SYSTEM INFRASTRUCTURE FOR INTEGRATED AND COLLABORATIVE PRODUCT DEVELOPMENT IN THE AERONAUTIC INDUSTRY

T. Nguyen Van*, P.L. Frossard*, P. Guellec, F.Féru*****
***Snecma, **Airbus CIMPA, ***EADS**

Keywords: **Extended Enterprise, Collaborative Environments, Data Management**

Abstract

In the development of partnership and sub-contracting activities, collaborative design and data management processes are crucial for faster and better product development. Collaborative environments and integrated design are now closely tied to product development activities. These practices are used to combine the strength, expertise and know-how of the best diverse, geographically dispersed technical teams in order to achieve better mission scenarios and designs, and to develop corresponding technologies more quickly [6].

This paper deals with the work done during the 6th European Union framework project called VIVACE (Value Improvement through a Virtual Aeronautical Collaborative Enterprise) which involves most of the large aeronautic enterprises in Europe. The work performed by the Engineering Data Management (EDM) Work Package of this project was motivated by the fact that data is still not well integrated and interoperable both within and between partners' systems. This paper defines an interoperable environment and the engineering context for data use. The contribution is based on a methodological framework and on the implementation of a collaborative platform for multi-partners and multi-engineering.

1 General introduction

The new paradigm of collaborative product development or collaborative enterprise [11, 14] can have significant implications for product development. These new collaborative manners of developing technologies and aeronautic systems create changes in the ways aeronautic systems are designed, produced, operated, maintained, and disposed of. Collaborative engineering is seen as the application of team-collaboration practices to an organisation's total product development efforts [3]. But, industrial collaboration suffers from the heterogeneity and diversity of software and the data exchanged, and their management between partners and alongside activities. Collaborative product development is slowed down by the lack of interoperability and integration of partners' systems [5]. Indeed, the communication efficiency and interpretability between systems must be improved. Thus, there is a need for an integrated reference frame allowing better control of the aforementioned properties. An intermediary platform ("middleware") to support collaboration is necessary.

There are many issues with regard to the support of collaborative efforts to define collaborative contexts between activities and partners. Indeed, collaborative methods and rules are an important aspect in identifying the characteristics of "integrated product development" [15]. Major research streams can be pointed out, such as:

- "Architectural Laws": transactions and interaction between systems and components of the collaborative architecture [10].
- Data dissemination strategy: to establish structural models (components and framework) for the migration and integration of data in heterogeneous systems [2].
- Data reference frame: to define the structure of the integrated environment and collaborative "objects". This also considers

the association of heterogeneous environments and sustains the definition of the migration between them [19].

In this collection of heterogeneous tools and methods for managing the multi-partners' data and environment, EDM is presented as a collaborative framework based on the following major requirements:

- A non-invasive framework: to preserve the "in-house" environment running with partners' processes. The target is to provide an interoperable "middleware" that integrates existing tools and methods.
- Standard-based communication: EDM targets domain interoperability using a semantic reference based on standards.
- Services to provide information in context: EDM proposes the definition of an Information Model providing the context of use and a domain model characterising the data involved in an activity.

This paper describes the necessary components to be linked for efficient communication. The definition of a multi-layer architecture is the basis for the progressive differentiation of data, built upon the notion of context. We then present the context notion defined for the aeronautic industry. Lastly, we propose to illustrate this architecture with a conventional use case for a design/simulation loop. This scenario serves as a basis for evaluating the efficient application of the standardisation of collaboration in industry, and especially the use of STEP (STandard for the Exchange of Product data) [7, 8].

2 Collaborative and integrated environment

Collaboration between two distinct engineering systems and/or sub-systems is the practice of combining functions and characteristics of a set of both systems and sub-systems so as to produce a single unified system that satisfies some overall need of an organisation [4]. Collaboration is the main way to allow team members to manage processes, data exchange and consistency from the early stages of a project. The collaboration then consists in analysing and coordinating business entities to ensure the consistency of the system.

The objective of defining an integrated environment is to provide an integrated view of the product – namely, a product reference frame for the partners of a project. In such an approach, the "collaborative view" is no longer owned by one partner but rather by all the partners. Constant refinement of the environment is essential to the improvement of collaboration. Teams are no longer working alone on the design of a module but rather designing it within the whole product environment [5, 16]. The integrated environment is also a means of integrating processes and activities. To ensure concurrent work on clearly defined project tasks, it is necessary to organise groups of designers who share a set of common design items. Their work should automatically be coordinated to avoid redundancies and inconsistencies.

2.1 Definition of collaboration typology

By analysing collaboration using 3D representations, Li [11] has characterised the collaborative environment adopting a two approach description: horizontal and hierarchical collaboration. This typology can be extended to our case.

2.1.1 "Horizontal" collaboration

This corresponds to the exchange and integration of data between equivalent tools. Performed at the same process level and for the same activity, this collaboration is based on the exchange of "development packages" that are integrated by partners and teams to rebuild the common "product environment". This collaboration is performed in two ways:

- By the use of translators: to transform data output from one system into data input for another. This method then requires N^2-N translators if we consider collaboration between N partners' tools [17].
- By the use of a standardised file: partners' data are pre-processed into standardised data. After the exchange, data then only needs to be post-processed to be integrated [20].

Reusing the previous example, this method requires only 2N translators.

2.1.2 "Hierarchical" collaboration

Nowadays, data management principles belong to design activities; collaboration using different tools and attribute typology is not well developed. This tends to change with the appearance of simulation data management systems. As digital simulation is closely tied to design, there is a need to integrate the data structures. Particular descriptions and attributes (e.g. simulation parameters) are still not integrated. Within this collaboration, the integration principles of horizontal collaboration remain applicable [13].

2.3 Targets of a collaborative platform

A collaborative platform now appears as a necessary element to allow data flows between applications used in companies. Ideally, its role would be to integrate and associate data created and managed during design and engineering activities. Considering data management systems (DMS), integration and association means that all partners' data can be merged to define the whole product. Indeed, considering the DMS, all engineering definitions and attributes of the product can be integrated and used in several systems. As the number of attributes linked to the product is also increasing, with the evolution of data management requirements and data exchanges, partners need more complete interfaces. As a consequence, there is also an increasing need to provide management for the data processed (e.g. managed, integrated…) by these systems.

This contributes to the definition of integrated infrastructures for the design of collaborative environments and multi-view (domains) application [5, 11]. The objective of defining an integrated infrastructure is to provide a conventional, efficient DMS, but extend it for collaboration purposes. Integrated infrastructure must also provide shared spaces for the partners, using dedicated workspaces for the different activities. Also, a collaborative platform must allow tool interoperability and data integration in the most meaningful way. The platform must provide services to create requests on activities and to define workflows between enterprises.

Collaborative infrastructures must create networked activities using product data representation (reference framework) provided by the collaborative workspace [15]. For the aeronautic industry, an integrated infrastructure must comply with 6 major requirements:

- Common data reference framework: to store and retrieve product data throughout its entire lifecycle. This also considers the consistency of data (product's characteristics).
- Management of information between partners: using standardised collaborative data, this defines the meaningful way to manage data. Models describe the attributes attached to the product and must manage the attribute links between partners' systems.
- PLM functions: define the major actions on data and guarantee management throughout the lifecycle.
- Data context has to allow retrieval of the context for data use (e.g. process, activity, data, parameters…).
- Providing data associativity has to define links between data created for activities.
- Definition of flows and process connections (engineering requests, engineering validation…): determine the different ways to access partners' processes.

2.3.1 Collaboration and development process

To achieve its definition, a product is refined throughout many activities. As aeronautics products are complex and developed in large partnerships, the number of persons involved in the product definition increases at each level of the lifecycle. Generally using processes specific to a single enterprise, collaboration lacks synchronisation between established and rigid individual processes. The harmonisation of processes is a priority step in defining new ways of developing a product between many partners. Thus, the main issues in collaborative processes can be defined as follows:

- Activity scheduling and synchronisation
- Workflow control

- Enabling concurrent engineering and collaborative design

2.3.2 Collaboration and tool synchronization

In the extended enterprise context, working on the same final product definition using many different tools is a complex issue. Even within a company, a product definition is processed through many environments. All these activities do not deal with the same data or methods. Hence it is important to synchronise everyone's work in order to optimise the product definition. Indeed, the number of tools and methods used tends to be infinite, and a multitude of heterogeneous tools and environments increases the difficulty of exchanging data between applications. The following issues arise from this project organisation and management:

- Standardisation of the product structure
- Definition of the information to be shared with partners
- Synchronisation of partners' product structure

3 Technological guidelines and drivers

3.1 Approaches to "designing" a platform

While collaborative environments were limited to design environments a few years ago, their use is now extended to the overall processes identified in the enterprise and to the different stages of a project [5, 11].

These collaborative environments provide mechanisms for distributed and integrated product development and engineering. Such mechanisms rely on the association of two major structural approaches:

- A static approach: to associate the static elements of the product definition (e.g. Id, name, maturity, version, exchange date…).
- A dynamic approach: relevant for a process/workflow definition. It defines the activity sequences that describe the actions relating to data.

Each environment is submitted to its own functioning laws. The objective is to create networked activities using product data representation, and above all, a reference frame for the product in the collaborative workspace [15].

3.2 A need for layered architecture

Regarding the need for the framework architecture, we noticed there were systems interacting with each other and with the collaborative framework. This led us to define a model for architecture layers.

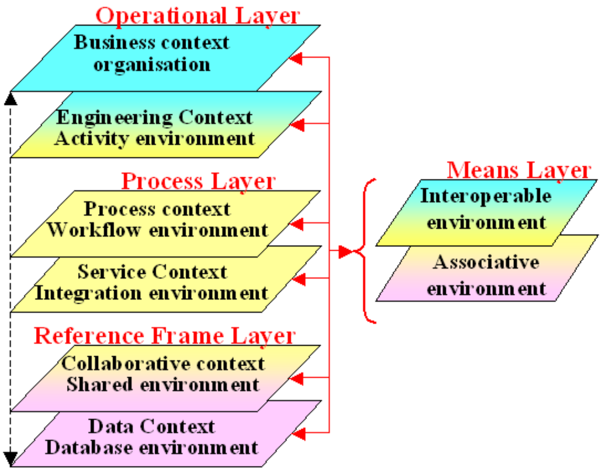

Fig. 1. Industrial system layers

Figure 1 represents the results and shows the integration of the layers in the industrial system.

- The operational layer corresponds to the "end-user" environment in which activities are performed. Composed of operational tools, this layer is grouped and included in a rigid infrastructure and possesses its own processes and methods.
- The process layer is the guarantor of activity changes. Composed mainly of workflow tools, this layer is the backward environment that provides the operational layer with the instructions for activity sequences.
- The means layer interprets the environment in which individual activities and actions are performed. Composed of information systems and translators, it allows for the design and redesign of environments and application domains.
- The reference frame layer is the common representation and interpretation of the

phenomenon that occur during a project. Providing a unified view of the product for activities and partners, it is acknowledged as the shared working environment.

4 The EDM framework

4.1 The engineering "context definition"

"*Context of Something consists of the ideas, situations, events, or information that relate to it and make it possible to understand fully*"[1]. More generally, context is not an object on its own. Context is rather a snapshot of relations between objects that are relevant to each other and their interactions. Context is therefore an unstable concept that relies on objects and interactions between them.

We can identify the important and relevant objects in order to set the engineering context in a collaborative process. As collaboration begins with a project, it appears essential to consider this, and we identified mandatory objects to define the engineering context. The product and its related process are essential in defining the engineering activity, and the role is fundamental in defining the context. Indeed, the person who has the right skills and knowledge performs a specific activity. Directly linked to the role, the user is the real performer of the activity. The relationships between users and roles must be mapped. Then, two final objects are identified to clearly define an engineering activity. We consider the tool with which the activity has been performed. Finally, we consider the resources the performer may use during his work. Resources may be of any type (specific data translator, physical resources and so on…). Figure 2 shows the seven identified objects that make up our contextual approach of engineering in collaboration. The dashed line represents the collaborative context and the dotted line the business context from the context typology. Links between the objects are explained below:

- Project-Context: this is the top level of the structure. The product is created within a project and a process is run on a product within this project;
- Product-Context: represents the product and its different stages (design, simulation…). It is managed by a project, and modification and manipulation are possible only through processes;
- Process-Context: defines the possible applicable activities on a product regarding a project and its progression. The process is also linked to a tool, a role and resource;
- Tool-Context: this element represents the gate that allows access to a role of a product for a specific process in a project;
- Resource-Context: represents any other systems, information, data type, Data Exchange Set (DEX) [12] or product that is crucial to the performance of a process. This is related to any other object of the context. It is a package that contains all the relevant information for the corresponding context;
- Role-Context: this element of the context is also the key to the activities applicable to the product within a project and open to a specific user. It represents the link between the organisation of the extended enterprise and the organisation of a project;
- User-Context: this final element is also the one that establishes the navigation through the context. Indeed, by connecting onto a platform, the user chooses his role in order to access the projects, the processes, the products and so on. Users are classified according to the organisational hierarchy.

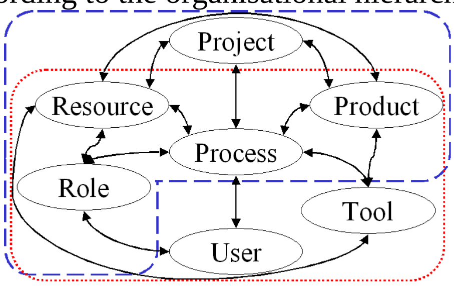

Fig. 2. Context Systemic Breakdown

4.2 EDM framework "blocks"

Regarding the European Union Project VIVACE context, the EDM framework has been developed.

This EDM relies on 6 main entities (see Figure 3):

- Applications (Apps): operational tools that are used to perform specific engineering activities.
- Information Model (IM): an object broker which is the basis for the EDM Server and that contextually defines the product, process, resources and links between them.
- Domain Models (DM): represent the attributes or data for each domain of expertise. Characterise a "product view".
- Consolidated Repository (CR): provides the persistency for the data of the virtual aircraft. It does not replace the repository of legacy application but provides a shared environment for the published data based on the Standard for The Exchange of Product model data (STEP) (Standards ISO TC184-SC4) [7, 8].
- Product Context Management (PCM): the collaborative interface that allows users to browse collaborative data and processes.
- Workflow (Wf): it ensures correct performance of activities regarding a product in a process and including team members and resources. This allows relevant monitoring of the product throughout its entire definition phase.

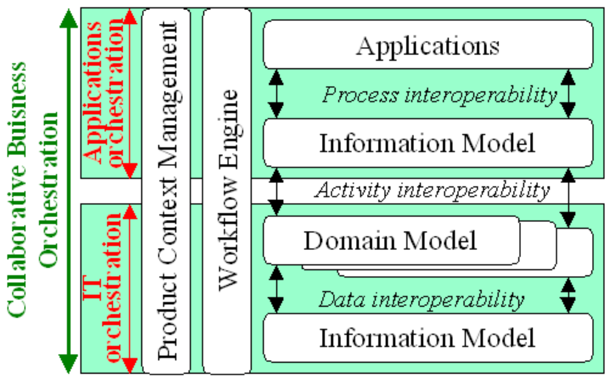

Fig. 3. EDM Frameworks

4.3 Functional overview of the framework

Let's take the example of a data export in order to identify the data flow (see Figure 4). The design engineer receives a change request in the PCM (1) and launches the Workflow (2) to carry out a design change. Once done, he saves the data both in his environment and in the collaborative environment (3). The Information Model breaks the native data attributes in order to map them to the standardised common definition (4). The mapping consists in associating native attributes and standardised attributes used in PLCS (Product LifeCycle Support – STEP ISO10303-239 [9]). As this mapping ensures the associativity between data through Domain Models, the data is changed in both simulation (Sim.) and design (Des.) Domain Models (5 & 5'). For each Domain Model, the product definition is rebuilt to fit the common definition (6 & 6'). The last activity is then the storage of data within the collaborative consolidated repository (7 & 7').

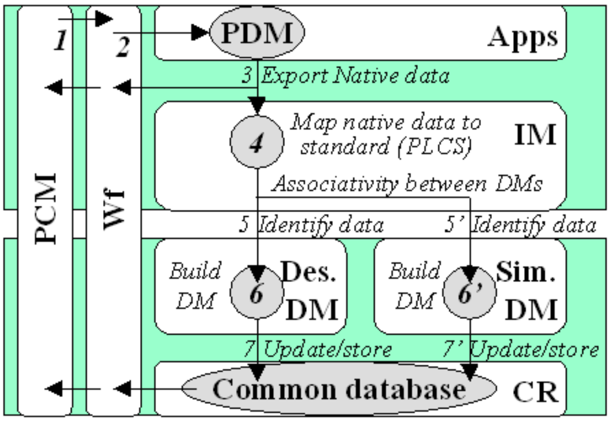

Fig. 4. Data export Use case

4.4 Definition of the data structure mapping – Domain model mapping

Based on a common database, commercial applications may rebuild their own environment or update the EDM environment. To do so, Web-Services (symbolised by arrows) are used to send or retrieve information based on the XML (eXtended Mark-up Language) standard.

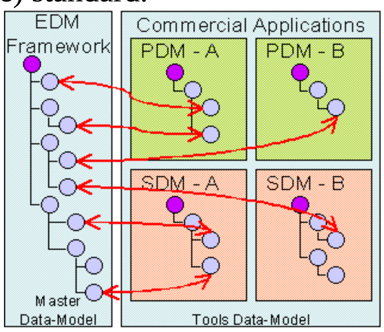

Fig. 5. Data mapping

This technology allows for communication between applications and ensures coherency of data structures by defining the mapping between EDM and commercial Applications. Hence the tool Data Models are different and specific to the COTS they are for. Those specific Data Models can be mapped and then instantiate a Master Data Model (see Figure 5).

4.5 EDM Graphical User Interface (GUI) for control – "Implemented" Information model

The collaborative application is called Product Context Management (PCM) and is composed of seven interfaces:

- User Workspace: lists user's tasks;
- Execution View: displays task details;
- Project Overview: displays workflow overview;
- Workflow Management: manages the workflow; is composed of three interfaces;
- Workflow Administration: manages processes and activities;
- User and Groups Management: manages users and groups;
- Mapping Management: manages mapping between users, groups and processes roles;
- Context Setting: sets the navigation context.

This interface represents the implementation of the Information Model we have created to manage the overall engineering contexts and the collaborative environment. The top layer of the interface is common to each interface. It is composed of tabs corresponding to the application interfaces. It displays details on user navigation: the navigation context. The navigation context defines the user name, the activity, the product, the workflow (activity) and the role of the user. If this navigation context is not set, the execution view and project overview are not available.

4.5.1 User Workspace

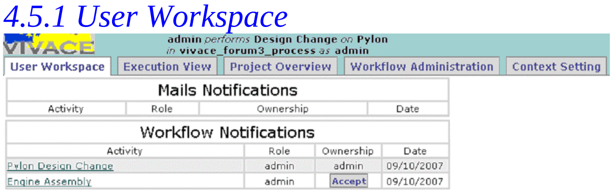

Fig. 6. User Workspace interface

This contains information about the activities the user has been assigned (Figure 6). Here, the user may switch off the interface by using the available tabs. The activity names are also a link. This link sets the user navigation context and directs the user to the execution view to see the activity details. If an activity is assigned to a role and not yet accepted, it appears in the workspace of every user that is linked to the role. Once the activity is accepted, it only appears in the workspace of the user that accepted it. Actions that are completed or not yet activated are not displayed.

4.5.2 Execution View

This interface (Figure 7) informs the user of the tasks he has to perform. With the Navigation Context, the Execution View displays all the activity details. Two frames display activity inputs and outputs. In each frame, tabs correspond to the number of inputs or outputs. Inputs and outputs depend on the workflow. Each input contains requirements, information and data that are relevant for the task. The output tabs have to be filled in by the user when he performs the activity. He creates outputs with data, information and requirements he has obtained during his task and that are relevant for the next activities.

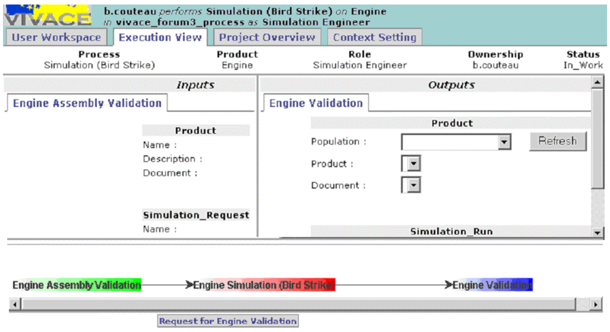

Fig. 7. Execution View interface

A short view of the workflow to which the activity belongs is displayed in the outputs/inputs frames. Indeed, the current activity is displayed with the previous activities on its left and the next activities on its right. The numbers of outputs/inputs tabs are equal to the numbers of previous/following activities. Finally, the bottommost line contains the activities' validation buttons. It corresponds to the output paths available for this activity.

4.5.3 Project Overview

This interface (Figure 8) enables the user to view the overall workflow. The displayed workflow corresponds to the workflow that is set in the Navigation Context. It is composed of two main frames: the workflow view and the activity details table.

Based on JAVA technologies, the workflow viewer enables dynamic navigation on the workflow. Clicking on an activity automatically displays its details in the activity details table. The activity details table contains relevant information on the activity. The product and the process on which the activity is based are displayed. A list of instantiated activities shows if the activity is currently running, accepted, not yet started or already completed. Clicking on the ID of an instantiated activity directs the user to the execution view to see the details. If the activity is completed, the execution view displays inputs and outputs that have been filled in, if the user is not the owner of the activity.

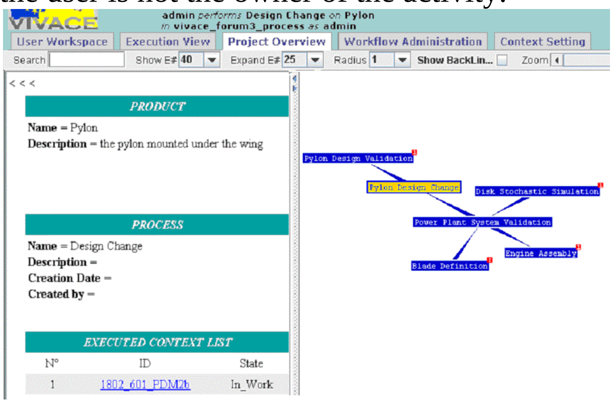

Fig. 8. Project Overview interface

4.5.4 Process Management

This interface (Figure 9) enables one to load workflow definitions from XPDL files, instantiate one or more of these workflow definitions and manage the running activities.

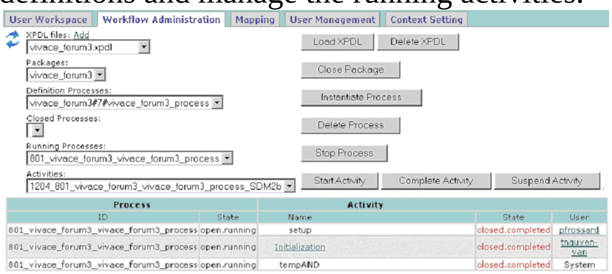

Fig. 9. Process Management interface

This interface is directly linked to the workflow engine. It displays the list of all activities with their status. The user also has access to the closed workflows to clear the workflow engine database. The activities table displays the name of the activity, the workflow to which it belongs, the status and the assigned user of the activity. Clicking on the activity name sets the navigation context and directs the user to the execution view to visualise the activity details.

4.5.5 Context Setting

In this interface (Figure 10), users browse the existing engineering tasks and their status. This consists in a query into the collaborative database by filling in some fields (project, process, product, role and status). The result is a list of workflows that corresponds to the query. From this list of workflows, all activities are available with their characteristics.

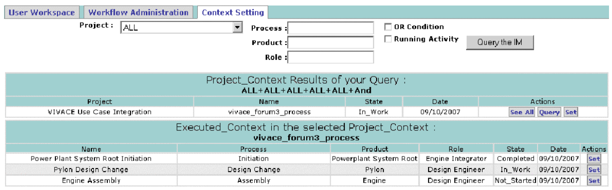

Fig. 10. Context Setting interface

5 Scenario description and test case run

5.1 Scenario overview

This scenario is a close up on the operational part of the global workflow. Before building it, a first assumption is made on the industrial purpose it can have. Let's take two technical components P and E that are assembled together into a product named A (P+E=A). Each component is a product from two different companies B and C. The company B in charge of the component P is also in charge of the integration between both components (A). Each company uses its own tools regarding activities performed on each component. The workflow must be simple enough to be seen at a program level and detailed enough for the operational manager (Figure 11). The grey boxes represent sub-workflows, in which the first activity is carried out through another application. This is where the workflow is controlled from an external application, while each company manager from the main interface pilots the other activities. Furthermore, this scenario takes into account two domains, i.e. the design and the calculation for each component.

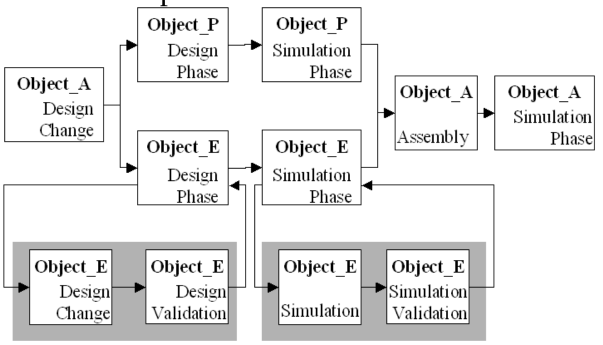

Fig. 11. Proposed Scenario

5.2. Scenario Implementation

To implement this scenario in the EDM Framework, the Information Model helps map the basic workflow concepts “As Planned” and “As Executed”. This implementation relies on the three following objects (see Figure 12):

- Context Definition: composed of a process applied to product, this is the basis element for the workflow implementation.
- Project Context Definition: instantiation of Context Definition and linked between each other, they represent the “As Planned” workflow.
- Executed Context Definition: sequence of instantiated Project Context Definition, they represent the “As Executed” workflow.

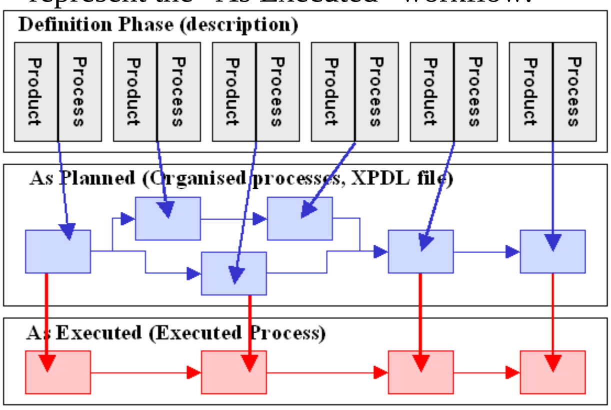

Fig. 12. Scenario Implementation

To manage all the applications together, the workflow engine has different behaviour regarding the current activity. As shown in Figure 13, the monitoring and controlling of a workflow can be represented as this activity’s sequence. A workflow is first instantiated within the workflow engine and is launched automatically (1). Corresponding to the first activity, a new task is sent to the manager. By accepting it, the manager launches the activity (2). Through the PCM interface, the manager creates a simulation request by gathering product and process information from the information model. When the simulation request is filled out, the manager completes the activity (3). Automatically, a new activity is instantiated (4) and the simulation request is sent to the SDM environment (5). The SE begins with retrieving the product structure. At the end of this activity, a status is sent to the high level workflow (6). This status automatically updates the activity progress (6’). This allows the manager to follow the progress of a lower level activity. The status transfer between both applications is based on a JAVA client. Then the SE begins to simulate the component’s behaviour. When the SE estimates that the simulation loop is over, a new status is sent to update the workflow engine (7 & 7’). In the next activity, the SE produces the simulation results report that will complete the simulation process (8). The validation of those results is then activated (9).

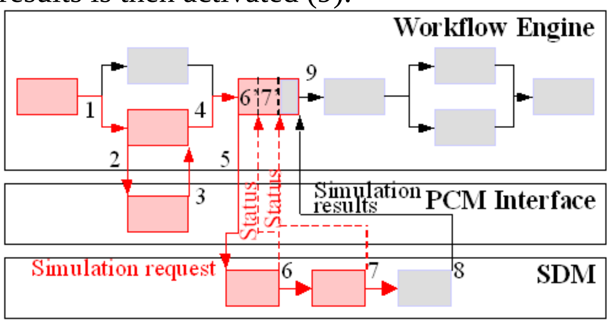

Fig. 13. Scenario Use Case

6. Conclusion

The communication between partners and activities in industry suffers from the communication between tools and systems and with regard to the contextual definition for processes and engineering environments. In the preliminary stages of projects, the interoperability and integration of engineering systems are essential in setting up an efficient partnership and reducing the time spent on conceptual stages. Here, we have addressed the following issues for the aeronautic industry:

- The increase of communication and semantic problems.
- The rationalisation of collaborative processes using multiple environments and product references.

Regarding these aspects, the contribution of EDM within the VIVACE Project corresponds to the major constraints presented in industry by:

- Increasing interoperability to support the definition and interpretation of information through domains and between:
  - Engineering systems.
  - Product and Processes.
  - Engineering data.
- Setting up an integrated environment to support the reference frame of the product:
  - The development of shared workspaces.
  - The definition of reference models using the Digital Mock Up.
- Defining engineering contexts.

References

[1] Akman, V. and M. Surav (1996). *"Steps toward formalizing a context"*

[2] Aziz, H., J. Gao, et al. (2005). *"Open standard, open source and peer-to-peer tools and methods for collaborative product development."* Computers in Industry 56(3): 260-271.

[3] Beckett, R. C. (2003). *"Determining the anatomy of business systems for a virtual enterprise."* Computers in Industry 51(2): 127-138.

[4] Dustdar, S. and H. Gall (2003). *"Architectural concerns in distributed and mobile collaborative systems."* Journal of Systems Architecture 49(10-11): 457-473.

[5] Fuh, J. Y. H. and W. D. Li (2005). *"Advances in collaborative CAD: the-state-of-the art."* Computer-Aided Design 37(5): 571-581.

[6] Hassan, T. M. and R. McCaffer (2002). *"Vision of the large scale engineering construction industry in Europe."* Automation in Construction 11(4): 421-437.

[7] ISO 10303-1 IS 1994 - *« Overview and fundamental principles »*

[8] ISO TC184/SC4/WG10 N326 - 2000-11-08 – PROPOSED ISO TC184/SC4 Standing Document - Technical Committee 184 for Industrial Automation Systems and Integration - Subcommittee 4 for Industrial Data

[9] ISO/DIS 10303-239:2004. *"Industrial automation systems and integration — Product data representation and exchange — Part 239: Application protocol: Product life cycle support"*

[10] Lehmann, H. and B. Gallupe (2005). *"Information systems for multinational enterprises--some factors at work in their design and implementation."* Journal of International Management 11(2): 163-186.

[11] Li, W. D., W. F. Lu, et al. (2005). *"Collaborative computer-aided design--research and development status."* Computer-Aided Design 37(9): 931-940.

[12] OASIS - Advanced E-Business Standard, http://www.oasis-open.org/committees/plcs/faq.php

[13] Oh, Y., S.-h. Han, et al. (2001). *"Mapping product structures between CAD and PDM systems using UML."* Computer-Aided Design 33(7): 521-529.

[14] Pardessus, T. (2001). *"The multi-site extended enterprise concept in the aeronautical industry."* Air & Space Europe 3(3-4): 46-48.

[15] Perrin, O. and C. Godart (2004). *"A model to support collaborative work in virtual enterprises."* Data & Knowledge Engineering 50(1): 63-86.

[16] Rosenman, M. A. and J. S. Gero (1999). *"Purpose and function in a collaborative CAD environment."* Reliability Engineering & System Safety 64(2): 167-179.

[17] Ruland, D. and T. Spindler (1995). *"Integration of product and design data using a metadata- and a rule-based approach."* Computer Integrated Manufacturing Systems 8(3): 211-221.

[18] Simon, H.A. (1969) *The sciences of the artificial*, MIT Press

[19] Vernadat, F. B. (2002). "*Enterprise modeling and integration (EMI): Current status and research perspectives."* Annual Reviews in Control 26(1): 15-25.

[20] Zha, X. F. and H. Du (2002). *"A PDES/STEP-based model and system for concurrent integrated design and assembly planning."* Computer-Aided Design 34(14): 1087-1110.

Copyright Statement