INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

DEFINITION OF AN ENGINEERING DATA MANAGEMENT FOR COLLABORATIVE PRODUCT DEVELOPMENT

T.Nguyen Van[1 2], P.Guellec[3], F.Féru[3], B. Maillé[2] and B. Yannou[1]

[1]Ecole Centrale Paris, Laboratoire Génie Industriel
[2]Snecma – SAFRAN Group
[3]EADS – Common Research Center (CRC)

ABSTRACT
Within 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 deeply linked with product development activities. These practices are used to combine the strength, expertise and know-how of the best diverse, geographically dispersed technical teams to reach better mission scenarios, designs, and to develop corresponding technologies in less time [16].
This paper deals with the work done within the 6th framework project called VIVACE (Value Improvement through a Virtual Aeronautical Collaborative Enterprise) which involves most of the large aeronautic enterprises in Europe. The approach used by the Engineering Data Management Work Package of this project is motivated by the fact that data is not still well integrated and interoperable between partners' systems. This paper focuses on the definition of interoperable environments and definition of context for use. In this paper, we develop the concepts reached during the VIVACE project. We present first a collaborative typology that is helpful to better understand constraints while partners attempt to collaborate. We then expose our guidelines for collaborative environment development by using the definition of a multi-layered architecture. We finally develop the components of this architecture and propose the definition of a scenario based on design/simulation loop.

Keywords: Extended Enterprise, Collaborative Environments, Data Management

1 INTRODUCTION

Presently, industrial collaboration suffers from the **heterogeneity and diversity** of software, data exchanges and their management between partners and along activities. Currently, collaborative development of product is limited by the **lack of interoperability and integration** of partners systems [14, 39]. It turns out that improving the communication efficiency and the interpretability between systems becomes necessary. Consequently, there is a need for an **integrated reference** for product development which would allow a better control of the aforementioned properties. The definition of an intermediary platform (kind of middleware) to support these constraints appears as a necessity to ensure better partnership.
Then, issues on the support of collaborative activities are related to the capability to **define collaborative contexts** between activities and partners. Indeed, the definition of collaborative methods and rules are also an important focal point to determine the characteristics of the integrated product development [6, 23, 30]. Three major research streams can be pointed out from our ground observation and the review of literature:

- The definition of architectural laws for collaborative structure: using the definition of transactions and interaction that exist between systems and components of the collaborative architecture [19, 20, 28].
- The definition of collaborative platform for data dissemination to elaborate structural models for the collaboration and migration between heterogeneous systems. It defines the characteristics attached to product development so as to define major components and layers to support such an architecture [2, 11, 18, 38].

- The definition of a data referential for the creation of associative environments to define how a collaborative environment is implemented and what it is composed of. This defines the structure of the integrated view based on the definition of collaborative objects. This also considers the association between heterogeneous environments and sustains the definition of the migration between them [9, 24, 40].

In the context of a collection of heterogeneous tools and methods to manage multi-partners data and environment, we propose to **define an Engineering Data Management (EDM)**. This EDM is presented as a collaborative framework based on the major requirements of:

- **Non-invasive framework**: Enterprises have already an "in-house" environment running with their processes. The objective is not to replace existing environments but, to provide an interoperable "middleware" that integrates existing tools and methods.
- A **standard based communication**: Our EDM framework targets enabling domain (activities) interoperability using a semantic reference based on standards.
- **Services** to provide information in context: Our EDM framework proposes an Information Model that provides the context of use and a domain model that determines data involved in an activity.

We then identify and define necessary components to be linked for an efficient communication. The definition of a **multi-layer architecture** is the basis for the progressive differentiation of data. This aspect of progressive determination of data relies in fact on the notion of context. We then present how the notion of **context** has been defined for the aeronautic industry. Moreover, this kind of architecture is also a requirement for the application of standardisation on data.

We finally propose to illustrate this architecture with a classical use case of design / simulation loop. This scenario is a basis to evaluate the efficient application of the standardisation for the collaboration in industry and especially the use of STEP (STandard for the Exchange of Product data – ISO 10303).

2 EXTENDED ENTERPRISE AND COLLABORATIVE ENVIRONMENTS

2.1 Typology of collaboration

Li [21] analyses the viewpoints associated to CAD activities and especially the collaboration using 3D representations. His work is based on the statement of systems heterogeneity, the use of tools and major functions that are needed in such systems. To better characterise the environment, he has adopted a collaborative description defining two approaches: the **horizontal and the hierarchical** collaboration. This characterisation can be extended in our case.

2.1.1 Horizontal collaboration

It designates the application of associating teams **from the same discipline or whose activities are located at the same process level** to carry out an aspect of the development (for example for design activities) This means enabling two collaborative teams to work on the design of the same object in synchronous or asynchronous ways [13, 34]. For example, let's consider the development of a product part on which two teams are working. These teams work to develop the same part of the product with similar tools. But modifications performed by each team and the integration of their work is made independently.

2.1.2 Hierarchical collaboration

It designates collaboration between teams **from heterogeneous disciplines**. It concerns the definition of **links and dependencies between upstream and downstream activities** applying rules and methods to migrate an environment to another. For example, this means integrating the work done from design processes to simulation processes [26, 43].

2.2 Definition of the « collaborative discriminant » for partnership

Trying to define the collaboration, many authors consider that it consists in **integrating parts of business or enabling unified methods and procedure**. Peña-Mora and Ruland [29, 35] define the collaboration as the capability to ensure the communication between business entities, whereas for Dustdar and Webster [11, 42], integration and collaboration deal with technological developments. Both approaches are necessary. Collaboration results in **merging technological and organisational** aspects.

Using the analogy with a mathematical concept, defining the relationships in business activities is supported by the **collaborative discriminant**. The collaborative discriminant is defined as the logical and physical connection of business components (e.g. activities, processes) with a collaborative environment. Indeed, retrieving this discriminant consists in defining an elementary structure (with rules, functioning laws and technical objects) that connects two defined business entities of the industrial context. If we consider two processes (for example design and simulation), the collaborative discriminant between both processes is the product definition (product structure, 3D models and attributes). Indeed, collaboration is not possible if we consider the sole product structure (that don't contain the 3D representation) or specific information (such as simulation parameters that are only relevant for simulation). Attempting to define the collaborative discriminant, we noticed that we had to consider three contributions:

- **The project discriminant:** Projects are collaborative structures locally and temporally defined. They are the backbone to group activities and teams for the product development (from conceptual stages to service). They are constrained by high level objectives that must be reached to satisfy the requirements of the client [36]. The project discriminant is characterised by high level business objects such as organisational aspects, governing rules for the development (resources, time schedule, and activities sequences) and financial aspects. Then, the discriminant is based upon the need to share responsibilities between partners. This aspect is also sustained by the willingness to share ideas, perform common developments and researches. The collaborative aspect relies on coordination and cooperation. Coordination defines the "who does what?" in the project and highlights the structural aspects. Cooperation deals with the definition of benefits while working together [8, 40].
- **The process discriminant:** Process defines an occurring or designed sequence of events or operations that are defined in a context determined by time, space, resource factors and which operates the transformation of an incoming object into an outcoming object. Adapted to an engineering view it becomes: «A connected series of actions, activities, changes etc, performed by agents with the intent of satisfying a purpose or achieving a goal» (ITIL - IT Infrastructure Library). The process discriminant defines the aggregation of numerous activities fragments, in which each fragment describes a part of the overall activity, with no overlapping between activity sets. In the collaborative industry context, each actor has a partial and overlapping view of the complete process [8]. The links between processes are ensured by the process discriminant. This discriminant is the visible part of the process which defines the functionalities. It defines how they are composed, coordinated and how they can cooperate [4].
- **The engineering discriminant:** It consists in executing an elementary activity to develop a part of the product. Engineering disciplines concern design, development, improvement, implementation, process and so on that are set up for product development. The crucial and unique task of the engineer is to identify, understand, and to integrate constraints on a design in order to produce a successful result [34]. It is usually not enough to build up a technically successful product, it must also meet further requirements. Constraints include available resources, physical or technical limitations, flexibility for future modifications and additions, and other factors, such as requirements for cost, marketability, productibility, and serviceability. Engineers derive specifications for the limits within which a viable object or system may be produced and operated.

2.3 Defining the "pillars" of the aeronautic collaboration

Aeronautic projects are supported by five major pillars (see Figure 1):

- The **strategy** defines the managerial set of decisions / actions and determines the long-run performance. The strategy in aeronautic projects relies on the appropriate risk sharing versus investment ratio that determines the go/no-go to join a project. This strategy is also based on the commitment of competencies and know-how of partners to determine the best partnership.
- The **business** which defines the constraints of the project (e.g. organisation, time-table, stages and milestones…). It defines the harmonization between working-group and rationalizes the collaborative activities.
- The **information** which defines the constraints attached to exchanges of product information (e.g. data and attributes, 3D models…). It defines the collaborative policies in terms of knowledge and know-how sharing.

- The **applications** which define the constraints raised by engineering applications. It defines the strategy to be built upon them to define the best practice for exchanging product data.
- The **infrastructure** which defines the set of interconnected structural elements (e.g. databases, collaborative repositories) that provides the framework supporting project. It defines the strategy of allowing communication between collaborating teams.

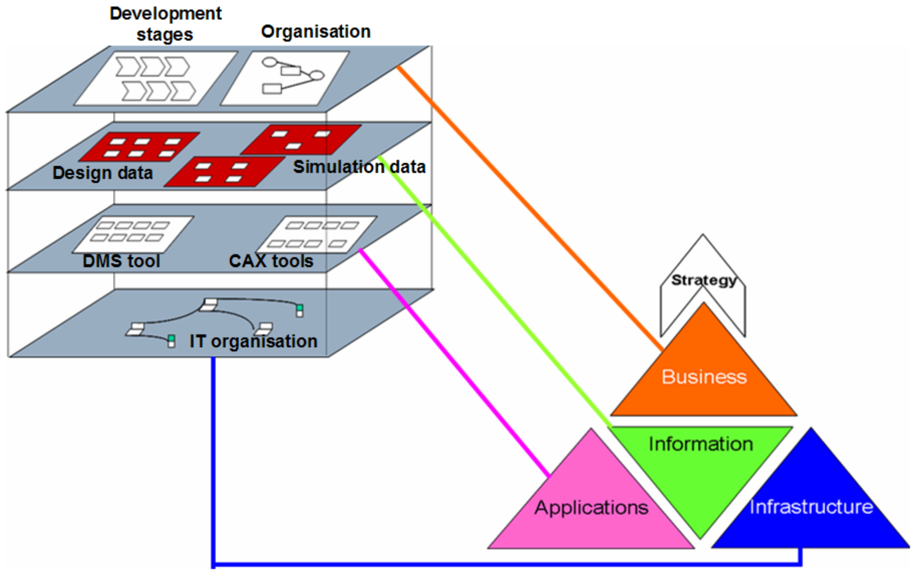

Figure 1 – The five pillars of the aeronautic projects

3 TECHNOLOGY GUIDELINES AND ORIENTATIONS

3.1 About the design of collaborative environments

While collaborative environments were limited to design environment a few years ago, such an environment use is now extended to processes identified in the enterprise (from design to manufacturing) and to the different stages of a project (from preliminary to service).
This is moreover true if we consider the approaches of Fuh [14] and Li [21] who propose mechanisms for distributed and integrated product development and engineering. Such mechanisms rely on the association of two major approaches for product development:

- **A static approach**: to associate the static elements of the product definition (e.g. Id, name, maturity, version, exchange date…). This is designed with classical modelling tools such as UML (Unified Modelling Language) [2, 22].
- **A dynamic approach**: Relevant for a process/workflow definition. It defines the activities sequence that describe the actions on data. This is mainly designed with classical modelling tools such as IDEF0 (ICAM DEFinition language) [5, 36].

The definition of a collaborative architecture must use both approaches. Bergman & Baker [3] describe a « shared virtual workspace » that ensure the data exchange through integrated environments. Each environment is submitted to its own functioning laws. The objective is to create networked activities using product data representation, and more especially a reference frame for the product in the collaborative workspace [30].

3.2 From previous studies toward an original concept

In the early '80's, there was little interest in the idea of Enterprise Reengineering or Enterprise Modelling and the use of formalisms and models was generally limited to some aspects of application development within the Information Systems community [5, 7]. Many of these legacy systems were and are still mission-critical: as businesses change to address the competitive pressures of today and tomorrow, these systems must also change. The problem is not just the complexity of technologies of yesterday, or of today: Open Architecture environments, Client/Server systems, CASE tools, Object-Oriented development etc. The problem is broader; organisations are undergoing rapid changes as they Re-Engineer to compete. Organisations and systems must be designed for change [12, 32, 27]. The subject of "architecture" was acknowledged at that time; however, there was little definition to support the concept. This lack of definition has primarily been developed in the concept of "**Framework for Information Systems Architecture**." Although from the beginning, it was clear that it should have been referred as a "Framework for Enterprise Architecture". This enlarged perspective now begin to be generally understood as a result of the relatively recent and increased world-wide focus on enterprise "engineering". "The Framework" as it applies to enterprises is simply a logical structure for classifying and organising the descriptive representations of an enterprise. These descriptions are significant to the management of the enterprise as well as to the development of the enterprise's systems [11, 17, 25].

One of the main challenges to solve is the use and the integration of the framework in the collaborative context of aerospace projects. It means, to identify the Business scope, the type of data managed, the associated processes, the type of contractual relationships, the physical architecture… Many implementations have been done such as association of a middleware using web services, association of models in integrated environment, creation of distributed business processes or platforms using enterprise modelling approaches. Perrin & Godart [30] expose in their work a centric approach based on middleware solution and implemented with web services. Such an approach results in defining a collaborative environment and repository for data. Middleware is a solution based on a thin server and a strong client solution. It ensures, through web services, to access the data. Sudarsan et al. [37] propose another solution based on models association in order to create a PLM structure that is a support framework for product information. Expected benefits rely on the capability to access, store, serve, and reuse product information all along the lifecycle.

At this level, **enterprise modelling and integration (EMI) is identified as a necessary approach to result in an efficient collaboration**. This consists in defining business processes and the explicit/implicit connectors that exist between them. After Vernadat [40], the enterprise integration consists in "breaking down organisational barriers" and the enterprise modelling consists in "making models of the structure, behaviour and organisation of the enterprise". In one sense, enterprise modelling and integration consist in defining models regarding different levels of the enterprise to characterise the processes exploited. A next step consists in organising these models so as to exploit the available connectors between processes and integrate them (modelling and state-of-the-art definitions of Shunk [36] and Reithofer [33], and examples of use by Delen [10] and Vernadat [40]).

3.3 Layer approach for referential

While analysing the framework, we noticed there were systems interacting with each other and with the collaborative framework. This has conducted us to define a model of architecture layers. Figure 2 represents the results and shows the integration of the layers in the industrial system.

- **The operational layer** corresponds to "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 activities evolutions. Composed mainly of workflow tools, this layer is the backward environment that provides to the operational layer the instructions for activities sequence.
- **The means layer** interprets the environment in which individual activities and actions are performed. Composed of information systems and translators, it ensures the design and redesign of environments and application domains.
- **The referential layer** is the common representation and interpretation of the phenomenon that happens during project. Providing a unified view on product for activities and partners, it is acknowledged as the shared working environment.

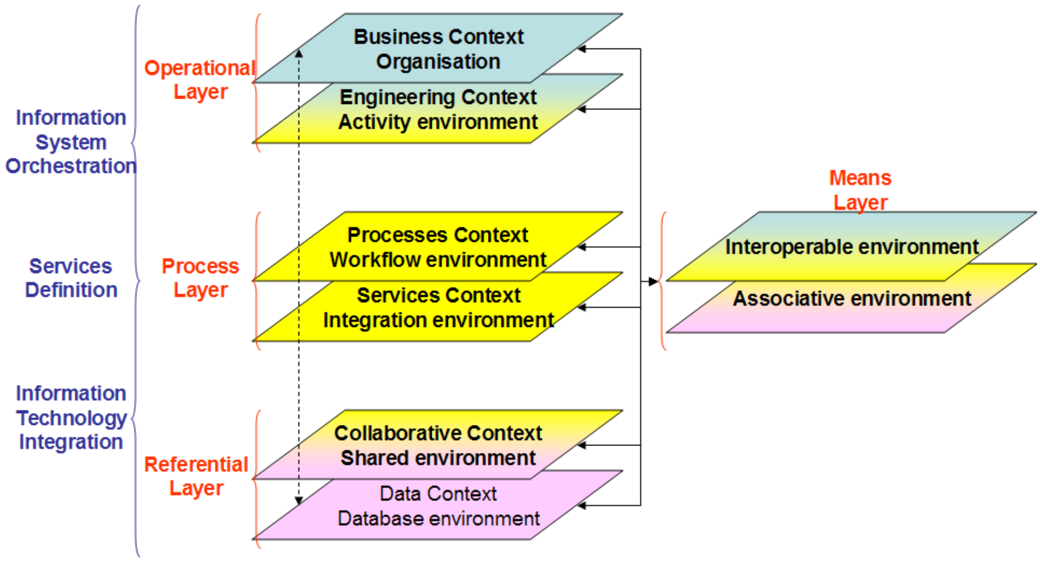

Figure 2 - Representation of system orchestration

4 ARCHITECTURE DEFINITION

4.1 Meta-view for collaborative framework architecture

The collaborative framework concept is deployed in a logical architecture composed of three layers: **the operational layer, the interoperable (or core) layer and the collaborative layer.** A graphical representation is presented in Figure 3.

1. The operational layer contains two sub-layers.
    - The first one is dedicated to the **activity tools** (COTS, legacy tools and portals dedicated to activity support). The framework targets integration and not substitution of legacy tools exploited by partners. The connectors are based on Web services, they ensure the exchange of information that needs to be consolidated. The activities shared by partners, domains or disciplines, are performed using collaboration with tools. The collaboration facilities are demonstrated in the collaborative framework using a dedicated user interface (Product Context Management User Interface) which allows the communication between the operational layer and the interoperable layer.
    - The second sub-layer is dedicated to process management, providing the dynamic aspect of the process in term of workflow implementation, tools communication and interoperability.
2. The interoperable layer implements the context management. This specific application called Product Context Management defines the infrastructure of the collaborative framework architecture and is composed of:
    - Communication component offering Web-services to build connectors with operational and collaborative tools,
    - Information Model component (associated with the information Navigation service) that offers the context of the information (Product, Process, Resource, applicable knowledge, Decision …),
    - Domain Model component that provides a frame for each domain or discipline so as to share the Information needed for collaborative tasks,
3. Finally, the collaborative layer is composed of the consolidated repository. The consolidated repository is also connected to databases which are called regarding the engineering domains.

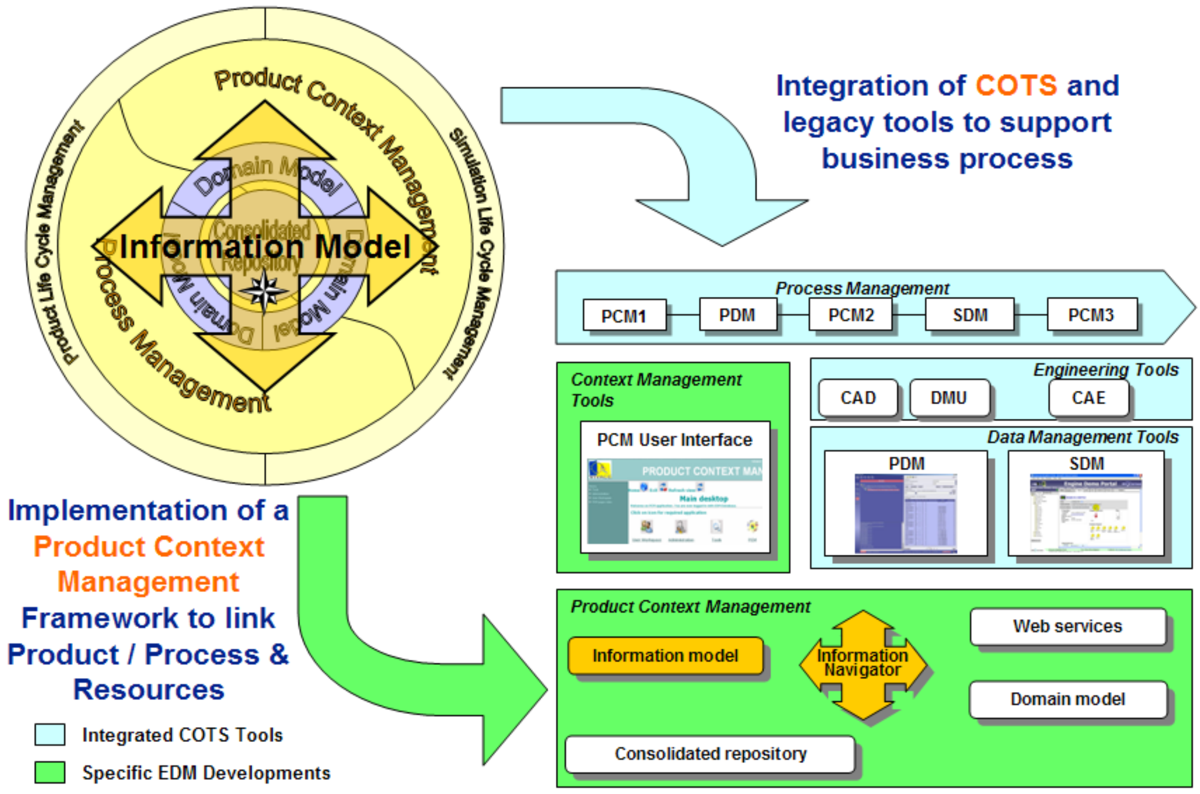

Figure 3 – Collaborative logical architecture

4.2 Modelling of the architecture – General structure

Our architecture is composed of three main layers that include the engineering and business objects necessary to perform concurrent product development. Figure 4 details the different layers and components that compose our collaborative environment.

1. The operational layer: The core concept of this layer is the operational context. It is the unified and integrated view that describes the context in which engineers are committed to perform an activity. It is identified as the meeting point between defined contexts, for a product development using defined resources.
   - The context is defined upon the integration of two components: the first one is the project environment and the second one is the process that provides information about the working transaction.
   - The product and more especially the data and the meta-data of the product are provided using the relationship with the Product Context Management. For the collaborative use, there is no direct links to a database.
   - Finally, the resources are mainly based upon the engineering tools (identified by Data management systems and computer aided technologies) that call the different services to allow data flows between the different layers.
2. The interoperability (or core) layer: The core concept of this layer is the Product Context Management. It defines the operational context and provides the different information to the operational layer using its User Interface. This layer is composed with the following entities:
   - *The definition of the context* at this level is done using the domain models. In fact domain models provide the attached view for the product on a given environment defined by the operational context and more especially by the process. The domain models correspond to the representation of product data using a generic schema that interprets the data of tools in the operational environment. The domain models can be then defined as the interoperable model for data.
   - *The definition of the product* is made using domain models, method objects and information navigator. As soon as the high level context is defined, the information navigator can navigate on all data available for this context. To determine the relevant set of data, the domain models are going to act as filters. Finally, check-in, check-out

of data and other actions on data are made using the method objects.

- *Resources* are available through method objects. Tools are calling services for collaborative actions, in the same time services access method objects to determine the corresponding actions on the interoperability layer.

3. *The collaborative layer*: The core concept of this layer is the consolidated repository. It defines the technological "database" containing the collaborative data and providing access to databases environments. Accessed using method objects that capture data relevant for the domain models, it is the source of collaborative and interoperable data.

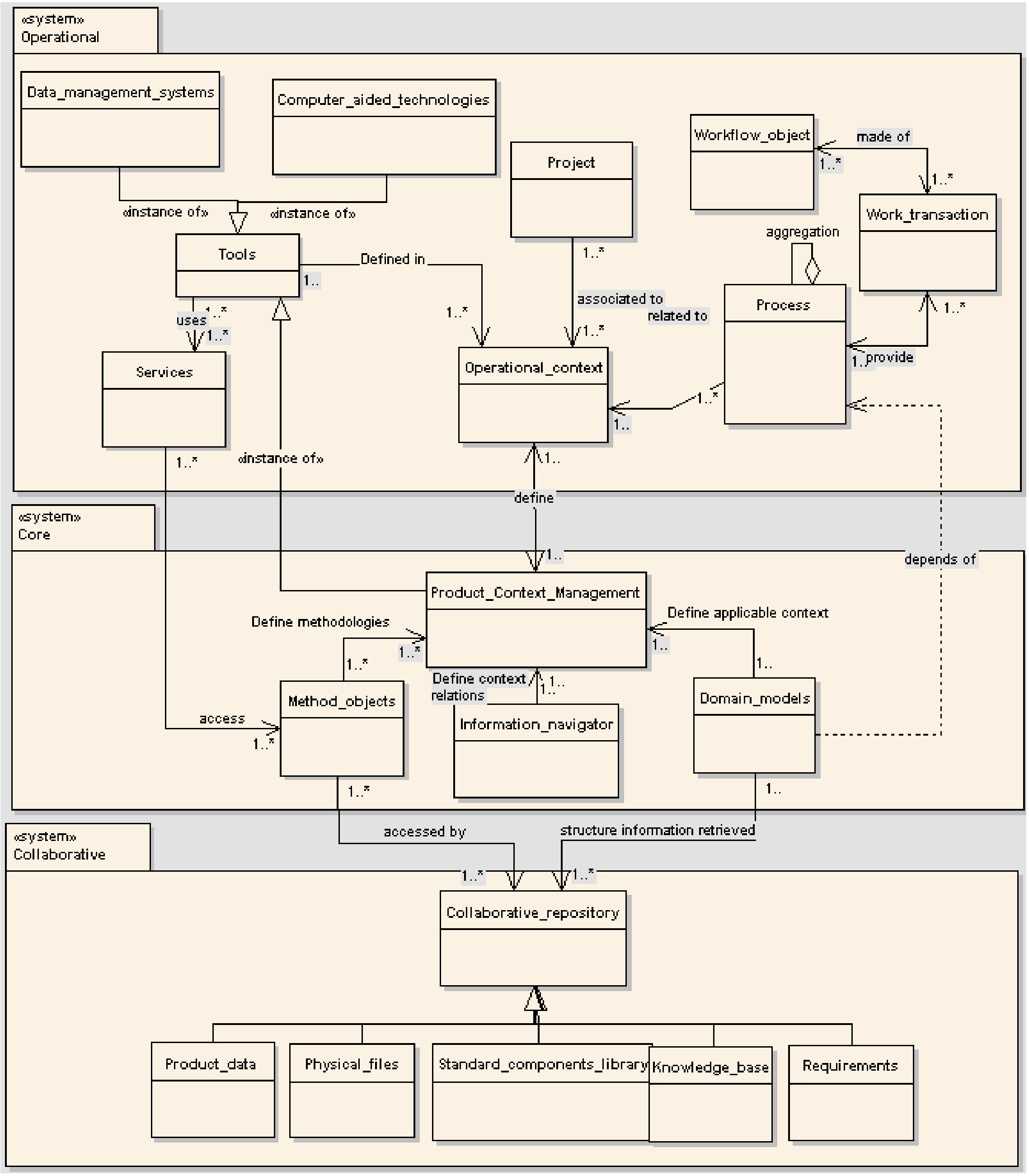

Figure 4 - UML representation of the layers and components model of the targeted industrial system

5 ARCHITECTURE PRINCIPLES AND SCENARIO

5.1 Architecture principles and functioning rules

In previous sections, we have defined an architecture based on three main layers: operational, core and collaborative. If we have to consider multi-tools and multi-activity environments, the approach remains the same. The operational layer will only be more consistent by increasing the number of variable objects and tools. The database will rely also on the core layer. What we propose so as to extend the global notion is to develop the intermediary level (core) made of models, references and workflow. In this way, the development of such an approach relies on a modular model that can be extensible. Figure 5 represents the vision for integrated CAD/CAE around the DMU. In this example, let us consider that a CAD model has been designed. Once this object is designed, current practices make that PDM is also implemented. This provides enough information to generate the integrated view for the DMU. Data are then extracted and migrated to database, they are also processed. This way, data can be migrated to partners so as to be integrated in their design environment. For the migration to simulation field, the DMU is the reference, because it represents the product at a given moment. It represents also the advance in design of the different parts, the configuration of the product and can provide information for bill of material. Then for the simulation object, models can be migrated (using for example the application protocol 209 from STEP) for simulation environment. That is quite the same thing while considering PDM and Simulation Data Management (Application Protocol 214). While models of the core layer are extracting the semantic objects, they can be migrated from the design to the simulation world. Loops can be implemented this way.

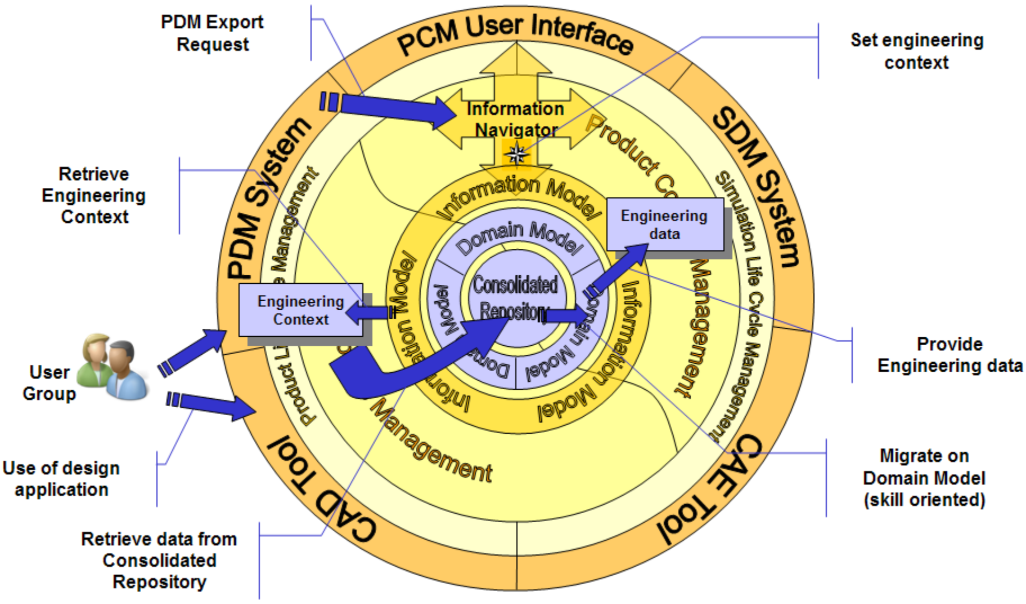

Figure 5 - Example of data context set up for design/simulation loop

5.2 Scenario definition and expected results

The mechanical scenario proposed is a sequential high-level process steering several operational sub-processes that are described here under:

The PCM1 (Product Context Management 1) makes the transition between two contexts of simulation. It first validates the previous simulation activity, then changes of context to work at the Whole Engine Model point of view (with aircraft pylon attach). It launches a design change request to start an optimisation loop. The sub-process PCM1, in the scope of collaborative framework, demonstrates:

1. The context tool: capability to define an overall context of process and capability to control the interoperability and associativity of information.
2. The interoperability with COTS or legacy tools

The PDM (Product Data Management) shows a design modification on a component in a multi-partner context. This part demonstrates in the context of collaborative framework:

1. Design and changes in context: Use of consolidated data for the collaboration and management of access authorisations
2. Use of in-house tools and configuration: Control of in-house system preservation through the use of Web services and use of standardised and directly operable data
3. Interface management for collaborative design task.

The PCM2 (Product Context Management 2) shows first a design analysis. Then after the validation from a multi-partner design point of view, a simulation request is launched to validate with a mechanical simulation. This part demonstrates:

1. How Product Context Management loads pertinent data: Use of Domain model that gets the

relevant information.

2. How Product Context Management facilitates the Navigation: Capability to enter context with product structure or process definition.

The SDM (Simulation Data Management 2) shows an analysis on collaborative data: Pylon and Whole Engine. This part demonstrates in the context of collaborative framework:

1. How collaborative framework integrates simulation environment to perform analysis with collaborative objects.
2. How Product Context provides interoperability between Design and Simulation worlds.

The PCM3 (Product Context Management 3) shows the analysis of the result of the simulation. This sub-process closes the loop of optimisation. This part demonstrates in the context of collaborative framework:

1. How the Product Context keeps a link between design and simulation information.
2. The different activities realised in this step are: the visualisation of design and simulation report together to perform the analysis, the closure of the analysis task.

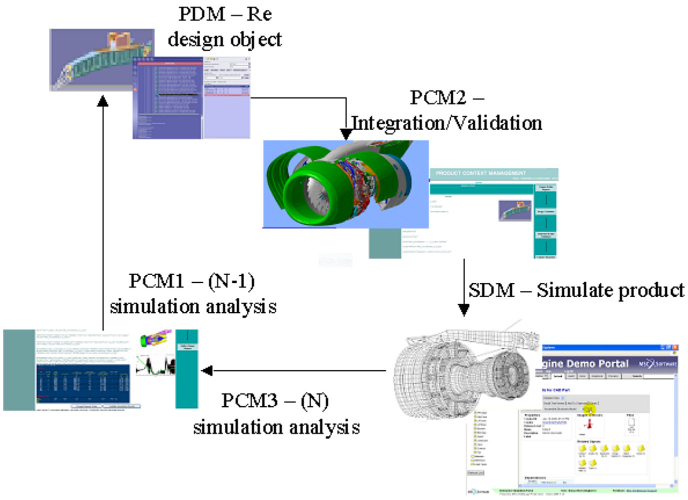

Figure 6 – Design/Simulation loop scenario

6 CONCLUSION

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

- The increase of communication and semantic problems. Related to the increase of the heterogeneity of systems and tools, it generates misunderstandings and wrong interpretations of data and information conveyed in information technology (IT) systems.
- The rationalisation of collaborative processes using multiple environments and product references. The dissociation and non associativity of these systems cause duplication and redundancies problems. Processes are no more guided upon the implementation of environments but by the need to perform the activity. Problems then appear while attempting to create a complete environment with available data.

Regarding these aspects, the contribution of the VIVACE Project on the development of a methodological referential and on the implementation of a collaborative platform for multi-partners and multi-engineering answers the major constraints presented in industry by:

- The development of the engineering interoperability that supports the definition and interpretation of information conveyed in the engineering environments with:
  - The development of interoperability between engineering systems.
  - The development of interoperability product/process.
  - The development of interoperability between engineering data.
- The definition of integrated environments that support the collaborative and referenced view on the product using:

- The development of shared workspaces. Domains of engineering are dependant but separated.
- The definition of reference models using the Digital Mock Up. Our work was mainly oriented on this solution because it is the convergence point of multiple activities.
- The definition of engineering contexts. The assimilation of conceptual domains is related to a defined typology of environments determined by the objectives of the activity and the needs of the engineering environment.

REFERENCES

1. Adrian E. Coronado M., Mansoor Sarhadi, Colin Millar (2002). “Defining a framework for information systems requirements for agile manufacturing” Int. J. Production Economics 75 (2002) 57–68
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. Bergman, R. and J. D. Baker (2000). "Enabling collaborative engineering and science at JPL." Advances in Engineering Software 31(8-9): 661-668.
4. Boujut, J.-F. and P. Laureillard (2002). "A co-operation framework for product-process integration in engineering design." Design Studies 23(6): 497-513.
5. Bradley, P., J. Browne, et al. (1995). "Business process re-engineering (BPR) -- A study of the software tools currently available." Computers in Industry 25(3): 309-330.
6. Burkett, W. C. (2001). "Product data markup language: a new paradigm for product data exchange and integration." Computer-Aided Design 33(7): 489-500.
7. Chen, D., B. Vallespir, et al. (1997). "GRAI integrated methodology and its mapping onto generic enterprise reference architecture and methodology." Computers in Industry 33(2-3): 387-394.
8. Chen, Y.-M. and Y.-D. Jan (2000). "Enabling allied concurrent engineering through distributed engineering information management." Robotics and Computer-Integrated Manufacturing 16(1): 9-27.
9. D'Adderio, L. (2001). "Crafting the virtual prototype: how firms integrate knowledge and capabilities across organisational boundaries." Research Policy 30(9): 1409-1424.
10. Delen, D. and P. C. Benjamin (2003). "Towards a truly integrated enterprise modeling and analysis environment." Computers in Industry 51(3): 257-268.
11. Dustdar, S. and H. Gall (2003). "Architectural concerns in distributed and mobile collaborative systems." Journal of Systems Architecture 49(10-11): 457-473.
12. Egelhoff, W. G. (1999). "Organizational equilibrium and organizational change: two different perspectives of the multinational enterprise." Journal of International Management 5(1): 15-33.
13. Fowler, S. and R. Karinthi (1996). "Remote access to CAD databases using an information sharing system." Computers in Industry 29(1-2): 117-122.
14. 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.
15. G. Loureiro, P. G. L. (2003). "A systems and concurrent engineering framework for the integrated development ofspace products." Acta Astronautica 53 (2003) 945 – 961.
16. Hassan, T. M. and R. McCaffer (2002). "Vision of the large scale engineering construction industry in Europe." Automation in Construction 11(4): 421-437.
17. Holland, C. P. (1995). "Cooperative supply chain management: the impact of interorganizational information systems." The Journal of Strategic Information Systems 4(2): 117-133.
18. Huifen, W., Z. Youliang, et al. (2003). "Feature-based collaborative design." Journal of Materials Processing Technology 139(1-3): 613-618.
19. Kishore, R., H. Zhang, et al. "Enterprise integration using the agent paradigm: foundations of multi-agent-based integrative business information systems." Decision Support Systems In Press, Corrected Proof.
20. 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.
21. Li, W. D., W. F. Lu, et al. (2005). "Collaborative computer-aided design--research and

development status." Computer-Aided Design 37(9): 931-940.
22. Merlo, C. and P. Girard (2004). "Information system modelling for engineering design co-ordination." Computers in Industry 55(3): 317-334.
23. Mervyn, F., A. Senthil Kumar, et al. (2003). "Developing distributed applications for integrated product and process design." Computer-Aided Design In Press, Corrected Proof.
24. Mitschang, B. (2003). "Data propagation as an enabling technology for collaboration and cooperative information systems." Computers in Industry 52(1): 59-69.
25. Montreuil, B., J.-M. Frayret, et al. (2000). "A strategic framework for networked manufacturing." Computers in Industry 42(2-3): 299-317.
26. Moorthy, S. (1999). "Integrating the cad model with dynamic simulation: simulation data exchange." Proceedings of the 1999 Winter Simulation Conference - P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.
27. Nurcan, S. and C. Rolland (2003). "A multi-method for defining the organizational change." Information and Software Technology 45(2): 61-82.
28. Pastor, O., J. Gomez, et al. (2001). "The OO-method approach for information systems modeling: from object-oriented conceptual modeling to automated programming." Information Systems 26(7): 507-534.
29. Pena-Mora, F., K. Hussein, et al. (1996). ": A system for facilitating communication in a distributed collaborative engineering environment." Computers in Industry 29(1-2): 37-50.
30. Perrin, O. and C. Godart (2004). "A model to support collaborative work in virtual enterprises." Data & Knowledge Engineering 50(1): 63-86.
31. Pierra, G. (2000). "Representation et echange de donnees techniques." Mecanique & Industries 1(4): 397-414.
32. Quattrone, P. and T. Hopper (2001). "What does organizational change mean? Speculations on a taken for granted category." Management Accounting Research 12(4): 403-435.
33. Reithofer, W. and G. Naeger (1997). "Bottom-up planning approaches in enterprise modelling--the need and the state of the art." Computers in Industry 33(2-3): 223-235.
34. Rosenman, M. A. and J. S. Gero (1999). "Purpose and function in a collaborative CAD environment." Reliability Engineering & System Safety 64(2): 167-179.
35. 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.
36. Shunk, D. L., J.-I. Kim, et al. (2003). "The application of an integrated enterprise modeling methodology--FIDO--to supply chain integration modeling." Computers & Industrial Engineering 45(1): 167-193.
37. Sudarsan, R., S. J. Fenves, et al. (2005). "A product information modeling framework for product lifecycle management." Computer-Aided Design In Press, Corrected Proof.
38. Tang, M. X. and J. Frazer (2001). "A representation of context for computer supported collaborative design." Automation in Construction 10(6): 715-729.
39. Valckenaers, P., H. Van Brussel, et al. (2003). "On the design of emergent systems: an investigation of integration and interoperability issues." Engineering Applications of Artificial Intelligence 16(4): 377-393.
40. Vernadat, F. B. (2002). "Enterprise modeling and integration (EMI): Current status and research perspectives." Annual Reviews in Control 26(1): 15-25.
41. Vikram S. Adve, R. B., James C. Browne, Ewa Deelman, Aditya Dube, Elias Houstis, John Rice, Rizos Sakellariou, David Sundaram-Stukel, Patricia J. Teller, Mary K. Vernon (2000). "POEMS: End-to-End Performance Design of Large Parallel Adaptive Computational Systems." A preliminary version of this paper appeared in the Proceedings of the First International Workshop on Software and Performance (WOSP) '98, October 1998.
42. Webster, J. (1995). "Networks of collaboration or conflict? Electronic data interchange and power in the supply chain." The Journal of Strategic Information Systems 4(1): 31-42.
43. Werner, C., R. Weidlich, et al. (2004). "Engineers' CAx education--it's not only CAD." Computer-Aided Design 36(14): 1439-1450.