface for the correlation matrix. This application is actually residing on the virtual machine and is connecting to the other web-services and the knowledge base.

5. FIRST PROTOTYPE SOFTWARE FRAMEWORK DEMONSTRATOR

This section presents the first software prototype and its functionalities applied to the two use-cases, i.e. the Pininfarina and the Fokker Aerostructures use-case.

5.1. Stage of framework implementation

The demonstrator that is presented in this section shows the user interaction in the data acquisition phase of the framework. Referring to Figure 5, the modules from the iProd flow (Figure 4) are already oper-
ational and connect to a central knowledge base, federating heterogeneous data and making it available to the user through a single system. In the background, the user-provided data is currently used in a manually constructed simulation workflow. However, in the future the workflow creation can be automated.

5.2. Pininfarina use-case

The Pininfarina use-case focuses on the design of a door for the NIDO concept vehicle. A picture of the Pininfarina NIDO is shown in Figure 6.

Pininfarina expects the iProd software framework to help in the PDP process of the NIDO by providing the following benefits:

- Capture and reuse of knowledge.
- Functional correlation matrix between product subsystems and product performance.
- Support task leader decisions, offering less risky and optimal decisions/planning.
- Innovative methodology for working in a collaborative environment.
- Speed up the PDP for managing projects, especially those related to different versions/variants of the NIDO.
- Integrate heterogeneous tools allowing to save time and cost, involving a simulation workflow and a KBE application, to support and enable MDO.

The Pininfarina KBE application (not discussed in this paper) involves:

- Automated geometry modification of door hinge under door sag test.
- Coupling of geometry and stress analyses: Catia, GenDL, Virtual.Lab, Nastran.

5.3. Fokker Aerostructures use-case

The focus of the use-case is on moveables (rudders, elevators, high lift devices, etc.), specifically the rudder component of a business-jet type aircraft, e.g. as shown in Figure 7.

For these types of rudders, different structural concepts can be considered. Where the most important performance indicators are price and weight, with the challenge of finding the optimal solution. For the first prototype, only a thermoplastic multi-rib construction is considered.

Figure 7 Business-jet Rudder Component on a Gulfstream G650 aircraft. [12]

FAE expects the iProd software framework to help in the PDP process of the rudder design by providing the following benefits:

- Provide a knowledge base for knowledge retention.
- Combine heterogeneous software tools into a multidisciplinary, standardized workflow.
- Allow for more design iterations and improve design space exploration to increase product quality, by supporting and enabling MDO including a KBE application (as explained by, e.g., Van Dijk et al. [7]).
- Provide flexible, optimized task planning.
- Speed up the proposal phase from around 12 Man Months (average) to several weeks.
- Achieve a lead-time reduction.

The Fokker Aerostructures KBE application (not discussed in this paper, for more information consult Van Dijk et al. [7]) involves:

- Fully automated sizing of wingbox structures.
- Coupling of geometry, stress, weight and price analyses. Through coupling of GenDL application for geometry modifications, FAE in-house Excel sheets and FEM analysis in Patran/Nastran (incl. automated meshing).

5.4. First prototype demonstrator functionalities

This section shows the functionalities of the iProd framework through screen-shots from the actual web-interface. The log-in screen of the iProd framework first prototype requests the log-in of a specific user. The framework supports different user roles. Not only are there different users, to show data specific to one company, it is also possible to have users with different rights. Certain users can only view data, whereas others may edit and others may even add, delete or restructure data. Next to this, it is also possible to select

APPLICATION CASE OF THE IPROD SOFTWARE FRAMEWORK 455
which data would be visible to whom, within the same company.

This user, in this prototype either Fokker or Pininfarina, determines which data is made accessible from the knowledge base, i.e. Fokker Aerostructures data or Pininfarina data. The knowledge base queries “request” data from specific “graphs” within one knowledge base. These graphs contain all triples, structured according to the ontologies, related to a specific use-case and, additionally, there is a graph for the domain independent data, as explained in sub-Section 3.2. The user of the framework would be a knowledge engineer, to update the product and process knowledge or a project task leader, to gain insight into the development of a product in his or her project.

The first step in the iProd framework considers the project set-up. This is illustrated in Figure 8, where the family selection is shown. Here, the user, in this case from Fokker Aerostructures, can select the family of products to which the project belongs, he or she wants to work on.

The next pane, also shown in Figure 8, shows the projects that are currently in the knowledge base. Additionally, new projects can be created from a template project. Also new families can be stored, to which legacy data from different sources, e.g. requirements management software, spreadsheets or PLM system, using the LDC.

The next step is the definition of the product structure as can be done in the pane shown in Figure 9. Here systems (S) can be detailed and broken down into several levels of sub-systems (SS), functional groups (GF) and components (C). The template product structure was defined in the use-case ontology and defines a standard structure for a Pininfarina door. This product structure has been queried from the knowledge base and can now be adapted to suit the specific NIDO door.

Once a project is selected, the requirements tree for this project can be loaded from the knowledge base, as shown in Figure 10. This requirements tree can be further detailed in this pane. The inset in Figure 10 illustrates the option to specify target details for a specific requirement. The requirements tree allows to categorize requirements in different sub-levels. This way, general, non-specific requirements can be detailed to a technical level, including specific technical performance requirements. The current version allows three levels of top level requirements (TLR) and a level with product technical specifications (PTS).
Figure 10  Pininfarina requirements tree, with inset for editing the target of a technical specification

Now that the product structure and the requirements tree have both been defined for the NIDO door, a Correlation Matrix can be made, to relate product structure elements and requirements and define the level of correlation. This is illustrated in Figure 11 and the figure also shows the color coding that specifies the level of correlation. In this case all correlations are of the level high, yet also the levels none, low and medium can be used and these are identifiable by a different shade of blue in the correlation matrix view.

The Correlation Matrix can be used to directly relate requirements to product structure elements and specific tests. In doing so, one can quickly assess which tests need to be performed in case a product structure element is changed, or which test and which product structure elements are concerned when a requirement is changed. Imagine exchanging part of the Leading-Edge Box (LE-Box) for a different type (e.g. different material). In the correlation matrix, one can see which requirements may be concerned when this is done. This is illustrated for a part of the Correlation Matrix in Figure 11. Additionally, from these requirements, the related tests can be quickly identified and a development engineer may decided upon which tests are to be performed again in order to verify compliance with the requirements. Another perspective of the correlation matrix is addressed once a requirement is changed. Now, the associated tests and product structure elements can be directly identified and a decision towards actions can be taken. Figure 11 shows an inset with the specification pane of this view for a certain correlation node, where also the test and target can be specified, next to the level of correlation. Selecting these tests results in a test list. This test list can be submitted to the work breakdown module. An example of the work breakdown module as used included in the iProd framework is shown in Figure 12 for the Pininfarina use-case. The work breakdown module is based on KE-chain [13], a web-based workflow platform under development by KE-works. KE-chain efficiently manages and defines a series of engineering tasks within an organization.

Figure 12 shows an overview of the tasks to be performed during the design of the car door. The tasks can be detailed according to the inset shown in Figure 12. This is the task plan that is stored in the knowledge base, which can now be altered. Subsequently, the task plan can be optimized in the task planning optimization module of the frame work. The tasks with associated duration, required resources and the available resources are passed to an algorithm that minimizes the total duration of the task plan and delivers the optimized task plan.

The work breakdown module can also visualize a flow through the different tasks to be performed and will in the future be extended to a Gantt-chart view. The op-
Figure 11  Fokker Aerostructure Correlation Matrix Extract, with inset for test specifications of a correlation node.

Figure 12  Pininfarina work breakdown module task overview, with inset for task details in KE-chain [13]
timized task plan can then again be loaded in the work breakdown module for visualization. The work breakdown module can also couple to a simulation workflow manager, as described by Nelissen et al. [14]. This way, for example an optimization of a car door by means of a KBE application can be triggered through the workflow.

5.5. Prototype evaluation

The prototype can be evaluated against the expectations of the end-users, both end-users expectations involve knowledge capture and retention for future reuse. A knowledge base that provides a single source of truth has been implemented and the knowledge can be captured through a single, innovative and interactive interface. This knowledge is structured according to the ontology structure shown in Figure 3. Additionally, knowledge from different sources can be connected using the LDC.

Product and design knowledge is captured by the framework when the framework is used. When a project is being edited and subsequently stored, the changes are reflected in the knowledge base. These projects can then serve as a template for future projects, hence reusing information from a previous project. This also ensures a human interaction, as a product design cannot be made automatically, but requires human input.

For both use-cases, KBE applications are being developed that automate sizing of the products, based on inputs from the engineer. The simulation workflows are already capable of automatically performing geometry modifications and integrate heterogeneous tools. These workflows and KBE applications also allow for more design iterations to be performed, and support the human developers by providing the possibility to speed up the PDP for both the rudder and the door in the future. The actual lead time reduction has yet to be validated and quantified. Additionally, for both use-case a functional correlation matrix between product sub-systems and product performance has been implemented. The flexibility of the task planning optimization is still to be improved, such that it can also support task leader decision making.

The methodology that will be used for iProd software evaluation is based on the following steps:

- Definition of end-users stories according to the standard structure.
- Definition of Key Performance Indicators (KPIs) for story evaluation.
- Definition of report sheets where bugs, new functionalities and comments are reported.

The validation process considers several stages. First, the stories which are divided into phases and steps, are executed by end users and are evaluated using specific sheets. The evaluations are collected, and passed to the development team (KPI evaluations, bug reports and request for new functionalities to be implemented). The development team will then correct problems in case of low values of KPIs, fix the bugs reported and set a priority list of new features to implement. Once this step is completed, a new release is delivered to the release manager, who will distribute it to end users to start a new validation test iteration. This process is iterated three times in iProd.

End users will evaluate the iProd platform following the story line which has been specified. After each test, the user will evaluate the behavior or his opinion of the platform, using a list of parameters described with the use case. During the test of the platform, all iProd functionalities will be checked and therefore iProd services are tested accordingly. Starting from the evaluation of the stories, a wide list of KPIs is derived and grouped in categories, in order to understand the levels of functionalities, reliability, usability, efficiency and portability of the entire platform.

The aim of the iProd platform is to reach about 75-85% of positive attitude expressed by the end users. After evaluating the first prototype, an average of 3 was scored for the two use-cases. Slightly higher for Pininfarina and slightly lower for Fokker Aerostructures. The functionalities and usability of the prototype are appreciated. However, in terms of reliability, efficiency and maintainability the prototype had to be significantly improved. These issues will be addressed in the next to versions of the software framework.

6. CONCLUSIONS AND FUTURE WORK

Using the first iProd software prototype, a PDP and its data can be managed from project set-up to execution of the virtual simulation. The application covers the product and requirements definition, specification of tests and tasks and the entire task flow throughout the PDP in one single environment. It has been shown that the prototype is already meeting several of end-user requirements for the use-cases that were presented in this paper. The potential of reducing lead time and increasing product quality and maturity, in addition to allowing for more design iterations is indicated. However, this is yet to be validated. Validation and testing has to
be performed by comparing the design process using iProd to the current design processes, for the use-cases considered in the project. This will be performed with end-users in the user group of the project, in collaboration with the software developers.

Future development will include a second and third prototype, improving and extending the framework modules to meet the end-user expectations. Thess will focus on updating and refining current functionalities, extension of the coupling between different modules and the coupling to the knowledge base, as well as the federation of heterogeneous data from different sources. More specifically, an update of all ontologies, to incorporate version management and STEP standards for view definitions and breakdown management and connectors to DOORS, Excel and TeamCenter. After the project end some partners consider the commercialization and further development of the software.

ACKNOWLEDGMENT

The research leading to these results within the iProd project has received funding from the European Community Seventh Framework Programme FP7/2007-2013 under grant agreement number 257657. The authors acknowledge all iProd consortium and user-group members for their support and work on the iProd project, and in particular the developers involved in the implementation of the software framework, without whose efforts this paper could not have been created.

References


