



HAL
open science

System engineering for collaborative data management systems: Application to design / simulation loops

Thomas Nguyen Van

► To cite this version:

Thomas Nguyen Van. System engineering for collaborative data management systems: Application to design / simulation loops. Engineering Sciences [physics]. Ecole Centrale Paris, 2006. English. NNT: . tel-00181449

HAL Id: tel-00181449

<https://theses.hal.science/tel-00181449>

Submitted on 23 Oct 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**ÉCOLE CENTRALE DES ARTS
ET MANUFACTURES
« ÉCOLE CENTRALE PARIS »**

THÈSE
présentée par
Thomas NGUYEN VAN

pour l'obtention du

GRADE DE DOCTEUR

Spécialité : Génie Industriel

Laboratoire d'accueil : LGI – Laboratoire de Génie Industriel

SUJET :

**Ingénierie système appliquée à la gestion des données techniques en entreprise
étendue : Application aux boucles de conception / simulation
System engineering for collaborative data management systems: Application to
design / simulation loops**

soutenue le : 6 décembre 2006

devant un jury composé de :

Ricardo GONCALVES – Professeur, Nouvelle Université de Lisbonne	Président
Bernard YANNOU – Professeur, ECP	Directeur de thèse
Jean-Pierre BOUREY – Professeur, ECLille	Co-directeur de thèse
Abdelaziz BOURAS – Professeur, IUT Lyon 2	Rapporteur
Benoit EYNARD – Maître de Conférences, HDR, UTT	Rapporteur
Anne-Françoise CUTTING-DECELLE – Professeur, IUT d'Evry	Examineur
Bruno MAILLE – Snecma	Examineur

2006-29

Remerciements

Je tiens à remercier en tout premier lieu Bernard Yannou, Jean-Pierre Bourey, Anne-Françoise Cutting-Decelle et Bruno Maillé qui m'ont encadré tout au long de cette thèse et qui m'ont aidé et inspiré autant au niveau méthodologique que scientifique. Je tiens aussi à les remercier pour leurs encouragements et leur soutien constant.

Je voudrais remercier Jean Claude Bocquet, directeur du Laboratoire Génie Industriel à l'Ecole Centrale Paris, de m'avoir accueilli dans son équipe de recherche.

J'adresse aussi de vifs remerciements à Abdelaziz Bouras et Benoit Eynard, qui ont bien voulu être rapporteur de ce mémoire. Merci aussi à Ricardo Goncalves qui a accepté de faire parti de mon jury et de le présider.

Merci encore à mon encadrement Snecma et particulièrement ma chef de département Agnès Gourillon ainsi que Claudine Planquet qui est responsable de la coordination du projet VIVACE chez Snecma.

Merci à mes « partenaires » du projet VIVACE qui m'ont aidé et supporté dans la réalisation de ces travaux. Un clin d'œil tout spécialement à Frédéric Féru (EADS) mon « leader » de Work Package et Pascal Guellec (AIRBUS) qui ont été tous les deux une énorme source d'inspiration et de savoir.

Merci aussi au staf de l'Ecole Centrale, en particulier Sylvie, Anne et Corinne. Un merci pour Carole, Aude, Yacine, Oualid, Georges (*2), Salma, Marija, Sandrine, Christine et ceux que j'aurais pu oublier. Un grand merci à vous tous, qui avez su m'accompagner pendant ces trois ans et qui avez donc contribué à la bonne réalisation de ce manuscrit.

Un grand merci finalement à ma famille, mes parents et ma sœur qui ont su m'aider à garder le courage même pendant les moments difficile. Une pensée aussi à ceux qui m'ont quitté pendant ces trois ans : mon grand-père et ma grand-mère.

Finalement un grand merci à Sandra avec qui a su me soutenir tout au long de la préparation et de la rédaction de ce mémoire sur la fin de rédaction de ce mémoire.

Résumé étendu

L'évolution actuelle des projets aéronautiques vers des projets collaboratifs intégrés nécessite une nouvelle approche de l'environnement collaboratif. L'hétérogénéité des outils et les besoins croissants de gérer l'interopérabilité entre eux et autour d'un référentiel obligent les industriels à se tourner vers les plateformes collaboratives. Les travaux présentés ci-après se sont déroulés chez Snecma (Groupe SAFRAN) et se sont orientés vers le développement d'un tel environnement collaboratif.

Mots clefs :

Entreprise étendue, interopérabilité, systèmes de gestion des données, plateforme collaborative

Introduction et Contexte de recherche

De nos jours, la collaboration pour le développement de produits est devenue une pratique courante. Le développement **du partenariat et de la sous-traitance** n'en est d'ailleurs que le reflet. Ainsi, le produit et sa conception ne sont plus l'œuvre d'un groupe de travail ou d'une entreprise, mais résultent d'un travail coordonné autour des objectifs de la réalisation de ce produit. Deux notions fortes se dégagent sur ce sujet : la notion **d'entreprise étendue** (qui caractérise le principe organisationnel attaché à cette collaboration) et la notion de **conception collaborative** (qui caractérise l'application au niveau des processus et des activités opérationnelles). Sur ces préceptes, les industriels se voient donc contraints d'adapter leurs processus et les outils d'ingénierie qui les assistent de façon à se tourner vers ces nouvelles structures de projet. Cette adaptation est cependant difficile et délicate. En effet, nombre d'entreprise fonctionnent sur des processus assez rigides orientés principalement sur leurs activités et leurs outils d'ingénierie. L'approche collaborative voit donc l'émergence d'un nouveau type de besoin: celui de **créer des structures intermédiaires**. L'objectif de ces structures intermédiaires est de jouer un rôle d'interface entre les entreprises aussi bien pour intégrer les pratiques de conception que l'ensemble des données du produit générées pendant celles-ci.

Ces nouvelles structures intermédiaires (autrement appelées plateformes collaboratives ou "hubs" collaboratifs) apparaissent aujourd'hui comme un élément crucial pour intégrer les processus inter-entreprise et les différentes données de conception. Le besoin principal lié à ces nouveaux outils est de permettre la définition

d'un espace de conception partagé par les différents partenaires. Actuellement, la conception de produit complexes (comme les turbomachines aéronautiques) requiert de multiples outils pour leur développement. Les **outils XAO** (pour les activités Assistées par Ordinateur) ou les **Systèmes de Gestion des Données Techniques (SGDT)** en sont l'illustration. Une problématique industrielle est alors de permettre la **communication entre ces outils** afin que ceux-ci intègrent les mêmes données et paramètres de conception du produit. En effet, ces outils de conception numérique ont évolué depuis leur apparition dans les années 1960. Actuellement, le marché de ces différents outils a connu un tel essor que désormais les systèmes des partenaires d'un projet sont hétérogènes. A ce titre l'application collaborative est en charge de résoudre deux problématiques issues de cette hétérogénéité:

- ❖ La première est de définir un **environnement intégré de conception** dans lequel les partenaires peuvent s'identifier au travers de ce qu'on pourrait appeler un "référentiel de données". Celui-ci correspond à la vue unifiée et actualisée du produit. D'un point de vue concret, cela peut correspondre à l'union de bases de données distribuées qui contiendrait l'ensemble des informations du produit sur son cycle de vie.
- ❖ La seconde est de définir les **termes techniques de l'intégration des pratiques et outils**. Cette intégration concerne dans un premier temps la définition de systèmes de gestion des processus (principalement établis en industrie sous les termes de "workflows"). L'idée est de fournir la possibilité aux partenaires de lancer successivement les étapes de la conception en réduisant au maximum les temps de latence entre activités. Dans un second temps, l'intégration est l'objet (ou tout du moins la notion) qui permet aux différents outils de communiquer. L'idée ici est d'améliorer la qualité des données qui sont fournies par et pour les différents systèmes.

L'application collaborative doit aussi être **le lieu "d'interface"** qui permet de créer les liens entre processus inter-entreprises. Dans cette approche, la problématique émerge du fait que les processus inter-entreprises font actuellement l'objet de procédures manuelles de mises à jour, sans garantie de la validité des données. Pour résoudre cette problématique, des processus de validation intermédiaires permettent de valider les données avant leur publication officielle aux partenaires. Ils permettent aussi de les avertir de cette validation afin qu'ils puissent poursuivre leurs activités sur les données les plus récentes.

Ces nécessités industrielles nous conduisent à considérer trois champs de recherche principaux:

- ❖ La première porte le nom **d'interopérabilité**. Elle définit et caractérise la facilité de communication d'un système à un autre. Dans notre cas, elle permet par exemple de caractériser la migration de données relatives à une configuration de produit d'un SGDT vers celui d'un partenaire. L'interopérabilité fait actuellement l'objet de nombreuses recherches sur les échanges de données [Augenbroe 2004, Davis 2002, Sudarsan 2005] et bénéficie d'une forte représentation dans les communautés scientifiques (Réseaux INTEROP, Communauté travaillant sur les standards comme STEP -STandard for the Exchange of Product data- Peng 1998).
- ❖ La seconde porte le nom **d'associativité**. Elle définit et caractérise les différents liens qui existent entre données. Elle permet de définir entre autre l'évolution d'une donnée sur l'ensemble du cycle de vie du produit. Dans notre objet de recherche, elle permet par exemple d'associer une même configuration de produit, que l'on considère une activité de conception ou une activité de simulation. Cette associativité des données reste un domaine assez proche de celui de l'interopérabilité car les solutions visées cherchent à déterminer un langage commun qui permettrait d'associer les données [Jasnoch 1996, Jeng 1998].
- ❖ Le troisième concerne **l'intégration des entreprises** (aussi connu sous les termes de « Enterprise Modelling and Integration » -EMI- Vernadat 2002). Cette approche aborde la modélisation des entreprises à l'aide de méthodologies spécifiques comme CIMOSA ou FIDO. L'intégration d'entreprise vise en premier abord une définition et une caractérisation des processus afin d'analyser les liens qui peuvent être dressés entre deux processus d'entreprises différents

Le contexte de recherche proposé par cette thèse porte donc sur l'analyse de la collaboration et plus particulièrement sur la conception collaborative et les échanges de données de produit entre les partenaires. Une partie de ces travaux est dédiée à l'analyse des échanges collaboratifs afin de développer le **maximum de cohérence** dans ces échanges et en regards des activités.

Positionnement scientifique de la recherche

Ces dernières années, le domaine de l'ingénierie de conception a évolué vers le numérique avec l'apparition de la Conception Assistée par Ordinateur (CAO) et plus particulièrement des activités Assistées par Ordinateur (XAO) [Demouzon 1998, exemples d'utilisation par Maillé 2001, Sadoul 2000, D'Adderio 2001]. La XAO consiste en l'utilisation d'outils informatiques qui assistent l'ingénieur dans le développement de

produit. La CAO permet par exemple la création, la modification et la simulation de plans 2D et la construction de produits virtuels tridimensionnels. Depuis son apparition dans les années 1960, les capacités de modélisation de produit ont évolué du tracé de plans 2D jusqu'au 3D incluant des éléments de connaissances de l'entreprise (paramétrage dimensionnel des pièces, utilisation de bibliothèques de pièces standardisées) [Dustar 2003, Huifen 2003]. L'introduction de ces techniques d'ingénierie numérique autour de la représentation 3D du produit a introduit de nouveaux processus de validation au sein de l'entreprise et ce, notamment, dans les validations de conception, les décisions et, plus récemment, dans la gestion de la collaboration multi-partenariat [Fuh 2005, Rosenman 1999]. **La maquette numérique** est un exemple de ce développement collaboratif de la CAO [Demouzon 1998, Sadoul 2000]. Elle permet, dans le contexte collaboratif, de spécifier les espaces de conception 3D pour chacun des partenaires en précisant les différents éléments (appelés interfaces) nécessaires à l'appui de cette conception. Utilisée comme support de l'information collaborative, elle est désormais **échangée entre les différents partenaires** de la conception dans le contexte de l'entreprise étendue [Sadoul 2000, Grandl 2001].

L'utilisation croissante de ces technologies a accru le problème de la gestion des données générées par ces systèmes. Ce problème de gestion associé aux besoins d'accès aux données par les différents acteurs de la conception a amené le concept de gestion des données techniques. Celui-ci est défini par les deux points suivants:

- ❖ **Fournir la bonne donnée au bon moment** avec les objets sémantiques suffisants pour l'utilisation dans une activité [Chen 2000]
- ❖ Permettre de fournir des **informations en cohérence avec le statut de développement** du produit [Rosenman 1999]

L'évolution de ces systèmes de gestion des données a permis d'asseoir les concepts de "**Product Data Management**" (PDM) [Hameri 1998], de **Product Lifecycle Management (PLM)** [Sudarsan 2005, Chen 1998, Chen 2000]. De nos jours, le concept de PLM permet de définir et de gérer les données techniques tout au long du processus de conception et d'industrialisation. Ainsi les notions de "workflow" permettent de contrôler les flux de données entre les différents utilisateurs ainsi que dans les "business systems" tout au long du cycle de vie [Kovacs 1998, Leong 2002]. Aussi, depuis l'avènement des solutions de communication et de l'apparition des "business structures" liées à la collaboration, les systèmes de gestion des données

sont utilisés pour maintenir des "images" unifiées du développement de produit [Chen 2000, Zanella 1996].

Ce besoin de collaboration a ainsi contribué à la définition **d'infrastructures intégrées** permettant de créer des environnements collaboratifs pour la conception [Bergman 2000, Fuh 2005, Li 2005]. L'objectif principal de ces structures est de fournir un environnement de gestion de données mais aussi de l'étendre à la vue collaborative. Ainsi, leur rôle est de permettre une intégration des différents partenaires en leur proposant des espaces partagés en fonction des activités qu'ils ont à remplir dans la conception du produit [Gou 2003]. Cette approche des "plateformes collaboratives" est justifiée au travers du besoin d'intégrer la "chaîne de conception" et de permettre à des entités autonomes de s'intégrer dans les processus de développement [Fuh 2005, Li2005, Bergman 2000].

Au delà de ces problématiques structurelles sur l'organisation au niveau des processus et des outils d'ingénierie se pose celui de la **communication des informations**. Les différents développements d'outils font qu'aujourd'hui le tissu de solutions est très **hétérogène**. Ainsi la communication entre les différents outils nécessite de mettre en place différents filtres et convertisseurs afin de permettre une communication efficace [Augenbroe 2004, Davis 2002]. Le rôle des standards de communication est donc devenu important pour permettre la circulation des informations entre les différents systèmes [Teeuw 1996, Peng 1998]. Aussi, au regard de cette problématique, deux approches existent : la conversion des données et la standardisation des données. La conversion, correspond à des échanges entre outils utilisant un format intermédiaire de données. La standardisation permet la conversion des données en un format universel compréhensible par un ensemble d'outils.

Cette problématique de **migration** des données est abordée par le terme d'interopérabilité. Selon la définition de IEEE (Institute of Electrical and Electronics Engineers), l'interopérabilité caractérise "*The ability of software and hardware on multiple machines from multiple vendors to communicate*" (<http://www.computer-dictionary-online.org/?q=IEEE>). Pour l'"*European Interoperability Framework for pan-European e-Government services*", "*Interoperability means the ability of information and communication technology (ICT) systems and of the business processes they support to exchange data and to enable the sharing of information and knowledge.*" (European Commission, European Interoperability Framework for Pan-European eGovernment Services, Luxembourg: Office for Official Publications of the European Communities, 2004, p. 5).

Face à ces différentes hétérogénéités dans les systèmes de conception, deux champs de recherche majeurs se dégagent:

- ❖ La définition des **environnements de conception intégrés**. En ces termes, les champs de recherche actuels visent l'intégration des environnements de conception partagés permettant aux différents acteurs de retrouver les environnements adéquats pour leurs activités [Bergman 2000, Fuh 2005, Li 2005]
- ❖ **L'interopérabilité des systèmes**. Ici l'objectif est de permettre la cohérence des informations proposées par les différentes activités et de permettre l'interprétation des attributs de conception du produit par l'ensemble des partenaires, pour toutes les activités et tout au long du cycle de vie du produit (notamment les études sur la standardisation et l'application de ces langages standards - An 1995, Peng 1998 -, stratégies sur les bases de données pour créer des relations entre attributs produits hétérogènes - Jasnoch 1996, Jeng 1998, Lim 2000 - et le développement de modèles de données pour l'instanciation d'objets sémantiques - Moorthy 1999, Peltonen 1993).

Notre positionnement par rapport aux champs de recherche évoqués ci-dessus est orienté par six besoins de base que nous avons identifiés :

- ❖ Définir un **référentiel** de donnée commun. Celui-ci doit permettre à tout acteur de pouvoir retrouver les informations nécessaires quelque soit la stage du cycle de vie. Il doit aussi aider à assurer la cohérence des données en utilisation, par exemple : permettre de retrouver une configuration...
- ❖ **Gérer l'information entre les partenaires** : c'est-à-dire permettre l'association des formats de fichiers. Cela doit permettre une gestion plus efficace de la sémantique qui doit être transférée entre les différents systèmes.
- ❖ Fournir les **fonctions principales d'un PLM** : permettre une gestion simplifiée par rapport aux outils existants mais sans pour autant les concurrencer. L'objectif est de permettre une gestion des informations nécessaires aux activités.
- ❖ Fournir le **contexte** des données : permettre l'association d'un contexte à une donnée, par exemple fournir un modèle 3D CAO pour la conception et un modèle élément fini pour la simulation...

- ❖ Permettre **l'associativité** des données: définir les liens processus-données sur l'ensemble des activités existantes. Par exemple permettre l'association d'un modèle 3D à un modèle élément finis.
- ❖ Définir les différents **flux de données et les liens entre processus**. Recréer l'équivalent d'un workflow mais orienté sur la collaboration entre entreprises et en fonction des activités.

Notre approche traite de l'élaboration de plateforme collaborative. Les solutions implémentées actuellement proposent ce type de gestion sous la forme de base de données partagées. Notre approche tente d'être plus complète que ce partage de fichier. Nous nous positionnons particulièrement sur les concepts de l'interopérabilité et de l'intégration des processus pour définir précisément le tryptique Produit-Contexte-Ressource. Celui-ci nous permet de mettre en avant sur un processus spécifique l'ensemble des éléments qui vont faciliter la conception du produit (à savoir l'outil d'ingénierie et les données). Concernant notre position par rapport à l'environnement de notre plateforme, nous considérons que celle-ci est un outil d'implémentation des systèmes existants dans l'entreprise. Elle doit contenir les outils d'interopérabilité, de gestion et les mettre en relation avec les environnements opérationnels.

Méthodologie de la recherche

Ce projet de recherche s'inscrit dans le cadre d'un projet Européen appelé VIVACE (Value Improvement through a Virtual Aeronautical Collaborative Enterprise). L'objectif premier de ce projet Européen est de s'inscrire dans la vision ACARE 2020 (recommandations sur l'industrie aéronautique Européenne à l'horizon 2020) qui a pour but de haut niveau:

- ❖ De diminuer par deux le « Time to market »
- ❖ D'améliorer l'intégration des divers fournisseurs,
- ❖ D'assurer la diminution constante du coût de transport par avion

A titre plus spécifique, notre sujet s'inscrit plus particulièrement dans le groupe de travail intitulé « Engineering Data Management (EDM) ». Ce groupe a pour objectif de définir, caractériser et développer une plateforme de collaboration pour les différents partenaires d'un programme. Ce travail est effectué avec les partenaires classiques de Snecma comme Airbus, Rolls-Royce, MTU, Avio...

Au niveau de Snecma, ce travail est effectué au service « maquette numérique » qui gère les échanges de données techniques, et en collaboration avec les ingénieurs de la dynamique d'ensemble et des méthodes. L'objectif visé à moyen

terme de notre action de recherche est de fournir des méthodes de communication qui permettront de faciliter les échanges et d'avoir une meilleure traçabilité du développement dans les phases d'avant projet. L'implication opérationnelle de ce type de développement est de permettre de développer et déployer plus tôt une solution de gestion partagée. L'implication de ces travaux porte aussi dans le développement d'un système de gestion de données de simulations pour le secteur de la dynamique d'ensemble. Celui-ci devrait permettre de valider les apports de ce type de technologie en mettant en avant les réductions de cycles dans les boucles conception-simulation.

L'analyse du milieu collaboratif a principalement été effectuée avec Airbus et a porté sur l'analyse des outils et « best practices » qui existent actuellement dans le domaine de la collaboration. Celles-ci nous ont permis d'identifier :

- ❖ Les outils classiquement utilisés dans la collaboration
- ❖ Les outils qui permettraient une amélioration des performances collaboratives (notamment en terme de récupération des données, qualités des données...).

Ceci nous a permis de nous orienter sur le choix des outils pour une architecture opérationnelle.

Les différents besoins identifiés lors de notre participation aux groupes de travail chez Airbus et Snecma nous ont permis de définir les composants essentiels pour une telle architecture. Finalement, l'architecture doit être constituée de quatre niveaux principaux :

- ❖ Le niveau opérationnel représentant les outils d'ingénierie déjà déployés chez les partenaires
- ❖ Le niveau collaboratif qui permet d'intégrer des processus au niveau inter-entreprise
- ❖ Le niveau d'interopérabilité qui permet d'extraire les différents attributs du produit depuis des fichiers dits « d'échange » (ces fichiers d'échanges sont les fichiers qui sont actuellement échangés par les différents partenaires et qui contiennent les informations sur le produit)
- ❖ Une base de données standardisée et commune dans laquelle les éléments collaboratifs sont stockés dans un format standard de façon à être exploitables par différentes applications.

Au-delà du principe d'architecture, nous avons aussi défini les principaux « connecteurs » qui vont permettre la communication entre les composants principaux. Par ailleurs les principes de fonctionnement de cette architecture ont été aussi définis

(définition de la migration des données, passage dans les différents niveaux de l'architecture...).

Les deux composants qui nous intéressent le plus dans cette architecture et qui nous sont apparus comme primordiaux à développer sont le composant collaboratif et le composant d'interopérabilité.

A l'analyse, le composant collaboratif est apparu comme étant un outil de service permettant aux utilisateurs de définir les processus dans lesquels ils sont inscrits et les données dont ils ont besoin. Ce composant se présente donc comme une interface utilisateur permettant de faire des appels sur les différents niveaux de l'architecture (en utilisant des « web services » qui constituent des formes de requêtes).

Le composant d'interopérabilité correspond quant-à-lui à un outil technologique de transformation. L'analyse de différents standards de communication nous a amené à considérer la norme **STEP – ISO10303** qui concerne la définition d'objets 3D ainsi que les données techniques. A ce titre, nous utilisons un langage spécifique qui est rattaché à cette norme : **l'EXPRESS**. Ce langage est comparable à un langage de modélisation des systèmes d'information de type UML. Il nous permet de définir les entités et attributs qui sont gérés dans un système et de les modéliser. Notre objectif est donc de définir l'interopérabilité entre deux domaines (conception et simulation) en décrivant les modèles de gestion des données, et en mettant en avant les règles de fonctionnement de l'intégration de ces données pour un usage collaboratif. Nous définissons aussi le principe de référencement des données vers une représentation partagée du produit afin que celle-ci soit accessible à toute activité et tout outil.

Les développements de ces différents niveaux seront ensuite intégrés et validés sur un cas d'usage mettant en avant le principe d'interopérabilité entre deux domaines (conception et simulation) et plusieurs partenaires.

Résultats de la recherche

Les travaux menés selon le plan évoqué dans le paragraphe précédent nous ont permis de mettre en avant l'architecture collaborative.

Au premier niveau, nous retrouvons l'acteur de la collaboration et son outil. Celui-ci est en communication avec les autres acteurs de la collaboration au travers de la plateforme collaborative. Le « workflow integration » permet la circulation des flux de données. Nous avons ensuite l'élément collaboratif de contextualisation. Celui-ci correspond au composant collaboratif. Il se présente comme une interface logicielle qui

permet à l'utilisateur de caractériser son contexte (notamment projet, processus...). Cette interface permet de créer un premier filtre sur les données. Le composant suivant (« data model ») correspond au composant d'interopérabilité que nous avons développé.

Basé sur le langage EXPRESS, il définit la structure des différents outils utilisés dans la conception et une structure de donnée qui sert de référence pour les imports et exports d'un outil vers un autre. Le dernier niveau est la base de données collaborative. Composée de documents standardisés, elle a donc la capacité d'être appelée par n'importe quel outil.

Ainsi le processus exploité dans un contexte partenaire va être défini au travers d'un « modèle d'information » qui va caractériser l'environnement. Ensuite, une analyse contextuelle va caractériser les informations qui doivent être mises en avant dans l'activité (par exemple faire remonter des cas de charge ou des conditions limites pour la simulation). La caractérisation fait appel à des « business objects ». Ceux-ci sont définis comme des requêtes standard qui vont être applicables aux données et vont permettre leur utilisation dans un contexte donné (par exemple cela peut correspondre au fait de remonter une configuration sur différents modèles éléments finis ainsi que les charges qui doivent être appliquées).

Validation des résultats et perspectives de fin de thèse

Actuellement, il reste encore quelques développements sur l'architecture. Les premiers tests de notre modèle interopérable ont été effectués avec des données d'un programme actuellement en cours, l'export d'un système PDM vers un système SDM (Simulation Data Management) a déjà été validé. La suite du travail de validation comporte deux étapes. La première correspond à la mise en place d'un démonstrateur commun avec EADS-CCR. L'objectif étant dans de pouvoir montrer les développements et principes de l'architecture collaborative à l'ensemble des acteurs concernés chez Snecma. Ce démonstrateur doit aussi nous permettre de valider les concepts auprès des utilisateurs finaux de la plateforme collaborative. Aussi, nous devons mettre en place un scénario d'usage. Celui-ci a été discuté avec les bureaux d'étude Snecma afin de définir et caractériser un scénario proche de la réalité. Nous avons donc défini les différentes étapes d'un processus de conception-validation/simulation. Ce scénario devra mettre en avant les capacités de notre architecture en termes :

Collaboratif : L'environnement de test est découplé entre deux sites, un sur Snecma Villaroche et un chez EADS-CCR Toulouse. L'objectif est de démontrer la

communication à distance sur un même référentiel technique. Nous démontrerons aussi l'intégration des processus via un workflow inter-entreprise.

Données : Nous avons créé pour les besoins du scénario une maquette numérique qui nous permet de simuler les échanges de données entre le site de Snecma et EADS-CCR. Afin que cette partie puisse être révélatrice de l'implémentation de nos travaux de recherche, nous exploiterons la vue données sous deux formes. La première constituera un test classique d'échange dans lequel les données seront migrées d'un système de gestion des données techniques vers un autre. Afin de valider la possibilité d'utilisation des données sur différents systèmes, nous mettrons aussi en place une validation de migration et d'import de données d'un SGDT classique vers un gestionnaire de données de simulation.

En parallèle de ce scénario, nous avons construit des grilles qui doivent nous permettre d'évaluer les performances de notre système par rapport à l'existant.

Actuellement nous mettons en œuvre les différents composants présents dans notre architecture. Les perspectives de fin de thèse sont donc orientées sur la mise en place du démonstrateur et l'évaluation selon le scénario d'usage validé.

Conclusion : Limites et pistes de recherche

Actuellement, la mise en place des travaux a concerné des outils communs à Airbus et Snecma, et concerne principalement deux types d'activités : la conception en bureau d'étude et la simulation. Bien évidemment, ces travaux doivent par la suite être étendus à d'autres domaines comme la thermique ou encore l'aérodynamique. Le besoin de ces deux activités en termes de données est encore différent de ceux abordés actuellement. Par ailleurs, leurs besoins en termes de maquette numérique (exploitée comme référentiel dans notre travail) est fort.

Au niveau de l'interopérabilité, le travail pourrait être poussé plus loin en permettant par exemple de développer des ressources spécifiques dans les langages de standardisation comme STEP en les implémentant pour exploitation dans d'autres domaines de la conception de produit. Aussi l'analyse des processus devrait permettre dans ce cadre de développer des interfaces spécifiques mieux intégrées que le développement de « workflows » en présentant par exemple des plans d'intégrations de processus inter-entreprise.

Notre travail se positionne à l'intersection de trois domaines principaux : la *gestion des données techniques*, l'*interopérabilité* et l'*entreprise étendue*. Notre apport principal se décline en deux points principaux :

- ❖ Le **développement de l'interopérabilité au niveau des domaines d'ingénierie** (notamment conception et simulation), dans laquelle nous définissons le support et l'interprétation des données.
- ❖ **La définition d'environnements intégrés qui supportent la gestion collaborative et le principe de référencement des données.**

Table of Contents

General introduction	37
I. Introduction	39
II. Presentation of our “research process”	39
A. Stage 1: Audit of the industrial system	40
B. Stage 2: Definition of the research frame	41
C. Stage 3: “New concepts” development	42
D. Implementation and application of concepts	43
Chapter I: Snapshot of the industrial context.....	45
I. Introduction	47
II. Partnership and collaborative product development	47
A. The partnership in aeronautic industry	47
B. The collaborative projects: a necessity for work harmonisation	49
III. Instrumentation for product development.....	53
A. Computer Aided Technologies (CAX)	53
B. The Digital Mock Up (DMU)	54
C. Data Management Systems (DMS).....	56
IV. Conclusion	59
Chapter II: Diagnosis of industrial collaboration in the product development	61
I. Introduction	63
II. Reference frame of the diagnosis	63
A. Introduction – Reasons of the collaboration	63
B. Analysing collaboration using the “collaborative discriminant”	64
III. Diagnosis at project level	66
A. Pillars for project definition	66
B. The project discriminant: the IT infrastructure	67
C. IT system in aeronautic projects.....	68
IV. Diagnosis at process level	69
A. Notion of process workspaces	69
B. Workspaces identification at Snecma	70
C. The reference workspace: Idealisation of Collaborative discriminant	71
for processes	71

V.	Diagnosis at engineering level	72
	A. Comment on the heterogeneity of tools for collaboration at engineering level	72
	B. Typology of data in the product development	72
	C. The data: collaborative discriminant at engineering level.....	74
VI.	Discussion about the results of diagnosis	74
	A. Current issues observed in industry	74
	B. Necessary evolution of the collaborative context	76
VII.	Conclusion	77
Chapter III: Research methodology		79
I.	Introduction	81
II.	Integration in industrial projects	81
	A. Working environment at Snecma	81
	B. Environment of the VIVACE Project.....	82
	C. Planning of our PhD intervention.....	84
	D. Choices in research methodology	85
III.	Scientific viewpoint.....	85
	A. Epistemological choices	85
	B. Methodological approach	86
	C. Refining issues perimeter	86
IV.	Conclusion	88
Chapter IV: State-of-the-art on multi-partnership and multi-activity collaboration and use of standards for communication.....		89
I.	Introduction	91
II.	Engineering tools in the Collaborative environment.....	91
	A. CAX tools and collaboration	91
	B. DMS tools and collaboration	96
III.	The influence of collaborative platforms and “integrated environments” 101	
	A. Collaborative Engineering versus Data Management Systems ..	101
	B. Collaborative Engineering versus Collaborative Process.....	102
	C. Evolution towards collaborative and integrated environments	103

	D.	Major studies on collaborative and integrated environments	104
IV.		Collaboration with standards to suppress interoperability constraints	106
	A.	Standards and main industrial issues in communication.....	106
	B.	Collaborative streams using standards	107
V.		STEP Standard (ISO – 10303).....	109
	A.	STEP Overview	111
	B.	Example of STEP application range.....	113
		Figure 36 – Example of STEP exploitation along product lifecycle	113
	C.	STEP structure	114
		Figure 37 – STEP (ISO 10303) structure.....	114
	D.	Toward a new structure for STEP – Introduction of STEP modules	120
	E.	STEP and EXPRESS G language	120
VI.		Conclusion	120
		Chapter V: Definition of preliminary specifications for multi-partnership and multi-activity collaboration	123
I.		Introduction	125
II.		Necessary methodology to develop the collaborative architecture	126
	A.	Formal description of environment for method implementation ..	126
	B.	From previous studies toward an original concept	126
	C.	Characterisation of the method	128
	D.	Defining the integrative approach.....	129
III.		Characterisation of the collaborative environment for multi-partnership and multi-activity	130
	A.	A three domain cross-vision	130
	B.	Toward Framework critical issues	131
IV.		Applicable concepts	133
	A.	Integration of the product and definition of the digital integration	133
	B.	High level constraints: definitions for a coherent architecture	134
	C.	Integrating the contexts	135
	D.	Integrating the processes	137
	E.	Integrating the resources.....	138

V.	Conceptual functions.....	139
	A. Architecture deployment on conceptual functions.....	139
	B. Multi-view approach	140
	C. Layer approach for referential	141
VI.	Conclusion	142
Chapter VI: Preliminary orientations on architecture and technologies		143
I.	Abstract of Chapter VIII.....	145
II.	Overview of the collaborative framework	146
	A. Meta-view for collaborative framework architecture	146
	B. Modelling of the architecture – General structure	147
	C. Modelling of the architecture – general processing.....	149
III.	Context implementation	152
	A. Logical model to define the context.....	152
	B. Components necessary to define a context	153
	C. Technological set up of the context.....	155
IV.	Product implementation	156
	A. Model to manage product.....	156
	B. Technological definition of the product.....	158
V.	Resources implementation.....	161
	A. Overall overview of resources implementation for the collaboration	
	B. Defining the services components	163
VI.	Rules and functioning principles	164
	A. Communication between components	164
	B. Collaborative framework functioning	166
VII.	Synthesis.....	168
Chapter VII: Proposal of the “collaborative schema”		171
I.	Introduction	173
II.	High level model support – Definition of Information Navigator	173
III.	Information data for product reference – Implementation for Domain models	176
	A. Description of the generic model for data management.....	176

	B.	Interface design/simulation.....	180
	C.	Data specific to simulation.....	183
	D.	Product views	184
IV.		Context implementation – Implementation of the information model ..	187
	A.	Link to process and project	187
	B.	The context definition in EXPRESS	189
V.		Resources implementation.....	192
	A.	Definition of tools structure.....	192
	B.	Services structure.....	194
VI.		Conclusion	195
Chapter VIII: Prototype implementation and preliminary results on industrial scenario			197
I.		Introduction	199
II.		Definition and Characterisation of the Use Case	199
	A.	High level definition of the scenario.....	199
	B.	General requirements.....	200
III.		Proposal of the testing Process	202
	A.	PCM1	202
Figure 78 – First Step: Product Context Management			203
	B.	PDM	203
Figure 79 – Second Step: Product Data Management.....			204
	C.	PCM2	204
Figure 80 – Third Step: Product Context Management			205
	D.	SDM	205
Figure 81 – Fourth Step: Simulation Data Management			206
	E.	PCM3	206
Figure 82 – Fifth Step: Product Context Management			206
IV.		The prototype “show”	207
	A.	Description of what is invisible for “public”	207
	B.	The user interface and the scenario.....	209
V.		Results and validation	214
	A.	General remarks.....	214

	B.	Potential benefits of the architecture	215
	C.	Potential benefits for enterprise partnership.....	216
	D.	Potential benefits for activities collaboration	216
VI.		Synthesis.....	217
Conclusion			219
I.		Contribution review	221
II.		Discussion about collaborative architectures	222
III.		Perspectives and related works	223
Bibliography			227
I.		Description of the VIVACE Project (From VIVACE Document of Work) 245	
II.		VIVACE Project Objective	245
III.		Project organisation	246
Annexe 2: EXPRESS - G.....			249
Information extracted from STEP APPLICATION HANDBOOK			261
ISO 10303			261
I.		Overview	261
II.		Data EXchange Sets (DEX)	261
III.		The contents of a DEX.....	261
IV.		Current set of DEXs	262
	A.	D001 - Product Breakdown for support.....	262
	B.	D002 - Faults related to product structures	262
	C.	D003 - Task Set	262
	D.	D004 - Work Package Definition	262
	E.	D005 - Maintenance plan	262
	F.	D007 - Operational Feedback	263
	G.	D008 - Product as Individual	263
	H.	D009 - Work Package Report	263
	I.	D010 - System requirements.....	263
I.		Application Protocol 209 – AIM diagrams details.....	267
	A.	Material_designation	267

	B. Material_property and Shape_definition / fea_model_definition	
relationships	268	
	C. State_definition.....	269
	D. Nodal_freedom_and_value_definition.....	270
II.	Application Protocol 214 – diagrams details	271
	A. Activity (ARM diagram).....	271
	B. Application_context (AIM diagram)	272
	C. Document (ARM diagram).....	273
	D. External_file_id_and_location (ARM diagram)	274
	E. Configuration_design (AIM diagram).....	275
III.	Application Protocol 239 – ARM diagrams details	276
	A. Product_identification	276
	B. Interface_connection	277
	C. System_breakdown.....	278
	D. Project	279
	E. Activity_method	279
	F. Process_property_assignment.....	280
	G. Person_in_organisation.....	281
	H. Document structure	282

Table of Figures

Figure 1 - The three major stages of our research process.....	40
Figure 2 - Multi-layer and multi-partner work breakdown into design packages	48
Figure 3 – Project definition for the engine development (in French)	50
Figure 4 – Parallelism between aircraft project stages and engine project stages	51
Figure 5 - Example of integration planning (deliberately fuzzyfied for confidentiality reasons)	51
Figure 6 – Principle of data exchange between partners and the integrator – Definition of the integrated environment and dissemination to partners of the project .	52
Figure 7 - Example of a rotor for engine designed with CAD tool.....	53
Figure 8 - Example of 3D model creation with CATIA V5	53
Figure 9 - Example of pre processed models for mechanical simulation.....	54
Figure 10 - Example of post-processed data - Results of bird strike simulation	54
Figure 11 – Digital Mock Up of the CFM56.....	55
Figure 12 - Example of collision analysis using the DMU	55
Figure 13 – Parallelism between project stages and DMU evolution.....	55
Figure 14 - Example of design in context	56
Figure 15 - Example of PDM interface (deliberately fuzzyfied for confidentiality reasons).....	57
Figure 16 - Example of product data (deliberately fuzzyfied for confidentiality reasons).....	58
Figure 17 – The five pillars of the aeronautic projects	67
Figure 18 -Classification of systems infrastructure applied to CAD collaboration (extracted from [Fuh 2005]).....	68
Figure 19 - Schema of IT infrastructure used in collaborative projects.....	69
Figure 20 - Illustration of workspaces for product development at Snecma ...	71
Figure 21 - Typology of data use for product development	73
Figure 22 - Example of consistency loss	75
Figure 23 - Process of determination of our scientific standpoint	81

Figure 24 - PhD development Plan - Integration of Snecma, VIVACE Project and Research work.....	84
Figure 25 - Design parameters during product development.....	87
Figure 26 - Cost impact of design evolution during development stages	88
Figure 27 - State-of-the-art decomposition	91
Figure 28 - Illustration of horizontal and hierarchical collaboration with CAX tools	92
Figure 29 - Representation of horizontal CAX collaboration using the example of CAD	93
Figure 30 - Representation of hierarchical CAX collaboration using the example of CAD and CAE	94
Figure 31 - Illustration of horizontal and hierarchical collaboration with DMS tools	96
Figure 32 - Horizontal collaboration using translators	97
Figure 33 - Horizontal collaboration using standards	97
Figure 34 - Simplified overview of the components of SC4 results (http://www.tc184-sc4.org/).....	110
Figure 35 - Description of the organisation of STEP (ISO 10303) parts - STEP on a page.....	112
Figure 36 – Example of STEP exploitation along product lifecycle.....	113
Figure 37 – STEP (ISO 10303) structure.....	114
Figure 38 - Representation of the internal structure of an application protocol	115
Figure 39 - Example of the AAM - develop product extracted from ISO 10303 – 214 [ISO10303-214]	116
Figure 40 - Example of the ARM – person in organisation extracted from ISO 10303 – 214 [ISO10303-214]	117
Figure 41 - Example of the AIM – organisation item extracted from ISO 10303 – 214 [ISO10303-214]	118
Figure 42 - Process of specification definition	125
Figure 43 - Overall view on CAD/CAE interface	136
Figure 44 – Definition of the process defining the interface between two components	137

Figure 45 - Expression of conceptual functions linked to high level objectives	139
Figure 46 - Expression of functions in regards with the framework related capabilities	140
Figure 47 - Representation of system orchestration	142
Figure 48 - Guideline of the chapter VII – From the high level description of the architecture to the definition of innovative components for collaboration	145
Figure 49 – Collaborative logical architecture.....	147
Figure 50 - UML description of the layers and components of the targeted industrial system	149
Figure 51 - High level description of the transactions in the architecture	151
Figure 52 - UML description of the context.....	153
Figure 53 - UML description of product management	157
Figure 54 - Conceptual description of the data structure for an integrated management.....	159
Figure 55 - Process principle for analysis and integration of partners' data .	162
Figure 56 - UML sequence diagram of transactions in the analysis and integration process	163
Figure 57 – Building of a Common Reference.....	165
Figure 58 – Engineering context setting	166
Figure 59 – Navigation on Engineering Data.....	167
Figure 60 - Definition of the “collaborative schema”	173
Figure 61 - High level model support of the collaborative schema – Information model for navigation	175
Figure 62 - Product structure management	177
Figure 63 - Physical models relationships	178
Figure 64 - Management of design data.....	179
Figure 65 - Management of simulation data	180
Figure 66 - Interface data management	181
Figure 67 - Interface definition	182
Figure 68 - Management of simulation parameters	184
Figure 69 - Aircraft structure description.....	186
Figure 70 - Configuration Management	187

Figure 71 - Description of Process	189
Figure 72 - Context definition.....	191
Figure 73 - VPM tool representation.....	192
Figure 74 - SimManager tool representation	193
Figure 75 - Services definition	194
Figure 76 - Roles and actors for the scenario.....	200
Figure 77 - General definition of the scenario.....	200
Figure 78 – First Step: Product Context Management	203
Figure 79 – Second Step: Product Data Management	204
Figure 80 – Third Step: Product Context Management	205
Figure 81 – Fourth Step: Simulation Data Management	206
Figure 82 – Fifth Step: Product Context Management	206
Figure 83 - XML file defining the content of the PDM called VPM.....	207
Figure 84 - Transformation file with instances of VPM using the transformation of the collaborative schema.....	208
Figure 85 - "Abom" file generated for use in SimManager.....	209
Figure 86 - Main desktop page of the Product context management interface	210
Figure 87 – User workspace page of the Product context management interface.....	210
Figure 88 – Interface pylon-powerplant page of the Product context management interface.....	210
Figure 89 – Interface pylon-powerplant page of the Product context management interface.....	211
Figure 90 – Confirmation pop-up of the Product context management interface	211
Figure 91 – Design change request page of the Product context management interface.....	211
Figure 92 - Main desktop page of the Product context management interface	212
Figure 93 – New pylon page of the Product context management interface	212
Figure 94 – Simulation request page of the Product context management interface.....	213

Figure 95 - Main desktop page of the Product context management interface 214

Figure 96 – New design and simulation page of the Product context management interface 214

Figure 97 - VIVACE Project breakdown 247

List of abbreviation

❖ A/C	❖ Aircraft
❖ AAM	❖ Application Activity Model
❖ AIM	❖ Application Interpreted Model
❖ AM	❖ Application Module
❖ AP	❖ STEP Application Protocol
❖ API	❖ Application Programming Interface
❖ ARM	❖ Application Resource Model
❖ ATA	❖ Air Transportation Association
❖ BOM	❖ Bill of Materials
❖ CAD	❖ Computer Aided Design
❖ CAE	❖ Computer Aided Engineering
❖ CAX	❖ Computer Aided X
❖ CM	❖ Configuration Management
❖ COTS	❖ Commercial-Off-The-Shelf
❖ cPDM	❖ Collaborative Product Data Management
❖ cPLM	❖ Collaborative Product Lifecycle Management
❖ CPR	❖ Context / Product / Resources
❖ CR	❖ Change Request
❖ DEX	❖ Data Exchange set
❖ DMS	❖ Data Management System
❖ DMU	❖ Digital Mock-Up
❖ E- BOM	❖ Engineering BOM
❖ EDM	❖ Engineering Data Management
❖ ERP	❖ Enterprise Resource Planning
❖ FEA	❖ Finite Element Analysis
❖ FEM	❖ Finite Element Modelling

❖	FTP	❖	File Transfer Protocol
❖	GUI	❖	Graphical User Interface
❖	HLA	❖	High Level Architecture
❖	HTML	❖	Hyper Text Mark-up Language
❖	HTTP	❖	Hyper Text Transfer Protocol
❖	HTTPS	❖	Hyper Text Transfer Secure sockets
❖	IDEF0	❖	ICAM definition Language 0
❖	IDEF1X	❖	ICAM definition Language 1 extended
❖	IS	❖	Information System
❖	ISO	❖	International Standards Organisation
❖	KBE	❖	Knowledge Based Engineering
❖	LCA	❖	Life Cycle Application
❖	LRU	❖	Line Replaceable Unit
❖	OBS	❖	Organisation Breakdown Structure
❖	OMG	❖	Object Management Group
❖	OMT	❖	Object Modelling Technique
❖	OOM	❖	Object Oriented Model
❖	P&O	❖	People and Organisation
❖	PBS	❖	Product Breakdown Structure
❖	PCM	❖	Product Configuration Management
❖	PDM	❖	Product Data Management
❖	PLCS	❖	Product LifeCycle Support
❖	PLM	❖	Product Lifecycle Management
❖	QCD	❖	Quality - Cost – Delivery
❖	SCM	❖	Supply Chain Management
❖	SDM	❖	Simulation Data Management
❖	SME	❖	Small and Medium-sized Enterprise

❖	STEP	❖	Standard for The Exchange of Product model data
❖	UML	❖	Unified Modelling Language
❖	V&V	❖	Verification & Validation
❖	VE	❖	Virtual Enterprise
❖	VPM	❖	Virtual Product Model
❖	WEM	❖	Whole Engine Model
❖	WBS	❖	Work Breakdown Structure
❖	XML	❖	eXtended Markup Language
❖	XSL	❖	eXtended Style Language

General introduction

I. Introduction

This doctoral thesis analyses the evolution of digital technologies and methods for data management systems applied to the context of extended enterprise.

Nowadays, in the aeronautics' product development context, projects are evolving towards large-scale partnerships. A large amount of data created for this product development has then to be processed and managed in a coherent way between the different partners. Indeed, the need of enabling data exchange and integration has evolved from the simple file to the management of an integrated design context. This way, closer integration between data created and managed is identified as a factor to shorten the development cycle. This integration involves as well the integration of the overall data created all along the project as the structure to support processes.

This PhD dissertation then presents the study of a centralised architecture to ensure the multi-partnership and multi-view engineering by providing a common referential for data semantic. The use of standards in this study also complies with a high level of interoperability issues in terms of consistency and easy migration of data between engineering systems and activities.

II. Presentation of our “research process”

Our “research process” has been guided by three major stages (see Figure 1):

- ❖ The audit of the industrial system which has permitted to identify and determine the overall environment in which our PhD has to be performed. This has resulted in a better understanding of what collaboration encompasses and what are the necessary changes to perform.
- ❖ The definition of our research frame which has permitted to determine more precisely the scope of our research environment. This was the opportunity to define our scientific approach of the “collaborative problem” and also the state-of-the-art focus.
- ❖ The “new concepts development” which was the result of the two previous stages. Using both requirements from the industrial environment and concepts already developed, we have defined the frame of our contribution. This has resulted into the development of new concepts and their implementation.

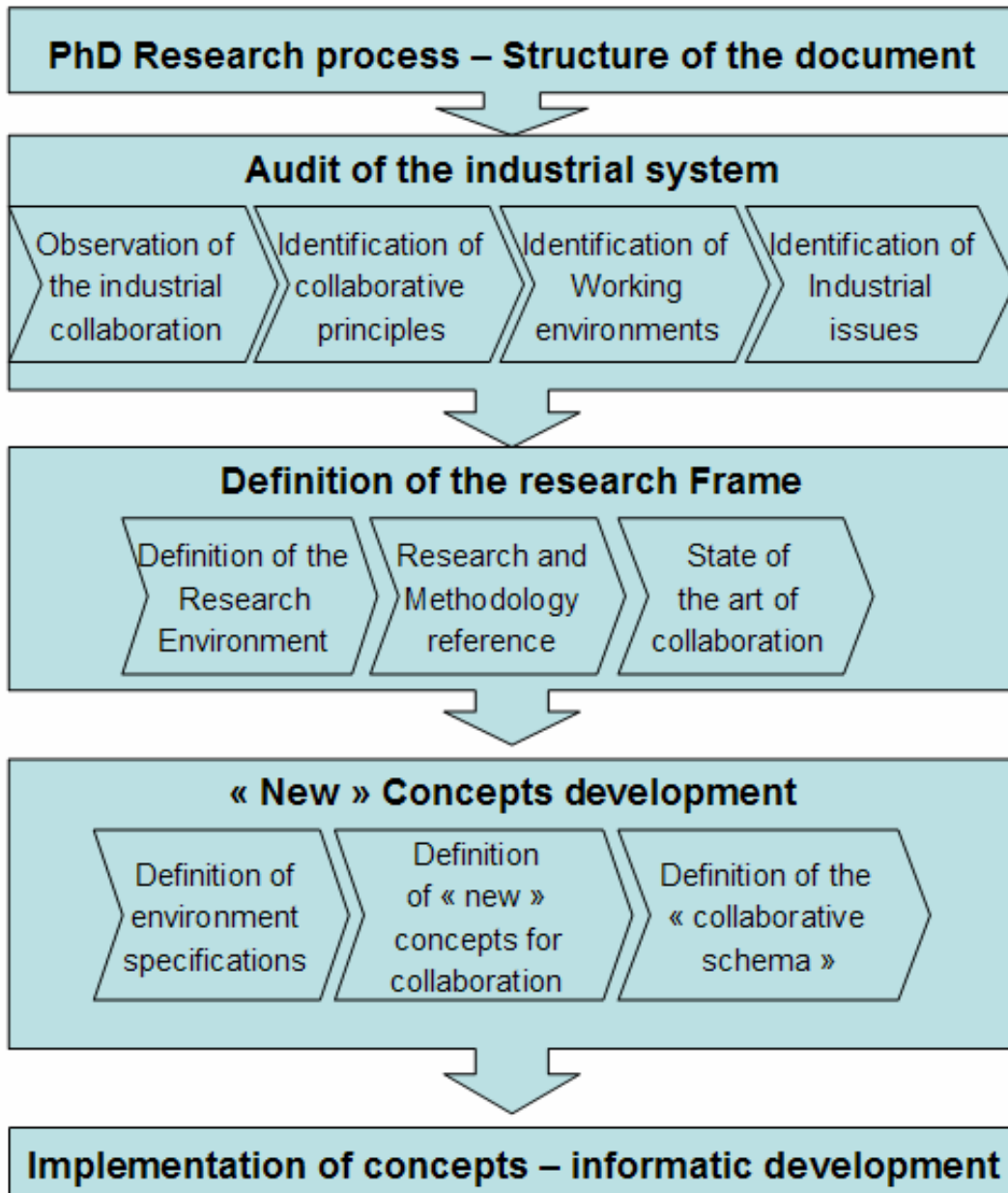


Figure 1 - The three major stages of our research process

A. Stage 1: Audit of the industrial system

The stage concerning the audit of the industrial systems is composed of four major sequences:

- ❖ The observation of the industrial collaboration. Indeed, when we arrived at Snecma, the notion of collaboration and partnership was associated to the definition of a “perfect world” in which partners of the product development work in a unified workspace. We have been disappointed when we saw that in industry the notion of collaboration relies on more basic practices. However, we were not familiar with the different tools exploited for engineering. From our first

observation, there were a lot of tools and they seem to work perfectly together. During some months, we have then analyse these tools and try to use them in a real engine program. Integration between tools was no more obvious, and we finally know what the main problem of collaboration was.

- ❖ This was also the opportunity to identify collaborative principles. Of course, they were also different of what we have imagined. A study of processes and collaboration protocols at Snecma have rapidly show us the current practices and constraints of the multi-partnership.
- ❖ Indeed collaboration is not considered as a whole, and this was more obvious when we have finally identified the notion of working environments. Indeed, activities are performed independently and the unit in which we worked was responsible of the integration of these working environments. But, we also rapidly identify that the action range of this unit was not so large. This was the first question we have: why not to extend this activity?
- ❖ This idea to extend the notion of reference for product development has rapidly appeared. But as we also worked in a European project called VIVACE, we have rapidly seen the limits of such extended system. The constraints and issues raised by our partners in the project didn't seem to be so hard to reach at first look. We then have studied more deeply the current tools and systems that exist and have discovered that there was currently many issues in three domains: the enterprise integration (which is related to the definition of making a model of a collaboration for enterprises), the interoperability (which is related to the principle of making two systems communicate) and the associativity (which defines the way data are associated each other).

B. Stage 2: Definition of the research frame

- ❖ These observations have finally defined our research environment. We were in the centre of three environments: an operational one (which needs rapid solution to enhance performances), a research and development one (which studies and analyses technologies applicable) and a theoretical one (which is our conceptual research). This was finally the opportunity to define guidelines for an innovative environment.
- ❖ But, we had to define our research and methodology reference. The constraints identified in our working environment have first resulted in a dualist attitude. On one hand, we had an operational environment and on the other a “perfect collaborative world”. This was the opportunity to identify that we were in a

constructivist epistemology in which the models we defined don't represent a universal "perfect collaboration" but our view of the "perfect collaboration" for the aeronautic industry. One more question then appeared: where is the main environment to impact? A deeper analysis has rapidly resulted in the preliminary stages. Where time was lost and when major efforts were provided for collaboration? Indeed it is the preliminary stages, where collaboration, communication technologies and strategies on product are set up.

- ❖ Our literature then had been oriented on four main subjects. Indeed, collaboration is performed with tools. When have then analysed major streams using them. Finally, it was also the opportunity to define major streams related to our research topic. It was also the opportunity to highlight and define the gap we had between conceptual views of the collaboration in literature and the industrial system we have to deal with. Also, we know that nowadays the notion of "collaborative platform" is highly represented for partnership. Our state-of-the-art has then considered the conceptual aspects of such environments. Finally, the constraints raised in the VIVACE Project have conducted us to study the STEP standard. We have then analysed this norm (ISO 10303 – STEP). As a result, we have finally concluded that it was adapted to our research topic.

C. Stage 3: "New concepts" development

- ❖ But, before the application of such technology, we had to define what we deal with. The definition of the environment specification has resulted in the definition of a collaborative environment. The process of defining (finally) our contribution was based on the definition of two main stages:
 - ❖ The definition of a methodology applicable to an operational environment. This provides the guidelines of implementation from analysis and concepts to the implementation of an operational environment.
 - ❖ The applicable concepts which define what are the components of such architecture, but also the functioning principles.
- ❖ This has finally resulted in the definition of the conceptual architecture. In this one, we have identified "new" concepts that have to be applied to reach a better collaboration using the already existing enterprise configurations (in terms of tools and engineering systems already existing). A development of these concepts using UML has finally been made. But the final objective was also to use the STEP standard. We have then only represented the main principles.

- ❖ Finally the development has concerned what we have called the “collaborative schema” (in reference to other existing schemas such as the “PDM schema” or the “SDM schema” that have been proposed by Charles in his PhD [*Charles 2005*]). The objective was then to develop and define this schema using EXPRESS-G. This has resulted in the definition of 15 schemas defining a potential environment for collaboration in aeronautic industry.

D. Implementation and application of concepts

Finally, as our work was also related to the European project VIVACE, the development and “computerization” of concepts is currently made. Some results are already available for the migration of data between tools.

Chapter I: Snapshot of the industrial context

I. Introduction

The present PhD dissertation is about two industrial stakes that currently impact the aeronautic industry: the collaboration between partners of a project and the rationalization of the environments for product development. For a few years, industry activities have seen a shift in practices and organisation due to the evolution of collaborative structures. The definition of **network of collaboration** [Cullen 2000, Frederix 2001] has modified the way activities were performed in order to meet new performances. Today, restructuring and implementation of communication techniques and technologies are essential to merge working teams [Beckett 2003].

Collaboration is a well know practice at Snecma. The most famous one concerns the collaboration with General Electrics since 1974 for the CFM 56 engine. But, since 1974, technologies have evolved and more particularly those concerning engineering tools for the product development. Our intent in this chapter is to present a snapshot of the present aeronautic system. We describe the partnership and illustrate how product development is presently performed. We also introduce the instrumentation of product development, and highlight the current industrial issues on collaboration.

II. Partnership and collaborative product development

A. The partnership in aeronautic industry

The aeronautic industry is composed of several companies involved in different specialised manufacturing systems. In recent years, they have lived a shift in their organisation to focus their activities on high added-values competencies. The result is that today, **aeronautic industries are highly specialised**. To build a new product, partnerships and sub-contracting become a necessity. Presently, in the aeronautic industry, the concept of “divide and conquer” is the strategic method that allows coping with the complexity of engineering and designing complex products such as engines. The strategy applied is based upon the **breakdown of the product** into separate design items also called modules (authors such as Demouzon, Pardessus and Monell [Demouzon 1998, Pardessus 2001, Monell 2000] illustrate this strategy). Then the work and responsibility for each design module is allocated to separate design teams.

The Figure 2 illustrates this practice. The product is decomposed into several modules using both functional and mechanical breakdown. Then modules developments and studies are distributed to partners and sub-contractors. Two layers of collaboration then appear:

- ❖ **The internal collaboration and sub-contracting** which correspond to an “in-house” collaboration and which is performed with subsidiary companies (for engine development, it is the case between Snecma for engine and Hispano-Suiza for transmission systems)
- ❖ **The external collaboration and sub-contracting** which correspond to the partnership between distinct companies (for example between Snecma and Rolls-Royce, MTU or Avio)

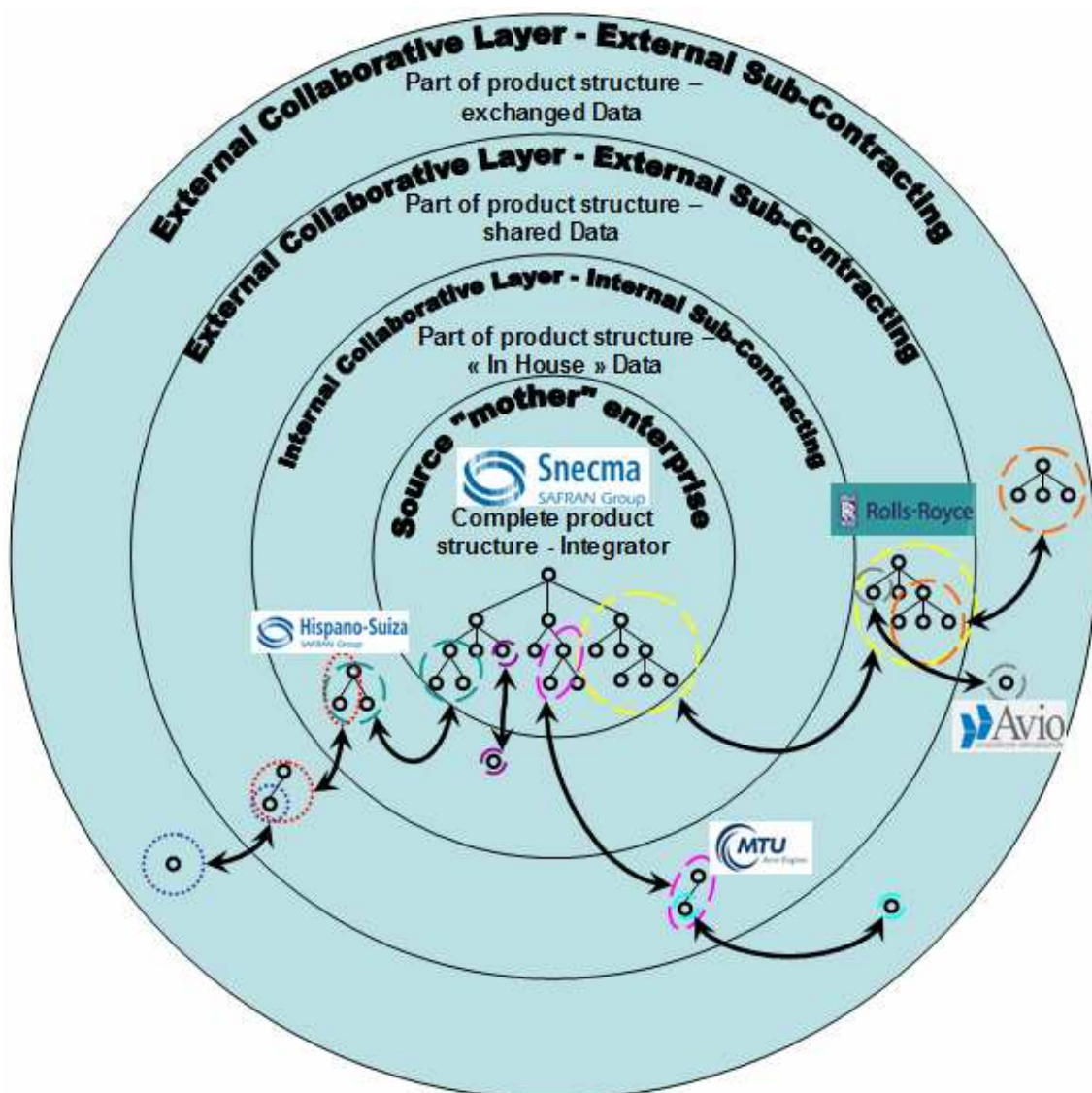


Figure 2 - Multi-layer and multi-partner work breakdown into design packages

B. The collaborative projects: a necessity for work harmonisation

Definition 1: A project is a temporary endeavour undertaken to create a unique product or service. It can also comprise an ambitious plan to define and constrain a future by limiting it to set goals and parameters. The planning, execution and monitoring of major projects sometimes involves setting up a special temporary organization, consisting of a project team and one or more work teams. (www.wikipedia.org)

The project consists of a set of **coordinated and controlled activities** organised to achieve an objective conforming to specific requirements. Presently, at Snecma, a project typically represents 30 months of development (or 60000 hours of studies) and involves about 350 millions of euros of research and development. The project as it can be observed at Snecma is composed of four main stages (see Figure 3):

- ❖ **The preliminary stage** is a pre-definition study of the product. At this stage are performed requirements analysis and performances specifications.
- ❖ **The definition stage** concerns the validation of the previous stage. Here, specifications on the product are detailed and, concepts and components are validated for studies.
- ❖ **The design stage** concerns the studies and development of the product concepts. Major studies on product are provided at this stage. Also, major certifications and tests are made to validate the product development.
- ❖ **The service stage** is the manufacturing stage in which engines are prepared for clients.

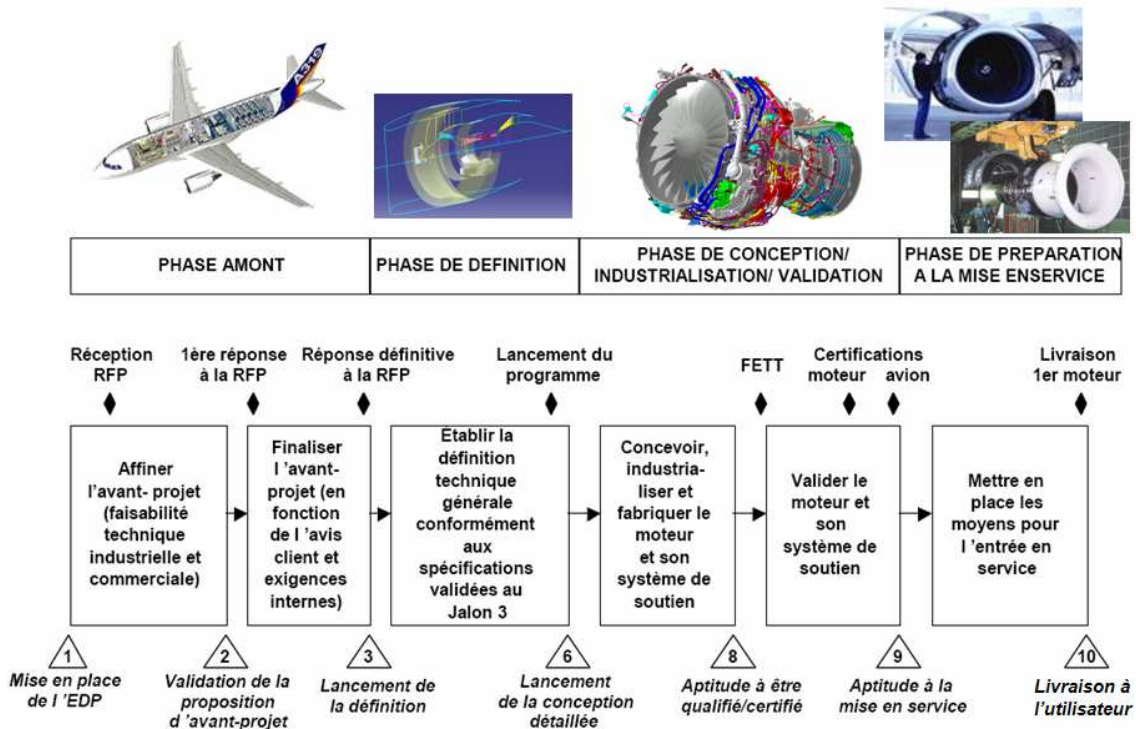


Figure 3 – Project definition for the engine development (in French)

1. Coordination of project stages

The stages previously exposed concern only each of the partners. The challenge is to **synchronize at best the activity sequencing for all partners**. The Figure 4 shows the different stages of an aircraft development in regards of an engine development. The difficulty when considering these technical specifications is to figure out how integration between both development cycles is performed.

To overcome this issue and the one of providing a high level of integration for collaborative stages, the different partners of the product development often propose integration planning between stages. The result of such thought is proposed by Figure 5 that shows the integration planning between the different partners all along the stages of the project.

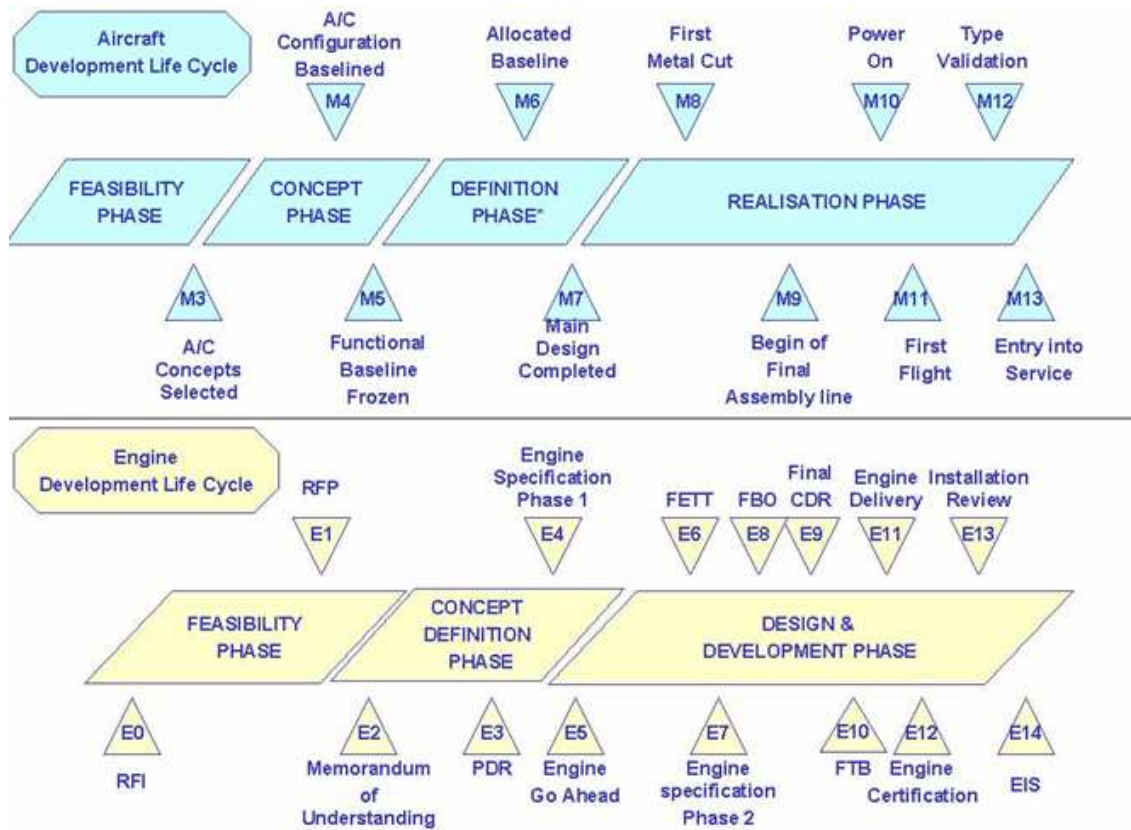


Figure 4 – Parallelism between aircraft project stages and engine project stages

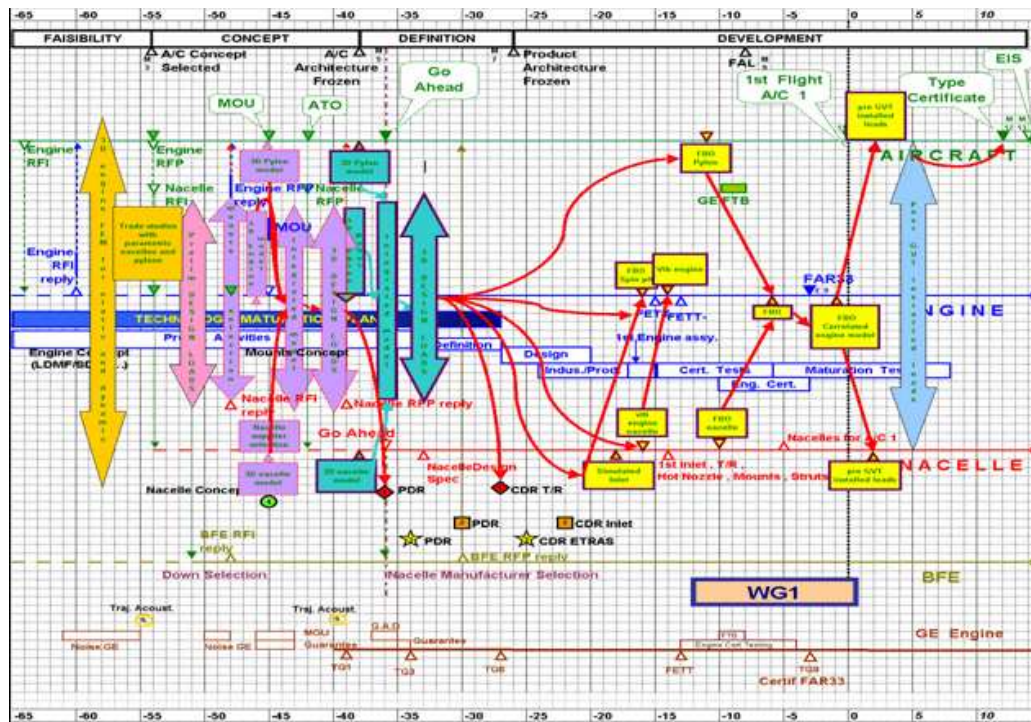


Figure 5 - Example of integration planning (deliberately fuzziyied for confidentiality reasons)

2. Project and product development

The integration planning is then the basis of the collaboration. It explores the constraints and nesting of the partners activities to perform the development of the product. These needs mainly concerns data and information that are produced during the studies of the product. To be able to perform the development and to be informed of partners developments, these data and information are then exchanged between the different partners of the project.

Figure 6 exposes this **principle of exchange**. This principle is based on the replication of data from a system to another. In the example of Figure 6, we can identify three partners (two partners plus one integrator partner). These partners have in their system their own data. Once they have validated their publishable data, they provide it to the integrator. The integrator is then in charge of the integration and validation of the partners data, and of the dissemination of data to other partners. In this example, the product structure is common to all the partners, and each partner is in charge of the development of a component for a sub assembly. For example the partner 1 is responsible of the definition of component 1 (e.g. definition of the 3D model and attributes which describe this component). Once the partner 1 has validated its component, he sends it to the integrator (partner 3) who integrates its data and to redefine the product environment. Once data are integrated, the integrator then disseminates the component 1 integrated in the environment to the partner 2.

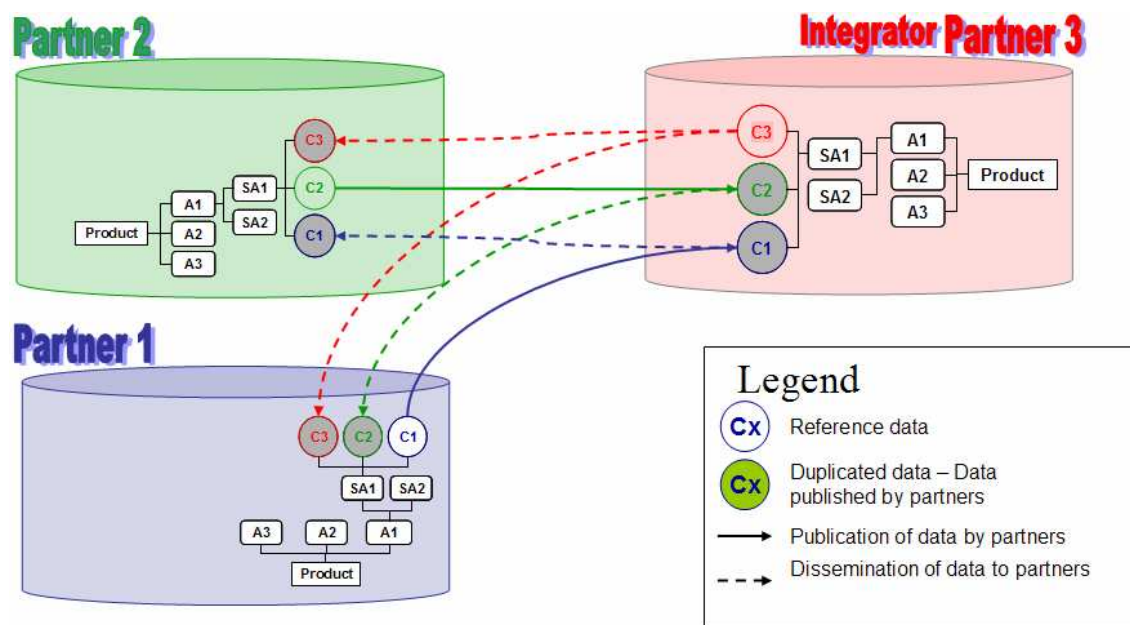


Figure 6 – Principle of data exchange between partners and the integrator – Definition of the integrated environment and dissemination to partners of the project

III. Instrumentation for product development

A. Computer Aided Technologies (CAX)

CAX acronym designates the three dimensional oriented technologies from the computational domain that assist engineers in the design and studies of a product. This technology is provided to help engineers to deal with the new constraints of design, analysis and manufacturing activities [Tang 2001, Rosenman 1999, Fuh 2005]. Presently, most of these solutions are based upon 2D and 3D representation of the product (see examples in Figure 7 and Figure 8). This adaptation is mainly due to the evolution of software and hardware capabilities driven by the need to represent the virtual product [Werner 2004, Wang 2002]. It figures a wide range of application tools that are used by engineers to perform the development (as an example of CAD tools, we can quote: CATIA V5, Pro-Engineer or Unigraphics). Nowadays, CAX tools define many other applications such as **CAD** (Computer-Aided Design), **CAE** (Computer Aided Engineering), **CAM** (Computer Aided Manufacturing) [Bacha 2002, Werner 2004].

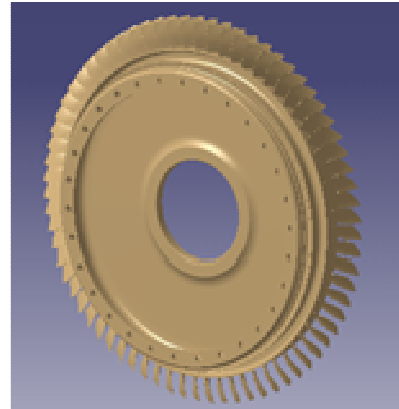


Figure 7 - Example of a rotor for engine designed with CAD tool

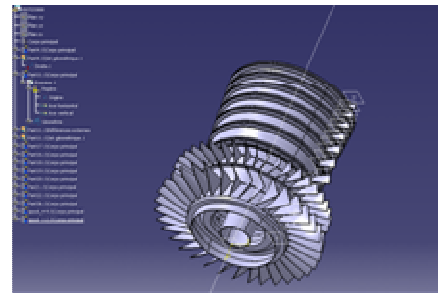


Figure 8 - Example of 3D model creation with CATIA V5

Insert 1 - Evolution of CAX tools

This process of digital design has risen in the early 1960's with the need of improving product development with design software. CAD applications have then started from 2D sketching to attain advanced integration and knowledge capture. In this evolution, implementations such as 3D function oriented, parametric modelling, assembling, and knowledge have evolved in regards with design and collaborative needs. The deployment of the digital engineering using the 3D representation has also allows better design, decision and validations. After Demouzon et al [Demouzon 1998], CAD activity is presently the current practice for design engineering. In industry, CAD is used for design activities from the early stages of the projects. Moreover, its implication in design processes is implemented with the capabilities to model the product, simulate, manufacture it and even design digital plants

[Sadoul 2000, Maillé 2002, D’Adderio 2001]. Today, the increasing use of design’s software allows the early use of digital models. It allows then to create poles of design and to specify in advance the space and the interfaces that are necessary for the definitive model.

Computer Aided technologies are currently used in most of activities and are widespread in the industry. As we already noticed, they are present in the design activities through CAD tools to provide engineers the necessary functions to design the product [Hower 1996, Fuh 2005, Li 2005, Sadoul 2000]. In simulation, they are used by simulation engineers for pre processing data (creation of **Finite Element Models – FEM-**, addition of simulation parameters such as loads, boundary conditions...see Figure 9) and post process visualisation (visualisation of simulation results, see Figure 10).

Finally, they are also used in manufacturing to guide the machining of product components. Their main objective is to permit the most realistic and precise description of the product using 3D definition [Rosenman 1998, Hower 1996, Werner 2004].

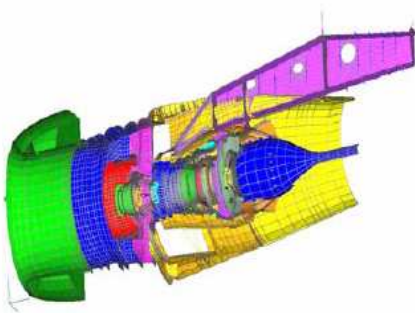


Figure 9 - Example of pre processed models for mechanical simulation

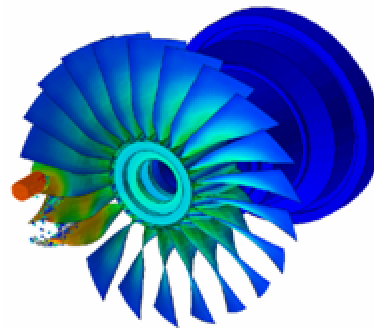


Figure 10 - Example of post-processed data - Results of bird strike simulation

B. The Digital Mock Up (DMU)

The DMU is an extended use of CAD. The DMU is a 3D virtual technology method for product visualization; it allows engineers to build large and complex assemblies of components and perform review placement, alignment, collision and clearances analysis. It uses tessellation of the geometry to allow real-time moving of components, “fly-through” and 3D dynamic sectioning of products. Figure 11 proposes the illustration of a virtual engine (DMU engine) and Figure 12 illustrates the capability of collision analysis.

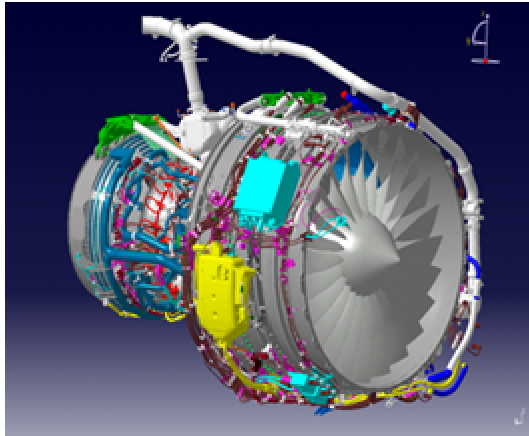


Figure 11 – Digital Mock Up of the CFM56

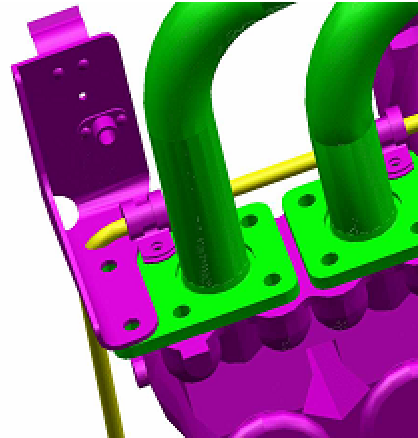


Figure 12 - Example of collision analysis using the DMU

The DMU is the **representation of the virtual product** all along the development process. The definition of the DMU then evolves with the refinement and development of product modules. Figure 13 illustrates the evolution of the DMU all along a development cycle.

In the preliminary stages, the geometrical representation is based on the 2D cross-section of the product. Once the concept is validated, the cross-section is transformed into 3D model using a rotation. At this stage the DMU represents the space allocated for the different components of the product [Hamdi 2006]. In the definition stage, the DMU is deeply modified. As design departments are working on the development of modules, the new design definitions are integrated into the DMU to provide a new representation of the product. Finally, before the service stage, the DMU represents the final product.

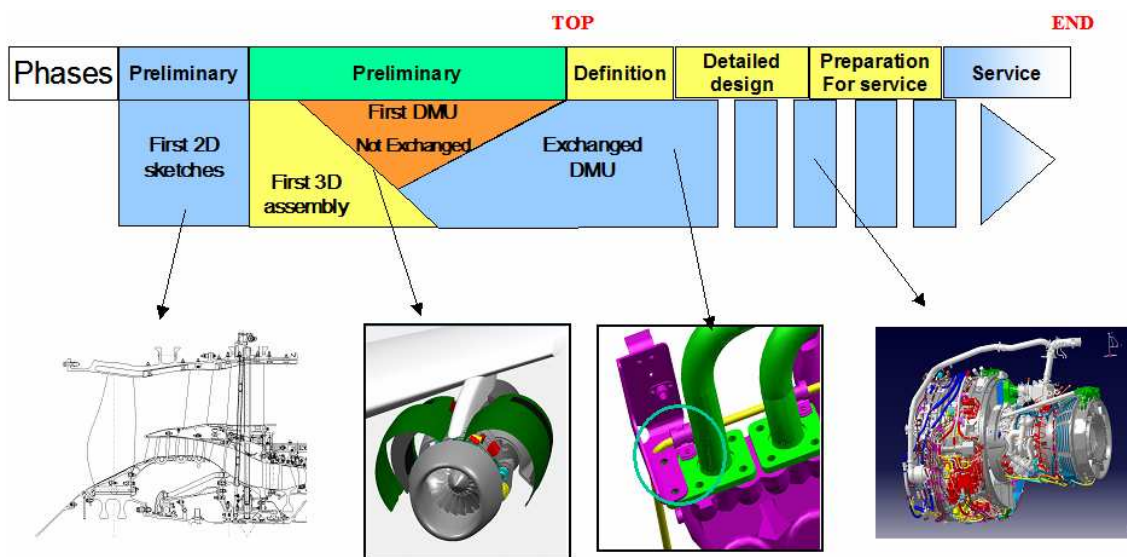


Figure 13 – Parallelism between project stages and DMU evolution

The DMU also allows **the design in context**. The design in context characterise the way to design a module using its adjacent environment and constraints. The product is not designed in a decontextualised 3D space but constrained with the other modules of the product.

Figure 14 represents a contextualised object. In this example, the designer uses the product environment to position and develop the product. The context is defined by the notion of interfaces and adjacent parts of the product. **Interfaces** are a particular category of 3D objects. They define the links between two 3D models. . For example they define the limit and positioning of an object or are specification for the design (for example an interface can define the axis of a cylinder...).

The adjacent parts of the product define the space constraints for design. Using their 3D definition, it defines the space of components to be designed and try to avoid collision between the different modules of the product.

C. Data Management Systems (DMS)

Definition 2: Data management designates the method that controls the creation, conveying and evolution of information that define the product (e.g. information that defines how the product is specified, designed, manufactured and use).

Definition 3: Data management systems (DMS) then designate the set of processes and information that ensure the control and quality of data all along its lifecycle.

Data management includes all the disciplines related to managing data as a valuable resource. The official definition provided by DAMA (Data Management Association) is that "Data Resource Management is the development and execution of

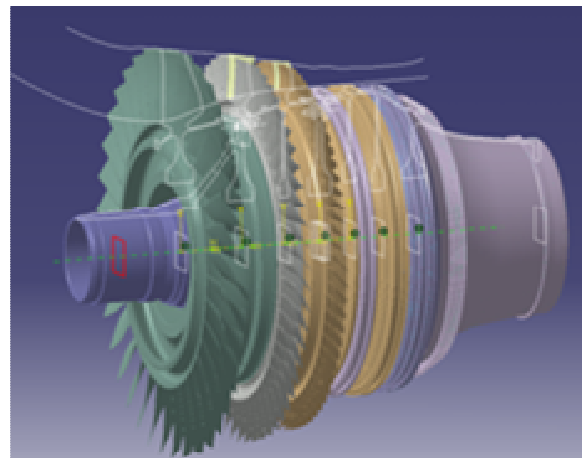
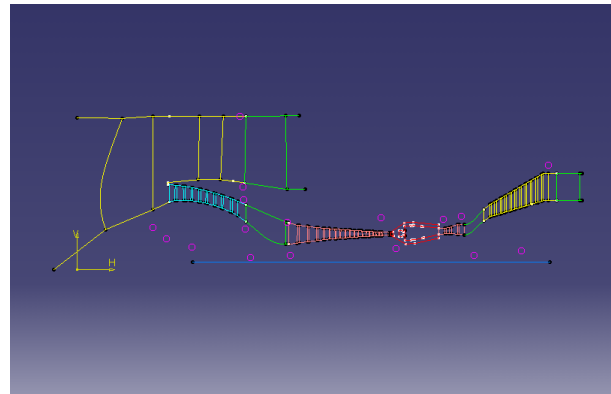
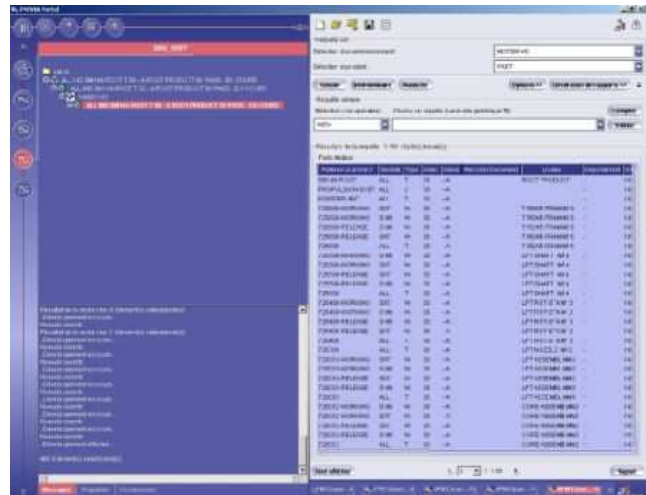


Figure 14 - Example of design in context

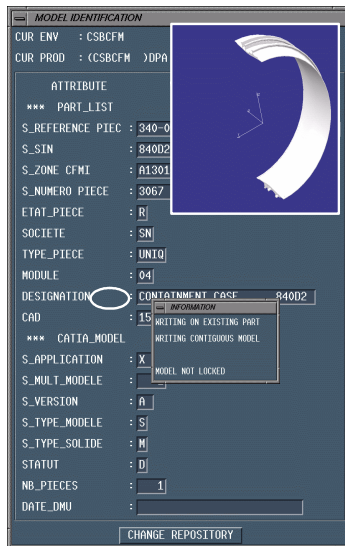
architectures, policies, practices and procedures that properly manage the full data lifecycle needs of an enterprise." Data management systems integrate many tools such as:

- ❖ **Product Data Management (PDM)** which is a category of computer software that aims to create an automatic link between product data and a database. The information being stored and managed includes engineering data such as CAD models, drawings and their associated documents [Eynard 2004, Hameri 1998]. As an example of PDM tools, we can quote: Enovia VPM or Enovia Smarteam
- ❖ **Product Lifecycle Management (PLM)** which is the process of managing the entire lifecycle of a product from its conception, through design and manufacture, to service and disposal. PLM is a set of capabilities that permit an enterprise to effectively and efficiently innovate and manage its products and related services throughout the entire business lifecycle [Kiritsis 2003, Wognum 2004]. As an example of PLM tools, we can quote: Enovia LCA, Windchill or Teamcenter.
- ❖ **Simulation Data Management (SDM)** which is an extension of PDM (and the information it manages). The SDM includes in addition to engineering data, the necessary information to perform simulation (loads, boundary conditions, scenario of simulation, results...) [Eben 2004]. As an example of SDM tools, we can quote: SimManager.

First, the data management tool manages the structure of the product. It defines the structural parts (that are associated to assembly of modules) and models parts (that define the different parts of the product – e.g. 3D models and related attributes). The Figure 15 illustrates the current PDM tool at Snecma.



**Figure 15 - Example of PDM interface
(deliberately fuzzified for confidentiality reasons)**



**Figure 16 - Example of product data
(deliberately fuzzified for confidentiality
reasons)**

Each part is defined by attributes related to the product definition. They allow retrieving the name of the parts or modules, their identification number and other attributes useful for the identification and traceability of the development (e.g. owner name, maturity, last release...). Figure 16 illustrates the different attributes contained in the PDM system and the relationships between these attributes and the 3D CAD model.

Insert 2 - Evolution of Data Management Systems

Manufacturing companies are usually good at systematically recording component and assembly drawings, but often do not keep comprehensive records of attributes related to these components (such as 'size', 'weight', 'where used' etc). As a result, engineers often have problems accessing the information they need (evoked by Peltonen [Peltonen 1993] for the definition of data management architectures, McFadden [McFadden 1995] for data management approaches and Eynard [Eynard 2004] for the use in enterprise). This leaves a gap in their ability to manage their product data effectively [Peng 1998]. Data management systems manage attribute and documentary product data, as well as relationships between them, through a relational database system. CIMdata defines PLM as: "A strategic business approach that applies a consistent set of business solutions that support the collaborative creation, management, dissemination, and use of product definition information". Today, this definition needs to be implemented with new notions such as the support of the extended enterprise (customers, design and supply partners, etc.), the management from concept to end of life of a product or plant, the integration people, processes, business systems, and information [Westfechtel 1998, Sellgren 1996, Liu 2001]. It is important to note that PLM is not a definition of a piece, or pieces, of technology. It is a definition of a business approach to solve the problem of managing the complete set of product definition information—creating that information, managing it through its life, and disseminating and using it all along the development of the product. PLM is not just a technology, but is an approach in which processes are as important, or more important than data. It is critical to note that PLM is as concerned with "how a business works" as with "what is being created" [Naka 2000, Sudarsan 2005].

PLM had its inception in product data management (PDM) applications during the mid 1980s. Early implementations primarily wrapped around engineering design data. But as the industry evolved in response to customer requirements, the scope has expanded far beyond engineering departments. In the 1990's, this lifecycle view expanded from managing primarily the mechanical elements of a product's definition to include the electronics and software elements that have become a larger part of many products [Lau 2003]. That expansion continued extend the perception of what "design" encompassed.

PLM includes management of all product-related information from requirements through design, manufacturing, and deployment. This information ranges from marketing requirements, product specifications, and test instructions and data, to the as-maintained configuration data. A PLM solution links information from many different legacy tools and other systems to the evolving product configuration. Also, in the 1990's, the lifecycle started to include production-focused attributes and information [Chen 1998, Chen 2000, Giannini 2002]. Today, PLM encompasses significant areas of process definition. It helps defining, executing, measuring, and managing key product-related business processes. Manufacturing and operational process plans are also now viewed as an inherent part of PLM. Processes, and the workflow engines that control them, ensure complete digital feedback to both users and other business systems throughout lifecycle stages [Kovacks 1998, Leong 2002]. Although, the final evolution of such a system is to integrate all the product data and analyse them in order to provide better management. This could allow filtering and delivering data in regards with activities needs. At the same time, and as the industrial context is now evolving to partnership, these solutions have turned to collaborative applications.

IV. Conclusion

In the definition of networked enterprises and activities we have presented, the need of **harmonization between practices and technologies is essential**. Besides, the constant evolution of enterprise organisation and of engineering tools constitutes the basis of the issues raised by this PhD dissertation. These issues can be defined by two basic questions:

- ❖ The first concerns **the harmonization of the collaboration between the partners** of a project. It means, what are the strategies of collaboration between dispersed teams? And how the collaboration between different activities is performed?
- ❖ The second concerns **the harmonization of the data exchanged between partners**. It means, what are the links between tools used by partners? And how data are migrated and integrated from a system to another?

Chapter II: Diagnosis of industrial collaboration in the product development

I. Introduction

In this chapter, we make the diagnosis of the industrial collaboration as it can be currently observed in the aeronautic industry. When we arrived at the Digital Mock-Up (DMU) unit at Snecma, we noticed that major collaborative activities were dedicated to file transfer and file management concerning the product definition (data and 3D models) between the partners of a program. The first definition we adopted was: collaboration refers to a process in which people work in a concurrent way to perform a common objective. With deeper audit of the collaborative processes and activities, we remark that collaboration is an activity performed in different ways, by different players at different levels of the company. We also identified that collaboration may be conceived in a broader sense, with the increase of sub-contracting and partnership activities, as the definition of enterprise networks [Cullen 2000, Frederix 2001].

Here, we present our methodology for the analysis of the collaboration. We identify the “collaborative discriminant” regarding three levels of collaboration: The project collaboration, the process collaboration, and the engineering collaboration. Finally, we discuss the results of this diagnosis, and identify current issues in industry.

II. Reference frame of the diagnosis

A. Introduction – Reasons of the collaboration

The new paradigm of collaborative product development or collaborative enterprise [Willaert 1998, Li 2005, Pardessus 2001] can have significant implications for product development. These new collaborative manners of developing technologies and aeronautic systems produce changes in the ways aeronautic systems are designed, produced, operated, maintained, and disposed of. By combining the strength, expertise and know-how of the best diverse, geographically dispersed technical teams, better mission scenarios, designs, and corresponding technologies can be developed in less time [Hassan 2002]. Collaborative engineering is then seen as the application of team-collaboration practices to an organisation’s total product development efforts [Browne 1995, Moss Kanter 1999, Beckett 2003]. It builds upon the systems engineering, project/program management foundations of primarily in-house cross-functional product development teams introduced by concurrent engineering [Dustdar 2003, Huifen 2003].

B. Analysing collaboration using the “collaborative discriminant”

Definition 4: In mathematics, the discriminant is the method used to define the results of algebraic equations

Definition 5: The collaborative discriminant is then the method applied to find out the characteristics of relationships between business entities (e.g. enterprises, processes...) and/or object (e.g. data, 3D models...).

Trying to define the collaboration, many authors consider that it consists in **integrating parts of business or enabling unified methods and procedure**. For example, Peña-Mora and Ruland [Peña-Mora 1996, Ruland 1995] define the collaboration as the capability to make sure that the communication between business entities, whereas for Dustdar and Webster [Dustdar 2003, Webster 1995], integration and collaboration deals with technological developments. Both approaches are necessary in fact, the 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 can be defined as the logical and physical connection of business components (e.g. activities, processes) with a collaborative environment. Indeed, retrieving the collaborative discriminant can be seen here as defining an elementary structure (implemented with rules, functioning laws and technical objects) that connects two defined business entities of the industrial context. If we consider two processes, for example the design and the simulation activities, the collaborative discriminant between both activities is the product definition (product structure, 3D models and attributes). Indeed, collaboration is not possible if we only consider product structure (because we miss the components that define the product) or specific information (such as simulation parameters, because they are relevant only for simulation). Attempting to define the collaborative discriminant, we noticed that it must be defined at three different levels: Project, Process and Engineering levels.

- ❖ **The project discriminant:** Projects are analysed as collaborative structures locally and temporally defined. They define the **backbone** to group activities and teams for the product development (from conceptual stages to service). Based on “multiplexing” coordinated activities, it is constrained by high level objectives that must be reached to satisfy the requirements of the client [Shunk 2003]. Collaborative 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 collaborative determining 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. Collaborative determining relies on two major ideas: **coordination and cooperation**. Coordination defines the “who does what?” of the project and highlights the necessary structural aspects. Cooperation deals with the definition of benefits while attempting to work together [Chen 2000, Vernadat 2002].

- ❖ **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 to an outgoing 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*), or an “ensemble of correlated activities that transform input elements in output elements” (*ISO 9000*). Process discriminant defines the **aggregation of numerous activities fragments**, in which each fragment describes different part of the overall activity, with no overlapping. In any large organisation and moreover for the collaborative industry context, each actor, or class of actors, has a partial and overlapping view of the complete process [Chen 2000]. The links between processes are ensured by 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** [Boujut 2002].
- ❖ **The engineering discriminant:** It consists in performing elementary activity to develop a part of the product. Engineering disciplines concern design, development, improvement, implementation and evaluation of systems of people, knowledge, equipment, energy, material and process that are set up upon 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 [Rosenman 1999]. It is usually not enough to build up a technically successful product; it must also meet further requirements. Constraints may 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. Many

authors such as Rosenman and Tang [*Rosenman 1999, Tang 2001*] illustrate this concept with the understanding of the constraints. Engineers derive specifications for the limits within which a viable object or system may be produced and operated.

III. Diagnosis at project level

A. Pillars for project definition

Aeronautic projects are supported by five major pillars (see Figure 17):

- ❖ The **strategy** which defines the managerial set of decisions / actions and that 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. Moreover, this strategy is also based on the commitment of competencies and know-how of enterprises to determine the most appropriated 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. Moreover, it is at business level that are defined relationships between lifecycle stages and integration planning (see Figure 4 and Figure 5 for illustration).
- ❖ 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. Moreover, it is at information level that exchange protocols on product data are set up.
- ❖ The **applications** which defines the constraints raised by engineering applications (e.g. CAX and DMS). It defines the strategy to be built upon these applications to define the best practice for exchanging product data.
- ❖ The **infrastructure** which defines the necessary 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.

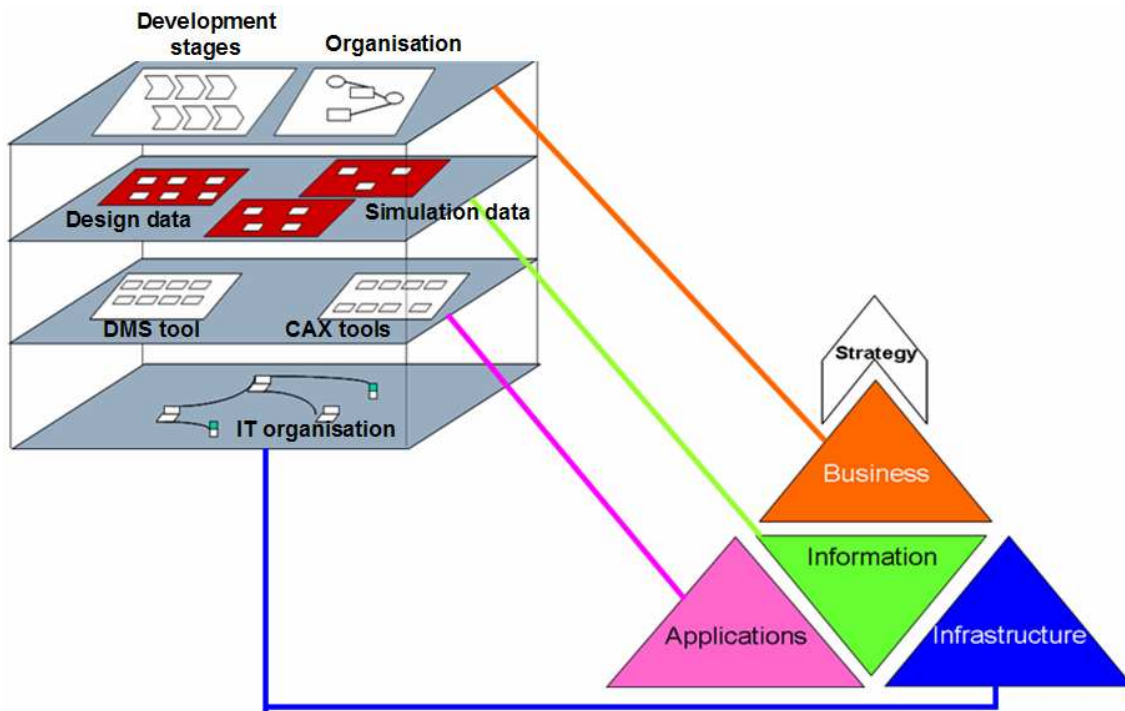


Figure 17 – The five pillars of the aeronautic projects

B. The project discriminant: the IT infrastructure

1. Reasons to consider IT infrastructure

The use of information technologies and their evolution in past years (with internet, intranet, transfer line...) have deeply influenced collaboration. Nowadays, collaborative projects and their organisation are mainly based on such Information Technology (IT) infrastructure. The development of such infrastructure in industry has enhanced communication of enterprises. They have also **allowed enterprises to rapidly develop a common and shared working environment**. This provides an efficient management of resources, processes, knowledge and competences in the meaningful way of goals attainment: “the success of the projects depends on all cooperating as a unit” [Camarinha-Matos 2001, Park Kyung Hye 1999].

2. IT infrastructure classification

Presently, it exists different ways to perform communication at IT level. Fuh and Li [Fuh05] propose a classification of integrated and collaborative environments based on three categories (see Figure 18):

- ❖ Thin server and strong client: Data are pushed from stand-alone environments to an integrated one

- ❖ Strong server and thin client: Data are directly handled in the integrated environment from the different standalone environments.
- ❖ Peer-to-peer: Data are directly exchanged through services between standalone environments.

Types of infrastructures	Functions	Characteristics	Diagrams
Thin server+strong client	<p>Clients are equipped with whole CAD systems and some communication facilitators</p> <p>A server plays as an information agent and exchanger to broadcast CAD files and commands generated by a client to other clients</p>	<p>Standalone CAD systems can be conveniently distributed through this mechanism</p> <p>Due to the heavy-weighted client mechanism, it is hard to be migrated to web applications</p>	
Strong server+thin client	<p>The data structures in clients are light-weighted and they primarily support visualisation and manipulation functions</p> <p>The main modelling activities are carried out in a common workspace in the server side</p>	<p>Modules can be rent out as an Application Service Provider (ASP)</p> <p>Data consistency is easily kept since the primary models are created and maintained in the server</p>	
Peer-to-peer	<p>The services or modules of a system can be shared and manipulated by other systems</p> <p>For the Inventor collaborative tool, an MS Netmeeting tool is embedded for application sharing</p>	<p>This mechanism enables a convenient manipulation on remote services or applications</p> <p>Due to the heavy burden of networks, the manipulation efficiency of systems is low</p>	

Figure 18 -Classification of systems infrastructure applied to CAD collaboration
(extracted from [Fuh 2005])

C. IT system in aeronautic projects

Presently, the collaboration at Snecma mainly **uses transfer protocols** (mainly peer-to-peer) for project collaboration. But collaboration in projects often has to resort to personal communication methods (phone, email...) or have to follow design guidelines (e.g. engineering coordination memos, protocols...). The Figure 19 illustrates these exchanges. Presently, both technologies are necessary for project collaboration, because:

- ❖ Concerning transfer protocols: they are essential to transfer product data. Because product data are heavy (often more than 200 Mo), this is the only way to share them. Two kinds of transfers are available:
 - ❖ The first is the direct share of data using servers' connections. Data stored in a server are transferred using transfer line to another server.
 - ❖ The second is the point-to-point connection. It means that transfer is ensured by a transfer line between two computer and / or tools.
- ❖ Concerning personal communication and design guidelines communication, they allow to share the status of the product development and to define coordination of activities.

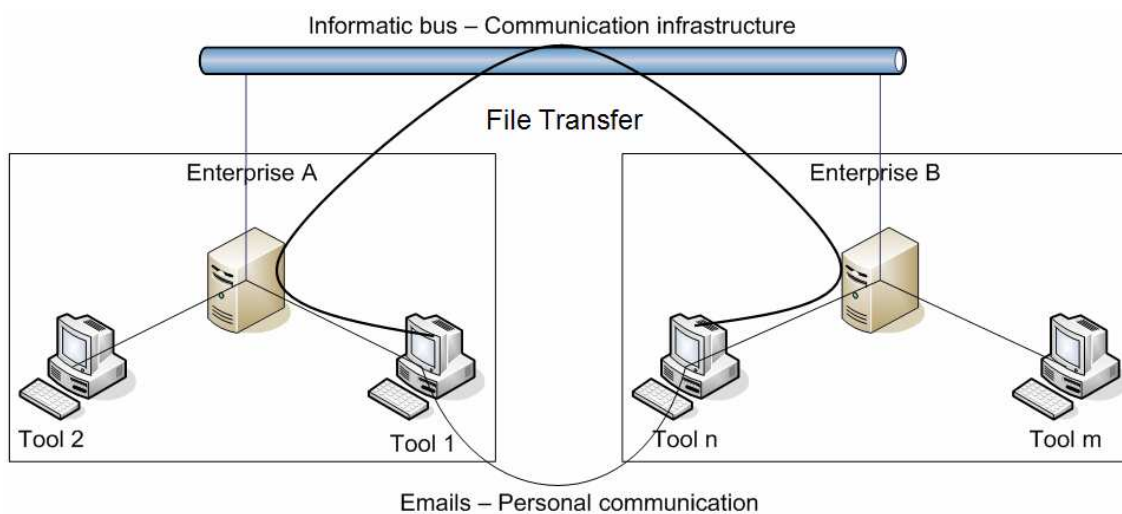


Figure 19 - Schema of IT infrastructure used in collaborative projects

IV. Diagnosis at process level

A. Notion of process workspaces

The project involves different departments working in different processes. Because of this sectioning of processes and activities, the notion of process workspaces appears. These process workspaces are composed of two kinds of business elements:

- ❖ The **activity workspace** in which engineering activities are performed (for example it corresponds to the design of a rotor, a combustion case or to the mechanical simulation of a component...)
- ❖ The **reference workspace** in which data useful for the activities are stored.

The definition of workspaces is also defined by the notion of **application context**. In this PhD, we will often reference the **context**. The context corresponds to

the application environment in which a transformation (e.g. process/activity/action) on product is performed.

B. Workspaces identification at Snecma

This analysis of the collaborative workspaces has been made analysing the current workspaces present at Snecma and the orientation of practices and methods. Since six months, a new environment for design and data management has been set up at Snecma. The representation in different workspaces is the result of this migration. Taking into account this migration of environments and their future development, there is currently at Snecma a coexistence of five main workspaces (see Figure 20):

- ❖ The reference workspace which contains the data to store, validate, share or exchange for product development. Its roles are:
 - ❖ To store the reference data (status of design and industrialisation data)
 - ❖ To provide and send needed data to other workspaces
 - ❖ To validate definition in coherency with configuration management
- ❖ The Configuration workspace contains:
 - The Bill Of Material (useful for the DMU definition)
 - The definition of product articles (stored in the reference workspace)
- ❖ The DMU workspace is the reference for design in context and is related to the Reference workspace and CAD workspace. It is the Digital Mock-Up workspace used for clash, functional, development reviews analysis.
- ❖ The CAD workspace correspond to the design departments workspaces. It is the reference for the 3D definition of the product.
- ❖ The CAE workspace is dedicated to simulation departments. It is the reference for the definition of Finite Element Models and for the simulation activities.

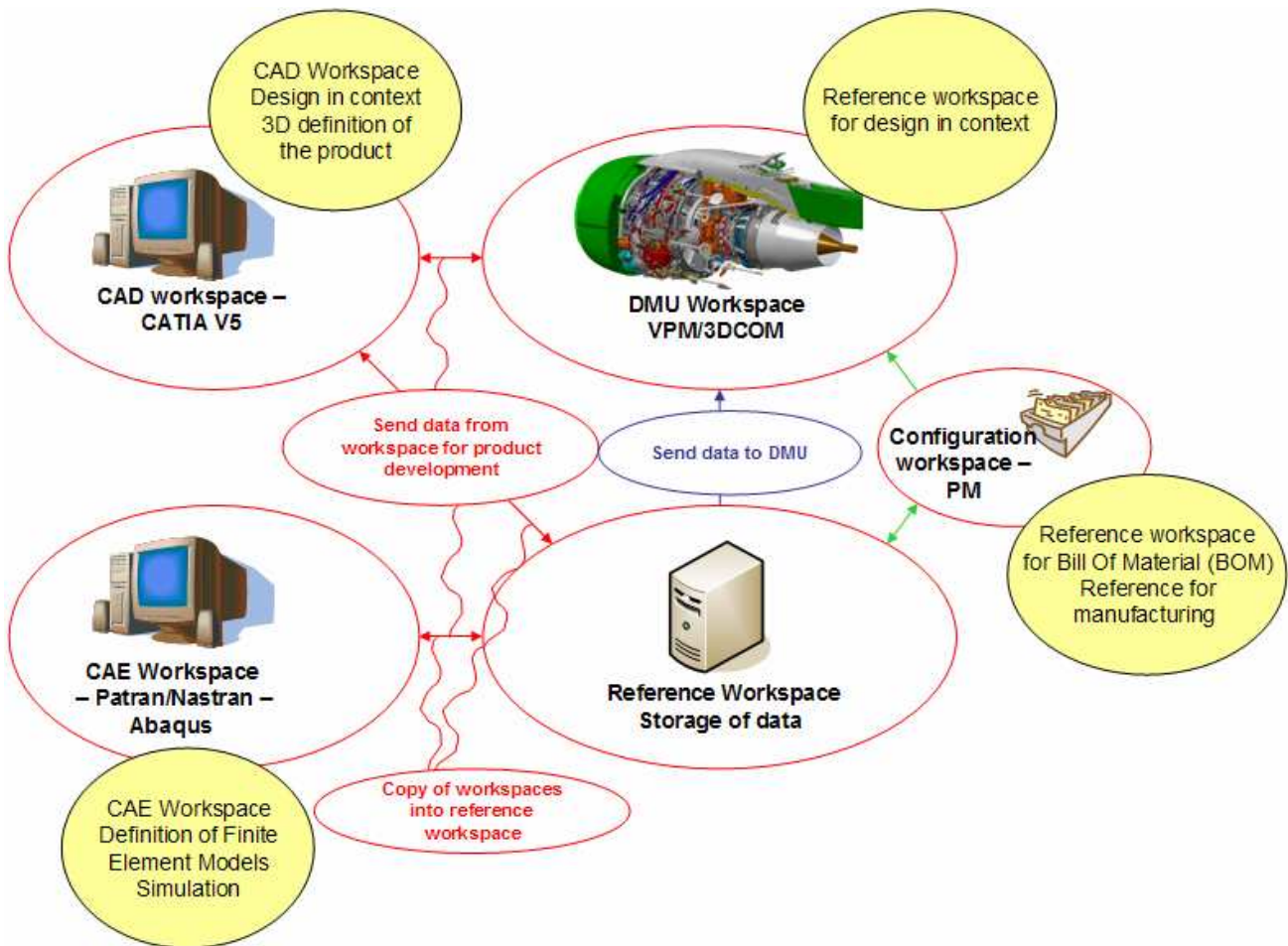


Figure 20 - Illustration of workspaces for product development at Snecma

Presently, this is the reference workspace that federates the different workspaces.

C. The reference workspace: Idealisation of Collaborative discriminant for processes

Indeed, a reference workspace is the centric component from which all the other workspaces are set up. It should permit to:

- ❖ **Initiate processes.** As its role is to gather all the information about the different workspaces, it can provide the necessary input to initiate processes.
- ❖ **Work using a reference on product development.** As the different workspaces are disconnected each other, the reference workspace is then the only workspace which can define the maturity of the product development.
- ❖ **Collaborate with partners.** As the reference workspace is the reference for all workspaces, it means that it contains the more recent data on product

development. It is then from this workspace that collaboration and exchanges with partners are initiated.

V. Diagnosis at engineering level

A. Comment on the heterogeneity of tools for collaboration at engineering level

In the collaborative environment, engineering tools have an important impact. Since a few years, multiple tools have been introduced in the design landscape and have gained importance. Then, industries are now confronted to a wide range of solutions. As collaboration is most of the time associated with the exchange of product data, we are facing the “**technical framework for collaboration between engineering tools**”. This notion depicts the environment defined by tools solutions adopted by partners of a project and that are necessary for the different activity to perform product development [Rosenman 1999, Tang 2001]. This environment as it can be observed in aeronautic industry **is heterogeneous**. There are as many tools as activities to perform. The communication between engineering tools is here an essential requirement for trans-activities communication. The framework is then characterised by the technological connections that exist between tools [Murphy 2002, Peng 1998, Xu 2002].

B. Typology of data in the product development

The typology analysis was made regarding the different departments needs in terms of data. We have represented a simplified typology of data required in Figure 21. The typology presented in Figure 21 illustrates the different data used by the different players of the product development. Here, we can distinguish five main environments:

- ❖ The data related to the configuration manager. These data are mainly Bill Of Material (that describes the product in terms of its assemblies, sub-assemblies, and basic parts) and attributes (significant identification number, id, description, name...).
- ❖ The data related to the designer. These data are specifications (that define guidelines for design), geometric data (3D models and 3D assemblies), context data (which represent the 3D models adjacent to current designer model), and interfaces. Interfaces are a particular category of 3D objects. They define the links between two 3D models. For example they define the limit and positioning of

an object or are specification for the design (for example an interface can define the axis of a cylinder...).

- ❖ The data related to DMU integrator. These data are mainly 3D models (that define the different parts of the DMU), assemblies (to build the DMU in coherency with the BOM), configuration and attributes (which define the different parts involved in the DMU).
- ❖ The data related to simulation engineer. These data are also 3D data but pre-processed (in Finite Element Models – FEM – in order to be used in simulation tool), simulation parameters and results on calculus. These data are also interfaces which define the different nodes to assemble 3D FEM.
- ❖ The data related to CAE integrator. These data are mainly related to the product structure and definition of interfaces for the simulation.

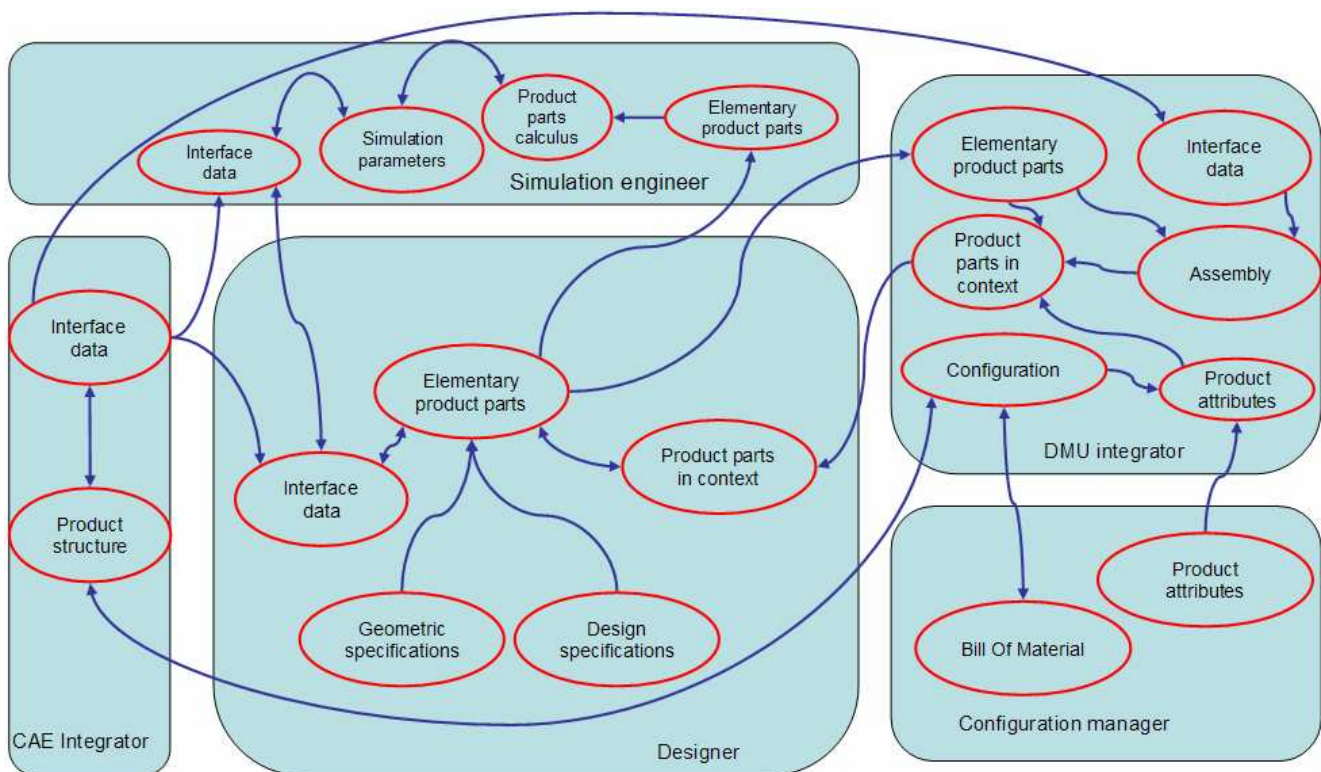


Figure 21 - Typology of data use for product development

The typology exposed in Figure 21 represents what we have been able to observe. At the end of this analysis, we have been able to simplify this one (the result is not presented here due to confidentiality reasons). Using this result, we have been able to define new ways for data dissemination between the different environments and then define new links between data and activities. The result of this new typology has been the basis of our thought about a new way to manage multi-activity data.

C. The data: collaborative discriminant at engineering level

Indeed, from our observation the same data can be used in different activities and by different players of the development. The data is then the **centric business object** that links two different engineering activities. Also, data can be defined as the collaborative discriminant because:

- ❖ At engineering level, the collaboration is performed by exchanging data about the product development
- ❖ Engineering data defines the current status of the product development and then allows to evaluate the maturity of the development
- ❖ Engineering data are the only link between engineering tools and then between engineering processes

VI. Discussion about the results of diagnosis

A. Current issues observed in industry

During the diagnosis we made at Snecma, we have been able to observe three major issues on collaboration. They are identified as the enterprise integration, the interoperability and the associativity. They are currently major research streams well represented in scientific literature on the subject of enabling collaboration and particularly engineering collaboration using collaborative structures and data.

- ❖ **Enterprise integration:** From our observation at project level, we have noticed that IT systems were still not integrated. The major links that exist between enterprises are based on simple exchanges. The enterprise integration, also identified in literature as “**Enterprise Modelling and Integration**” (EMI – *Vernadat 2002*), defines enterprise modelling using specialised methods such as CIMOSA [*Zelm 1995, Kosanke 1999*] and FIDO [*Shunk 2003*]. Enterprise integration aims at defining and characterising the structure and processes of an organisation so as to define the links that can be built between them.
- ❖ **Interoperability:** From our observation at process and engineering levels, we noticed that there was still a lack in integrating efficiently tools and then data. This lack of efficiency is currently a problem while attempting to communicate in a meaningful way. As an example of interoperability consequences, we can illustrate the loss of **data consistency**. The data consistency concerns the quality and the semantic (that refers to the aspect of meaning of the design

representation) of the data conveyed through information systems. In fact, the migration from a system to another causes the problem of data interpretation. In the collaborative development, all the partners must have the same representation of the product at a given time. But the tools used in activities are technically different concerning data representation and most of the time the consistency of a data is damaged. This loss of consistency often produce false interpretation while read.

An example is presented in Figure 22. The red ellipse surrounds a part in surface representation while it was defined in volumes before. This is not an isolated example. The same problem can appear in data management systems where product attributes can be lost during the migration.

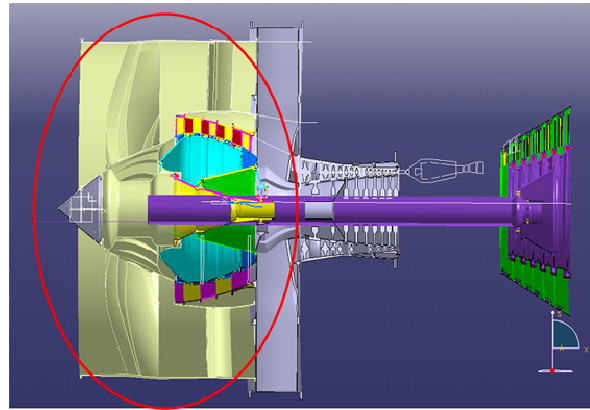


Figure 22 - Example of consistency loss

The interoperability characterises and defines the **ability of two systems to communicate**. For IEEE (Institute of electrical and electronics engineers), it is "The ability of software and hardware on multiple machines from multiple vendors to communicate" (<http://www.computer-dictionary-online.org/?q=IEEE>). For the European Interoperability Framework, "Interoperability means the ability of information and communication technology (ICT) systems and of the business processes they support to exchange data and to permit the sharing of information and knowledge." (*eGovernment Services, Luxemburg: Office for Official Publications of the European Communities, 2004, p. 5*). Interoperability of systems concerns multiple domains. It ranges from the definition of providing PDM communication to the e-Business (see for example ATHENA Project www.athena-ip.org [Goncalves 2006]).

- ❖ The **associativity** of data: From our observation data can be used in several engineering domains (e.g. design, simulation, DMU integration...). Also, the data associativity defines and characterises the different **links that exist between different data**. Associativity then maintains seamless information flow of data between applications. By providing explicit links, it allows for example CAD

geometry to be migrated into analysis models using the most meaningful communication way with the last information of configuration, maturity...

B. Necessary evolution of the collaborative context

1. Managing the data: towards the reference frame for product development

Unfortunately, current distributed Computer Aided environments often suffer from a **lack of coordination** among the current design activities. As this coordination in design is mainly supported using design and product data, their faulty integration causes some problems of coherency and consistency. In addition, there are no means to define and consistently manage a shared work and information space that is crucial to any kind of collaboration and cooperation [Chen 2000, Rousset 2004].

This is how integration of data between systems appeared. Our approach (primarily based on Kovacs and Peña-Mora [Kovacs 1998, Peña-Mora 1996] work about integration of systems) is to define the technical terms of the integration of design practices and of tools used to perform activities. This integration by engineering data management systems first concerns the integrated process management. The need is to provide the partners the capability to evaluate their work and then to be able to launch the stages of design process with the minimum time-out in between. At the data level, this is characterised by the need to permit communication from one tool to another whether they are homogeneous or not. The challenge is to provide the best “data quality” (e.g. for their use in different systems) [Sudarsan 2005, Merlo 2004]. But we must face two well known issues:

- ❖ The **propagation of data** from separate sources into a local collaborative source or a global collaborative source. This means that collaborative data (data that can be published) are replicated from one system to another. The local collaborative source is relevant to make data available for different activities in the same enterprise whereas global collaborative source concerns data published between partners. Data must be consistent to be used in a coherent way [Mitschang 2003, Lee 2004, Aziz 2005].
- ❖ The **controlled access, management and manipulation of data** in the collaborative source. This role still belongs to an “integrator” but the strategy of sharing data is under partners’ commitment.

2. Definition of integrated environments

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. The associated context of this study of collaboration is defined as a **multi-partnership and multi-activity environment** [Dustard 2003, Tang 2001]. Collaboration is the main way to allow multi-participants to control 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 aim of an integrated environment is to provide an integrated view of the product namely a product reference frame to the partners of a project. In such an approach, the collaborative view is no longer owned by one partner but by all the partners. The integrator is then only in charge of the validation. This view is the most adapted to perform the “contextual design”. As soon as each partner is updating the collaborative view, the latter must be straight forwardly made available to the other partners. The constant refinement of the environment is a condition to perform better collaboration, teams are no longer working alone on the design of a module but perform it within the whole product environment and in interaction with the work of others [Eder 1995, Fuh 2005, Rosenman 2001]. The integrated environment also aims at enabling an integration of processes and activities. To be able to concurrently work with a clear project breakdown, it is necessary to organise groups of designers who share a set of common design items. Their work should be coordinated automatically to avoid redundancies (for example duplicated design environments and data) and inconsistencies (for example loss of attributes, 3D models shaded off...).

VII. Conclusion

From our observation **collaboration lacks nowadays an integrated environment** for product design. Then, engineers working for the product development are not certain to get the right information. While analysing the product, we notice that there are several references on the product commonly transformed through the different activities.

A **reference frame appears then as a necessity** to allow the different players both from partners side and activities side to access the same development view. However, one missing compromise is done on resources tools. As we have been able to analyse it in program contexts, there is still not an integrative way to consider

exchange and sharing of data both between partners, activities and engineers. Integration is performed by engineers who need the data. Going towards collaborative structures, the main issue is that current configurations and systems in industry are not flexible enough.

Chapter III: Research methodology

I. Introduction

In this chapter, we introduce the industrial context and the projects we have worked in. This chapter is composed as described by Figure 23.

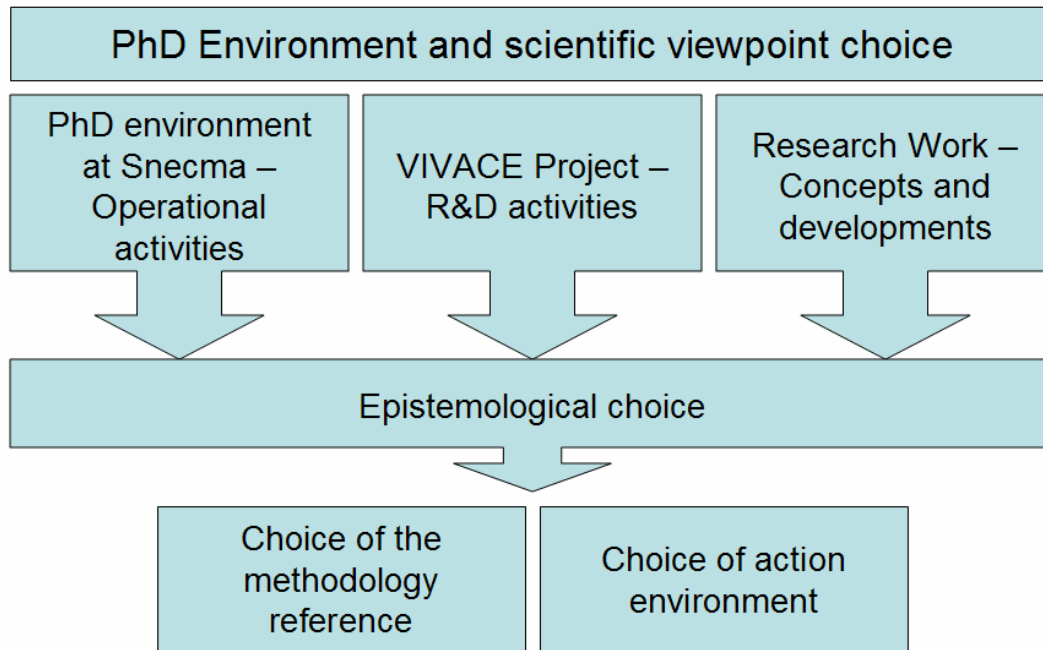


Figure 23 - Process of determination of our scientific standpoint

In a first time, we then describe the different environment in which we have work. Analysing the constraints and necessary orientation proposed by our operational environment, we have then been able to choose the overall viewpoint we expose in our PhD. We then define the methodological reference we use and justify the frame of our intervention.

II. Integration in industrial projects

Our PhD has three interventions fields: Snecma and the operational environment, VIVACE Project (see ANNEXE 1 for project description) and the development of “innovative capabilities” for collaboration and our research work as it is presented in this PhD dissertation. The overall planning of this intervention is defined in the Figure 24.

A. Working environment at Snecma

This PhD has been made in partnership between the **Ecole Centrale of Paris** and **Snecma** (SAFRAN Group). Our work was made at the Digital Mock Up Unit in the department of configuration management. The DMU unit is currently in charge of three major engine programs: the GP7200 (engine for the Airbus A380), the TP400 (engine

for the military aircraft A400M) and the SaM146 (engine for Sukoï regional planes). At the beginning of this PhD, we have dedicated much time to the analysis of Snecma partnership and development processes. We also have participated to the program TP400 in order to better analyse the current collaborative processes and problems while collaborating. Also, during this PhD, we have been able to discuss with people involved in engine programs. At this occasion, we have had a strong collaboration with the mechanical simulation unit (from which we have also analysed needs and current issues in activities and collaboration). Our intervention at Snecma was also the opportunity to collaborate with the methods department. Our research was then **strongly dependant of the constraints and needs exposed by the different Units we met**. This way we have decided to adopt a **dualist attitude**. This seems to be the most adapted to provide a constant confrontation between concepts developed in our work and reality of industrial system. From one part, we faced the definition and characterisation of conceptual systems (including theories about building collaborative contexts) and from the other part the experience of characterising from a physical viewpoint the interoperability and the collaborative system.

B. Environment of the VIVACE Project

Our work in the **VIVACE project** was included in the Work Package 3.4 called **Engineering Data Management**. The overall aim is to develop and validate a framework enabling several partners involved in the aircraft development process to share simulation data (simulation must be understood in a large sense, covering the functional aspects, the behavioural aspects as well as the geometrical aspects) all along the life cycle, starting from early stages. Configuration Management over the various involved disciplines, as well as maturity management, is considered as key elements supporting this framework. Depending on the simulation type, these Configuration Management capabilities must be applied either to the product components (structure components, system/equipments, software components and product trees) – the idea here is to rely on the object concept to reach a full object-oriented context – or to large data sets used within “heavy” simulation types. The idea is thus to manage the link between these two types of simulation, taken into account the traceability constrains required for certification purpose.

In this work package our responsibility was to manage two tasks:

- ❖ The first one is the evaluation of the prototyping of the architecture and results
- ❖ The second one is the aircraft / engine integration (in terms of processes, data, practices...)

Our work was mainly related to this project. Many of the concepts exposed and defined in this PhD have been applied to the VIVACE Project. Indeed, we have participated to the definition of the integrated environment / platform. We have specified and develop the architectural concepts. Also, we have proposed functioning models to implement the collaborative environment defined in VIVACE. Finally, we have proposed the interoperable models to be applied for a cross-activity data migration (e.g. between design and simulation). We have also proposed the first demonstrator for a collaborative environment that has been taken as the initial specification for the VIVACE demonstrator. We have also participated to the development and implementation of the SDM called SimManager.

Our contribution in the VIVACE project was very important. We have deeply participated to the development of concepts and deliverables that were audited by the European commission. Our work in the VIVACE project has also an important impact in our understanding of the collaborative context. Indeed, it has resulted in several meetings mainly with EADS-CCR to identify, create and evaluate the concept of collaborative architecture.

C. Planning of our PhD intervention

	Snecma Intervention		European Project VIVACE			Research Work		PhD development Plan
2004	Audit at Snecma		Introduction to VIVACE			Diagnosis of the collaboration and environment (Chapter I and II) State of the art on collaborative engineering, DMU, CAX, DMS (chapter V)		Audit and diagnosis
	Interview of departments identification of needs and issues in industry. DMU and configuration management Mechanical simulation Method department	Collaboration analysis Processes and protocols analysis	Tools and best practices identification and definition			Diagnosis of the collaboration and environment State of the art on collaborative engineering, DMU, CAX, DMS	Identification of STEP as a solution for the interoperability issue. Study of the Norm (AP203 -209-214-239) (Chapter V and VIII)	Specification and preliminary stage of design
Participation to the TP400 program		Preliminary architecture set up	Analysis of technologies available	Development of collaborative concepts				
2005	Work on mechanical simulation requirements and specifications for data management		Development of the SDM with MSC Software	Models development for the collaborative architecture			Preliminary specifications for a collaborative environment Identification of technologies to be set up (Chapter VI)	Concept development
	Work on implementation of the SDM - 3D models, test data, scenarios	Implementation of requirements for collaborative environment into specifications	Sub-contractor needs	Development of Conceptual models	Interoperability principles and models specification - Tools environment analysis			
			Prototyping of the future architecture, assessment of concepts			Concepts development Methods for managing multi-activity Transactions (Chapter 7)	Interoperable models development - Definition of the "collaborative schema" (Chapter VIII)	
2006	Demonstrator planification and preparation		Implementation of Conceptual models	Definition of transactions and interoperable model development		PhD dissertation		Evaluation
	Work with methods and demonstrator, participation to specification and needs meetings		Prototype set up, informatic developments			PhD dissertation		

Figure 24 - PhD development Plan - Integration of Snecma, VIVACE Project and Research work

D. Choices in research methodology

Our work was motivated by two major objectives:

- ❖ First, in the VIVACE Project, we worked on **innovative collaboration between partners of an aircraft program**. The objective was to propose new technologies and principles for managing engineering data and working environment in the context of multi-partners and multi-activity. The results of such thoughts have also to be available in order to use them in an operational environment. As we were also working on new tools (particularly the SDM), the objective was to integrate the requirements of Snecma in their development.
- ❖ The second one is to develop a prototype concerning a “**new generation**” of **collaborative work**. The idea was to analyse and implement new collaborative practices and functionalities and provide to Snecma a demonstrator in order to analyse the potential of such solution.

III. Scientific viewpoint

A. Epistemological choices

Our research work is motivated by the definition of a collaborative environment for product engineering. This objective is pointed up by the analysis, definition, design of the collaborative framework implemented with methods and processes. This focuses our research towards an adapted framework. But as this study is also a part of an industrial project, we have to deal with the complexity and constraints of engineering environment. Here, the use of the constructivism epistemology seems to be the most adapted to represent our system. **The constructivism epistemology** is an approach that doesn't characterise the “universal model” of a system: it can exist different ways to represent it considering the different perspective it can be associated with. It refers to the hypothesis that a model cannot “correspond to” reality but merely “be viable in” reality. This concept of contextual and multi-view (defining nature of the system, objectives to reach, discipline involved...) modelling seems to be the most appropriate to:

- ❖ **Characterise the physical system** and model the interaction between the components of a collaborative architecture
- ❖ Define **the integration in terms of business views, processes, data** and then tools. It defines critical points to implement with both technologies and

methodologies advocated in literature and implemented structures and systems in industry.

Then our development approach is based on the interaction and the integration of concepts for collaboration with physical components (mainly represented by systems and tools exploited for product development) and engineering developments.

B. Methodological approach

The epistemological attitude we have adopted is a consequence of the coherency need between developments and direct applications in industry. While looking at available approaches, this work is guided towards **system engineering** that is the most adapted to design a relevant solution to our problem of collaboration. Here, we determine our system as the management of the overall information and product data in the entire lifecycle. Moreover, we also deal with the constraints incoming from the extended enterprise.

Then, analysing the collaborative requirements identified by partners, the development of our collaborative solution has to be defined in two different ways:

- ❖ The first is requirements and tools analysis and the definition of engineering environment to determine optimal functions and technological objects to be implemented.
- ❖ The second is the concurrent analysis of both systems present in the enterprise and constraints exposed in the norms to be applied.

In this broad environment of software proposals (CAX solutions, Data Management Systems), we identify system engineering as a necessary approach to build a collaborative infrastructure for product development. Based on the identification of major implementations for communication and exploitation of meaningful objects, we have to adapt our method to define and characterise models of activity context, product and resources. The use of system engineering is motivated by the need to extend collaborative platforms out of the boundaries of enterprises so as to **ensure consistency and integration of resources for projects**.

C. Refining issues perimeter

This part refines the perimeter of scientific issues on collaborative engineering and on the need of collaborative platforms. Here, we explicitly define the major impact of such technologies on design and product development. During a project, different stages exist. The need in terms of data quality and of collaborative needs and

exchanges varies during these different stages. Figure 25 presents the evolution of data (regarding criteria of flexibility, number of alternatives and development resources, of modification and variants) concerning design activities during the different stages of a project. It is inspired from the car industry, but concepts and needs remain the same in aeronautic industry. Considering the pre-project stage, the data exchanged and shared are customer specifications. The need for collaboration starts at this stage. The conceptual design stage and the development stages are also relevant for the use of such technologies. During these stages, the major design and engineering studies are performed. The number of design alternatives increases in order to study, validate and choose the best concept. Consequently, the frequency of modifications also increases, while the flexibility of product design still decreases to converge towards the final product. As engineering activities are supported by collaboration with design tools, the need to manage data is the highest at development stage.

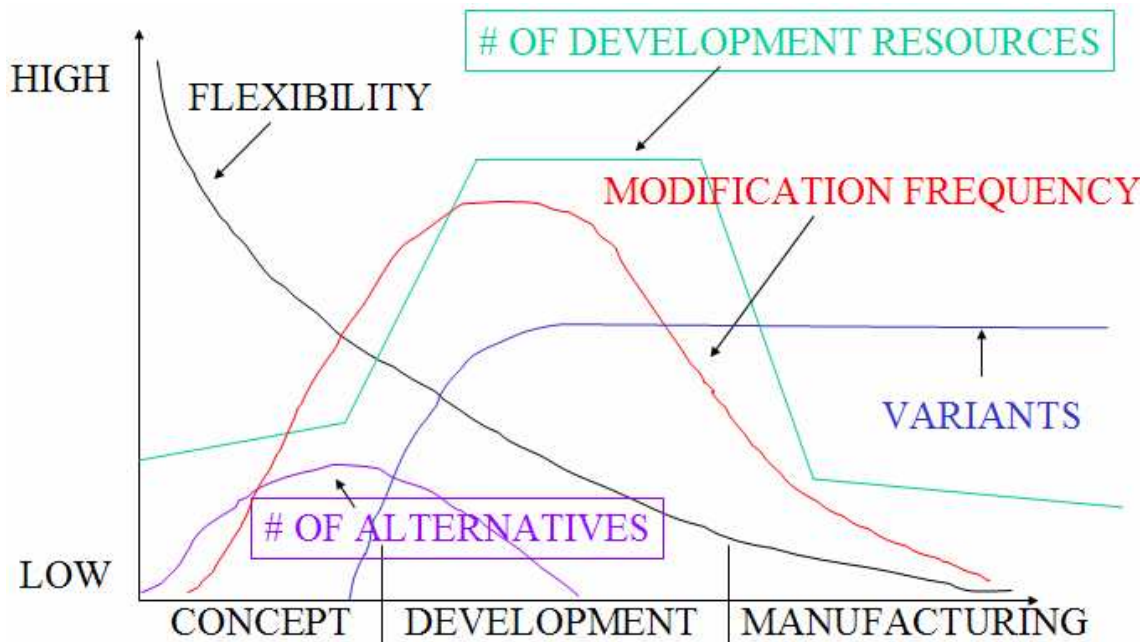


Figure 25 - Design parameters during product development

(Pininfarina - F. Furini –1st presentation on: “Collaborative Design And Learning”

February 16, 2005)

Figure 26 presents the evolution of the cost impact of modifications during the development stages. If a modification requires \$1 at requirements stage, the same modification will cost \$1000 at the production stage. Why this amplification? Because of the different studies and stages generated by propagation after considering an elementary modification.

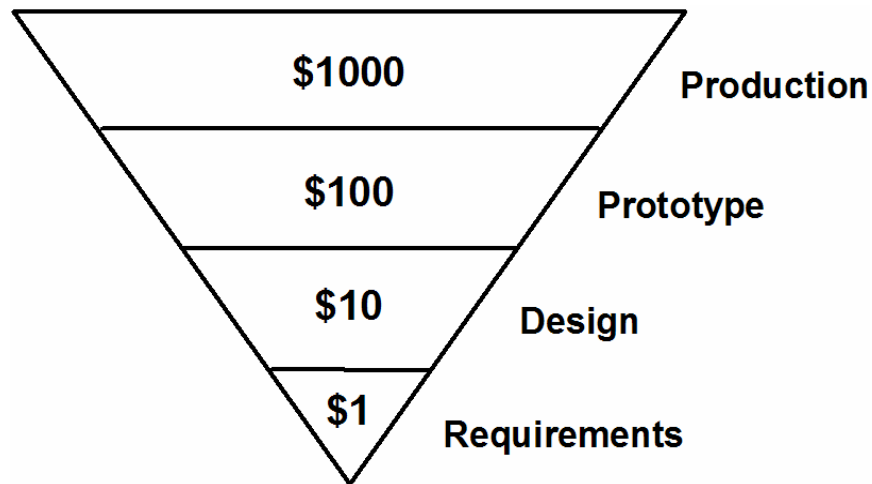


Figure 26 - Cost impact of design evolution during development stages
 (Pininfarina - F. Furini –1st presentation on: “Collaborative Design And Learning”
 February 16, 2005)

Then, we particularly consider the **early stages of the project**. Indeed, it is in these stages that the impact of data management technologies can be the largest. The early development of communication between tools, the creation of a referential, and the development of inter-enterprise processes are a necessary condition to improve collaboration and finally product development. This should also save time-out between stages and then reduce overall cost of product development.

IV. Conclusion

Our work was then divided in two tasks: the development of principles and proposal of new technologies for collaboration and the development of tools directly applicable to an operational environment. Also, we have to deal with the strong interaction with classical partners of Snecma in the VIVACE project. These constraints have obliged us to take advantages of our PhD work by developing conceptual models and applying them in the context of Project.

Our position in projects has also allowed us to have a strong influence on the development of the diverse tasks of the EDM Work Package of VIVACE. We have also try to document each of our development by using as much as possible the existing literature in the domain of collaborative engineering.

Chapter IV: State-of-the-art on multi-partnership and multi-activity collaboration and use of standards for communication

I. Introduction

The present state-of-the-art is decomposed in four major parts (see Figure 27). This state-of-the-art results of the logical analysis of the current collaboration in the industrial environments. First, analysing the collaboration means analyse tools and their relationships. The analysis of the collaboration is at this stage decomposed into two fields: the CAX activities (in the box 1) and the DMS (in the box 2). Then, we focus on the definition of the backbone that supports these tools and finally activities (in the box 3). We finally detail the standardisation and the particular case of STEP (ISO-10303 [ISO10303-1]).

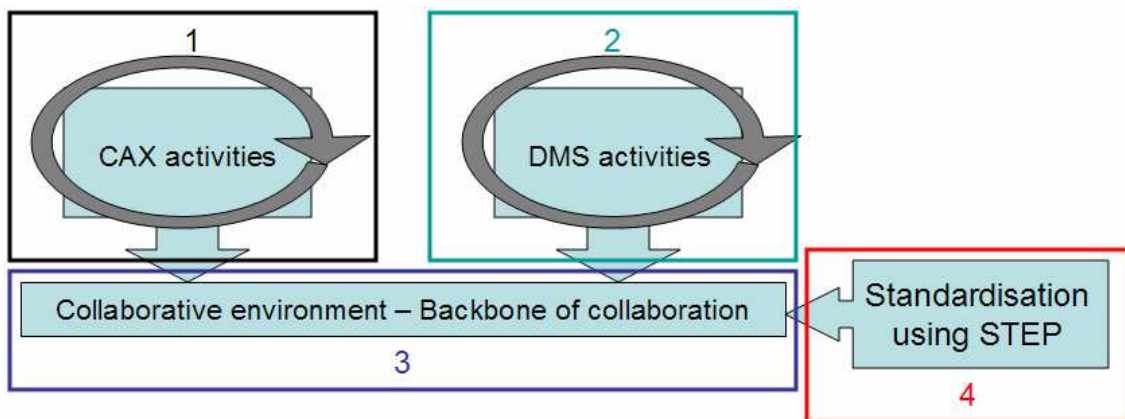


Figure 27 - State-of-the-art decomposition

II. Engineering tools in the Collaborative environment

A. CAX tools and collaboration

1. Typology of collaboration with CAX tools

Li [Li 2005] analyses the different viewpoints associated to CAD activities and especially the collaboration using 3D representations. His work is based on the statement of systems heterogeneity, of the use of tools and major functions that need to exist in such systems. To better characterise the environment, he has adopted a collaborative description defining two approaches: the **horizontal and the hierarchical** collaboration. Based on his work, we have decided to reuse and adapt such a classification to describe collaboration with CAX (see Figure 28):

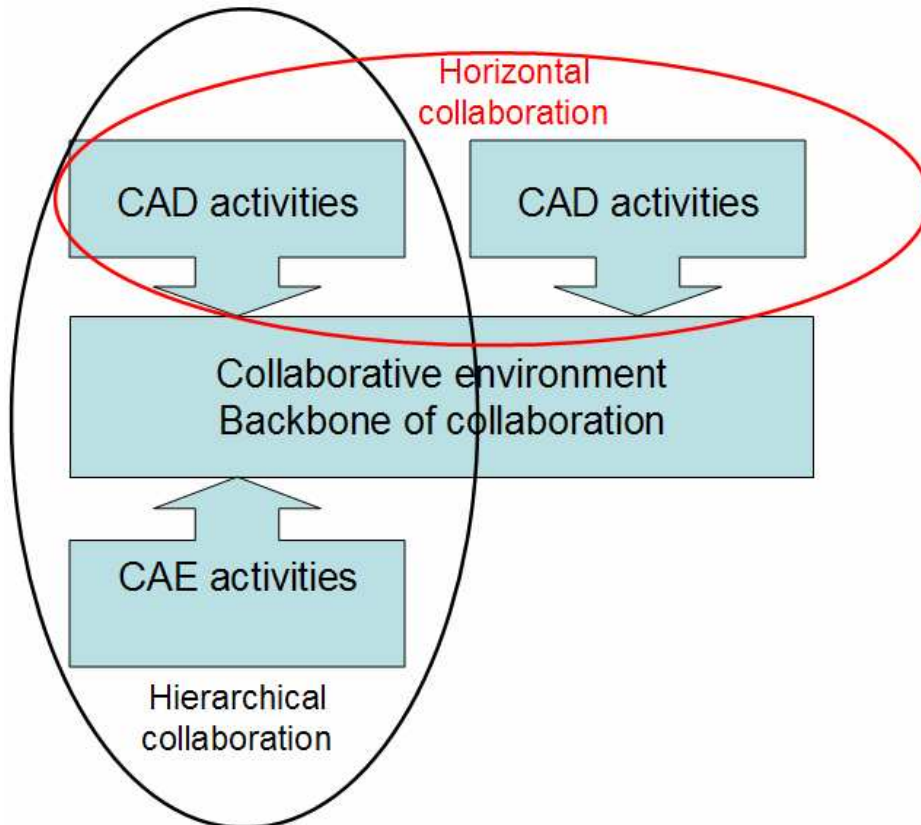


Figure 28 - Illustration of horizontal and hierarchical collaboration with CAX tools

- ❖ **Horizontal CAX collaboration:** Horizontal CAX designates the application of associating teams from the same discipline or which activities are located at the same process level to carry out a particular aspect of the development. This means enabling two collaborative teams to work on the design of the same object in synchronous or asynchronous ways [Fowler 1996, Rosenman 1999]. For example consider the development of a product part on which two teams (e.g. one from the enterprise responsible of the design and the other from a sub-contractor) are working. These two teams work to develop the same part of the product with equivalent tools. But the modifications performed by each teams and the integration of their work is made independently. This implicates two major topics:
 - ❖ The definition of **the collaborative view** associated: It means that the different working teams should access the same view of the product. Also identified by data reference frame, it provides the same context for designing [Peng 1996, Fuh 2005] (see collaborative view in Figure 29).
 - ❖ The definition of **the collaboration support**: As a consequence of making teams working together and collaborating, it defines the collaborative structure and allows interoperability and systems integration

[Augenbroe 2004, Valckenaers 2003] (see collaborative support in Figure 29).

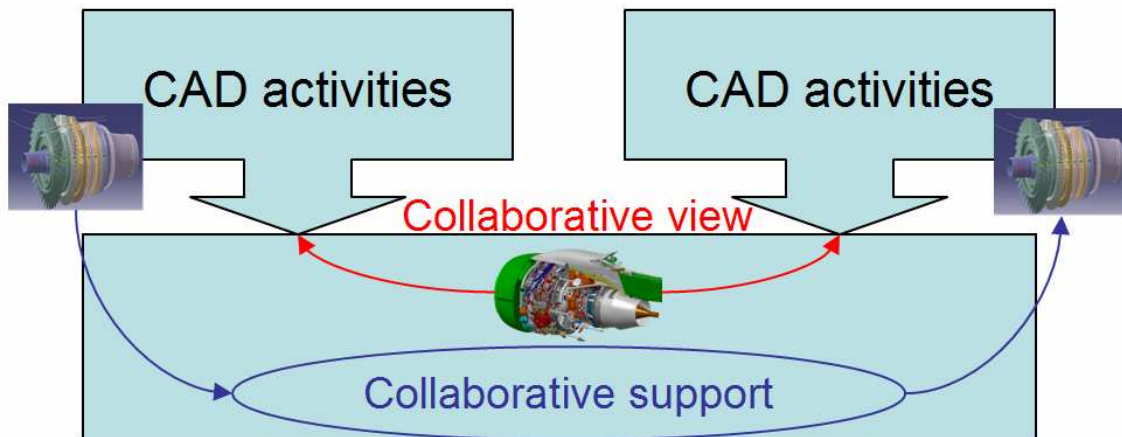


Figure 29 - Representation of horizontal CAX collaboration using the example of CAD

- ❖ **Hierarchical CAX 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 in design processes to simulation processes [Moorthy 1999, Werner 2004]. This implicates also two major topics:
 - ❖ The definition of **the integrated view**: To integrate processes and resources, a major need is to define a common integrated view to be propagated between activities. This relies on a common referential that should be the same as the one described for horizontal CAX [Moorthy 1999, Wang 2002].
 - ❖ The definition **of migration principles**: The CAX tools exploiting data in hierarchical collaboration are not the same (for example hierarchical collaboration between CAD and CAE). Then, the support for the development (e.g. common 3D model or DMU) needs to be transformed into the most meaningful way to be exploitable whatever the application required [An 1995, Shephard 2003, Eben-Chaime 2004].

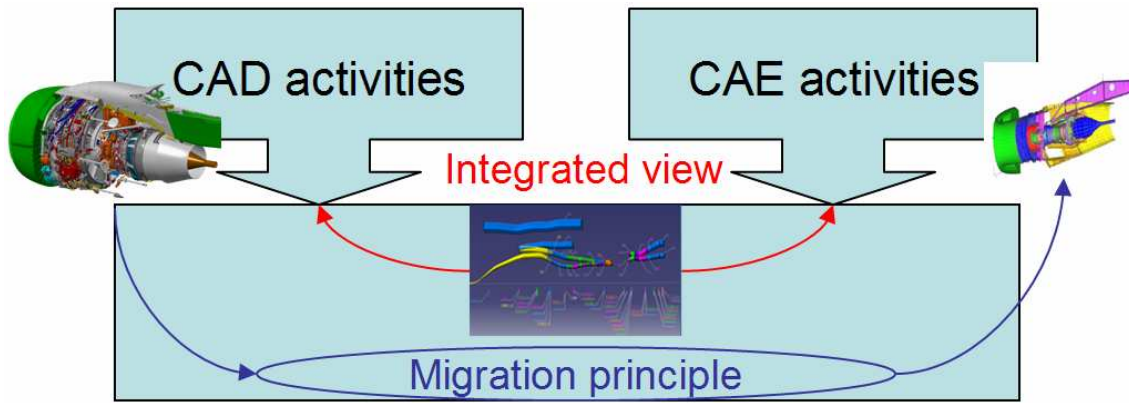


Figure 30 - Representation of hierarchical CAX collaboration using the example of CAD and CAE

2. Collaborative streams using CAX

In the field of collaboration with CAX, many solutions are proposed for the definition of integrated and collaborative environments. These environments are supposed to provide the “collaborative interfaces” that guarantees the partners to access the same data on the most recent release. The collaborative CAD environment is defined as a structured area in which properties and geometrical definitions (including collaborative interfaces) allow partners to associate their design objects on the whole product [Fuh 2005, Li 2005, Zhu 2004]. This contributes to the definition of the **integrated context and the design in context**:

- ❖ Integrated context corresponds to the collaborative view on the entire product.
- ❖ Design in context corresponds to the design definition of a partner in the whole product environment.

Integration consists in merging two business contexts (e.g. design context, mechanical simulation context) and to create only one meta-business context (for the example of design and simulation contexts, they can be merged in a context called “development and analysis”). A business context is defined by activity and process that define it. It defines the main characteristics of the integration in part of the information and data needed to answer the development need [Merlo 2004, Rosenman 1999].

Concerning CAD/CAX, integration ensures the development of the working environments with the virtual prototype. Then, this integration allows partners to access and use the CAX/CAD data he needs. "If an integrated CAD system, based on a CAD framework, is adopted, a common design database ideally holds all the information needed to run any CAD tool. Each tool in turn needs an interface towards the database." [Zanella 1995] "Hence, the new open CAD environment must provide strategies to integrate both the system and data." [Jasnoch 1996].

3. Industrial gap to reach a better collaboration with CAX

Studying the literature on collaborative CAX and auditing the industrial contexts of Snecma (especially the collaboration with partners) has conducted to the identification of a gap between conceptual studies and real capabilities of system implementations in industry. Then two major topics have appeared concerning this need of implementation. The first one concerns the capability to exploit a referential for CAX activities and the second one the system support for these environments.

- ❖ **Referential for CAX:** Attempting to design and analyse a product in a collaborative environment, each partner needs to get a unified view of the product (example of the DMU exposed by Sadoul and Demouzon [*Sadoul 2000, Demouzon 1998*]). CAX referential is then composed of three main views: **The integrated environment, the interface notions and the DMU referential.**
 - ❖ Considering the integrated environment (following the description of Mervyn or Hiufen [*Mervyn 2003, Hiufen 2003*]), it should be built as the collaborative backbone. Based on common representation and view shared by partners, it favours the integration of data in a consistent and meaningful way.
 - ❖ DMU referential [*Sadoul 2000, Demouzon 1998*]: Attempting to create a common and integrated view for all partners in a project, the DMU appears as the best candidate to be the referential for design and collaborative activities. The DMU referential is one of the first tools which allow concurrent engineering. The opportunity of its exploitation is strengthened by the fact that DMU can be built and defined early in the project.
 - ❖ Interfaces are 3D objects that result in a good understanding of the design context by the partners. In design, interfaces are objects that specify the location of the different collaborative items [*Thong 2002*]. As a consequence, such interfaces can also be migrated to other activities to provide for example an add-on of behaviour regarding the simulation topic.
- ❖ **System support:** This is the supporting object that implies the factual integration and communication between the different objects and entities of the system. Regarding the system support, we notice that it is submitted to two integrated notions: the first one concerns the **operational system and the second one the interoperable system** [*Merlo 2004, Fuh 2004, Wognum 2004*].

- ❖ Operational system: This refers to the set of activities that can be performed using a unified view of the product. This is the concept that adapts multiple CAX objects using the same referential and view. For example, this concept should allow the migration from DMU and CAD objects to Finite Element Models (FEM) and integrated FEM [Moorthy 1999].
- ❖ Interoperable system: This refers to the capability to build a multi-view (activity and engineering support) and a multi-partner reference for communication. Based on a common description of semantic objects used, this facilitates the migration between the views necessary for the development [Valckenaers 2003, Augenbroe 2004].

B. DMS tools and collaboration

1. Typology of collaboration using DMS

So far, data management has been devoted to the design use. The need of industries to also manage simulation parameters, models and results has resulted in defining simulation data management. Indeed, the development of such system increases the heterogeneity of the environment [Bouras 2006]. This also results in a complexity of classification for such tools. Based on the work done by Li [Li 2005] for CAD systems, we can adopt the same approach of **horizontal and hierarchical** collaboration for data management systems (see Figure 31).

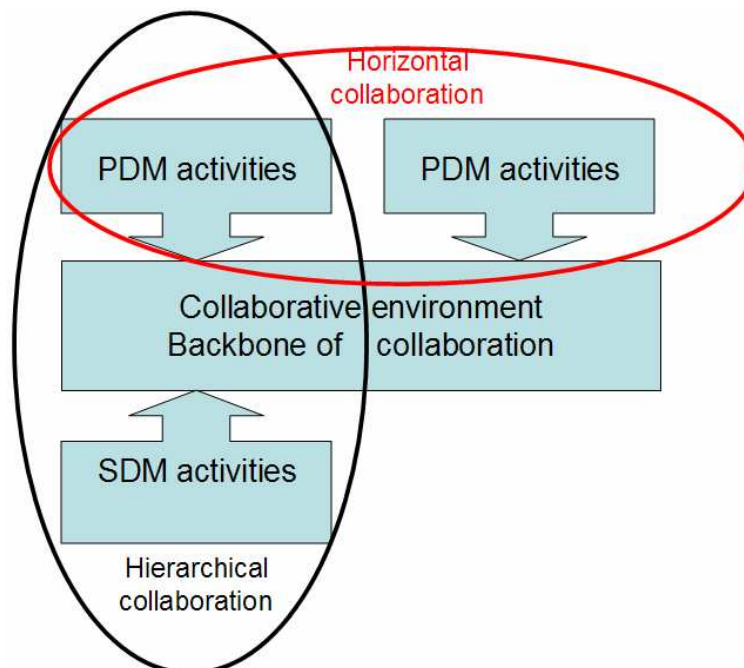


Figure 31 - Illustration of horizontal and hierarchical collaboration with DMS tools

❖ **Horizontal collaboration:** This corresponds to the communication and integration of data between equivalent tools. Usually used at the same process level and for the same activity, collaboration corresponds to the exchange of “development packages” that are integrated by the different partners and teams to rebuild the common view of the product. To date, this collaboration is performed in two ways:

❖ The first way corresponds to the **use of translators** that ensure to transform data output of a system into data input of another. Using this method requires using N^2-N translators if we consider the collaboration between N partners each one using a different data management system [Ruland 1995] (see Figure 32)

❖ The second way is performed using **standardised file** description between partners. Partners that export data need to pre-process them in order that they can be reused and integrated by the other partners [Liang 1999, Zha 2002]. Partners have then only to post-process data to integrate them in their system. Using the previous example if we consider N partners, we need to define $2N$ translators.(see Figure 33)

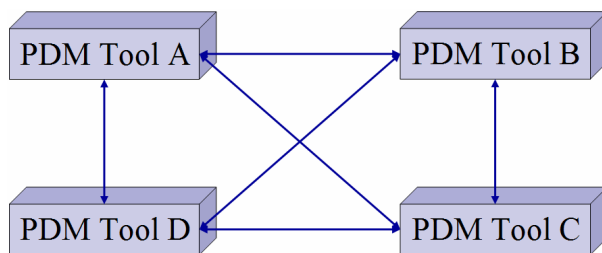


Figure 32 - Horizontal collaboration using translators

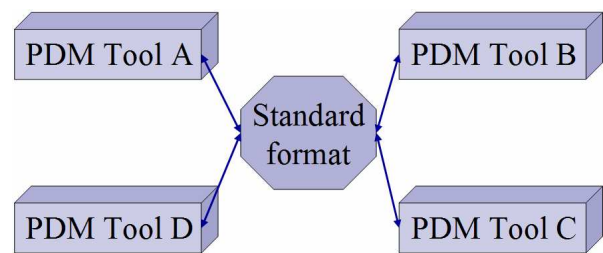


Figure 33 - Horizontal collaboration using standards

❖ **Hierarchical collaboration:** Today, data management principles belong to design activities, collaboration using different tools and attributes typology is not well developed. The appearing of simulation data management systems tends to change this. As simulation depends on design, the need to integrate the different data structures is appearing. They are presently performed using transformation codes to gather necessary information. Particular aspects and attributes such as **simulation parameters are then still not integrated**. Considering the migration between heterogeneous categories of tools, integration principles of horizontal collaboration remains applicable [Pernul 1995, Oh 2001, Burkett 2000].

2. Collaborative streams using DMS

While looking at present literature review, we notice that we are still in the logic of the horizontal integration which mainly corresponds to the definition of integrated design. Looking forward and identifying new trends and developments for simulation activities, the hierarchical collaboration is appearing. This kind of collaboration is necessary to provide capabilities of semantic interpretation. This semantic interpretation is finally made concrete with the integration of data in the different data management systems.

Up to now, major studies have focused on three major techniques [Boujut 2002, Chalmeta 2001, Dustdar 2003, Fuh 2005, Peltonen 1993, Zhao 2003]:

- ❖ The definition of a common repository for data, also defined as a “**shared database**”.
- ❖ The developments of **virtual platforms** that link together different objects contained in partner databases that are accessible using web service.
- ❖ The application of **integration strategies** using translators so as to provide the data in the right format for derived application on other tools.

These different techniques are also submitted to the following requirements [Li 2005, Peltonen 1993, Tang 2001]:

- ❖ To provide a constant workflow of data between partners. Indeed, when a data is created, it is expected to link it to a referenced object. Referenced objects permit to disseminate an integrated environment into partners' environments.
- ❖ To create associative links between data. The objective is to create constraints between data so as to link them in a meaningful way. This approach is often studied through the use of data models representing the semantic parsing of data extracted from tools [Case 1996].
- ❖ To filter out context. It means to be able to determine which data can be applied in which process and for which activity. Indeed, this links the process to the product and the integrated resources. This is used to define strategies that have to be applied in databases and semantic links built on data models. This strategy on data is crucial for multi-view engineering [Boujut 2002].

So, the main related issues on such approaches for collaborative engineering using “integrated environments” are [Teeuw 1996]:

- ❖ The development of a **semantic approach for collaborative engineering**. This is related to the analysis of data needed for activities and of the capabilities to link the semantic created by the different activities [Case 1996]
- ❖ The development of the integration of data. This is related to the definition of the **application of a standardised language** between the different applications [Fuh 2005].

The increase of COTS solutions in the engineering domain since the 1980's has enhanced communication needs. This way, some strategies have been followed such as:

- ❖ **Standardisation studies and applications** [An 1995, Burkett 2001, Peng 1998] - Case of STEP (STandard for the Exchange of Product Model Data) since 1984, or XML (eXtended Mark-up Language) that became a W3C¹ recommendation on February 1998. The definition of standard for communication imply that data are expressed using the same language schema. Then they can be linked using this common language schema. Application of standardised language between tools aims at providing the referencing between semantic objects to result in their linking through a common ontology.
- ❖ **Application of databases strategies** for collaboration and communication [Jasnoch 1996, Jeng 1998, Lim 2000 and Lim 2004] – The literature review shows case studies and implementation on databases architectures for interoperability and consistency and relation instances from heterogeneous databases for object oriented relationships. When considering such databases, the objective is to classify and characterise these relations by using some relational schemas to link instances in database.
- ❖ **Development of information management and instantiation of semantic objects** in product data models [Peltonen 1993, Moorthy 1999, Hameri 2001] – Bibliographical review presents case studies on document strategies and association of product models instances. Such managements and strategies define transfer protocols using data, and distributed architecture to support communication, exchange and sharing of data between partners and activities.

¹ W3C was started in 1994 to lead the Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability

3. Industrial gap to reach better collaboration using DMS

Analysing collaboration with data management systems in literature and regarding their implementation and fitting with industrial system, we can notice that some studies are still not well developed or don't exist. Major authors focus on the development of the communication between tools (that is quite essential) [*Bloor 1996, Teeuw 1996, Leong 2002, Lee 2004*] or have more architectural considerations [*Jasnoch 1996, Madhavaram 1996, Lim 2000, Hubel 2001*]. Indeed, in industry, efforts are made on these topics but as we have identified, they have to evolve on two major topics:

- ❖ The first one is the data semantic broker: A recent evolution towards new technologies of data management let us think that **data semantic is not enough decomposed and characterised** in terms of entities to be applicable in multiple cases and/or migrated in different tools [*Métais 2001, Rousset 2004, Yoo 2002*]. Some efforts have to be made in terms of:
 - ❖ **Semantic for communication.** Indeed, some techniques already exist. However, practices in industry make that each time a new partnership is considered, a new definition of the formal semantic has to be recomposed. The main solution to shrink the time taken by those proposal and its corresponding developments is to define a common semantic reference adapted to the definition of the product for any available environments.
 - ❖ **Semantic for redesign environments.** The environments proposed by tools (such as the comparison of PDM and SDM systems) are not composed in the same manner and then their redesign is not submitted to the same semantic rules. Their implementation should consider the definition of mapping between the different semantic objects that exist.
 - ❖ **Data referential and integration.** A data referential provides the essential structure of data and the template of redesigning its structure. Then, integration is in charge of the interpretation of the data regarding the template to rebuild the data structure in a given system.
- ❖ The second one is to consider **rules and methods** within data management systems and their application for collaboration. As data management practices and methods can be assimilated as processes to rebuild data from different environments [*Chang 2001, Kornblum 2001, Beynon-Davies 2003, Goranson 2003*]. Two topics are emerging:

- ❖ System supports for integration and rules implementing the collaboration upon such structures. The need is to define more precisely the supporting systems that are going to ensure:
 - ❖ The management of **relationships and dependencies** at different levels of granularity across different domains.
 - ❖ The **synchronisation process of tasks and data flows** with process modelling across sub-processes.
 - ❖ The inclusion of **configuration management and the dynamic definition of structures to accommodate multiple views.**
- ❖ The architectural considerations to build the collaborative exploitation and interoperability upon systems:
 - ❖ The support for integration with other applications
 - ❖ The definition of a standardised communication infrastructure between the components and with external modules.
 - ❖ The definition of a modular architecture so that components can be easily plugged in and out of the system.

III.The influence of collaborative platforms and “integrated environments”

A. Collaborative Engineering versus Data Management Systems

In the literature, collaborative engineering mostly refers to the definition of methods, techniques, and processes that intend to link two activities. Indeed, Data Management Systems are seen as the executive tools of the engineering. Then, these two views describe different levels of integration need. Considering the exchange of product environments between partners, data management systems are then seen as the opportunity to create the collaboration [Lehmann 2005, Merlo 2004, Xu 2002]. From this viewpoint, data management includes the principles, technical objects and methods for collaboration. The standpoint we adopt regarding collaborative engineering is to consider tools in the way they facilitate collaboration. From the design viewpoint, this is characterised by:

- ❖ How data management systems can be integrated in the most collaborative way? Which are the methods to analyse technical objects? And how is created coherency between data and meta-data used during project?

- ❖ How could data management systems and associated tools (mainly CAX tools) reproduce the collaborative engineering view?

In the literature collaborative engineering is often represented by the integration of data. Bergman & Baker [Bergman00] describe a « shared virtual workspace » that aims at enabling the data exchange through integrated environments. Each environment is submitted to its own functioning laws.

We may summarize the expected results on the performance of collaborative engineering through data management and integration considerations by the following objectives:

- ❖ To deliver management mechanisms required for heterogeneous sets of data
- ❖ To build an environment including the required data translators to translate data from a system representation to another
- ❖ To develop inter-discipline mechanisms to retrieve data quality and consistency
- ❖ To implement rules on managed data, such as maturity rules for data exchange
- ❖ To manage collaborative data in a consistent way

B. Collaborative Engineering versus Collaborative Process

A collaborative process corresponds to the dynamic approach of collaboration. Collaborative processes define the activity sequences that describe the collaborative actions on product development. Collaborative processes are built upon the need to:

- ❖ **To maximise the flexibility and the adaptability to environmental changes**
- ❖ **To develop a pool of competencies and resources**
- ❖ **To optimise the overall supply chain of the product development**

In the literature, collaborative engineering ensured by processes is mainly described as an integration of business processes which define and lead the development. Many methods are then implemented to integrate the enterprises. Gou et al. [Gou 2003] propose to structure distributed business processes. Based on a UML description, they propose a multi-agent solution to develop operations on business processes. Loureiro and Leaney [Loureiro 2003] propose an approach of a “total view framework” that defines enterprise model at different levels. Their approach relies on

using the CIMOSA [Kozanke 1995, Zelm 1995] approach implemented with the Zachman framework [Perkins 1997, Zachman 1987 and 1996].

The architecture of such a collaborative framework relies on the definition of multiple layers based on the integration of multiple components. Then, the links between components serve the business processes with the use of models templates [Perrin 2004, Loureiro 2003, Sudarsan 2005].

C. Evolution towards collaborative and integrated environments

A collaborative platform appears now as a necessary element to permit data flows between applications used in enterprises. The heterogeneity and diversity of the software used during the different stages of product development have increased their need of “collaborative platforms”. Their role would be **ideally to integrate and associate data created and managed during engineering activities**. When considering data management systems, integration and association means that all partners’ data can be merged to define the whole product. Indeed, considering the data management systems, all engineering definitions and attributes of the product can be integrated and used in different systems. As the number of attributes attached to product is also increasing with the evolution of data management requirements and data exchanges, data management between partners needs more complete interfaces. As a consequence, there is also an increasing need to provide control on 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 application (design, simulation...) [Berden 2000, Fuh 2005 and Li 2005]. The major objective of an integrated infrastructure is to provide efficient data management systems such as those proposed by classical Product Lifecycle Management (PLM) systems but in extending them to collaborative environments. Integrated infrastructure must also **provide shared spaces for the different partners, using dedicated workspaces for the different activities** (e.g. workspace for design, simulation...). That way, collaborative platform must allow tools communication and data integration in the most meaningful way. The platform must provide services to create requests on activities and define inter-enterprises workflows. Furthermore, they have to comply with the NIST (National Institute of Standards and Technology) recommendations mentioned by Sudarsan et al. [Sudarsan 2005] on:

- ❖ Formal semantic or appropriate ontology to result in automatic reasoning
- ❖ Generic: deals with general entities so as to be universal enough

- ❖ Repository: to contain all the data relative to product
- ❖ To foster the development of novel applications and processes
- ❖ To incorporate the explicit representation of design rationale
- ❖ To convert and/or interface the generic representation schemes with a production-level interoperability framework

Following the thought of Perrin on collaborative infrastructures [*Perrin04*], they must create networked activities using product data representation (referential) provided by the collaborative workspace. For the aeronautic industry, an integrated infrastructure must comply with 6 major requirements:

- ❖ **Common data referential:** To store and retrieve product data in its entire lifecycle (for example, simulation activities presently lack the notion of referential and the association of the current design of a product within all simulation activities doesn't exist). This should also allow the coherency of data while considering product's characteristics (presently the coherency of product configuration between design and simulation is implicit and defined by engineers that need it).
- ❖ To **manage information between partners:** ensured by standardisation of collaborative data, this should define the meaningful way to manage data. Models should describe the attributes attached to product and must manage the attributes links between partners systems.
- ❖ **PLM functions:** must define the major actions on data and guarantees the management all along lifecycle.
- ❖ **Data context:** must permit to retrieve the context for data use. This defines necessary elements (e.g. process, activity, data, parameters and models describing the product) for data use.
- ❖ To provide **data associativity:** must define links between data created for activities.
- ❖ Define **flows and processes connections** (engineering requests, engineering validation...): must determine the different ways to access partners' processes.

D. Major studies on collaborative and integrated environments

From our literature review, we remark there is a growing need of analysing and providing methods to (deeper) integrate product development [*Perrin 2004*]. To

date, many studies deal with collaboration through CAX (mainly CAD) tools. We can quote for example Rosenman [Rosenman 1999], Fuh [Fuh 2005], Li [Li 2005]: they explore the different functions necessary to ensure the collaborative development of the product and especially design. They propose new methods and processes to make advance in collaboration using CAD. Rosenman [Rosenman 1999] proposes a description of the different domains using CAD and defines the properties for communicating in collaborative design. He exposes models to propose adapted functions and behaviour for collaboration with CAD. Fuh [Fuh 2005] and Li [Li 2005] propose a more pragmatic approach of the resolution of collaboration with CAD. While Fuh [Fuh 2005] explores the way to ensure collaboration using 3D visualisation with the implementation of architecture based on web and on exploitable formats for visualisation, Li [Li 2005] defines collaboration using data communication. Analysing visualisation capabilities he proposes an analysis on the topic of data sharing and exchange.

Other authors propose a higher level of analysis based on processes and activity analysis to define collaborative design. Many authors such as Monell [Monell 2000], Case [Case 1996], Favela [Favela 1994], Wang [Wang 2002] have worked on defining the common product representation in a collaborative environment. They analyse the trends and capabilities of coupling technologies of using hypermedia for design. Wang [Wang 2002] proposes a review of technologies already implemented based on web interfaces. As a consequence of its popularity, the web is often used by authors as a support for hypermedia processes and especially for collaborative design.

Finally, current trends are moving forward the creation and management of integrated design environments such as those suggested by Benford [Benford 1997], Aziz [Aziz 2005], Toye [Toye 1993], Yeomans [Yeomans 2006], Fuh [Fuh 2005], Li [Li 2005]. Integrated environments for design collaboration appear as a crucial need to allow design and process integration. Looking at their associated functions, such environments rely on collaborative and shared databases.

Main capabilities illustrated here are basic requirements to build an integrated design environment based on common visualisation of the product by partners. Moreover, integrated environments must provide also a multi-activity (really CAX oriented) platform to transfer and manage the use of tools using a common reference.

This contributes to the definition of integrated infrastructures for design in collaborative environments with multi-view application (design, simulation...) purpose [Berden 2000, Fuh 2005 and Li 2005].

Looking forward, collaboration with data management systems is not only initiating collaboration between databases, it also means to make them communicate. Three methods are available:

- ❖ First, the collaboration needs a **common language reference**. Today, available solutions have their own communication languages and models to define the different attributes of product. In this method, standards and translators are necessary. Their role is to retrieve in data files the semantic objects (e.g. meta-data or attributes) to transform and characterise them into another formal semantic that can be understood by another system [Altmann 2002, Liang 1999, Zha 2002].
- ❖ Second, collaboration needs that **an integration process** be able to duplicate attributes and meta-data from a system to another. Currently, analysing the different tools and data to be instantiated, we notice that analysis and integration of data is made step-by-step in the different tools. This integration performed step-by-step is a condition to integrate only required data. This has to be defined for each system and each partner for the referential for tool attributes to remain the same [Hameri 2001, Moorthy 1999, Peltonen 1993].
- ❖ Third, collaboration needs a **semantic interpret** to be able to interpret the data managed. The first requirement is to provide to share necessary and sufficient data. Due to intellectual property rights, all data are not shared by teams and partners. In other words, looking at activities performed to develop the product, we notice that the need of activities (in terms of data) is not the same. The semantic interpret has an important role to play in order to permit migration of requested data so as to perform an activity [Davis 2002, Augenbroe 2004].

IV. Collaboration with standards to suppress interoperability constraints

A. Standards and main industrial issues in communication

The increase of software and engineering solutions has increased the systems heterogeneity. Currently, industrial systems are facing problems in:

- ❖ The capability of communication: Tools exploit different semantic and data representation to describe an item. As there is not a standardised way of describing engineering data, software vendors then exploit different schemas that

are not able to communicate with each other. This problem is described by two major issues:

- ❖ The interoperability: This means that communication between the different systems is not or badly ensured. Then the representation of data has a poor quality which does not allow a good use [Augenbroe 2004, Valckenaers 2003].
- ❖ The integration: Is the idea that data or information on any given device can be read or manipulated by another device using a standard format. Currently, a meaningful representation is not obvious. While attempting to make two schemas communicate, we are facing the problem that only few semantic elements are recognised and then the interpretation can be shading off [Mervyn 2003, Paige 2000].
- ❖ The capability of operation: As tools uses a given range of semantic objects, there is no common representation of data from a system to another. The problem concerning exploitation is that they are only applicable to a small range of activities [Linnington 2004, Pierra 2000].

Having a look at these problems, industries have faced the need for a common language and semantic reference to communicate using the same data representation.

B. Collaborative streams using standards

The role of standards and formats is to permit systems to **communicate in an efficient way**. Because products are becoming more complex, the amount of product data is increasing for a project. One main trend can be observed: the format of the different data. Teeuw and al. [Teeuw 1996] underline this aspect of data formats through the recommendation of standards use. Because enterprises use different COTS tools, one problem is to permit the communication in a semantic way between all applications. This way, two main approaches are available: the data translation and the data sharing [Liang 1999].

For data translation, the product data are translated from an application A to an application B by using a pre-processor that transforms the product data in a neutral format. This neutral format, in order to be understood by application B, needs to use a post-processor that translates the product data in the format of application B. For the data sharing, the product data is conveyed from application A to application B using a common shared database. Data are then translated from tools and stored in the

database. This database contains the information in order that all applications may access the same neutral data via some kind of data access interface.

The loss of semantics is a real problem that engineers must take into account when transferring data from one application to another. "The same conceptual product data are represented in a variety of ways on different CA-tools. Furthermore, the same conceptual product data is stored simultaneously in different CA-tools. These redundancies cause consistency problems" [Ruland 1995].

In bibliography, this compatibility effort is also highlighted by interoperability. This interoperability is mainly defined by three main aspects [Armstrong 1999, Augenbroe 2004, Davis 2002]:

- ❖ The component: may be a system or a tool or even functionality. It is the major object that has to be linked with another to ensure interoperability of system;
- ❖ The link or connector: that defines the main way to create interoperability. Connector is mainly defined as an interoperable use of format.
- ❖ The technological object: ensures communication between components using connector. It is a kind of "request" addressed to the components by the user. It is considered as a service function.

Therefore, major studies have employed methods applied on these objects to create interoperability. The four main methods are related to:

- ❖ The **definition of schemas and their integration** in a product or data process model [Augenbroe 2004, Burkett 2001]. This method consists in specifying and defining data models attached to the product representation regarding given processes.
- ❖ The definition of **interfaces schema for integration** [Sudarsan 2005, Armstrong 1999]. As interoperability is provided by connectors, they may be designed as interfaces that need to be connected from both parts to components.
- ❖ The definition of **functions oriented for interoperability services** [Sudarsan 2005, Augenbroe 2004]. The definition of these objects ensures request executions from the user to a part or to the entire system.
- ❖ The definition of **database interoperability** [Armstrong 1999, Zhao 2003]. The interoperability is defined through the use of a standardised database (in the case of homogeneous databases), or define a semantic link and recognition from a database to another (in the case of heterogeneous databases).

V. STEP Standard (ISO – 10303)

Insert 3 - Introduction to ISO TC184/SC4 (<http://www.tc184-sc4.org/>)

ISO TC184 is one of the committees' members of the ISO ((International Standardisation Organisation, Geneva, CH). Its scope is: "Standardisation in the field of industrial automation and integration concerning discrete part manufacturing and encompassing the applications of multiple technologies, i.e. information systems, machines and equipments and telecommunications" [Cutting-Decelle 2004]. The mission of SC4 is to develop and promulgate standards for the representation of scientific, technical and industrial data, to develop methods for assessing conformance to these standards, and to provide technical support to other organizations seeking to deploy such standards in industry.

The scope of SC4 includes all the industrial data related to discrete products including, but not limited to the Geometric design and tolerance data, the Material and functional specifications, the Product differentiation and configuration, the Process design data, the Production data (including cost), the Product support and logistics, the Life cycle data, the Quality data, and the Disposal planning data. It includes organizational data such as the relationship between enterprises or the relationship between components of a single enterprise for the purposes of supplier identification. It includes personnel data to the extent of identification of approvals. Specifically excluded is business planning data such as profit projections, cash flow, etc., and any other personnel data or organizational data.

The goal of SC4 is the creation and maintenance of standards that enable the capture of information comprising a computerized product model in a neutral form without loss of completeness and integrity throughout the lifecycle of the product.

Specific objectives include the Flexibility to permit expansion without invalidating existing portions of the standard, the Efficiency for processing, communication, and storage, the Rigorous and unambiguous documentation, the minimum possible set of data elements, the Separation of data content from physical format, the logical classification of data elements, the Compatibility with other existing relevant standards, the Implementability, and the Testability.

In the Figure 34, below, the application standards (B) form the essential results in which industry shows a primary interest. The information requirements of the application standards are met by using an integrated set of common resources for applications (C) which support all the applications. The resources (C) are defined using the EXPRESS data specification language (D) and exchanged using a variety of implementation forms (D) under an overall technical architecture (E). The processes used to develop the standards to meet the application requirements are defined in the SC4 quality system (A) and the corresponding organization and procedures are described in the Handbook (F).

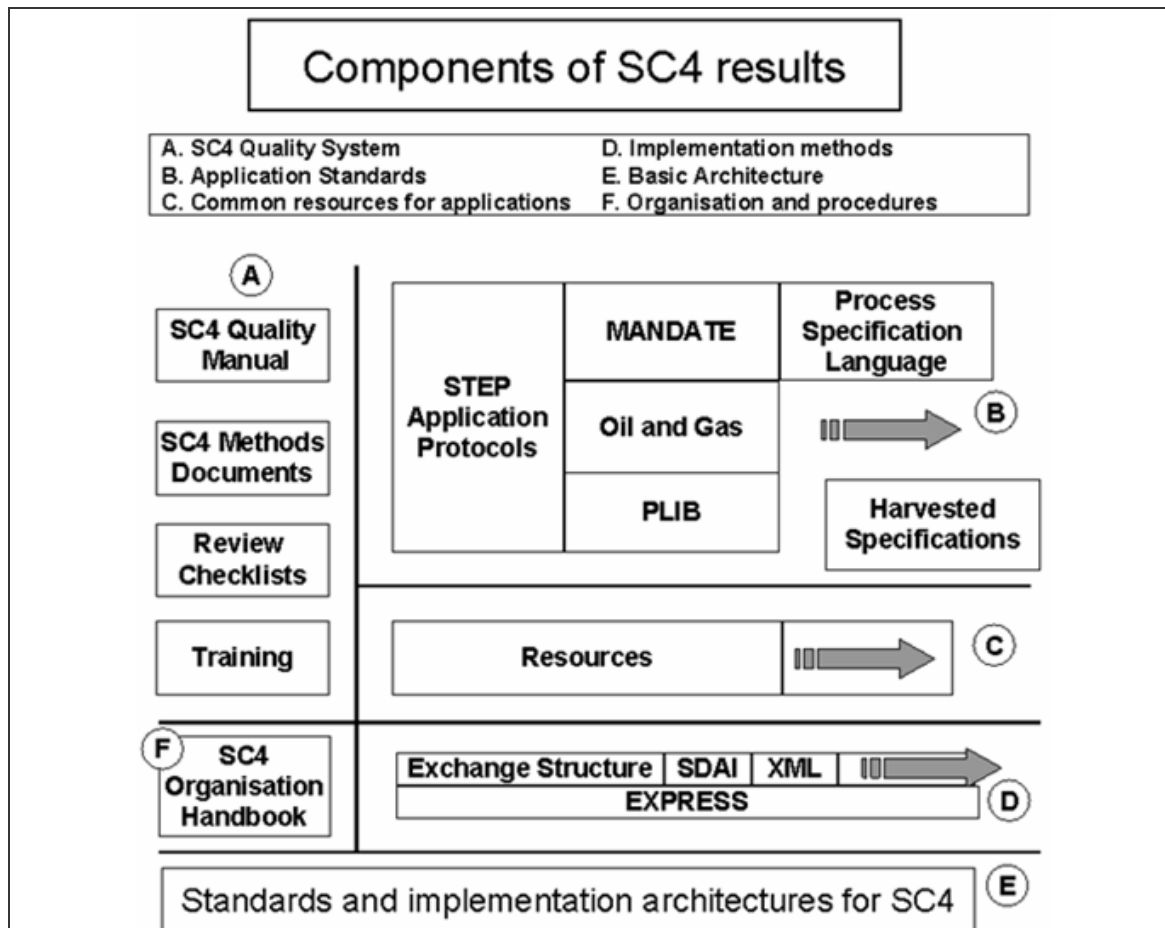


Figure 34 - Simplified overview of the components of SC4 results (<http://www.tc184-sc4.org/>)

TC184/SC4 currently develops five standards

1. STEP (ISO 10303 - Industrial automation systems and integration - Product data representation and exchange)

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving. This International Standard is organised as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, application modules, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1.

2. PLIB (ISO 13584 - Parts Library)

ISO 13584 is a series of International Standards for the computer-sensible representation and exchange of part library data. The objective is to provide a mechanism

capable of transferring parts library data, independent of any application, which is using a parts library data system. The nature of this description makes it suitable not only for the exchange of files containing parts, but also as a basis for implementing and sharing databases of parts library data.

3. MANDATE (ISO 15531 - Industrial manufacturing management data)

MANDATE covers industrial manufacturing management data, which is exchanged inside industrial manufacturing plants. Its scope covers the three areas of data exchanged between an industrial manufacturing company and its environment of manufacturing management activities; data able to reside in an industrial manufacturing company's resources database; data controlling and monitoring the flow of materials within an industrial manufacturing company [Cutting-Decelle 2006].

4. OIL & GAS (ISO 15926 - Industrial automation systems and integration)

The purpose of this International Standard is to facilitate integration of data to support the life-cycle activities and processes of oil and gas production facilities.

5. IIDEAS (ISO 18876 - Integration of industrial data for exchange access and sharing)

ISO 18876 is designed to satisfy the following high level requirements:

- 1. sharing and integration of data from multiple, heterogeneous sources;*
- 2. interoperability between applications and organizations that implement different standards.*

A. STEP Overview

STEP - Standard for the Exchange of Product model data is an ISO (International Organisation for Standardisation) / TC 184 (Technical Committee: Industrial automation systems and integration) / SC4 (Subcommittee: Industrial data) International Standard (IS) officially identified as ISO 10303, for the computer-interpretable representation of product information and for the exchange of product data [ISO 10303-1 1994]. The objective of this International Standard is to provide a **neutral mechanism capable of describing products throughout their life cycle**. STEP publishes a proposal for a methodology for development, implementation and validation of an open architecture for exchange and share of product data, together with a set of public data models identified as Application Protocols (APs). This standard is mainly contributing **for worldwide open systems adopting neutral networking communication for product data exchange between heterogeneous systems**, both in-house and with third parties. Also, assists on implementation of system independent architectures, flexible migration policies, contributing to long-term archiving in paperless and life-cycle maintenance support [ISO13584-1, Fowler]. The

complete architecture of STEP can be found at STEP On A Page (SOAP) [SOAP, 2000, <http://www.mel.nist.gov/sc5/soap/>] (Figure 35).

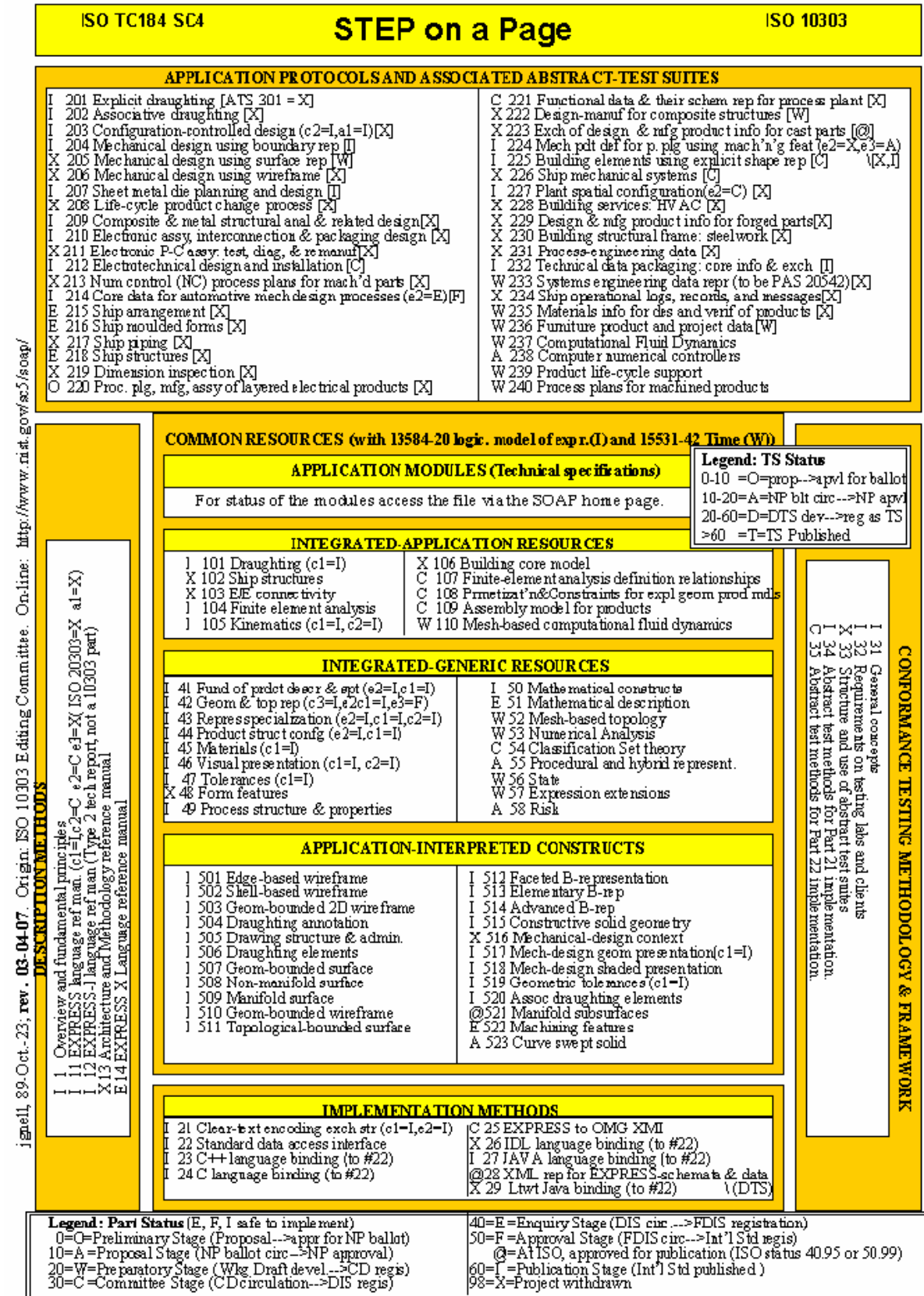


Figure 35 - Description of the organisation of STEP (ISO 10303) parts - STEP on a page

B. Example of STEP application range

Design/analysis integration and partners' data integration is typified by requirements to share and exchange representation data (mainly geometrical models) and data/parameters to perform activities [Pratt 2001, Bhandarkar 2000]. Integration is more difficult if we consider more components, activities and constraints as part of the problem. In large-scale engineering, standard-based solutions for activities and processes are becoming an imperative. Looking forward a solution to define the overall integration of system and architecture, we have identified STEP as the best candidate to provide a powerful integration. ISO 10303 STEP (STandard for the Exchange of Product Model Data) is the standard that provides a complete, unambiguous, computer-interpretable definition of the physical and functional characteristics of a product throughout its lifecycle [ISO 10303 -1 1994, An 1995, Peng 1998, Zha 2002]. Figure 36 represents the possible use of the STEP standard all along the product lifecycle

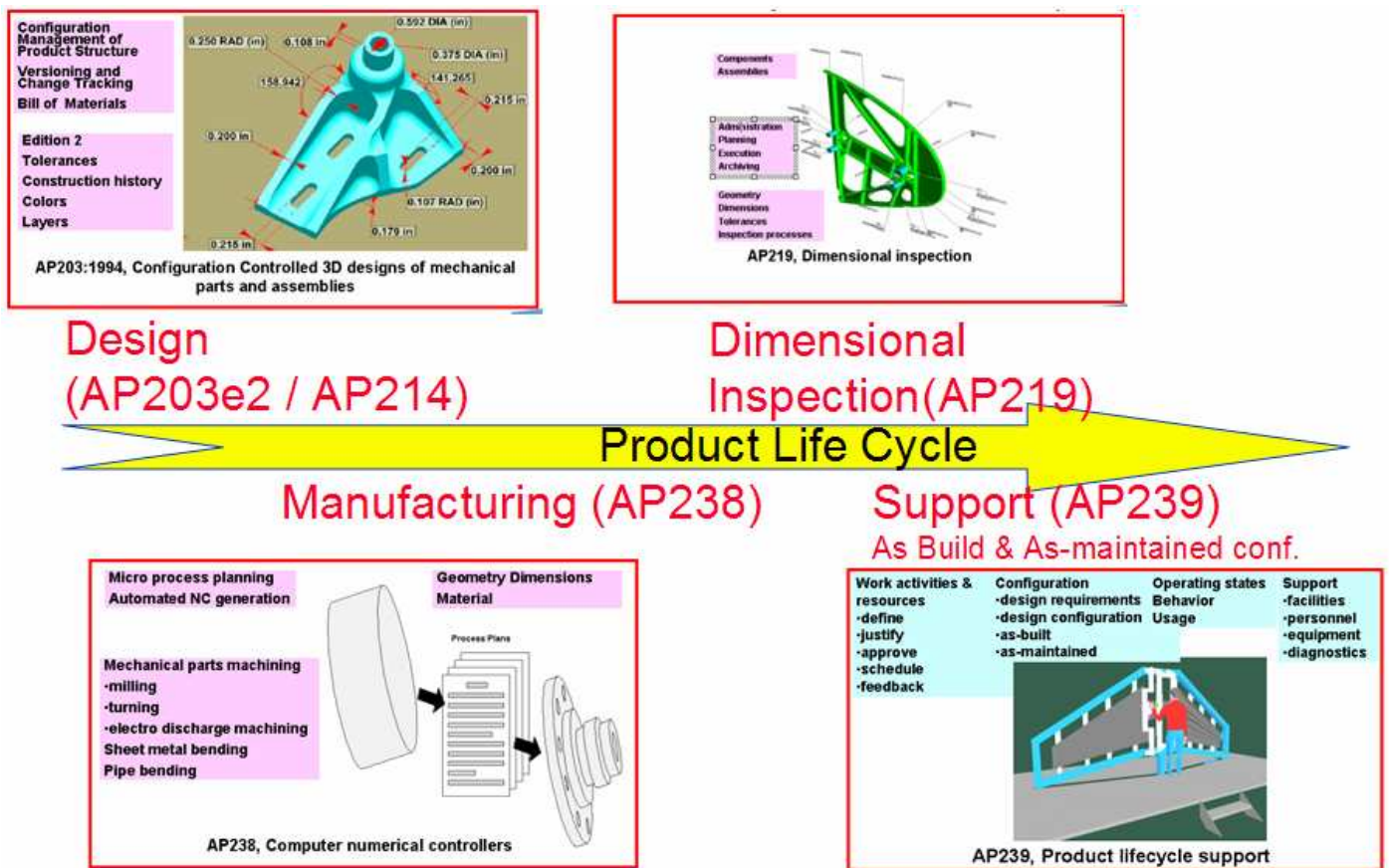


Figure 36 – Example of STEP exploitation along product lifecycle

C. STEP structure

STEP is organised in six part (for more details on published documents see: [http://www.tc184-sc4.org/SC4_Open/SC4%20Legacy%20Products%20\(2001-08\)/STEP\(10303\)/](http://www.tc184-sc4.org/SC4_Open/SC4%20Legacy%20Products%20(2001-08)/STEP(10303)/)), in which each one contains different part related to a specific application.

Figure 37 illustrates the various aspects of the STEP standard. The core of the standard is a series of integrated data models that provide resources for information such as product design, geometric and topological representation, and some specialised representations. These data models are all written in EXPRESS, an implementation-independent computer-sensible language that is also an ISO standard. There are then **Application Protocols** (APs) that define application specific views of the integrated resources in a clearly defined context. Some examples of APs include AP203 [ISO 10303-203] Configuration Controlled Design, and AP209 [ISO 10303-209] Composite and Metallic Structural Analysis and Related Design. There are two implementation methods defined within STEP, the first is a flat ASCII file (Physical File), and the second a standardised application programming database interface (STEP Data Access Interface (SDAI)). Finally, each AP has an associated set of Conformance Testing documents to provide a method to test and certify translators and interfaces.

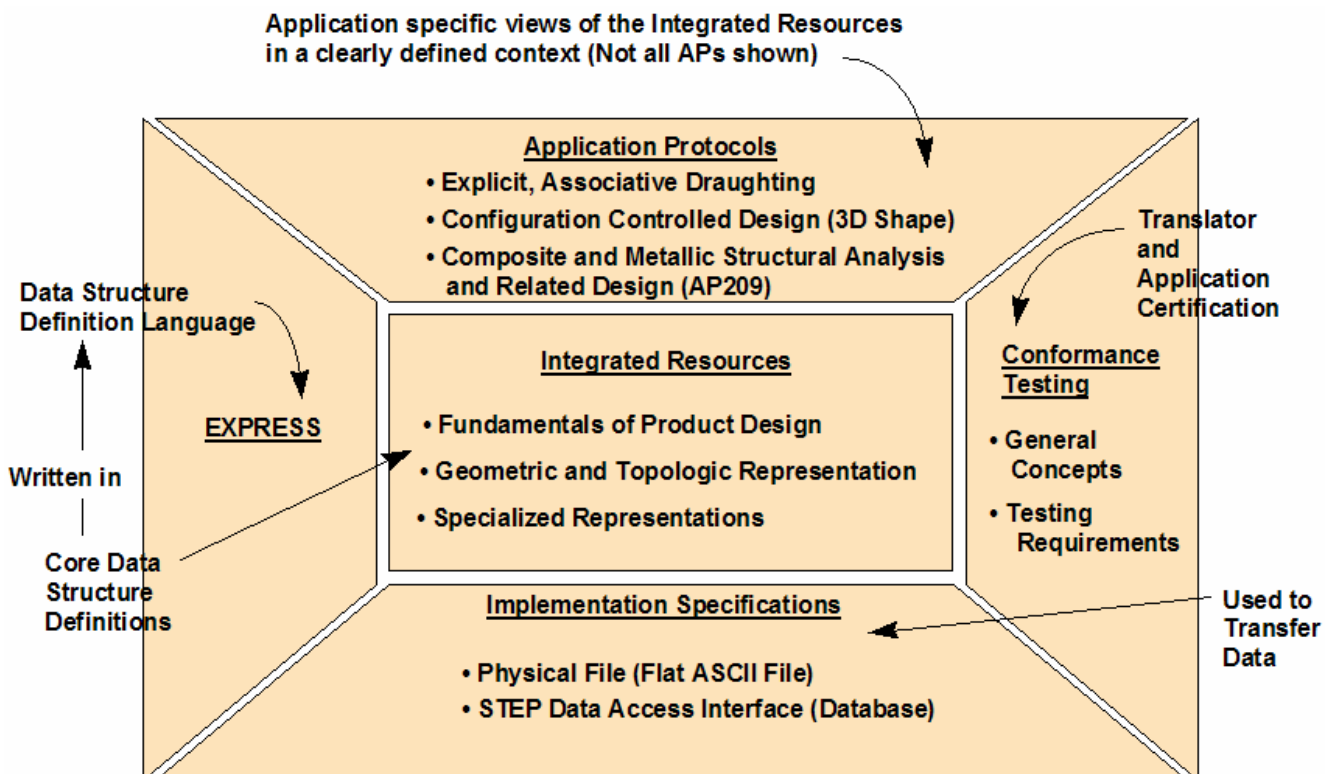


Figure 37 – STEP (ISO 10303) structure

The internal structure of an Application Protocol (AP) is illustrated in Figure 38.

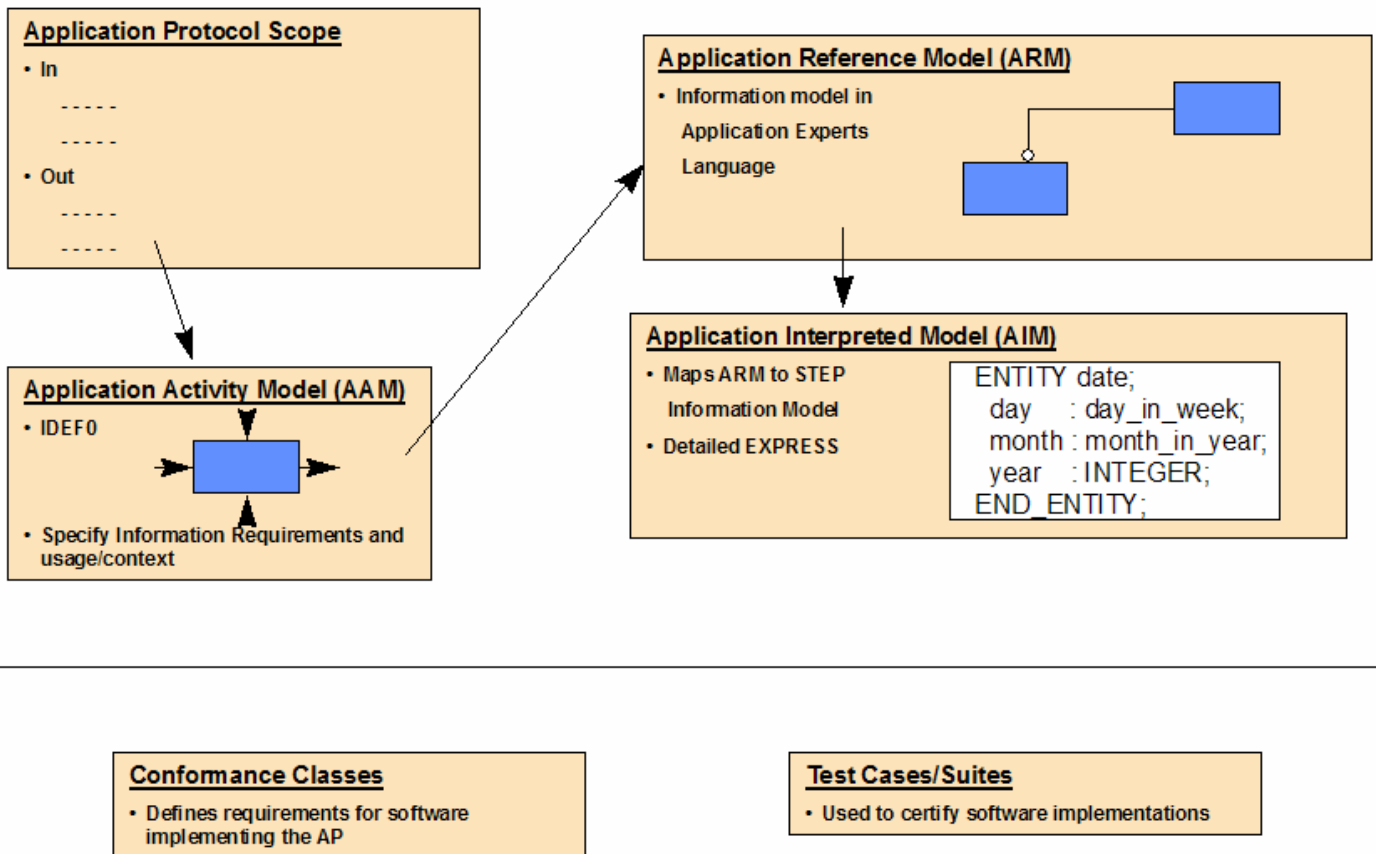


Figure 38 - Representation of the internal structure of an application protocol

The first section of an AP describes what is in and out of scope for data exchange. There is then an **Application Activity Model (AAM)** of the process that the AP which is written in the IDEF0 modelling language. The AAM is primarily used to set up the context for the data exchange and to provide a basis for data exchange requirements (see Figure 39).

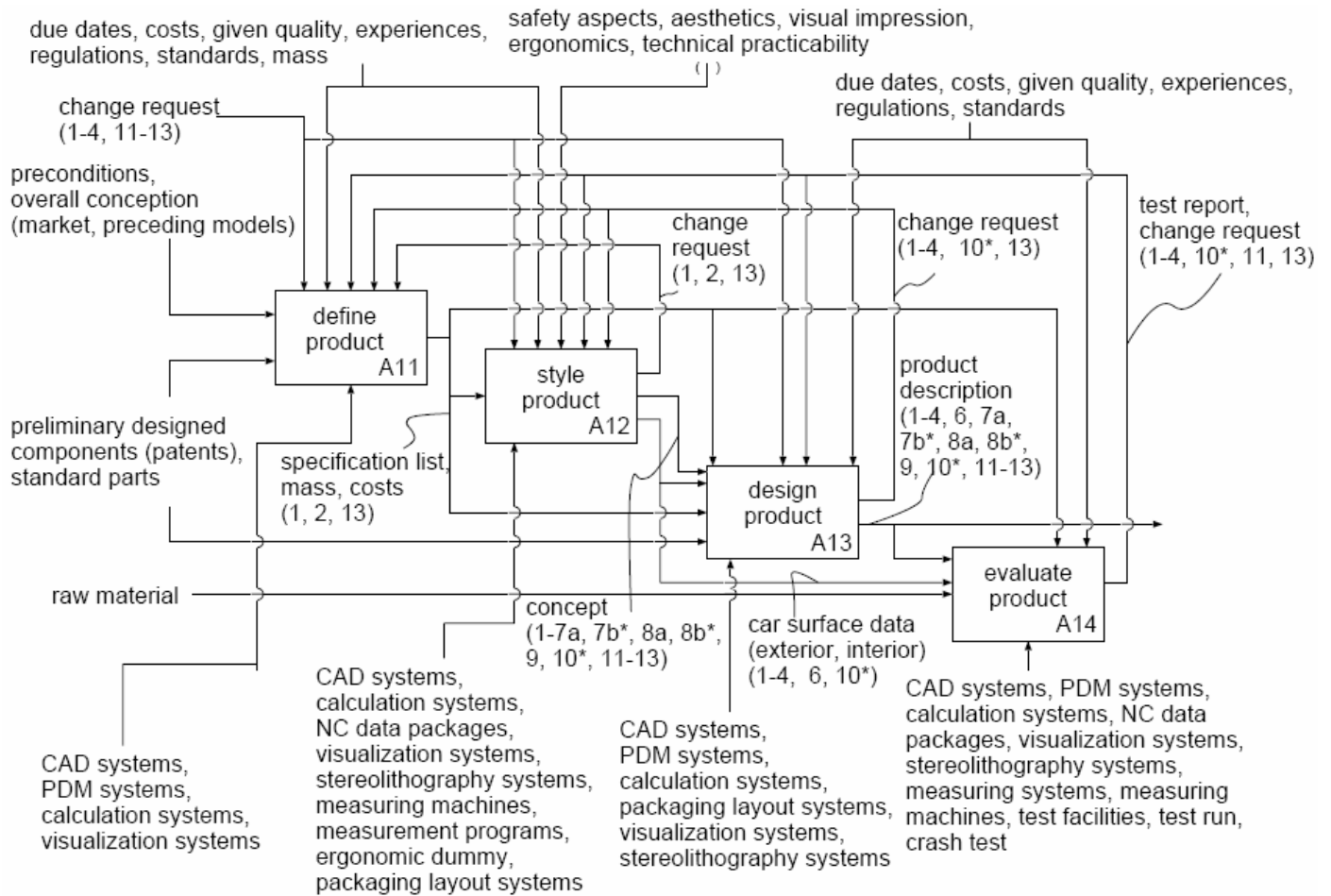


Figure 39 - Example of the AAM - develop product extracted from ISO 10303 – 214
 [ISO10303-214]

The **Application Reference Model (ARM)** is an information (data) model of all the information requirements of the AP. The ARM is the part of an AP that a user would find the most useful (see Figure 40).

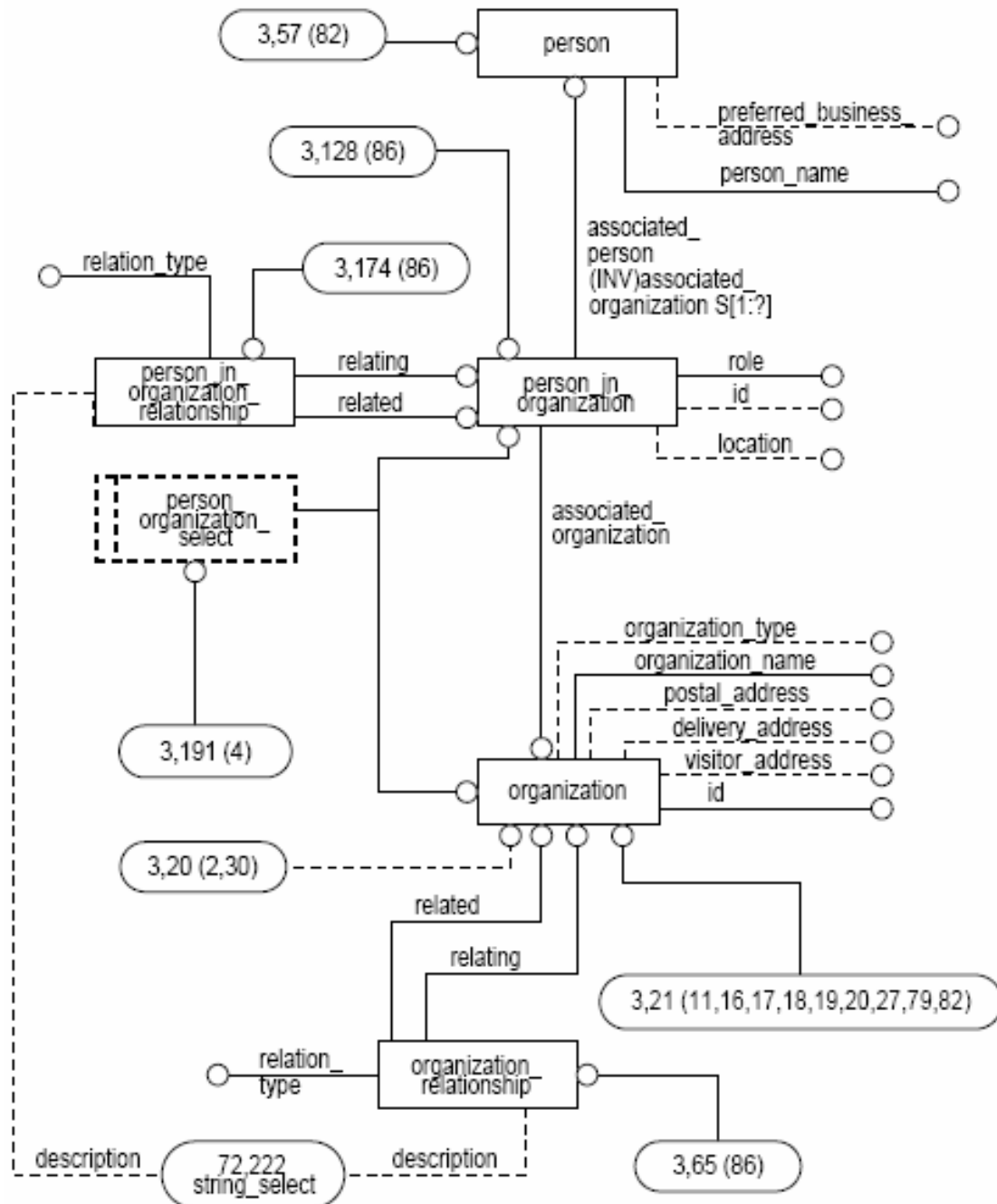


Figure 40 - Example of the ARM – person in organisation extracted from ISO 10303 – 214
[ISO10303-214]

Finally there is the **Application Interpreted Model (AIM)** that is the result of the mapping of the ARM requirements information model to the STEP integrated resources information models. The AIM is written in EXPRESS and is useful only for implementers (see Figure 41). There are also definitions of conformance classes for applications implementing the AP, and there are associated documents that define the test cases/suites for certifying the implementations.

```

TYPE organization_item = SELECT
  ( action, action_directive, action_method, action_property,
    action_relationship, action_request_solution,
    alternate_product_relationship, applied_action_assignment,
    applied_classification_assignment,
    applied_identification_assignment,
    applied_organization_assignment,
    applied_person_and_organization_assignment, approval,
    approval_status, assembly_component_usage_substitute,
    certification, class, class_system, configuration_design,
    configuration_effectivity, configuration_item,
    configured_effectivity_assignment, contract, document_file,
    draughting_model, drawing_revision, effectivity,
    event_occurrence, executed_action, general_property,
    material_designation,
    mechanical_design_geometric_presentation_representation,
    organization_relationship, organizational_project,
    person_and_organization, presentation_area, product,
    product_concept, product_concept_feature,
    product_concept_feature_association,
    product_concept_feature_category,
    product_concept_feature_category_usage, product_definition,
    product_definition_formation,
    product_definition_formation_relationship,
    product_definition_relationship,
    product_definition_substitute, property_definition,
    requirement_for_action_resource, resource_property,
    security_classification, security_classification_level,
    shape_representation, versioned_action_request);
END_TYPE; -- 10303-214: automotive_design

```

Figure 41 - Example of the AIM – organisation item extracted from ISO 10303 – 214
[ISO10303-214]

1. The description methods

The description methods provide the essential guidelines and the specification languages of the standard.

- ❖ The Part 1: **Fundamental principles**: exposes the scope and structure the STEP norm
- ❖ The Part 11 to 14: Description languages: define the modelling language EXPRESS. It precise how the language must be used to describe data

2. The Application Protocols (APs)

The application protocols (APs) describe the different uses of the STEP. They define the data relative to product in different context of application. They use different integrated resources to describe the particular application in an activity.

Also identified by the parts from 201 to 240, they describe the design domain, the simulation domain, and manufacturing domain. Indeed, they have an important application in the CAX and DMS domains.

3. The application modules

Application modules are the key component of the modularization of the initial ISO 10303 architecture. There are three types of application modules: foundation modules (level 1), implementation modules (level 2), and AP modules. Foundation modules provide lower level reusable structures that are not likely to be implemented alone, but are highly shareable and reusable. Implementation modules define a capability that can be implemented and against which conformance classes may be defined. Each AP references a single root module that is an AP module. An AP module is an implementation module, and the contents of an AP module are the same as other implementation modules, the only documentation difference is in their name and title. The AP module from one AP may be used by another AP.

4. The Integrated resources

The integrated resources are divided into three families:

- ❖ The integrated application resources which define the resources applicable to a specific domain (Part 101 to 111)
- ❖ The integrated generic resources which define the resources independently from an application domain (Part 41 to 58)
- ❖ The application interpreted constructs: that are necessary library that implement the integrated resources (Part 501 to 523)

5. The implementation methods

In STEP are also defined methodologies of implementation (for application). They define the conversion and application for the C++, Java languages and define the transformation of files incoming from other systems (for example XML...). These parts are numbered from Part 21 to 29.

6. The conformance testing methodology and framework

The conformance testing defines the guidelines analysis to evaluate the conformity of the application of STEP in the computational domain. These parts are numbered from 31 to 35.

D. Toward a new structure for STEP – Introduction of STEP modules

There is a bigger overlap between APs because they often need to refer to the same kind of products, product structures, geometry and more. And because APs are developed by different groups of people it was always an issue to ensure interoperability between APs on a higher level. The Application integrated constructs (AIC) solved this problem for common specializations of generic concepts, primarily in the geometric area. To address the problem of harmonizing the ARM models and their mapping to the AIM the STEP modules were introduced. They contain a piece of the ARM, the mapping and a piece of the AIM, called MIM. Modules are built on each other, resulting in an (almost) directed graph with the AP and conformance class modules at the very top. The modular APs are:

- ❖ AP203 - Configuration controlled 3D design , TS and 2nd edition
- ❖ AP209 - Composite and metallic structural analysis and related design , upcoming 2nd edition
- ❖ AP210 - Electronic assembly, interconnect and packaging design , 2nd edition
- ❖ AP221 - Functional data and their schematic representation for process plant
- ❖ AP236 - Furniture product data and project data
- ❖ AP239 - Product life cycle support

E. STEP and EXPRESS G language

EXPRESS-G is a graphical notation for the display of information models. The notation only supports a subset of the EXPRESS language, and therefore EXPRESS-G models are normally abstractions of EXPRESS models [Pierra 2000]. There are equivalent notations for bit-mapped graphics and ASCII character or "typed" diagrams. EXPRESS-G has symbols to represent EXPRESS entities, user-defined types, base types, relationships (required and optional), and inheritance. This notation is based on the standardized EXPRESS-G notation, which is itself described in ISO 10303-11 [ISO 10303-11]: Industrial Automation Systems – Product Data Representation and Exchange – Part 11: Description Methods: The EXPRESS Language Reference Manual. For more details see ANNEXE 2.

VI. Conclusion

In this chapter, we have provided an overview of systems called CAX for Computer Aided Technologies and data management systems. Both aspects are

System engineering for Collaborative Data Management Systems: Application to design simulation loops

nowadays linked due to the number of data created during the product development and all along its lifecycle. Using such tools reveals a major heterogeneity and integration issues. Indeed, industrial applications are developed by different software vendors and then do not use the same application semantic. Then, a referential is the basis for defining an integrated view. Concerning data management systems, they have already reduced engineering issues while considering the capabilities to create, modify and retrieve data. But with the increase of their development, new problems have appeared: the interoperability between systems which presently limit a “powerful communication”. This problem is going to increase with the appearing of more specialised data management systems such as those for simulation. Once again, the issue is related to communication aspects. Then techniques and processes have to evolve toward the referential (unique source for activities).

STEP standard is a solution to reach collaborative concepts considering industrial context. Using modelling capabilities, standardised references and semantic objects should cover most of activities performed for product development and data used in these contexts. Two major advantages for using STEP may be mentioned:

- ❖ The interoperability and communication aspects: STEP description and models that are already implemented provide a first way to migrate the different data using the same schemas.
- ❖ The range of use and the capabilities of development: In regards with industry need, STEP offers a wide range of application and interoperability principles between domains. The capability of using EXPRESS language to design new concepts and link them to already existing entities is also an advantage to customise regarding users requirements.

Chapter V: Definition of preliminary specifications for multi-partnership and multi- activity collaboration

I. Introduction

The definition of previous specifications for the development of a collaborative application is the opportunity to set up our working environment and delimit our action range. The Figure 42 illustrates our overall process of specification definition for the definition of a collaborative environment.

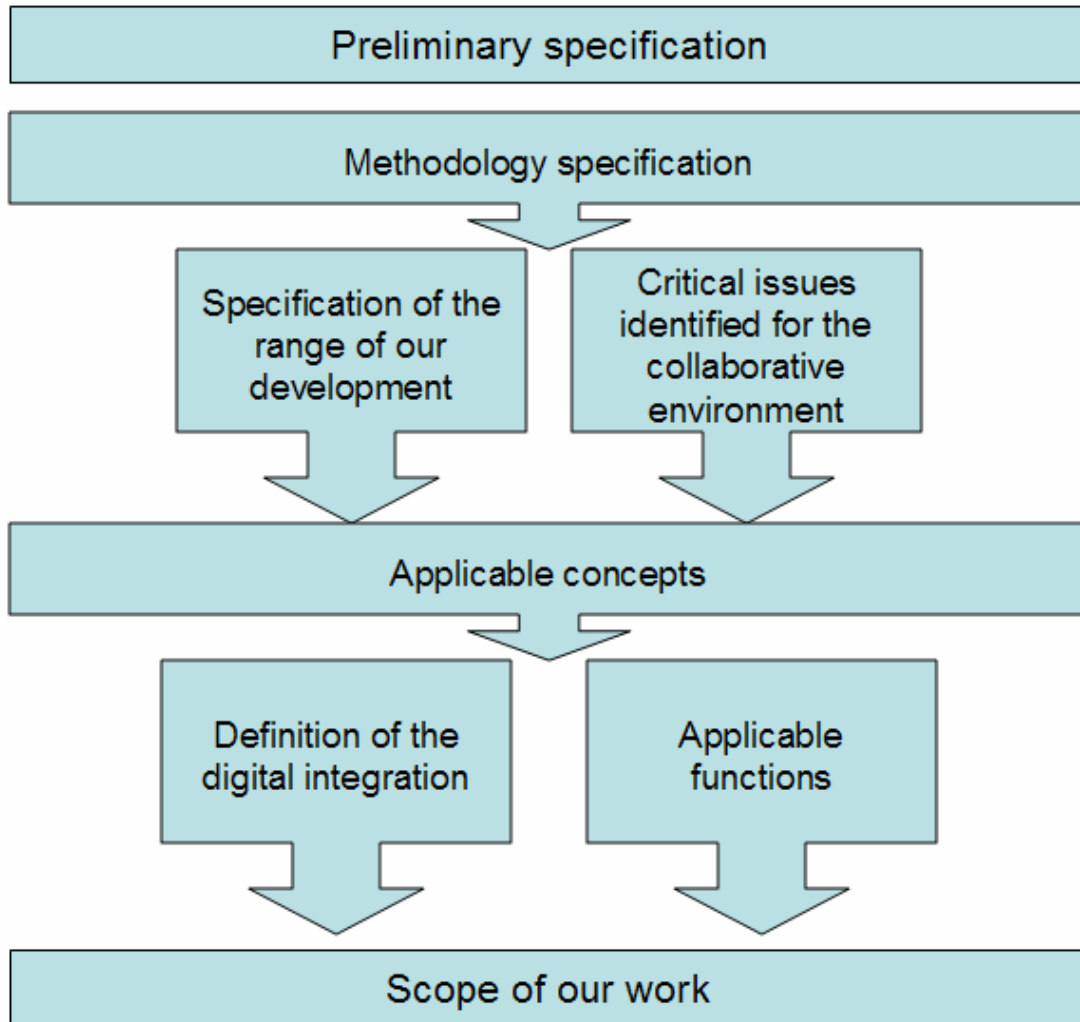


Figure 42 - Process of specification definition

As a beginning, the present specification precise the methodology specification. This provides the overall application process which has to be applied. Then we delimit our application scope defining the range of the development and the critical issues on developing such environment. Finally, we precise the applicable concepts which allow us to define the applicable functions and then the definition of the digital integration.

II. Necessary methodology to develop the collaborative architecture

A. Formal description of environment for method implementation

While collaborative environments were limited to design environment a few years ago, such an environment use is now extended to the different 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 [Fuh 2005] and Li [Li 2005] 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:** That is defined to associate the different 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) [Aziz 2005, Merlo 2004].
- ❖ **A dynamic approach:** Relevant for a processes/workflow approach. It defines the overall activity sequences that describe the different actions on data. This is mainly designed with classical modelling tools such as IDEF0 (ICAM DEFinition language) [Bradley 1995, Loureiro 2003, Shunk 2003].

The definition of a collaborative architecture must use these two kinds of approaches. Bergman & Baker [Bergman 2000] describe a « shared virtual workspace » that aims at enabling the data exchange through integrated environments. Each environment is submitted to its own laws of functioning. The overall objective is to create networked activities using product data representation, and more especially a reference frame created for the product in the collaborative workspace [Perrin 2004].

B. 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 (because it existed needs for complex application systems built to satisfy application processing) and the use of formalisms and models was generally limited to some aspects of application development within

the Information Systems community [Bradley 1995, Chen 1997]. 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 [Egelhoff 1999, Quattrone 2001, Nurcan 2003]. 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 [Coronado 2002, Dustdar 2003, Holland 1995, Montreuil 2000].

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, for instance, 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 [Perrin 2004] 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. [Sudarsan 2005] 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 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 [Vernadat 2002], enterprise integration consists in “breaking down organisational barriers” and 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 different processes exploited. A next step consists in organising these models so as to exploit the different available connectors between processes and integrate them (modelling and state-of-the-art definitions of Shunk [Shunk 2003] and Reithofer [Reithofer 1997], and examples of use by Delen [Delen 2003] and Vernadat [Vernadat 2002]).

C. Characterisation of the method

Our method is then composed of five stages. These stages follow a bottom-up approach from end-user view characterised by data topic, going through tools, processes and finally high-level processes. The last stage is the integration that characterises the connectors that need to be implemented in the first four stages.

- ❖ **Data typology:** Defines which data is accessible and who is allowed to access it. This defines the views to associate to data and the links that exist between them.
- ❖ **Definition and tools analysis:** Characterise the tool data models and available connectors that ensure the connection to a collaborative platform. This makes the inventory of attributes exploited and how they are managed to extract the data structure.
- ❖ **Processes structure definition:** Define processes and associated models to identify the available process interfaces that are accessible to set work requests.
- ❖ **Context/Product/Resources coupling:** Define the elements for contextual approach. It defines the links available between high-level objects. This coupling between context, product and resources precise what we called the environment.
- ❖ **Integrative approach:** It is the method that characterises the connectors available all along the definition of objects. From data connection viewpoint, it is characterised by attribute links between models. For tool connections, it is performed with API's programs or formats definition. For processes, it is defined by workflow interfaces, and for business it is ensured using project relations.

The data typology defines the attributes of product use in an activity. As attributes are relevant for a tool, they are then associated from a tool to another in regards with the previously established typology. Then the tool is used to transform a

data in a defined activity or process. The association between two processes results in defining an association between data using work requests through workflows. At a higher level, the enterprise model will define connectors at the project level.

D. Defining the integrative approach

This notion leads us to consider three major approaches:

- ❖ What is evoked through integration, what is it? What are the conceptual models and kinds of integration?
- ❖ How integration is achieved? What are the development strategies, mechanisms?
- ❖ How the integration is checked? What has to be exploited to analyse achieved goals?

The major issue on integration is to analyse the system regarding:

- ❖ Determination and analyse of the major flows (physical or non) and objects that have to be integrated - Determine major input/output and highlights constraints of the system.
- ❖ Determination and analyse of the major flows interfaces that exist - Determine of major connectors that can exist between two flows.
- ❖ Determination and analyse of the major "black box" component between flows - Determine the components and methods that can be used to match at best the defined connectors between flows.

The integration between two distinct systems and/or sub-systems is the practice of combining the functions of a set of system and/or subsystem, to produce a single unified system that satisfies some need of an organisation. It corresponds to the link between product, context and resources components to merge their functional and technical characteristics into an interoperable and associative system. Integration is seen as a process glue composed of a set of interfaces and rules of interaction that govern communication among components [*Kosanke 1999, Mertins 1997, Wang 2002*].

III.Characterisation of the collaborative environment for multi-partnership and multi-activity

A. A three domain cross-vision

Here, we identify major requirements for collaborative architecture design and development. Distinct but related domains have to be taken in consideration: Concurrent Engineering, Digital Mock-Up and Data Models

Nowadays "**concurrent engineering**" or "**simultaneous engineering**" are a major requirement for the development of product. They characterise the intention to group different working teams around the same product and characterise the simultaneous development of their product parts [Werner 2004, Dustard 2003]. Concerning concurrent engineering, the main requirement is to provide a simple environment enabling the creation of virtual spaces for concurrent design [Fuh 2005, Yeomans 2006]. This means defining working interface [Shephard 2003] to define the limits of shared spaces for the development and provide the capabilities for concurrent work using workflow (for more integrated activities) and data update (for more integrated product views). Although in the domain of aeronautic industry this spaces have to provide capabilities to ensure data security and "know how" confidentiality.

As a consequence of the need for concurrent engineering, many approaches and methods have been proposed. Our attention is particularly focused on **the Digital Mock Up (DMU) which is an example of the development of collaborative tools** as suggested by Sadoul [Sadoul 2000]. Defining collaborative spaces and integrated environment, the DMU seems to be the most adapted tool to support the common view of the product [D'adderio 2001]. Although, defining the DMU as a collaborative reference means defining the allocated spaces for design partners using the references of interfaces (this way, interfaces are the collaborative objects that define the limits of virtual spaces for development). Moreover, the capabilities of the DMU must be extended in order to ensure the development of the configuration management (to access a virtual configured product) and the capability to import and export different representations (this point is important to characterise the multi-view development using the DMU, it should ensure the migration of product representation between different working domains).

But while increasing the development of software capabilities and functions, we also increase the need of data management (more objects and attributes used in

different processes involves the need to define more complete structures). Although, defining a common reference for concurrent engineering duplicates this problem of data charge in systems [Werner 2004]. In fact, the collaborative platform intends to lower this problem. Instead of attempting to create a complete solution of "universal data management", the platform creates bridges between systems using **data models**. This allows then to charge the partners systems using the minimum amount of information so that they can perform their activities. As a consequence of the collaborative platform, the data management systems should also support interoperability between existing tools on different OS and provide a centralised access and repository for heterogeneous data. It should then provide classical PLM functions such as history on information, configuration management, links between activities and data life cycle, and should support the data and engineering systems evolution...

B. Toward Framework critical issues

The framework critical issues explore the implication of communication capabilities in the decomposition of the system. They have to point out constraints and links to provide the best communication between activities. We list in the following, the major requirements that must be consider for the development:

- ❖ Management of **relationships and dependencies** [Mitschang 2003, Murphy 2002]: must provide the management of critical dependencies across data and information involved in product engineering. One solution is to manage these dependencies using relationships between elements of design process. In collaborative environments, a fusion of these types of relationship management systems is required, and the relationship management needs to be extended to the knowledge level. To facilitate this, a relationship management at object-level is required for activities. That would allow management of information and data at a more detailed level as compared to the current file level management.
- ❖ **Update and data flow**: must support concurrent process modelling and the management of tasks and data flows. All must be coordinated between sub processes involved in activities. The modelling must also support multiple people working in a concurrent manner at different levels of business abstraction in the organisations [Jasnoch 1996, Perrin 2004, Sage 1995]. This notion relies on different views of the product information across disciplines that must be supported in concurrent environments. Critical points to take care of are:

- ❖ All applications rely on sub processes that can be launched independently. The idea is to provide a support for these applications so as they can address the level of process involved independently.
- ❖ Create an integrated process that is able to address the multiple levels of processes. These levels can be divided into:
 - ❖ Analysis execution level
 - ❖ Design methodology level
 - ❖ Inter-organisational design level
 - ❖ Enterprise level
- ❖ **Configuration Management** [*MacClean 1997, Wang 2002*]: The framework must permit to maintain several versions, histories, as well as different variants and components used in the overall context of the product (i.e. to preserve the main and invariant product structure and characteristics and to associate them with the options for configuration). The right configuration must be managed to describe a particular version or variant of an assembly built on those components within each domain. A common structure (e.g. skeleton), may define the necessary items for design and analysis, but may be viewed through multiple perspectives according to the rules of a specific domain. The tree structure corresponding to a view, implicitly hierarchical, should not be fully built from the origin; its final presentation will appear when all necessary items are identified. A dynamic support must be provided to ensure the coherency between the multiple domains. Each item should include parameters, attributes, and rules applicable to each domain.
- ❖ **Integration**: represents how components of the architecture are going to communicate [*Mervyn 2003, Pierra 2000, Xu 2003*]. The need from activities is to adapt the overall context (i.e. environment and evolutions that can happen from internal and/or external events) with their environment (variety of analyses, process management, and management of both information and knowledge). The notion of internal environment and evolution of the framework components refers to the other (or new) components that have to be integrated. The idea of open framework should facilitate the addition of a new component without requirement of changes in other components. The notion of external environment and evolution of the framework refers to other applications that the user employs in conjunction with the framework being considered. Therefore, the framework must be integrated with the external applications.

- ❖ **Standardised Interface** (communication infrastructure): describes that the framework must support the standardisation of the communication interfaces. Mainly the communication between CAE applications relies on files in specific formats, or program-specific APIs. In this case, files are generally used to transfer large units of information such as analysis outputs. In order to communicate relevant information to the user containing more details specifics to his needs, connections are used to extract and filter necessary and sufficient information derived from the original raw source. Adjunction of meta data will also maintain the required context for the targeted application [*Jaafari 1998, Johanson 2004, Li 2005, Werner 2004*].
- ❖ **Modularity**: the ability of collaborative framework to be open to internal and external sources is achieved using the modularity and the standardisation of interfaces [*Praehofer 2001, Leitao 2001*]. The main idea here is to provide sufficient flexibility and generic aspects to the framework for the purpose that if one component changes then no change are required for the rest of the system (it is often called independence of modules). This notion of modularity must enable the system to be re-configurable for different usage scenarios, reducing the customisation and providing users the opportunity to plug their different modules into the framework. In terms of the framework design, the ideal architecture associates one module with one function, reducing interdependencies and complexity.

IV. Applicable concepts

A. Integration of the product and definition of the digital integration

Digital Integration is essential for a project with several partners. Now, the aim is to build a product structure with common methods to administrate CAX's on concurrent engineering and multi partnership contexts. Configuration is also part of this virtual product integration. The objective is to use standard to manage CAXs' data and to allow an accurate exploitation. Virtual reviews should allow partners to set up and run collaborative reviews with the same shared set of 3D CAX or assembly. Formal and informal reviews are seen as a potentiality to reduce non-productive iterations.

An overall set of 3D CAX or assembly should be identified as the central tool for all engineering activities. The definition of a networked structure and the building

and application of methods, processes and tools are necessary to share data among partners.

This remote and direct access will allow all partners to:

- ❖ Create, share and update geometrical models organised in a product structure with configuration management rules, (containing weight and inertia characteristics or more),
- ❖ Identify and analyse Product dynamic behaviour
- ❖ Organise reviews with other partners, nacelle and aircraft manufacturer,
- ❖ Optimise the process for Product dynamic co-operation
- ❖ Create and share specific documentation on time
- ❖ Share data of test results in order to update modelling and simulation

Nowadays, virtual (meaning digital) integration of large product models within the product development process is - simply expressed - limited to the following considerations:

- ❖ Integration is done on geometry level, only (CAD software)
- ❖ Integration is limited by current STEP interfaces, since various and different CAD software packages are used throughout the supply chain
- ❖ Integration is mostly about aircraft structure - systems integration is limited yet (space allocation, typically)
- ❖ Costly Frontier Models (specifying the interfaces) are needed, either as digital data or as hardware (even worse)

Future requirements are to virtually integrate a complete Digital Engine to several engineering disciplines and interfaces with several aircraft partners, system suppliers and engine manufacturers. The challenge is to digitally integrate not only geometry related data, but also as much as possible the various kinds of data supplied on functional aspects.

B. High level constraints: definitions for a coherent architecture

An implementation of the collaborative framework needs to consider:

- ❖ Right level of **breakdown responsibility** to choose between Operational Layer and Business Layer: There is an overlap between the product data used both by operational layer and business layer. The collaborative framework should propose a method to avoid an overlap on product data management in order to

avoid multi-definition of the same information in several repositories. The ownership of each level of information has to be clearly identified. The product data used both by operational layer and business layer (or used between different disciplines) has to be published at the consolidated repository level.

- ❖ **Scalable architecture** to support collaboration between distributed disciplines or organisations: The different components of the collaborative framework will be distributed on the enterprise network.
- ❖ **Data access right** to manage regarding user authorisation: data sharing implies control of access right. The collaborative framework will propose multiple accesses to consolidated product data, all of these accesses will have to be controlled by a centralised mechanism compatible with each local access right policy of the different component of the collaborative framework.
- ❖ **Unique ID mechanism to identify data** must be provided by operational and business tools: this is a mandatory technical requirement of collaborative framework architecture. Each tool integrated in collaborative framework has to provide a mean to identify its published data. Without this functionality, collaborative framework will not be able to guaranty a consistent consolidated repository.
- ❖ **Workflow integration** to support integrated process across heterogeneous environment. Collaborative framework should provide communication mechanism at workflow level to guaranty the continuity of the process between the business and operational layer. Collaborative framework should also provide mechanism to ensure the traceability of the product data regarding process evolution.

C. Integrating the contexts

The definition of context for collaboration is described in literature as the support of engineering processes to perform a specific activity [D'Adderio 2001, Tang 2001]. The integration of engineering contexts is the procedure of associating product data instantiated for an activity into an environment defined by the working objectives [Boujut 2002, Sudarsan 2005, Tang 2001]. Integration of contexts targets the completion of engineering characteristics and business objectives characteristics.

To complete this integration of contexts, technological objects called **contexts interfaces** are defined (see representation in Figure 43). The integration of contexts using interfaces can be performed in two ways. The first one considers the structure and sequence of activities using processes. Then contexts depend of the process currently used by the engineer. The definition and migration from a context to another

is ensured by defining the characteristics of the process flow at a given time [Vanderhaeghen 2006]. The second considers the product interfaces. These interfaces describe the product environment and characterise the activity.

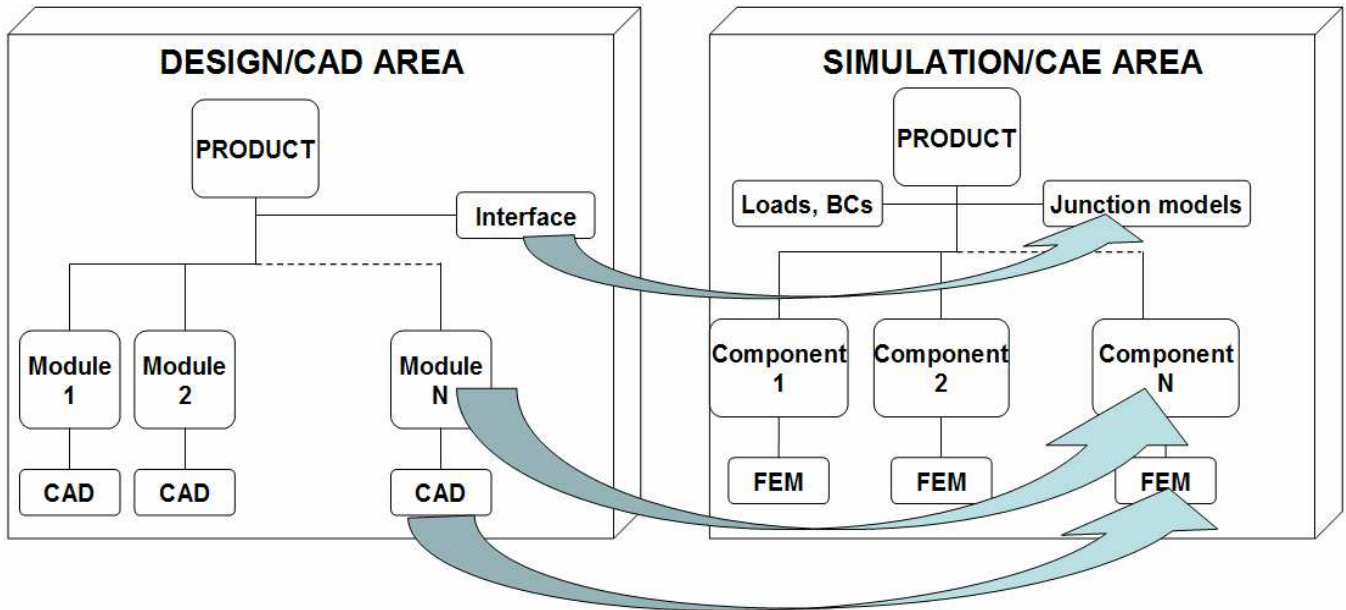


Figure 43 - Overall view on CAD/CAE interface

In our case, the consideration of an interface between design and simulation worlds involves many things to consider in the product definition. As we are looking at the structure of the product in the design area and trying to migrate it to the simulation area, three main transformations exist:

- ❖ Design world Interface: Corresponds to a physical definition (CAD file) that allows designers and partners to share the context they have to design in. This “CAD interface” is derived later in *junction models* that are going to ensure the creation of the simulation product and to the definition of specific parameters.
- ❖ Migration of product architecture (e.g. configuration): To validate product architecture, engineers must be able to get access to the valid configuration and must be able to migrate it in their context. Moreover, they must be able to incorporate the attached semantic (such as maturity, model review and status...).
- ❖ Finally, the migration concerns also 3D models that have to be managed. As the process of simulation also implies the geometrical structure of the product, we must transform it using the most meaningful information related to 3D product geometry. This way, we must also be able to manage the meta data that are attached to this model.

D. Integrating the processes

In this case, interfaces rely on the process of enabling two components to communicate in an efficient way regarding the environment considered and the characterisation of the data to be used. It is possible to determine a generic process map of component translation and communication from an environment to another.

The functionalities targeted in collaborative framework to support business processes are:

- ❖ The ability to define a process at Information Model Level
- ❖ The ability to instantiate a process in the collaborative environment and in the operational tools
- ❖ The ability to store product-process history in the consolidated repository
- ❖ The ability to monitor an operational process from the collaborative environment

There are two kinds of communication in collaborative framework architecture. **The communication inside the application composing the framework and the communication needed to guaranty the interoperability between the different applications.**

The communications inside the application composing the framework is the communication inherent to the architecture provided by the application (client-server, 3-tiers, multi-tiers), and the “on the shelf” communication provided by the software editors to realise a functional integration between complementary tools in a specific domain.

The Figure 44 proposes the definition of the integration process.

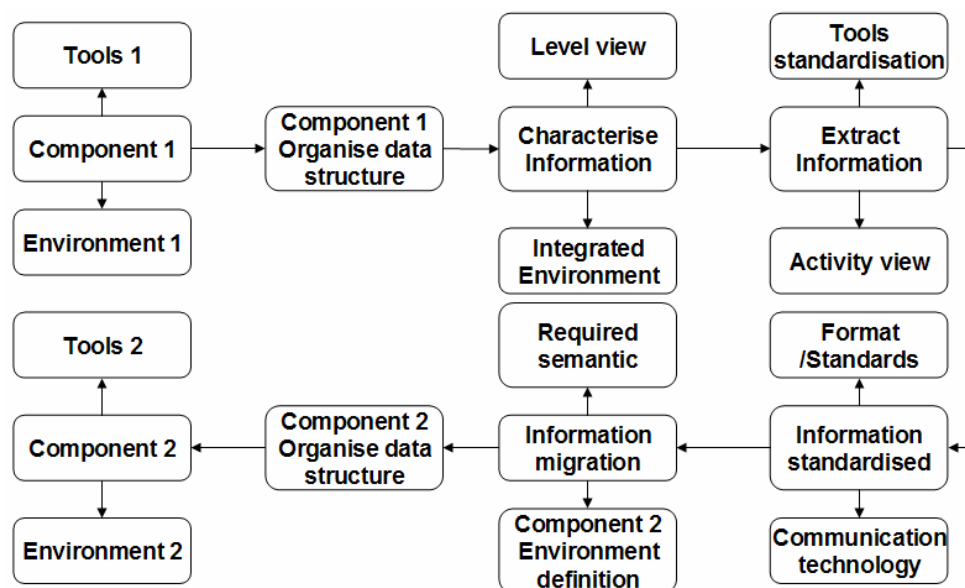


Figure 44 – Definition of the process defining the interface between two components

The process starts from the definition of an initial component (component 1) that is designed with a specific tool (quoted 1) in a specific environment (quoted 1). One important point to begin the process is to map and organise data structure so as to be comprehensive and standardised between system environments. The following step is the characterisation of data to be transferred. The latter sets of data have to convey semantic as well as context definition so that they can be migrated easily from one to another environment. Extraction of information is then applied when the information is consistent, organised enough and pre-processed regarding the activity it is going to be used in. This extraction will be made in regards with information that is required for an activity and of tools standardisation. The next step to consider is the standardisation of data on two main views. The first relies on the communication technology used and the second on the standardisation applied. Finally the component has to be prepared for the migration through the two next steps that are first the information migration which defines and characterises the semantic level to implement and the characterisation of environment definition that allows the next step of data structure organisation for the migration in component 2 used for tool 2 in environment 2.

In the collaborative framework the workflow integration consists in allowing the implementation of business processes across the different tools necessary to cover them. To insure the continuity of a process between different tools, it is necessary to develop interoperability between the workflow capabilities of each tool. This interoperability is built through workflow connectors.

E. Integrating the resources

The topics concerned by the definition and the method implementation are:

- ❖ The management of the large data sets between the activities of simulation and product data management. This part identifies the repositories in which information is shared by activities and partners.
- ❖ The management of the large data sets between the activities of simulation and design. This identifies the relation and coherence between design data (and associated models) and simulation data (and linked models).
- ❖ The management of the functionalities required for simulation activities.
- ❖ The management of the different functions and applications required in the fields of simulation and design.
- ❖ The management of the hardware architecture and infrastructure in order to have a physical view on the interface definition.

- ❖ As the different applicative tools identified to build the collaborative framework architecture are not based on open standard, the implementation approach will be incremental.

V. Conceptual functions

A. Architecture deployment on conceptual functions

The framework we define is characterised by different functions. Some are related to the general characterisation, others to the functions of components, and capabilities for exploitation in a system. Figure 45 groups the high level objectives. Based on our audit of the system, we have four main capabilities to develop:

- ❖ Development of cross-activities exchanges for a more coherent development
- ❖ Management of exchanges and interchanges between activity teams
- ❖ Management in configuration of all engineering data
- ❖ Development of the use of standards in the industry

High Level Objectives	AS IS	TO BE
Manage and Share Engineering data from Design to Test & Validation stages between multi-disciplines	Informal exchange of data based on proprietary format between not related tools. Local management of engineering data for each disciplines	Management and control of engineering data shared through: formalised data flows and an information model (for Virtual Aircraft) based on a consolidated repository
Share / Exchange Engineering data between partners and/or with contractors	Engineering data exchange based on files (proprietary or open format)	Engineering sharing based on an Information Model for Virtual Aircraft (open format)
Manage in configuration all Engineering data	Independent configuration management for each disciplines and partners	Manage an Information Model (for Virtual Aircraft) in configuration and in coherency with the configuration management of each discipline or partner
Take standards into account and influence them	Discipline model not based on standards. Data Exchange partially realised through standard. Workflow not based on standards	Support the virtual aircraft by an Information Model based on standards and able to integrate proprietary data

Figure 45 - Expression of conceptual functions linked to high level objectives

These functions, once implemented, should provide the capabilities of managing transaction objects in the architecture. Figure 46 identifies the different framework capabilities in regard with the functions to implement.

Framework related capability	Function
workflow integration	workflow communication: managing workflow orchestration, sending message between workflows transaction management: linking rights management on product with activity
Simulation Environment	Simulation & test management CAE interface framework access: import PDM structure, CAD files, Mesh files (for mechanical simulation) from framework, export results, simulation parameters
Collaborative environment portal	multi-view, workflow enabling, user links editor

Figure 46 - Expression of functions in regards with the framework related capabilities

Finally, in this deployment, we have identified four main innovative points to develop:

- ❖ Link design-simulation-result for traceability: To ensure a traceability chain from design to simulation and test activities
- ❖ Product Context Management: An end-user view on Engineering data and process to reach the environment required to perform a given activity
- ❖ Information Model: To define the semantic and the overall semantic links that exist between product, process and knowledge
- ❖ 3D&FEM Interface management: To permit multi-partners activities on the same product by defining logical and technical connection between components

B. Multi-view approach

Huifen [Huifen 2003] illustrates in his work the need of enterprise integration using product information. In fact, as far as we consider product design in multi-enterprise view, we face the integration of information and processes to optimise the supply-chain in a meaningful way of reducing time-to market, and enhancing quality and costs. One approach to characterise this integration is based upon the conveying of product data at the different levels of enterprise. A multi-view approach becomes today a major consideration for enterprise performance. Its justification is based on the associativity, coherency and synchronisation of data between activities performed

during project. This systemic approach is supported by integrative processes [Presley 1997, Shunk 2003]. This approach relies on the integration of four main models:

- ❖ Information model that provides the most relevant engineering data
- ❖ Data model that provides coherent engineering data (version and configuration)
- ❖ Application model that provides engineering data in the right format definition
- ❖ Processes that enable engineering data access in regards with activity, rights...

Multi-view must provide interoperability and coherence of processes with engineering data, linking them in a collaborative and reactive structure based on information systems [Shunk 2003, Noran 2003]. This must ensure the coherence of processes involved in activities while considering workflows between disciplines and, at higher level, enterprise.

C. Layer approach for referential

While analysing the framework, we noticed in chapter V that there were systems interacting each other and with the collaborative framework. This notion of systems that need to be integrated leads us to define layers to exploit. Figure 47 represents this layered approach and analyses the integration of the different 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 furnishes 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.

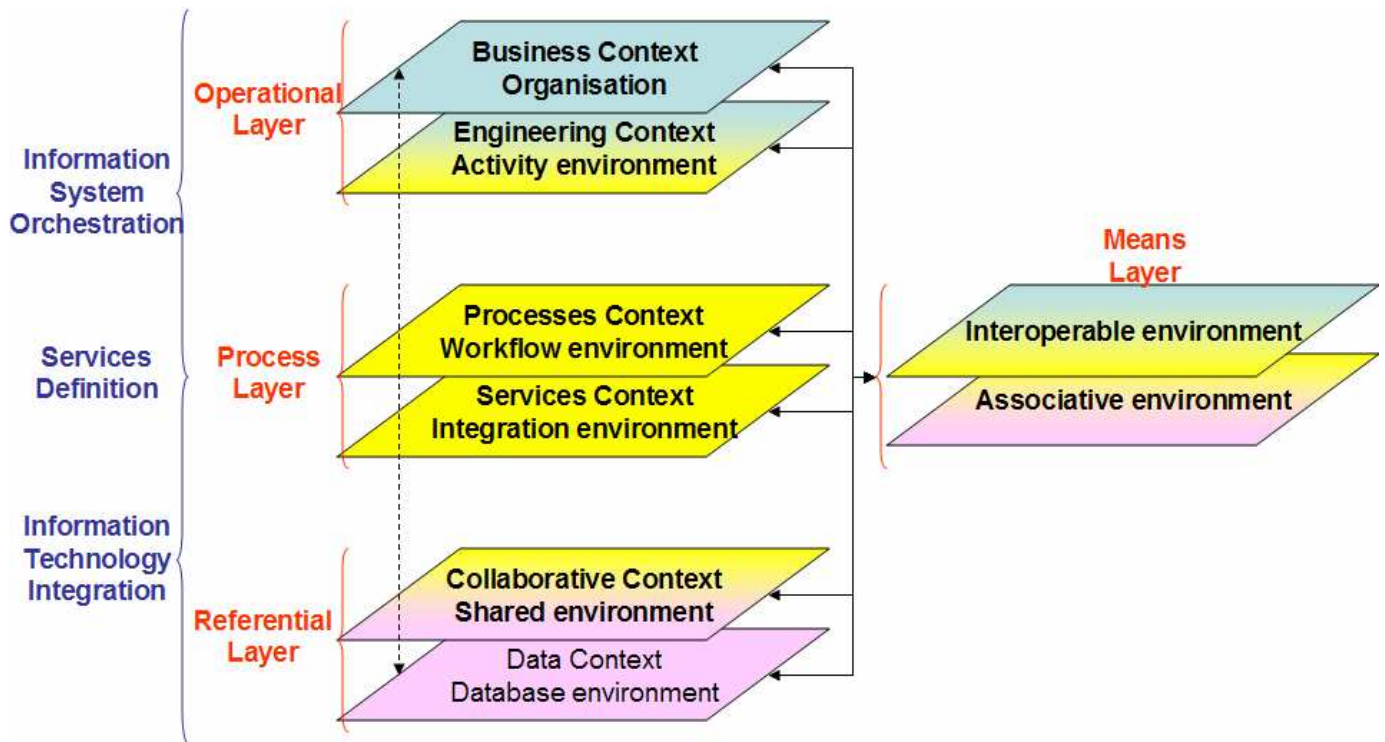


Figure 47 - Representation of system orchestration

VI. Conclusion

In this chapter, we have defined the first step of building the architecture of our platform. Here, three conceptual layers are identified:

- ❖ The operational layer concerning the tools used in the activities
- ❖ The Interoperable layer providing capabilities to set up the context and provide capabilities of data exploitation
- ❖ The collaborative layer providing the reference for activities

We have identified that their independent implementation is a condition to fulfil the major requirements. The use of transaction models is a condition to allow the different modules to work independently.

The integration of data, processes and workflows is then based on the identification of contractual objects that influence and implement the different systems. The need to provide a multi-layered platform should then manage the transaction objects whose role is to implement the business systems.

Chapter VI: Preliminary orientations on architecture and technologies

I. Abstract of Chapter VIII

In chapter VIII, we present the development of the conceptual layers and components that are going to be implemented to define our collaborative environment. In a first part, we propose to highlight the conceptual deployment based upon conceptual functions in order to provide the general composition of the architecture and its major functioning principles. This chapter is composed of the part presented in Figure 48.

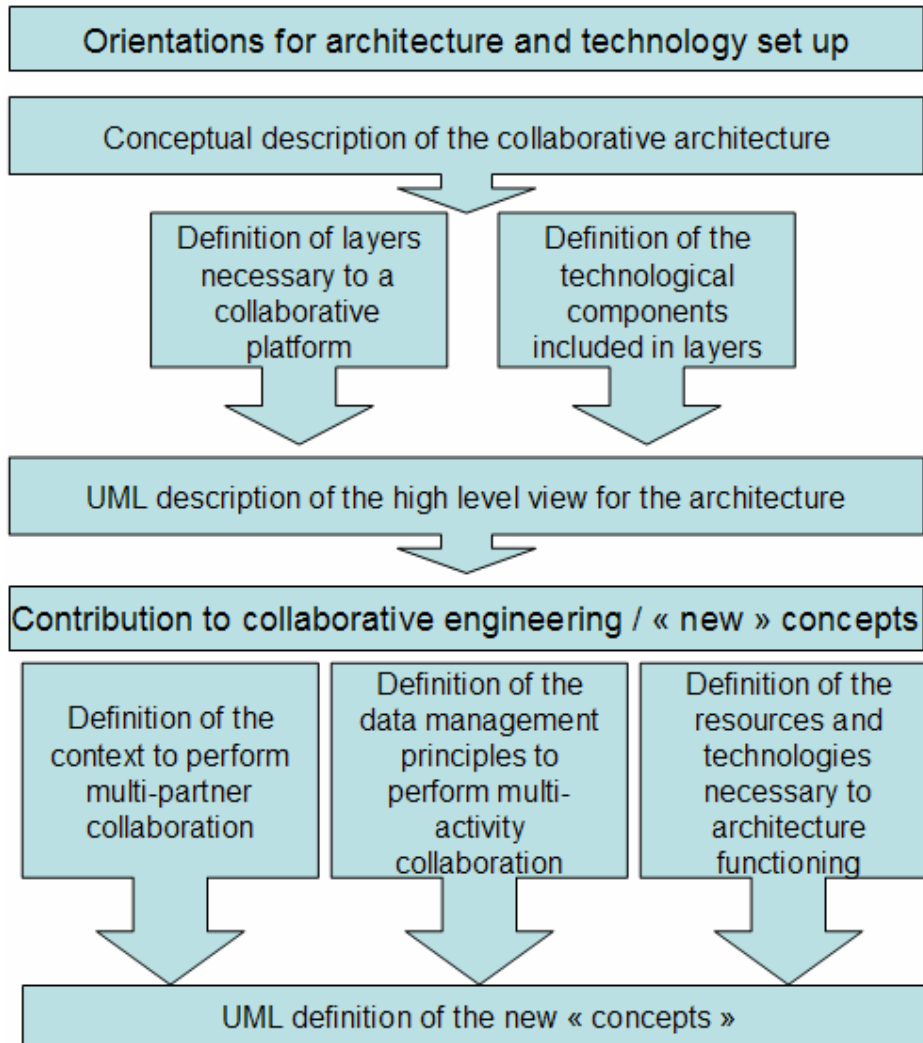


Figure 48 - Guideline of the chapter VII – From the high level description of the architecture to the definition of innovative components for collaboration

This chapter is composed of three main parts:

- ❖ The first concerns the definition of the high level view of the architecture. It defines the necessary components to ensure multi-partners and multi-activity collaboration

- ❖ The second concerns the definition of new concepts in a collaborative architecture. It defines the notion of context to set up the engineering environment, the notion of data management in the context of multi-activity needs, and the notion of resources necessary to the architecture functioning.

II. Overview of the collaborative framework

A. 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 49.

- ❖ 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 different partners, different domains or different 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 supplies 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.
- ❖ The interoperable layer implements the notion of 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 (associate with the information Navigation service) that offers the contextualisation of the information (Product, Process, Resource, applicable knowledge, Decision ...),

- ❖ Domain Model component that provides a frame for each domain or discipline to share the Information needed for collaborative tasks,
- ❖ Finally, the collaborative layer is composed of the consolidated repository. The consolidated repository is also connected to different databases which are called regarding the engineering domains.

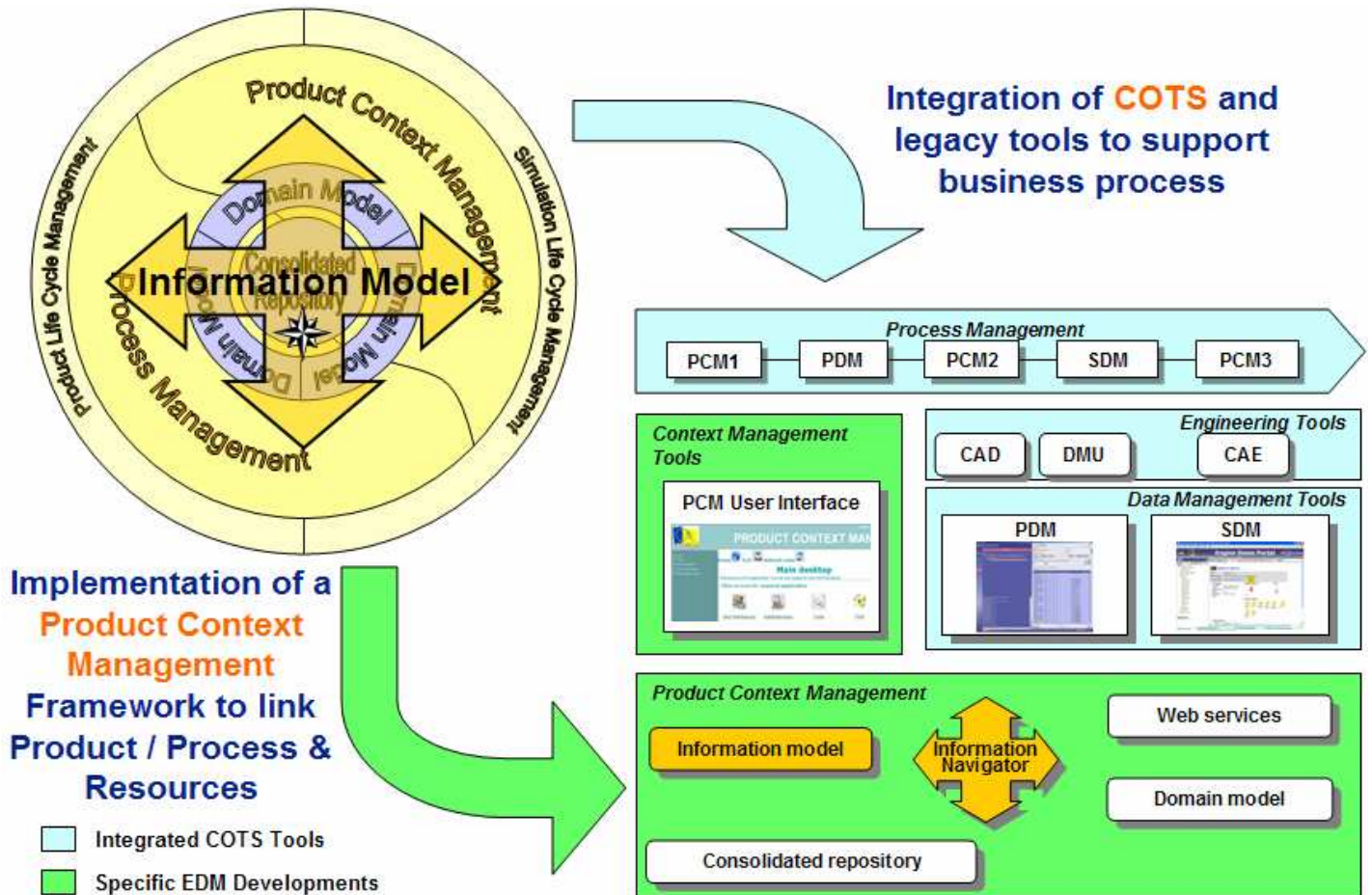


Figure 49 – Collaborative logical architecture

B. Modelling of the architecture – General structure

Our architecture is then composed of three main layers that include the different engineering and business objects necessary to perform concurrent product development. Figure 50 details the different layers and components that compose our collaborative environment.

- ❖ 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 inscribed 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.
- ❖ 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 able to interpret the different 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.

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 different databases environments. Accessed using method

objects that capture data relevant for the domain models, it is the source of collaborative and interoperable data.

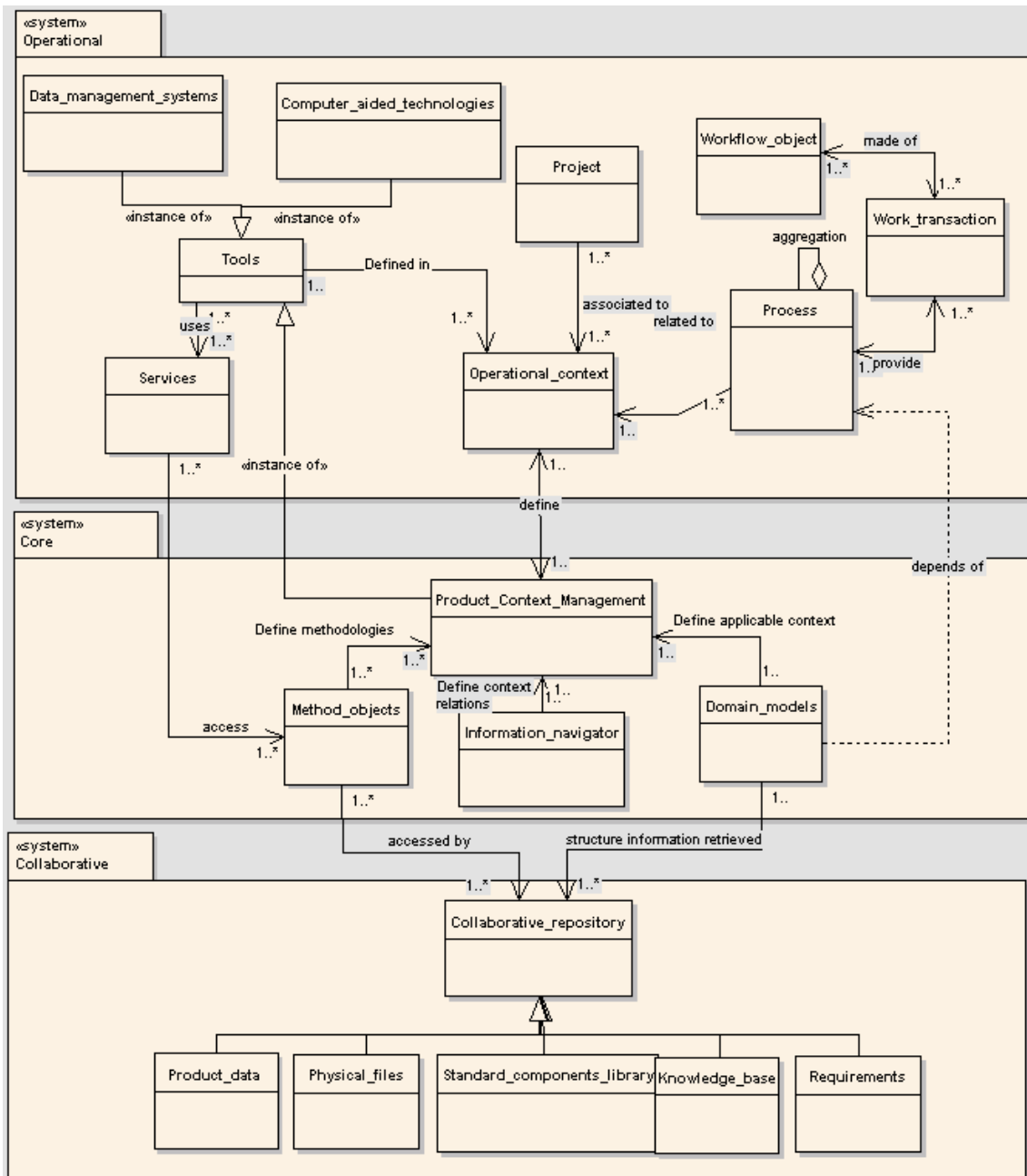


Figure 50 - UML description of the layers and components of the targeted industrial system

C. Modelling of the architecture – general processing

During the collaborative activities, many actions are performed on the data through their flows in the different layers of the architecture. Figure 51 shows the different objects that influence data and the different transactions and methods used in

order to be considered as "collaborative". Following the layers, three main processes are involved: the operational business activity, the core analysis process and the collaborative integration. The operational business activity is performed in the operational layer. Using the product resources (composed mainly of product data and their environment) it transforms and creates information using resources available (e.g. organisation and technological resources). Besides, this environment is also constrained by the activity. These constraints are the basis of the definition of the context.

Once an activity is performed, a transaction with the interoperable environment is launched. The result of this transaction validity sends back the request as a work order to reinitiate an operational business activity or goes to the core analysis process.

The core analysis process analyses the data incoming from the operational environment. After the validity check passed on the compliance of the system with the system, data are decomposed and analysed using the models structure. This means that data are first analysed regarding the domain they represent using the domain model, then they are decomposed. The information navigator will finally provide the capability to associate a data to a context in the consolidated repository.

The publication transaction is then launched. According to the result of data analysis, transaction is sent back in order that data are processed again in the operational business activity, or is migrated to the collaborative environment to be integrated.

The collaborative integration is in charge of integrating the collaborative data using the context determination. First, data are consolidated to ensure their persistence, they are then integrated in the database using the database governance schema. A check of this integration is then launched on the data before they become released data.

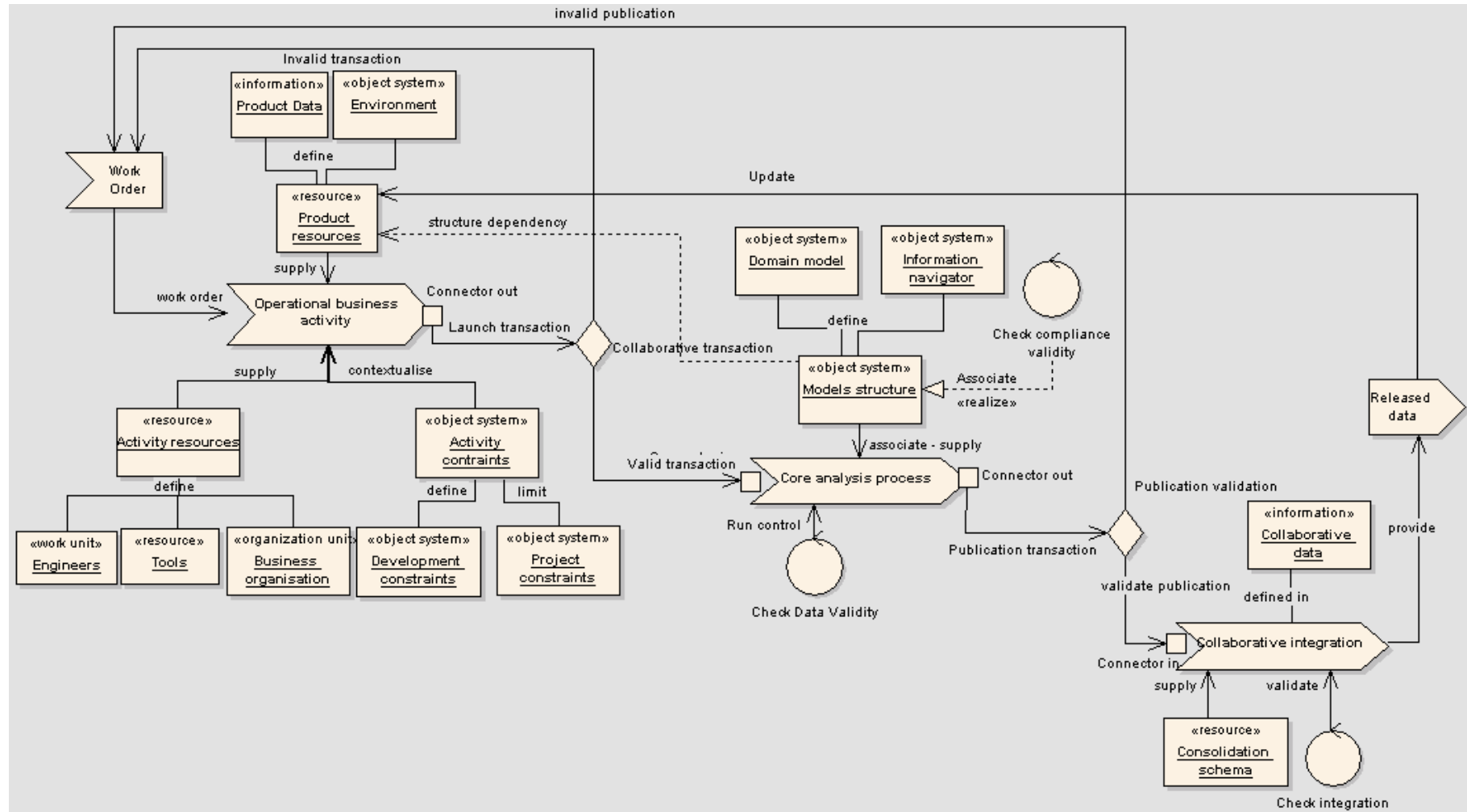


Figure 51 - High level description of the transactions in the architecture

III.Context implementation

A. Logical model to define the context

The major objective of the context definition is to provide an access for the end-users to the collaborative data, going through the different models layers. The context definition is represented by Figure 52. This context is composed of the interaction of three main environments: The engineering environment, the product environment, and the models environment.

- ❖ The **engineering environment** is represented by the composition of project and process. The workflow object is an element interacting with an activity. It creates the sequence of activities using the parameters transmitted by decision and event instances.
- ❖ The **product environment** represents the way transformations and developments are made. Interacting with the references and the resources, it defines the databases (and then necessary data and information objects) and tools used to perform the transformation.
- ❖ The **models environment** defines the filter applicable to define the context. The PCM links the different information provided by the information model such as product, process/activities, knowledge, and resources...The way PCM is linked to the information model ensures interdependencies between the objects. Its major role is to retrieve and to provide a high level view of the high level model provided by the information model. In the product environment, it is possible to retrieve also the product attributes and data concerned by an activity in a specified context.

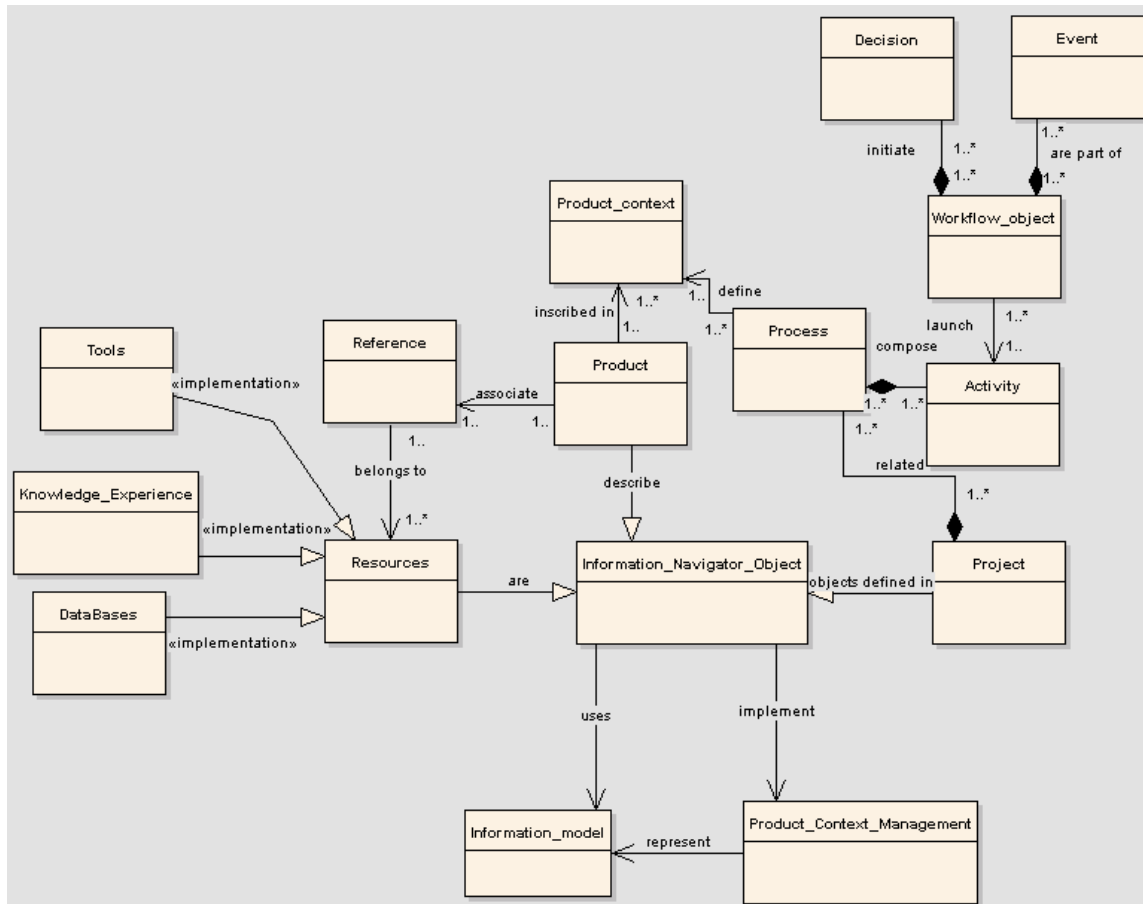


Figure 52 - UML description of the context

B. Components necessary to define a context

1. Definition of the information model

The Information Model provides a mechanism to define an engineering context linking product, process, resources and the applicable knowledge to perform a business activity within a project. The Information Model is generic in the sense that it is independent from a particular engineering domain, but also because it is reusable (instantiable) in all projects. It provides services of query and persistency in this engineering context. The Information Model bridges the concepts found in process modelling and product data modelling in order to allow business units to access product model data in an intuitive way. The goal is to provide a data model, which allows users to define reusable process modules, which can be formally approved and reused within the design and analysis process. Interdependencies between these process modules can be defined, as well as the required product data that must be used and produced by each process module. Defining the essential links, it contributes to the definition of the dynamic view with components such as workflow, query and

schema. Finally, the aim of the Information Model is not to provide a process description mechanism but to offer services to manage the history of the process in consistency with the product evolutions.

The Information model is generic enough to be applicable both at the business and the operational level of the processes as well as at the different stages of the product lifecycle. The users of this data model will be knowledge engineers, engineers responsible for defining approved processes within the manufacturing company and operational end-users.

2. Definition of the process management

The Process Management is shared by:

- ❖ The Product Context Management for the “static” implementation of the Process (the link of Product-Process interaction)
- ❖ The Workflow Engine for the “dynamic” implementation of the Process. The Workflow Engine component ensures the integration of different layers of Process and the integration of the operational workflows provided by COTS (PLM or SLM).

3. Definition of the Product Context Management tool

The PCM belongs to the information model. It represents the user interface designed to use the context approach provided by the different layers. **The Product Context Management (PCM)** represents an extension of the PDM systems. In the PCM the Information Model takes the role of the PDM product structure and provides a development context oriented access to the different kinds of product information stored in the consolidated repository. The PCM provides all the necessary services concerning the User Interface (UI), workflow and information management capabilities, resolving dependencies and establish view concepts (capability to filter data according defined development parameters). It determines product/activity context through the information model, to determine data approach through data models, to view the product related information of consolidated repository and to manage the different collaborative applications and workflows. The collaborative framework represents a kind of administrative tools to access the different data provided by the different partners. Constructed as a PDM/PLM system, it also provides the essential of information workflows and context of the data of the consolidated repository.

The Product Component Management offers interoperability services to integrate an “in house” user environment. Using an external reference mechanism Product Context Management is able to keep a link on the proprietary information managed by COTS, legacy application or process management system. The information provided in the Product Context Management is the necessary subset of Domain Model to perform integrated activities. The user of the proposed system is responsible for the definition of the data subset he wants to provide. Nevertheless the link to the proprietary environment from which they have been extracted is maintained by the Product Context Management.

C. Technological set up of the context

The major functioning principle is based on the relationships between the information model and the tools for the enterprise environment, the data model for the interoperability environment and the consolidated repository for the collaborative environment.

Once data has been generated by the tool, the information model is in charge of mapping the different context information's. It defines the different parts of the data model used to characterise the data attributes. It also describes the different methods applied through the business objects. Once this task is completed the context for data use is then defined.

The information model is the first interface object that ensures the data communication from the tools to the consolidated repository and vice versa. Here, we are attempting to define the communication rules between the information model and the surrounding components.

The business objects represent the executive objects of the architecture; they are in charge of the request conveyed from an application to another. The business object is an object that belongs to an applicative context or aspect defined. Methods are then conveyed through the business objects to the targeted population data. The business object represents a viable solution for linking process, product and knowledge data. The major objective of these objects is to define the associated methods to data actions. Business objects are defined regarding the information model context, the data and current processes. Once an action is engaged in the tools environment, data are disseminated in the information model layer, on order to links these data with a context. This context is defined regarding relevant methods described in the business objects. Then, when in a specific context an action is launched on data, methods are assigned

in regards with the application. The result is that business objects are in charge of determining and assigning required data from the different layers definition.

There are 4 main components to a business object:

- ❖ **DEX (Data Exchange Set)** – A subset of the Enterprise Product Model. Here the Enterprise Product Model is the ISO 10303 – 239 [ISO10303-239] data model, but the Enterprise Product Model typically contains additional extensions to cover the lifecycle of the product as identified by the gap analysis.
- ❖ **Reference data library** – This is a set of standardised datum referenced from a standardised product model. The distinction and the need become apparent when thinking of supplier data. Reference data are a recognised solution to this problem addressed in ISO 13584 and ISO 15926.
- ❖ **Business rules** – In addition to rules defined in the Enterprise Product Model and the DEX there may be the need to define company specific rules. This can either be done by extending an existing DEX or through a method rule and method schema.
- ❖ **Methods** – These are functions operating on the DEX and on the data. A business process means, defining how an activity will be carried out, will typically interact with the product model data via function calls on a business object. Methods are typically transactional in a read/write operation.

In this architecture, requests are applied through the “Business objects” called using the Business process means they result in the call of methods to import; export data...The call of business objects results in applying strategies on the consolidated repository.

IV. Product implementation

A. Model to manage product

The main objective of the collaborative architecture is to define a structure for the product management including data and meta-data in multiple domains. Figure 53 represents the logical model of product management. This structure is composed of two main objects: the technology objects (related to the implementation for a given context) and the engineering objects (related to data management).

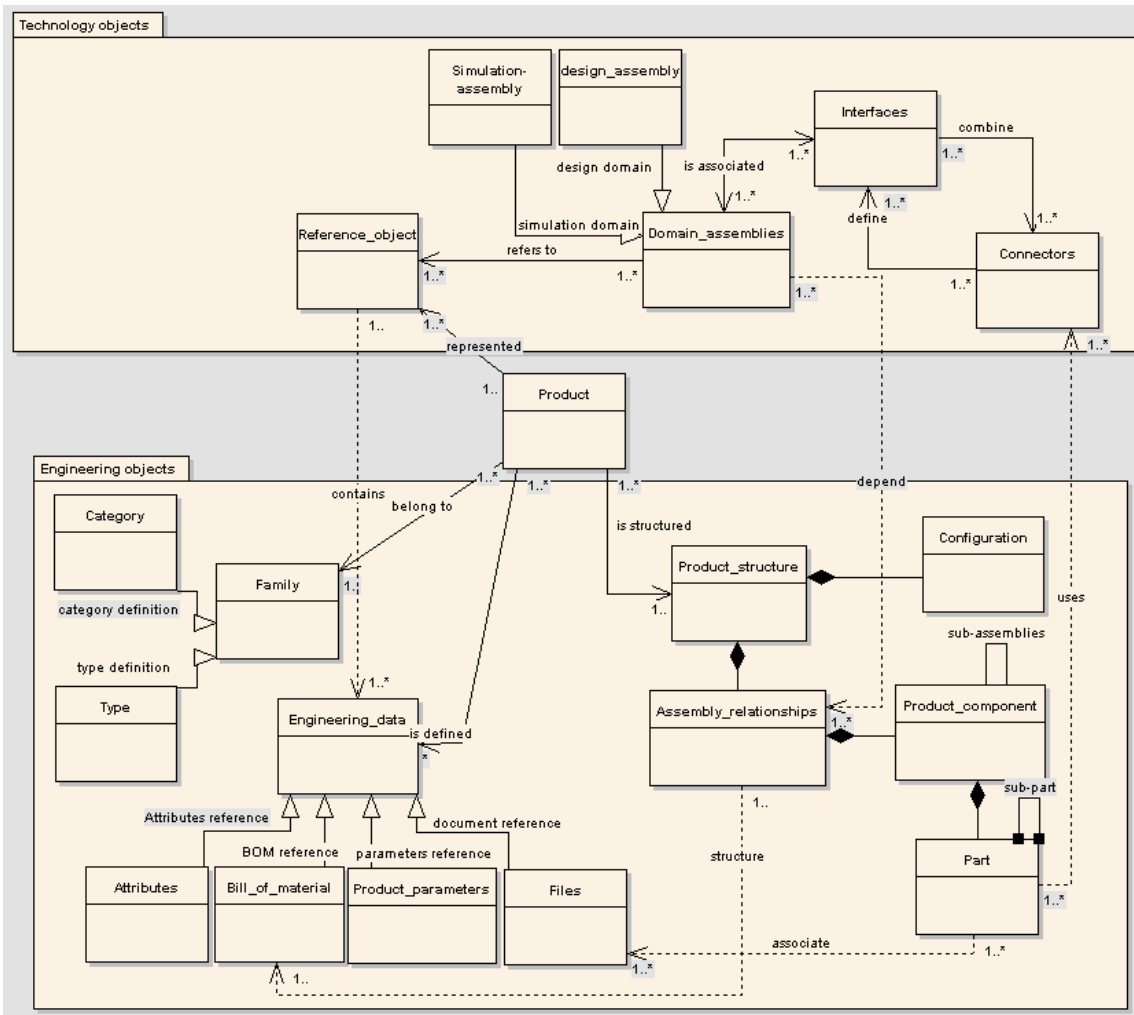


Figure 53 - UML description of product management

- ❖ The technology objects define **the management of the product references** in different environments and applications domains. These objects are identified as the technological management of associativity and relationships between engineering data.
 - ❖ The first component is the reference: built on DMU capabilities, it represents the referential object for the dissemination of the engineering data in the applicative domains represented by domain assemblies.
 - ❖ The domain assemblies. It is the cross-point between the domain assemblies. It defines the links between objects assembled in different applicative domains.
 - ❖ The associativity connectors and interfaces: the interfaces represent the logical and physical connection between the components of the product. These interfaces are defined by the association of connectors. In fact, as

the connector defines a logical or physical connection of a product part, the interface is in charge to associate similar connections.

- ❖ The engineering objects represent the **different data and meta-data attached to the product**. They represent the logical schema governing the product structure.
 - ❖ First the identifiers for the product. They define the product family using a classification into categories and types. This is a first characterisation which is deeply linked to the notion of project and implicitly involving the notion of context definition.
 - ❖ The engineering data represent the data created during the product development. It is composed of the attributes, bill of material, product parameters (mainly used in simulation environments) and the files attached (such as the 3D files).
 - ❖ Finally the product structure. Associated to the definition of domain assemblies, it defines the assembly relationships between elements and associates the notion of product configuration.

B. Technological definition of the product

1. Definition of an integrated product structure

For a better compliance and integration between the multiple engineering domains that use the product definition and data, we need to define an integrated structure for the data management. Figure 54 represents the conceptual definition of the integrated data management structure for a cross-domain integration.

The definition of the product structure must follow the definition of classical PDM tools. The structure must also follow the organisation of part definition and assemblies detailing the composition of the product. Here, we have defined two specific environments for the management:

- ❖ The **integrator view** (or administrator of the product development): The integrator is in charge of the exploitation and of the check-in of the whole product. The access is performed at the root level enabling both the view on partners' product definition and on the interfaces.
- ❖ The **partners view**: As partners are in charge of the definition of a product part, their view on the collaborative environment is restricted to the definition of their product.

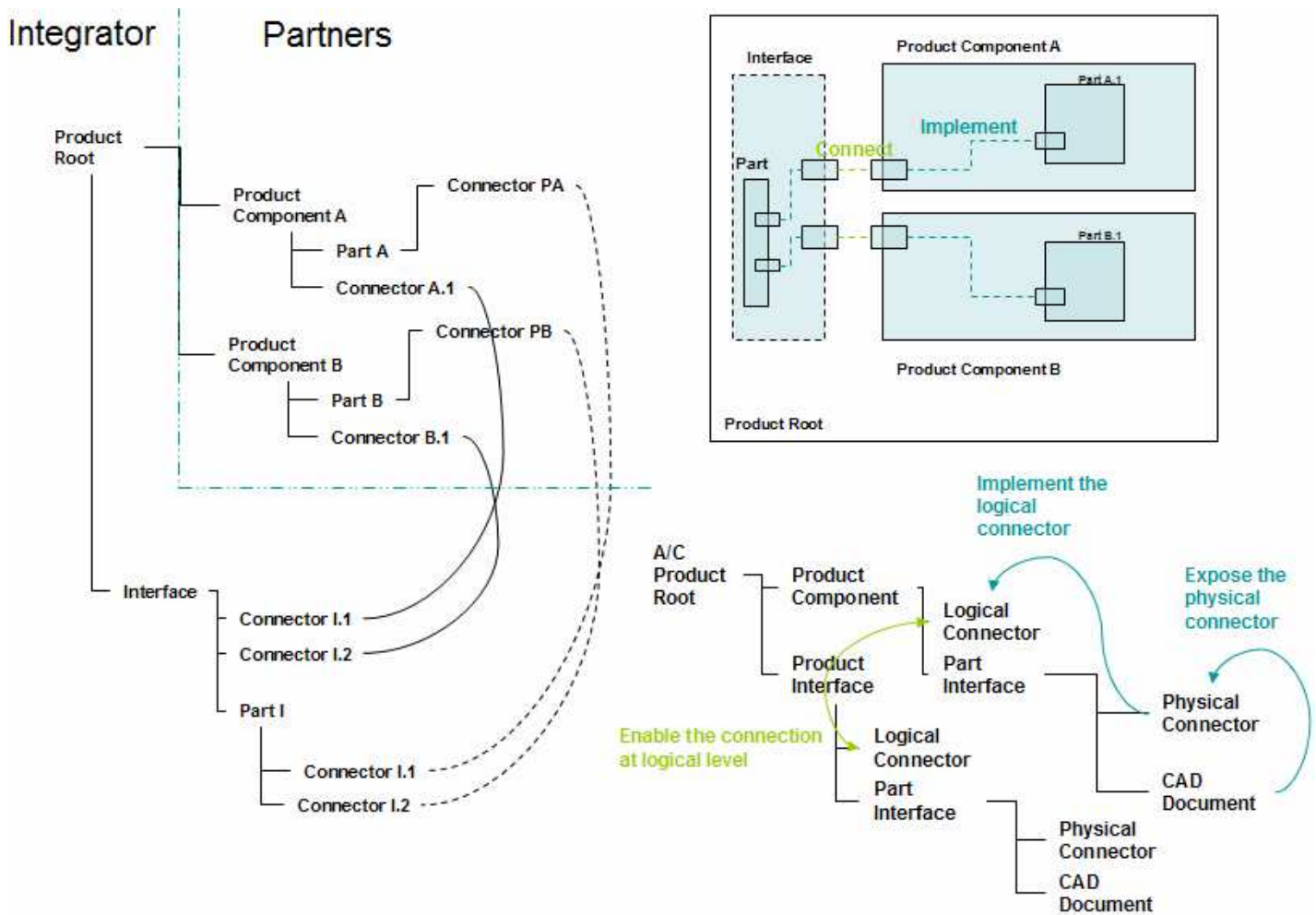


Figure 54 - Conceptual description of the data structure for an integrated management

As described by Figure 54 (right part and bottom of the schema), two kinds of connectors must be implemented:

- ❖ The **physical connector**: described using 3D components they allow to create an external reference in the design space. They also provide the design limits of components to the design teams.
- ❖ The **logical connector** describing the logical connection of product elements. They define for example the type of assembly constraints, the absolute or relative position...

Note that a logical connection is implemented by a physical one. This means that the definition of the elements is implemented using both a physical description and the characterisation of the relation.

Figure 54 (right part and top of the schema) also reveals the construction of the interface. A part of a product is implemented using a connector defining the properties of the physical connection. This physical connector implements a logical connector that defines how structural elements have to be plugged into each other.

Logical connectors both from product and interface are going to result in the integration. Finally, the logical connector of the interface is related to a physical one. The interface part manages the relationship between two physical connectors and results in their integration to finally connect two logical connectors defining two product components.

2. Architectural components of the product definition

The targeted objective is to ensure the use of the different data created during activities to be efficiently integrated between different domains. The integration is processed using two kinds of models:

- ❖ The domain models defining the semantics used in a specific context and activity
- ❖ The consolidated model defining the connection between the different semantics and components of activities

The Domain Models provide mechanisms for business users to describe subsets of their ontology (exhaustive conceptual map that describes a domain). The domains users (mechanical, aerodynamic, system ...) from the different engineering disciplines (design, simulation, test ...) describe subsets of their model. They define their view on the product and they delimit the information they want to share for collaborative engineering activities. The formalism and the vocabulary used to describe the Domain Models remain very close to the business ontology. The Product Context Management component uses Domain Models to produce the business views expected by the business users.

Once product data have been formalised using the domain models, they have to be consolidated to be used by different activities. The Consolidated Model provides a mechanism to standardise the Domain Models. The Domain Models are translated into a standardised ontology: the STEP application protocol 239 (PLCS-Product Life Cycle Support). This translation into an interoperable format based on STEP results in merging (or a partial merging) of the different Domain Models in a Common Reference Model at the project level. Multi-domains and multi-disciplines optimisation and validation are possible at the Consolidated Model level. In the case of a minimal consolidation of the Domain Models, Consolidated Model can be seen as a Common Reference Dictionary of the project. The Product Context Management component uses the Information Model (by the Business Object methods) to navigate and retrieve

Data in the Consolidated Model and produces a view of these data in the Domain Models formalism.

The data model represents the interoperability interface necessary to define the links between attributes defined in the different tools. The data model is in charge of the data attributes parsing and referencing in order to permit the storage in the consolidated repository. The data model represents the different objects and attributes manipulated by the different tools/activities. It defines and links the attributes that have to be associated from a model to another to a standardised ontology and models. The data model represents the operational view directly linked to the information model and that will provide, through the requests of business objects, the required data in the system. The data model is composed of multiple models defining each one an associated view on the product. In our framework architecture it was defined through an approach of design/simulation. The different models associated with these views are relevant for the information model view. The data model is a consequence of the contextualisation made by the information model because it is relevant for only one view of the product.

The objective of this layer is to clearly define the different object and data models relevant for activities. It defines the tool data models and links them to the reference data model that will define the implicit links between attributes.

V. Resources implementation

A. Overall overview of resources implementation for the collaboration

The overall process worked out for the analysis and the data integration is defined in Figure 55 (general description of the process) and Figure 56 (UML sequence diagram). It represents the decomposition of the “core analysis process” into a set of sub-systems analysis enabling three major actions:

- ❖ **The characterisation of the environment for engineering**
- ❖ **The definition and data semantic breaking**
- ❖ **The application of methods enabling the flow of data in the system**

Once exported from the operational system, data have to be checked. This is achieved by the process of compliance analysis. Using rules and schema templates that define both the structure of requests and determine the input structure of data, the

system performs an analysis of the compliance of the request and checks data to analyse the possibility to integrate them.

Once the compliance defined, the first step is the definition of the environment. Using the information navigator, it defines the environment conformity regarding the data processed. The conformity is performed using the product template to identify the compliance of the data exported from operational tools. Then, it uses the context template to determine the processes and the projects involved. Finally it exploits the resource template to determine the different queries.

Once the environment defined, a method is applied to determine the domain analysis. The definition of the environment defines the structural resources and environment for the product definition. The resources involved determine the different business objects available for the management of the product. Once a method is determined, it is applied to the determination of the domain. The domain template defines and characterises the data semantic exploited regarding the tools templates defining the data structure. This enables to break the semantic objects contained in the data exported from operational tools, and finally their migration into the collaborative environment.

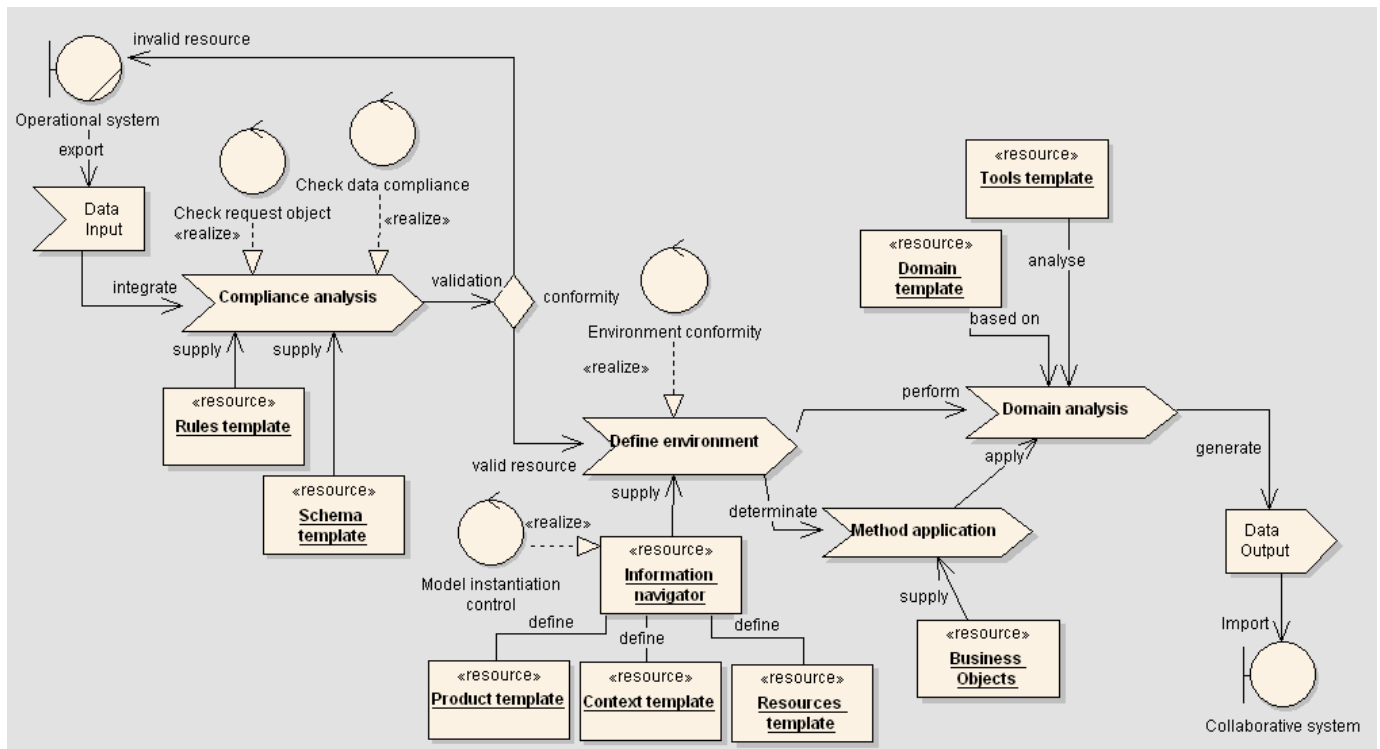


Figure 55 - Process principle for analysis and integration of partners' data

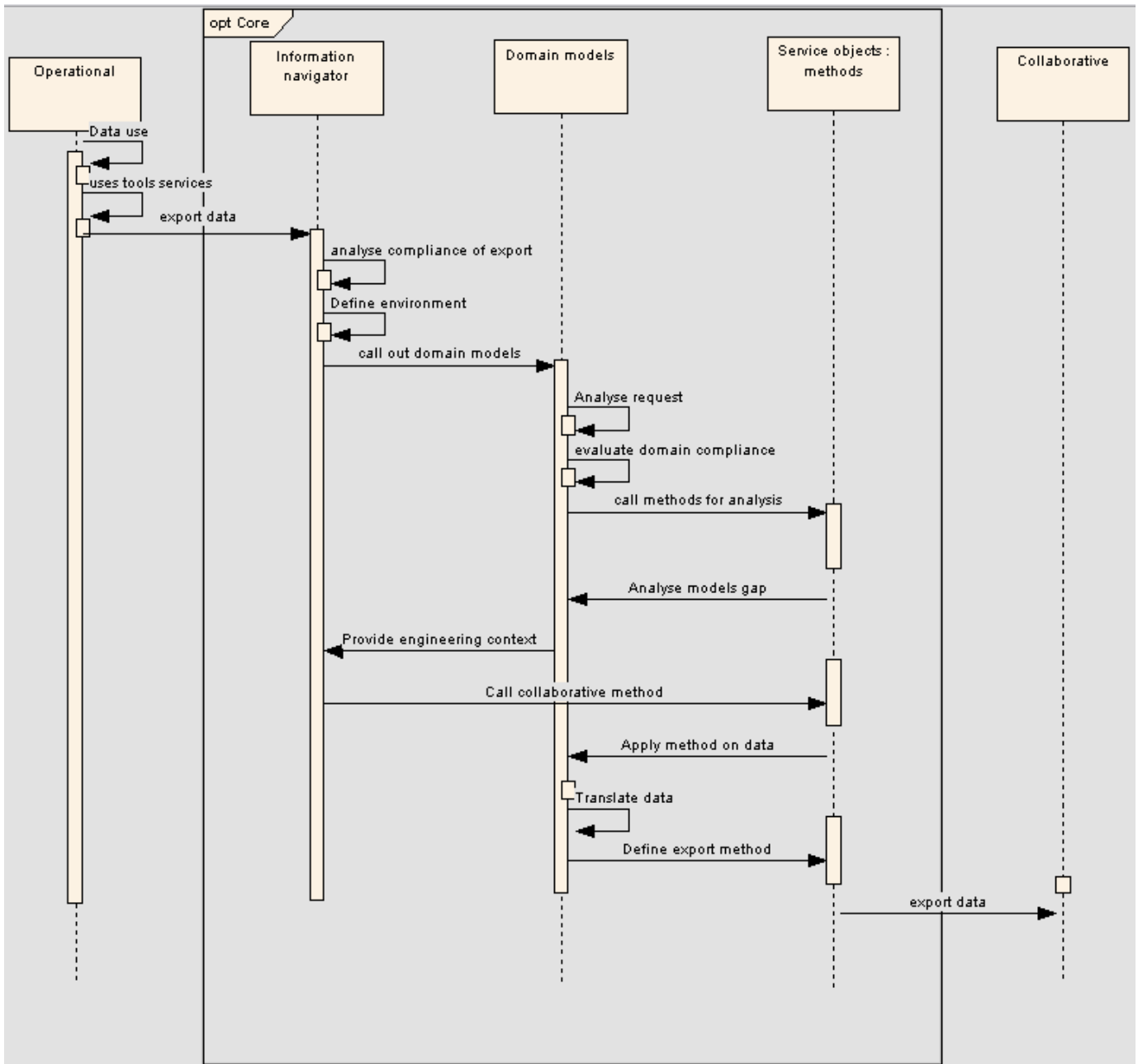


Figure 56 - UML sequence diagram of transactions in the analysis and integration process

B. Defining the services components

The collaborative server provides the services to access the infrastructure and to navigate on the consolidated Information. The Web-services provided are:

- ❖ Login – Log into collaborative environment by using user, group (role at extended enterprise level) and password
- ❖ Logout – Log out of collaborative environment

- ❖ ExecuteQuery – Execute a predefined query; note that queries can also update data; a parameter of the query is the action to perform on the result of the query.
- ❖ ExecuteMapping – Execute a predefined mapping in order to convert Information for communication purpose.
- ❖ ImportData – Import XML data by applying an XSL pre-process (optional) before importing in Part 28
- ❖ ExportData – Export XML data by applying an XSL post-process (optional)
- ❖ ExecuteRules – Execute business rules on a sub-population of data

VI. Rules and functioning principles

A. Communication between components

The mechanism to build the different models (see Figure 57) is made of:

- ❖ The description of the Information Model using the EXPRESS formalism: description of the different concepts defining the engineering context, description of the methods enabling the navigation on Data
- ❖ The extraction of the semantics from the different domains and disciplines: the extraction is performed based either:
 - ❖ on a set of real data if this set exists
 - ❖ on a proprietary model describing the data structure of an operational tool used for the business activity
 - ❖ On a standardised representation of the domain model (UML or EXPRESS formalism).
 - ❖ Then the semantics of the domains and of the disciplines is described in EXPRESS. The link with Information Model is built by defining a query schema in EXPRESS-X. The Information Model navigates on the Domain Models through the query schema.
- ❖ The consolidation of the Domain Models in the Consolidated Model consists in the definition of a mapping. It corresponds to the attachment of the different business objects from the Domain Models to a standardised object defined in the Application Protocol 239 [ISO10303-239]. The mapping leads to EXPRESS-X functions.

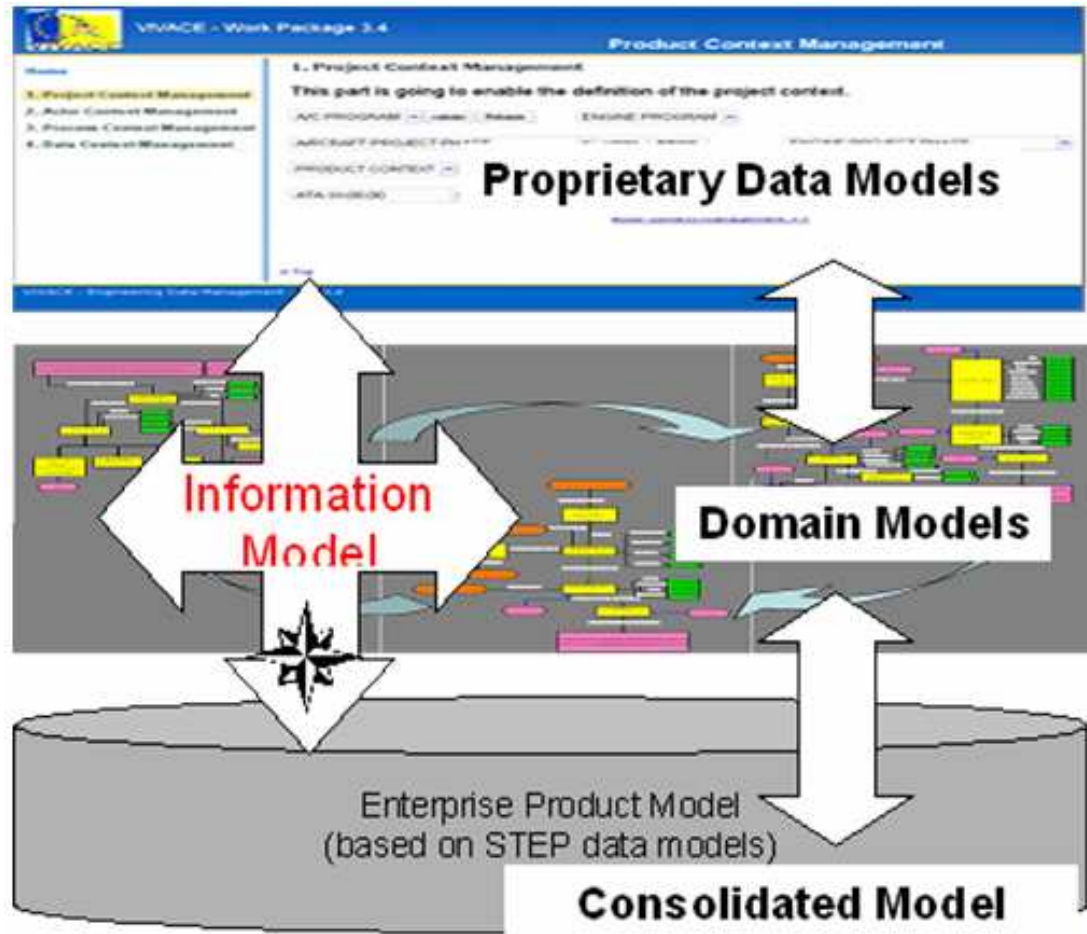


Figure 57 – Building of a Common Reference

The possible clients of collaborative framework architecture are COTS tools such as PLM applications, SLM applications and operational or collaborative portals based on Web technologies.

The PLM applications (Product Life Cycle Management) are classically used for design activities; they are associated to CAD and DMU tools (Computer Aided Design and Digital Mock-up). The PLM applications manage the Product from the design point of view, the Engineering Data are organised around the Product Structure tree and the processes are ensured using a workflow engine dedicated to follow the activities of Product design inside the application.

The SLM applications (Simulation Life Cycle Management) manage the simulation activities and provide integration with CAE tools (Computer Aided Engineering). SLM applications ensure the traceability of Simulation Data produced during a simulation activity inside the application.

B. Collaborative framework functioning

Figure 58 illustrates the mechanism to navigate on Engineering Data and depicts how the different models are accessed:

- ❖ The collaborative client queries the system for an Engineering Context linked to the activity to perform: The Information Navigator retrieves this context in the Information Model and loads it.

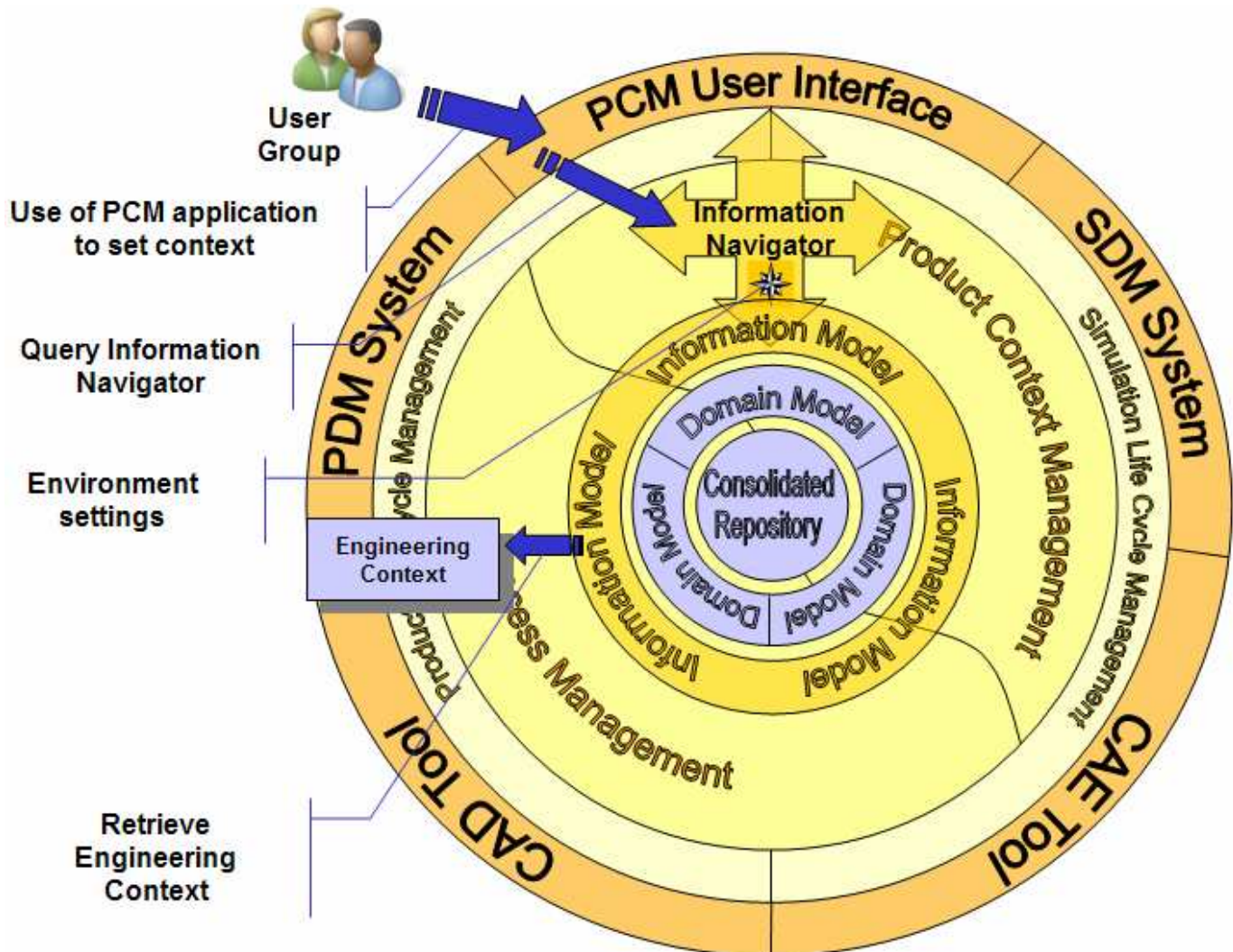


Figure 58 – Engineering context setting

- ❖ From this Engineering Context the collaborative client is able to send a query on a sub-population of the engineering data: The Information Navigator performs the query on the Consolidated Model (repository) to retrieve the subset. The data are then translated from the Consolidated Model formalism (PLCS) to the Domain Model formalism and loaded.
- ❖ The collaborative client directly works on the sub-population identified, all along the transaction (transaction: time frame to interact with the system for an

elementary activity) (see Figure 59). The modifications are directly performed in the Domain Model. At the end of the transaction the Engineering Data are consolidated through a committing operation: a reverse translation is applied on the Engineering Data from the Domain Model formalism to the Consolidated Model.

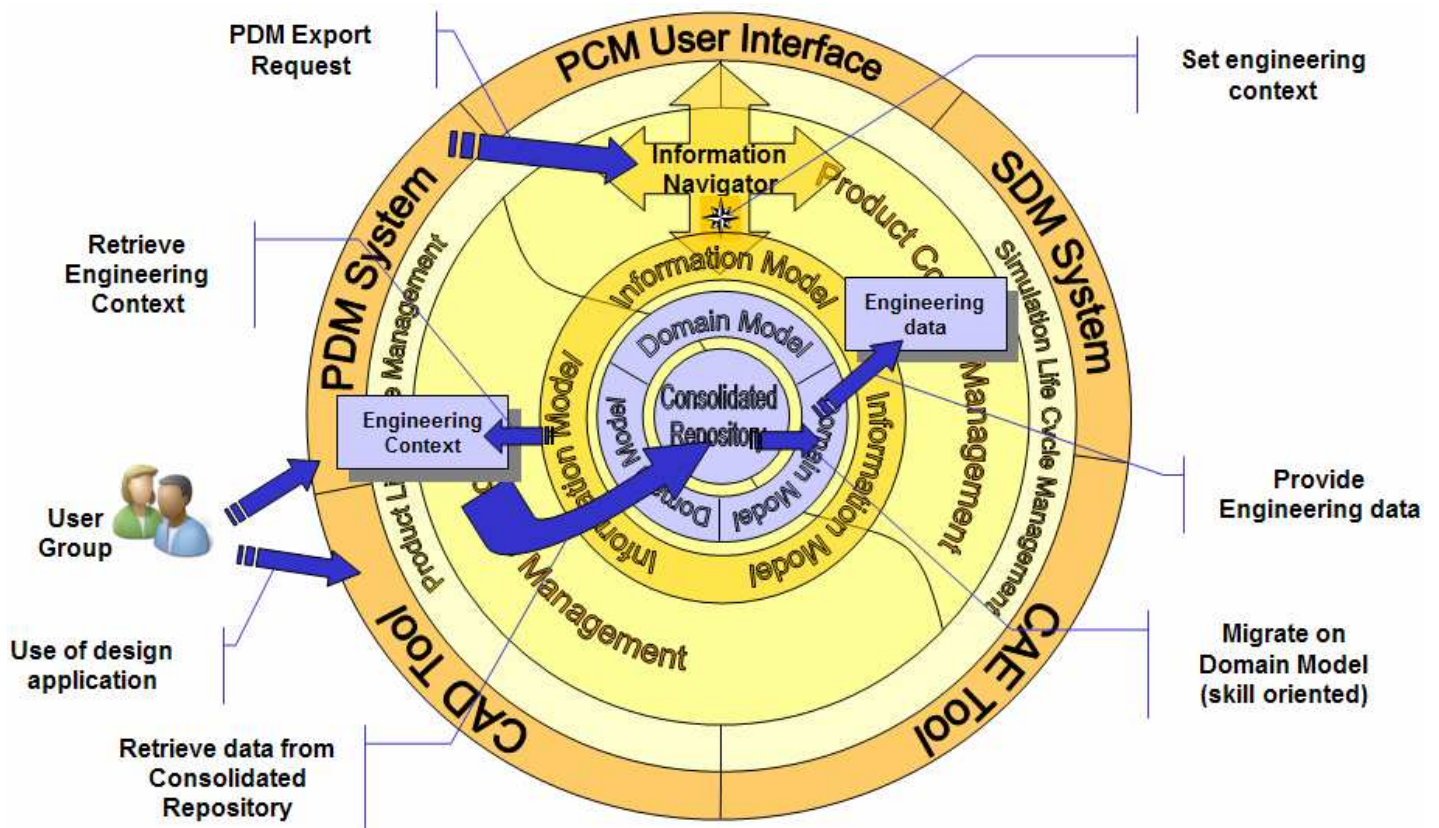


Figure 59 – Navigation on Engineering Data

The Information Model bridges the concepts found in process modelling and product data modelling in order to allow business units to access product model data in an intuitive way. The goal is to provide a data model, which allows users to define reusable process modules, which can be formally approved and reused within the design and analysis process. Interdependencies between these process modules can be defined, as well as the required product data that must be used and produced by each process module. So, its major role is to link high-level objects of the activity context such as product, process, resources and applicable knowledge. Defining the essential links, it contributes to the definition of the dynamic view with components such as workflow, query, schema and business objects.

The process pattern composed of a request acceptance, an activity execution and a result evaluation (request assessment / perform activity / result assessment) is used in the collaborative framework as an elementary brick to build up a complex

nested process. This pattern illustrates how the Information Model creates and maintains a link between a product context and an instantiated process. The Information Model manages the work requests, and allows the creation and management of the context of the activity to perform by defining the overall means, resources and constraints.

The role of Business Object is to implement the contextualisation of the necessary environment to perform an activity. The Business Object provides methods to retrieve Information (Persistent Object) in different repositories. These repositories could be the consolidated model of the collaborative infrastructure for the Information involved in collaborative activities, or external repositories (PDM, SDM, Knowledge database, Decision database, process library ...).

Each process managed by the Information Model has a context. The contexts of nested or chained processes are linked together. These links provide a multiple disciplines integration transferring through the workflow all the environment of the process. The links provide a traceability of the process and allow for the transfer of the information to share of a provider activity to a receiver activity the Information to share.

The Process Management is shared by the Product Context Management for the “static” implementation of the Process (the link of Product-Process interaction) and by a Workflow Engine for the “dynamic” implementation of the Process. The Workflow Engine component of collaborative framework ensures the integration of different layers of Process and the integration of the operational workflows provided by COTS (PLM or SLM).

VII. Synthesis

In the chapter VIII, we have described the major specifications of the development of a collaborative framework. The specification of such an intermediary system enabling the connection between elements in different layers has critical issues related to:

- ❖ The relationships between the components involved in the architecture
- ❖ The management of the information

This leads us to consider the notion of creating the reference frame and possible uses to reach those performances. Looking further into this description, we have described the notion of interfaces and generic processes, first step towards the consistency and association of data for a use in multiple domains.

The different components described contain the technological objects providing the different functions required. They are also the interface ensuring the identification, the analysis and the validation of the different transaction objects exchanged between the working teams and the industrial systems.

Chapter VII: Proposal of the “collaborative schema”

I. Introduction

In this chapter, we propose to develop the conceptual views described in the previous chapter using the EXPRESS-G formalism (See ANNEXE 2). The choice of this formalism is motivated by the need to create a standardised architecture (see the “innovation process” for the definition of the “collaborative schema” proposed in Figure 60). Many concepts and elements have already been developed in different STEP compliant with the ISO 10303 application protocols. This describes only the major implementations to provide. Indeed, this schema is based on the definition of the aeronautic environment, but it is generic enough to be applied in another industrial domain.

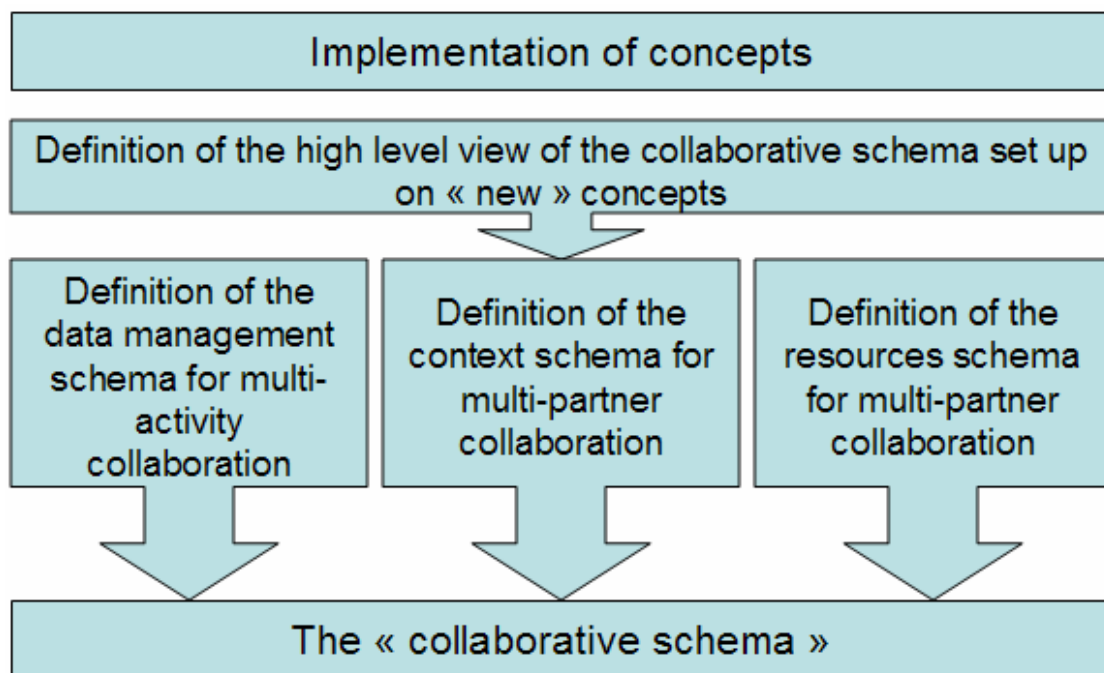


Figure 60 - Definition of the “collaborative schema”

In the first part, we propose to describe the high level model that supports the definition of our architecture. We then describe the models attached to the definition of the product, to the context implementation and finally to the generic resources providing the serviceability of the framework for system engineering.

II. High level model support – Definition of Information Navigator

The central concept defined here is the **Aircraft Product Model** (see Figure 61) which supports the link between the **Aircraft data structure**, **aircraft activity context** and **aircraft activity resources**. Using the definition of the AP239 Product

identification (For details on STEP references, see ANNEXE 3), it defines the link between engineering activities and contexts.

- ❖ The **aircraft data structure** represents the logical structure for multi-engineering data management (**aircraft data management structure**) and the product structural aspects (such as the configuration represented **by aircraft product information view**)
- ❖ The **aircraft activity context** represents the definition of the engineering environment. It depends on the **application context** defined in AP214. Here, we define the **aircraft process definition** that links elementary activities performed during development (with the definition of the activity from the AP214). The **aircraft context definition** defines the application domains (here design and simulation)
- ❖ The **aircraft activity resources** represents the various models enabling us to:
 - ❖ Define the data structure from tools with Tools data structure
 - ❖ Define the application of services with services description

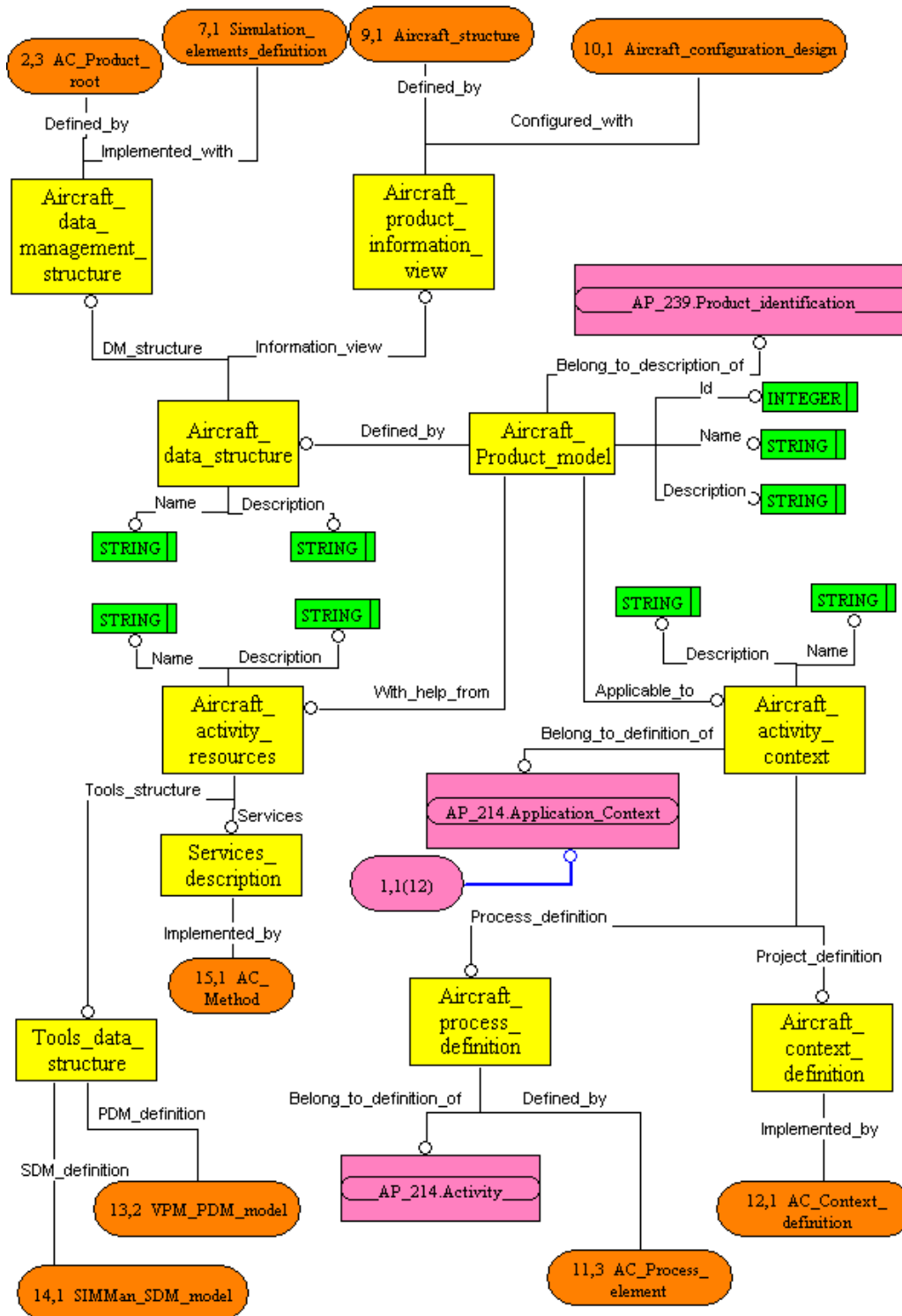


Figure 61 - High level model support of the collaborative schema – Information model for navigation

III.Information data for product reference – Implementation for Domain models

A. Description of the generic model for data management

Figure 62 represents the logical structure of the data management system involved in the collaborative framework. It is decomposed into two domains under the **product root** (representing the starting point of the whole product definition): the definition and the characterisation of product parts, and the definition of the product composition. Note that **product root** is also related to the notion of configuration shared by the partners (referenced configuration).

- ❖ The **product root part** characterises the structural part representing the entire product. It contains all the attributes to identify the component. It is extended to the definition and attachment of documents with **product root part model** and **product root document**. The documents are then characterised using the definitions of document and **external file id and location** of the AP214.
- ❖ The product composition is performed using **the AC node component** to define the multiple levels of product composition.
 - ❖ A first characterisation is that **AC node component** supports the structure for both design and simulation (**node part design** and **node part simulation**). It also supports the **node part interface** that defines the logical and physical connection between elements. **AC node effectivity** characterises the effectivity used for the node. Finally a logical component for the connection to interfaces exists also with the **AC node connector**.
 - ❖ The second characterisation concerns the definition of the component situation in the space revealed by the **component position matrix** that defines the 3D localisation using **component position point** and **component translation**.

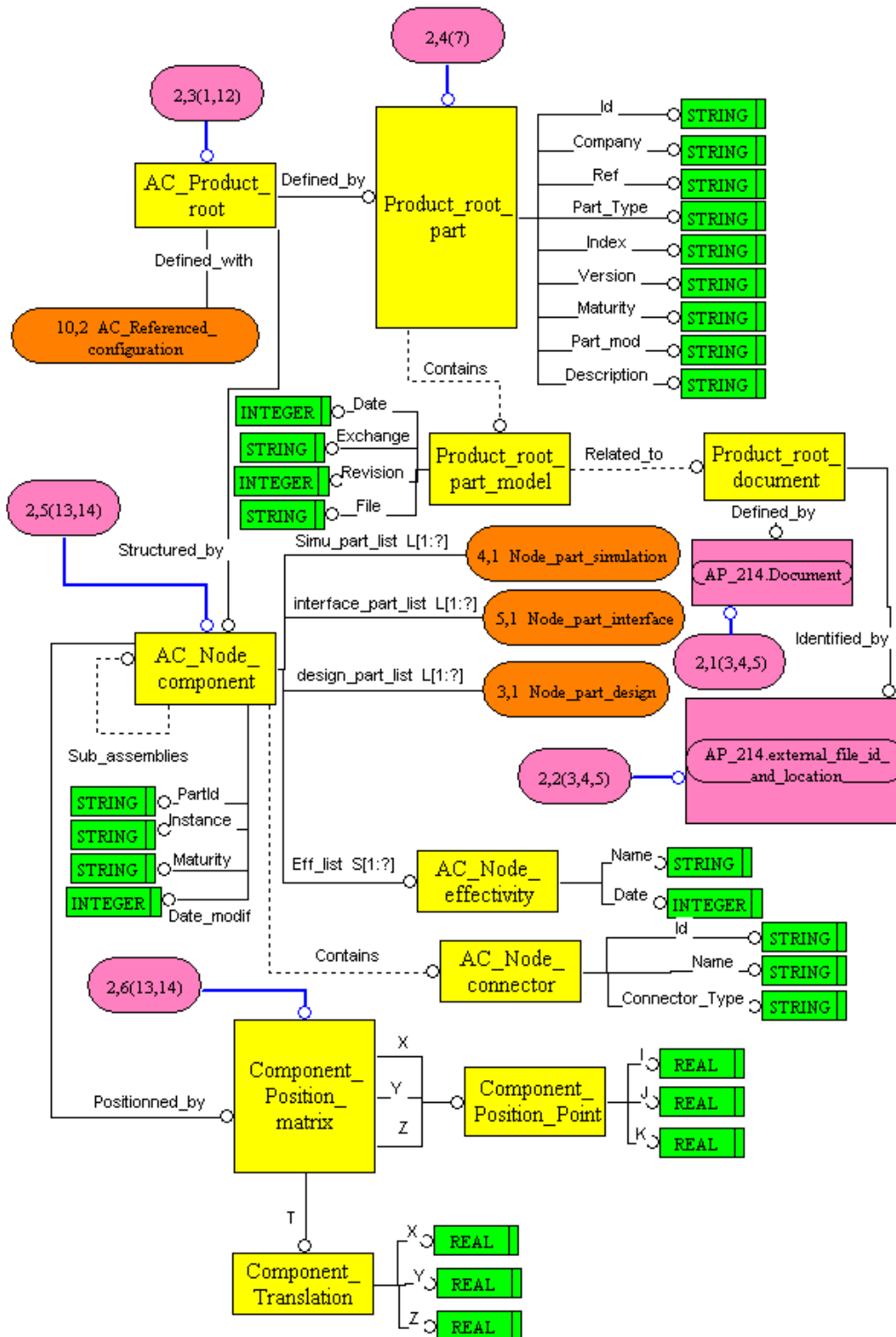


Figure 62 - Product structure management

Figure 63 defines the interconnection and the relationship between 3D CAD models and 3D FE models. This is deeply defined in the AP209 with the connection between **shape definition** and **fea model definition**. The result of this interaction allows the migration from documents contained in the design environment to the simulation environment.

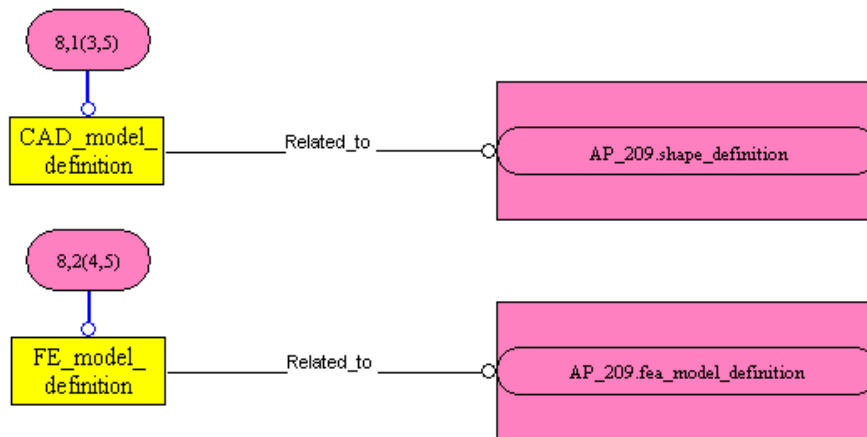


Figure 63 - Physical models relationships

1. Design view

Here, we define more precisely the design part. The composition is quite similar to the definition of the root (see Figure 64). Starting with the definition of structural part (**node part design**) containing product attributes, it supports both the notion of connectors and product definition.

- ❖ **Part design connector** characterises the logical connection between the different components that must be connected.
- ❖ **Node part design model** characterises the models attached to the product component concerned by the **node part design**. Documents are related to the component using **node part design document** and the identification via AP214 components (document and external file id and location).

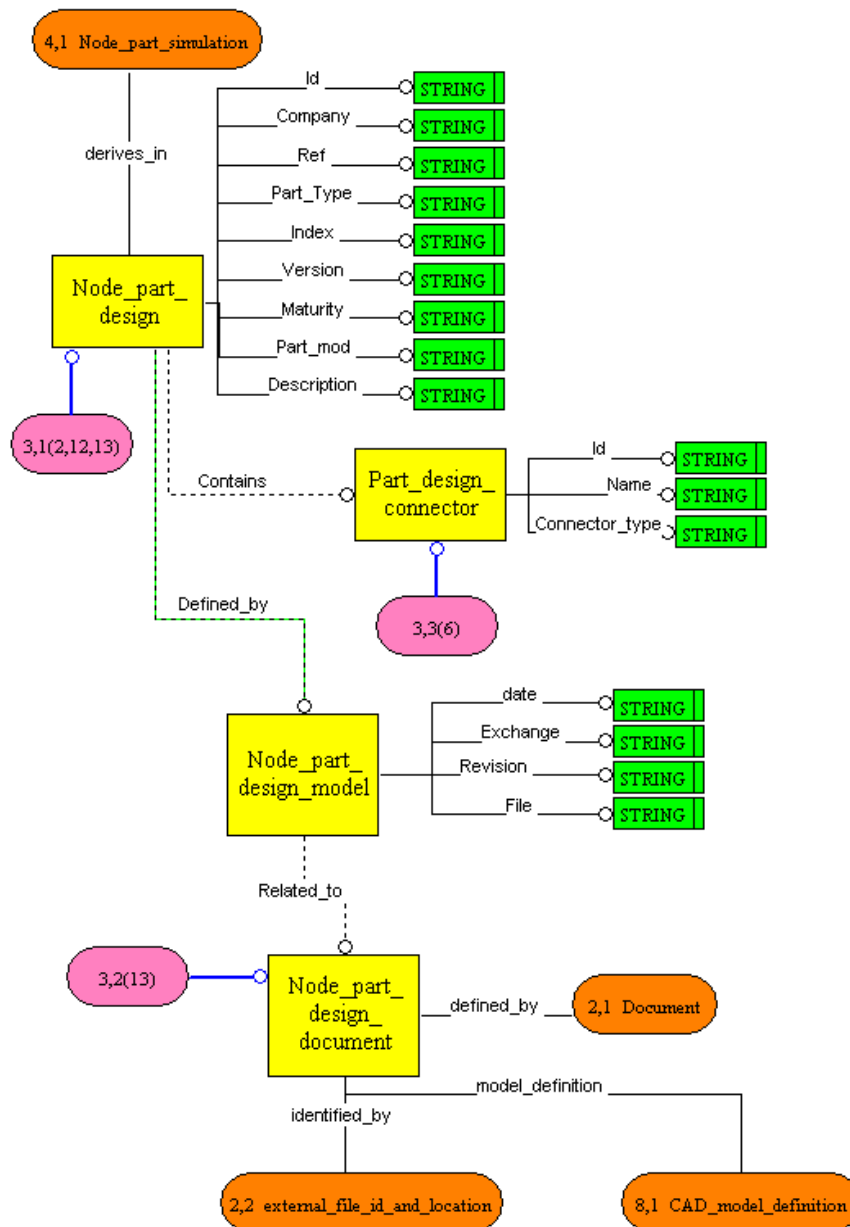


Figure 64 - Management of design data

2. Simulation view

The composition of the simulation data structure (see Figure 65) is the same as the design one. The only comment we can make here is that this composition derives from the design world. The definition of a structural part is supported by **node part simulation**.

- ❖ **Part simulation connector** characterises the logical connection between the different components.

- ❖ The **Node part simulation** model contains the definition of documents using **node part simulation document** and the identification via AP214 components (document and external file id and location).

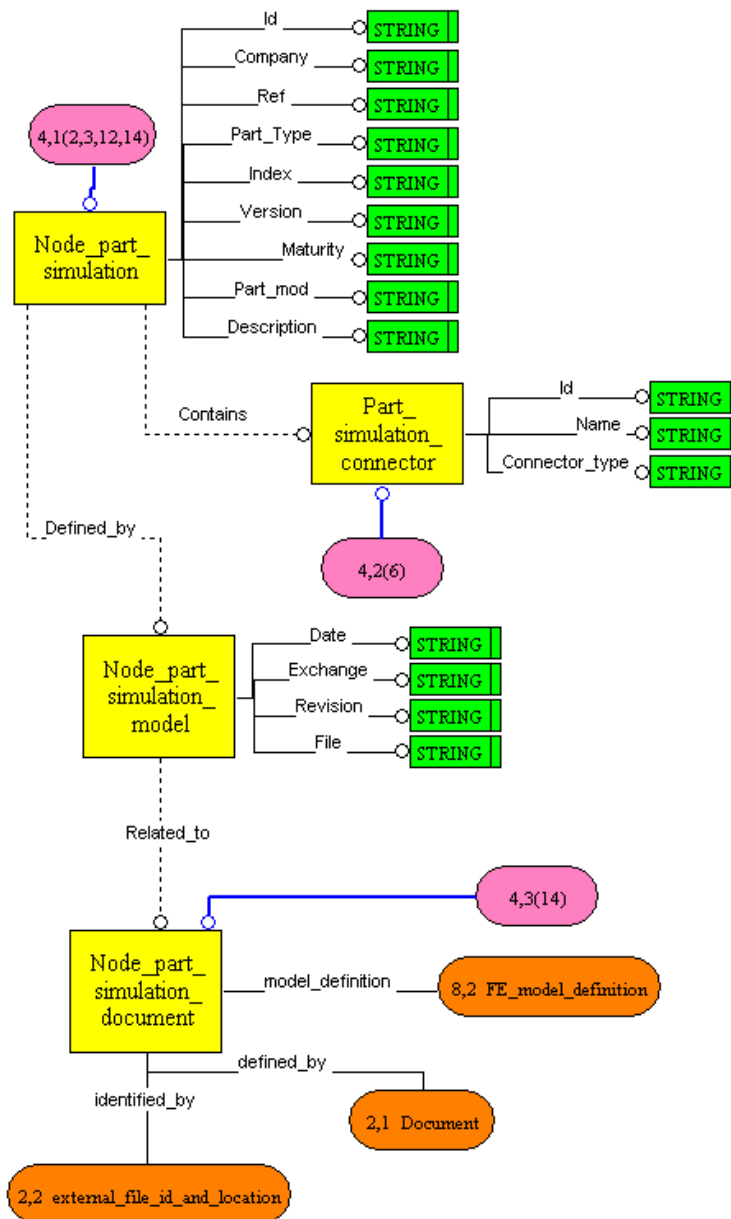


Figure 65 - Management of simulation data

B. Interface design/simulation

Figure 66 details the structure of the interfaces. This model considers interfaces as components of the product with the restriction that they are only structural objects. The entity **node part interface** supports both simulation and design interfaces. For each of the activities, there are two structural entities.

- ❖ The **part interface design connector** entity and the related entity for simulation **part interface simulation connector** ensure the connection within the connectors of the product components.
- ❖ Here, the entity **node part interface design model** (and **node part interface simulation model**) characterises the models attached to the definition of the components interfaces. The **node part design document** is defined (and **node part simulation document**) with the physical 3D CAD (and 3D FEM) and the physical interface. They are also managed with the AP214 entities of document and external file id and location.

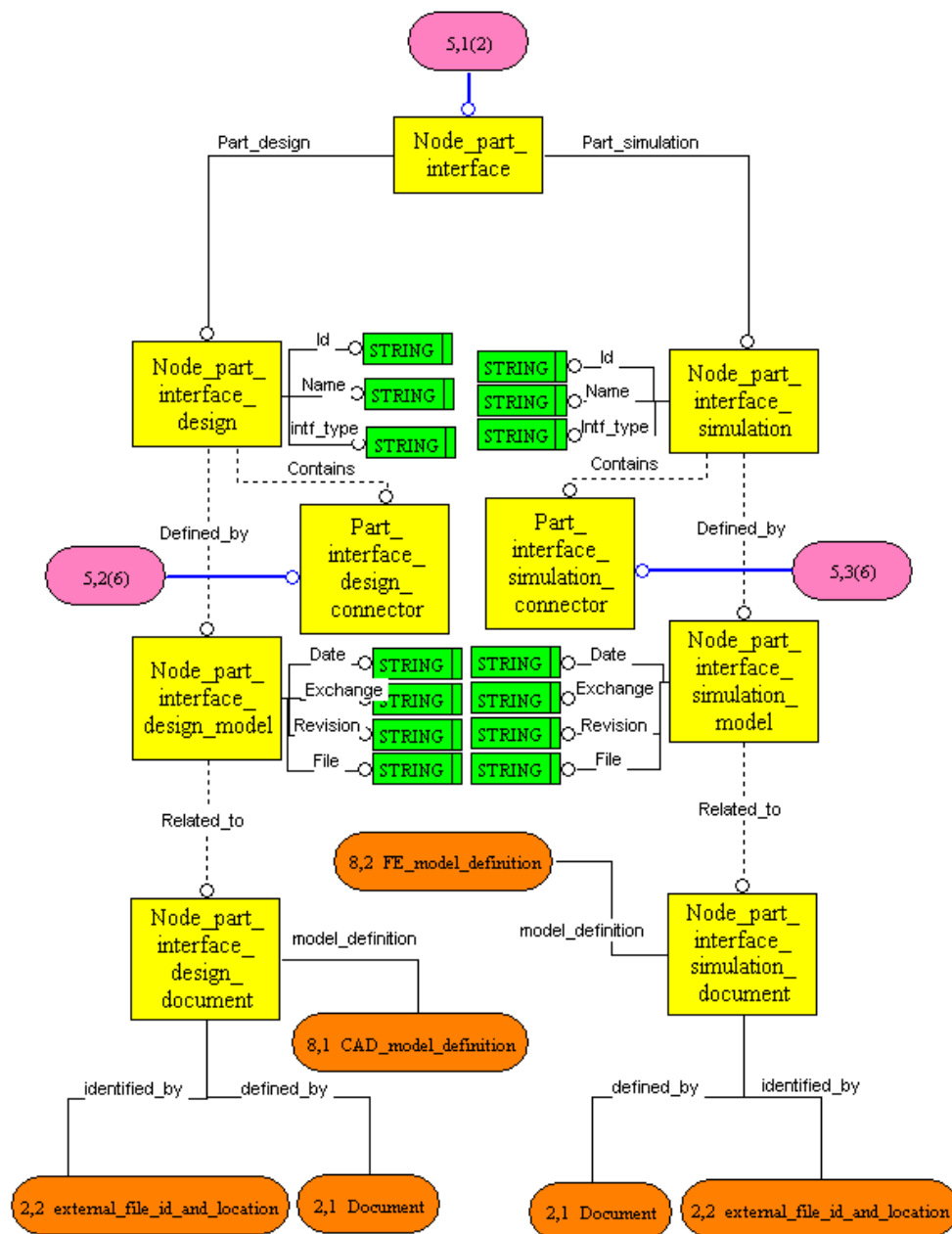


Figure 66 - Interface data management

Figure 67 details the relationships between design interface and simulation interface. The **related design connector objects** entity links the **part design connector** and **part interface design connector** in order to define the logical and physical connection by interface. The process is the same concerning the entity **related simulation connector objects**. **Interface part design** defines the structural interface that connects two components of the product. It is derived in **interface part simulation** to obtain the same principle in the simulation domain. To better characterise their interactions they are also characterised by the **interface connection** from the AP239.

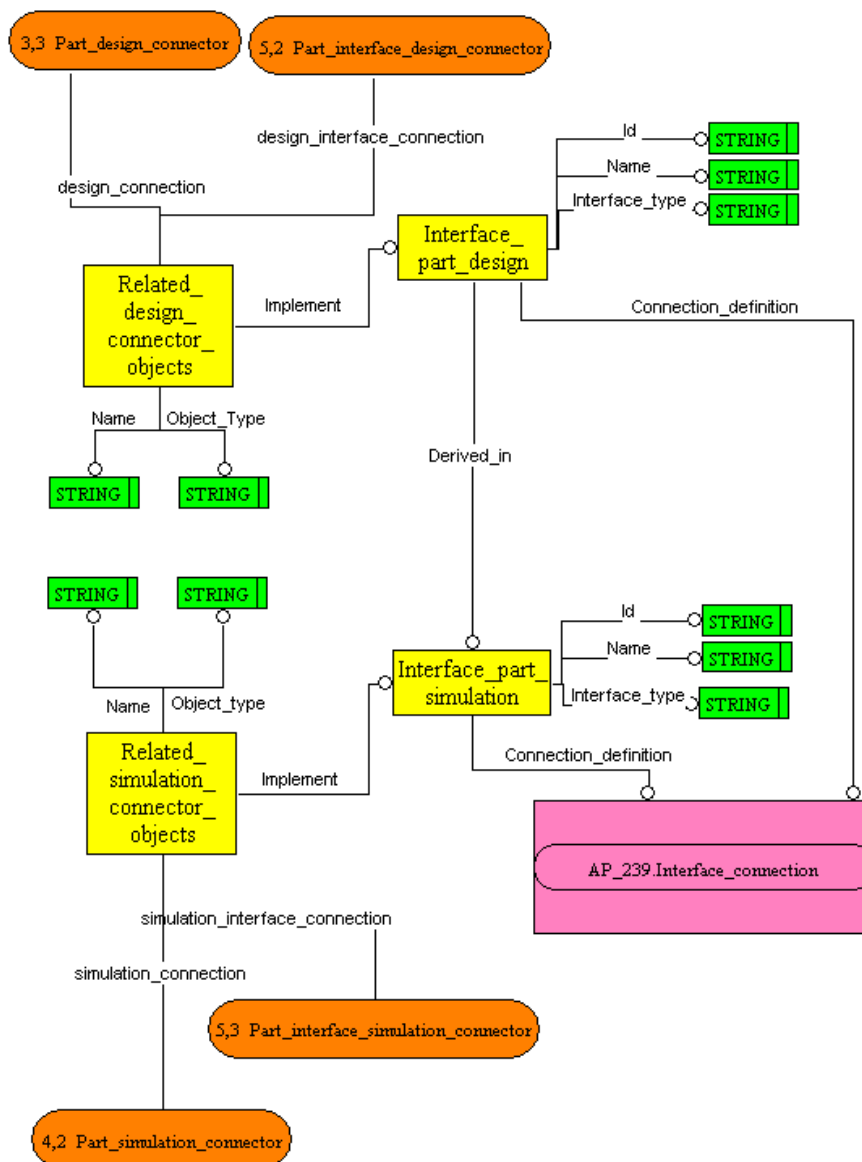


Figure 67 - Interface definition

C. Data specific to simulation

Figure 68 defines the implemented objects necessary for the simulation. All of them are grouped under the notion of **simulation elements definition**, specifying:

- ❖ The **analysis material identification** using the notions from the in AP209 by **material designation** and **material property**
- ❖ The **sim analysis parameters** defining specific parameters for the simulation. This keeps track of previous simulations
- ❖ The **boundary conditions** using the AP209 **state definition**
- ❖ The **load cases for simulation** corresponding to the AP209 **nodal freedom and value definition**

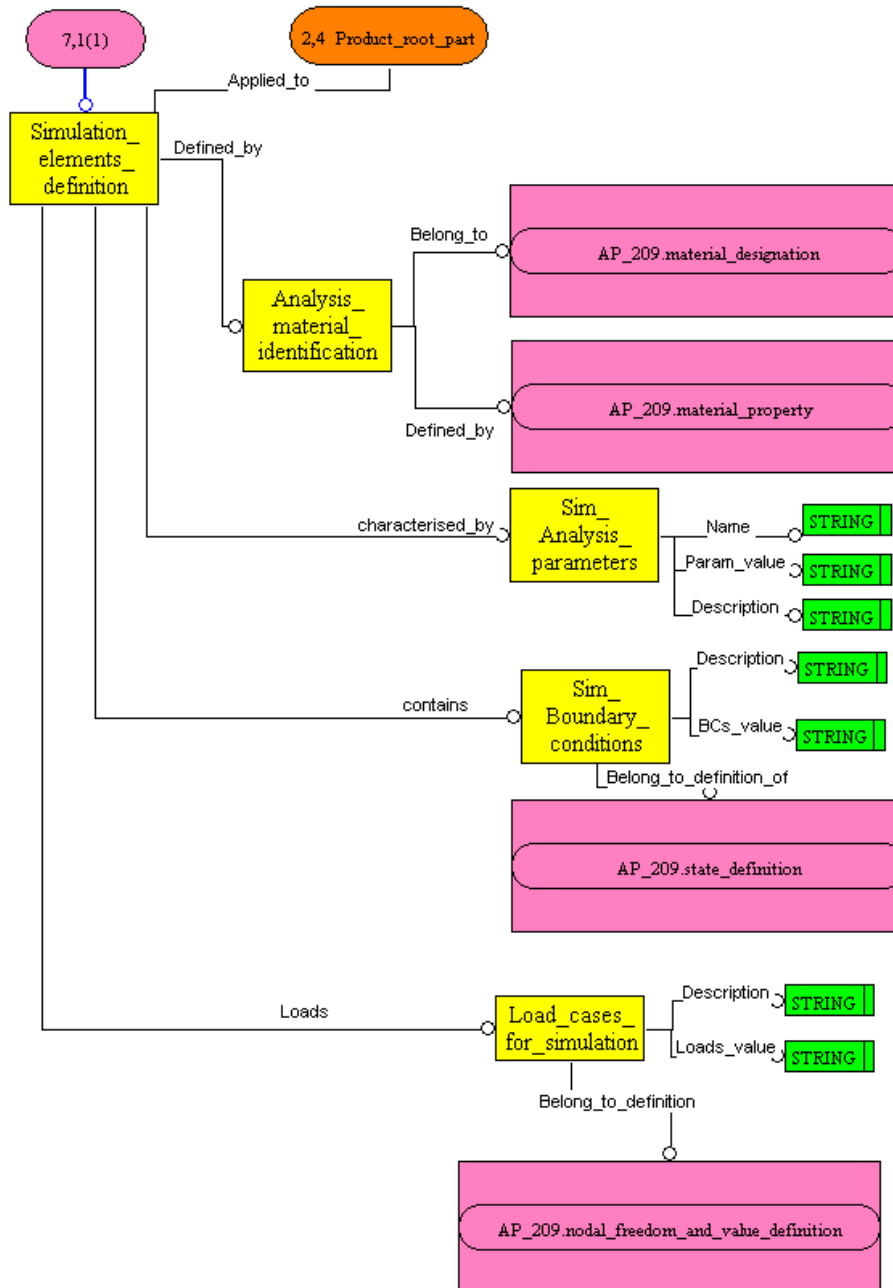


Figure 68 - Management of simulation parameters

D. Product views

This model details the structural aspect of the product. Defined under the **aircraft product information view**, it characterises the definition of the aircraft following two methods (see Figure 69).

- ❖ The first one defined by **ATA100 breakdown structure**. It defines the standardised structure of an aircraft following the decomposition of the **ATA chapter**, the **ATA section** and the **ATA description**. (ATA describe the breakdown of an aircraft using system breakdown).

- ❖ The second one is the **aircraft physical breakdown** using the definition of the AP239 **system breakdown**. The whole aircraft is decomposed into **aircraft modules** composed of **aircraft systems**, **aircraft sub systems** and **aircraft elements**.

These two decompositions are important since they are the sources which structure the product in different domains. Engineers from system engineering are more interested in the ATA100 decomposition that provides the system structure of the aircraft. They are able to find out all the components related to a function of the aircraft. The **aircraft physical breakdown** is more related to the structural design of the product without the final function of the product but with the logical organisation of the elements into account.

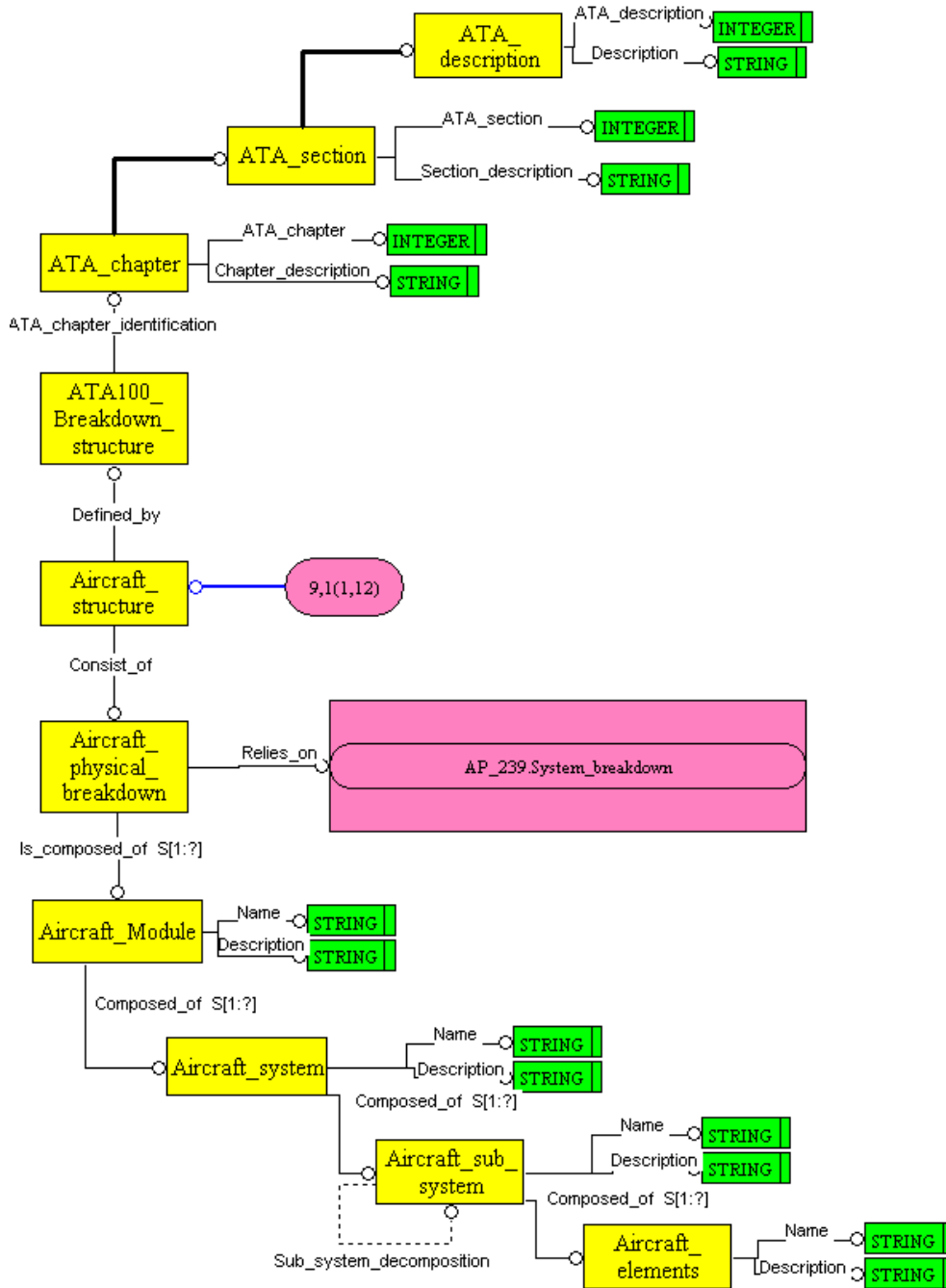


Figure 69 - Aircraft structure description

This model briefly details the notion of configuration (Figure 70). The configuration is deeply detailed in the AP214 with **configuration design**. Our model is related to this notion of AP214 and takes into account and defines only the relevant points for the collaborative configuration. The **aircraft configuration design** entity is the reference to retrieve the same configuration between partners and domains. It is composed of three entities:

- ❖ **Configuration item characterisation** defining different alternatives to design and product definition.

- ❖ **Configuration item solution** characterising the best collaborative configuration.
- ❖ **Referenced configuration** is the collaborative configuration provided for all partners and activities.

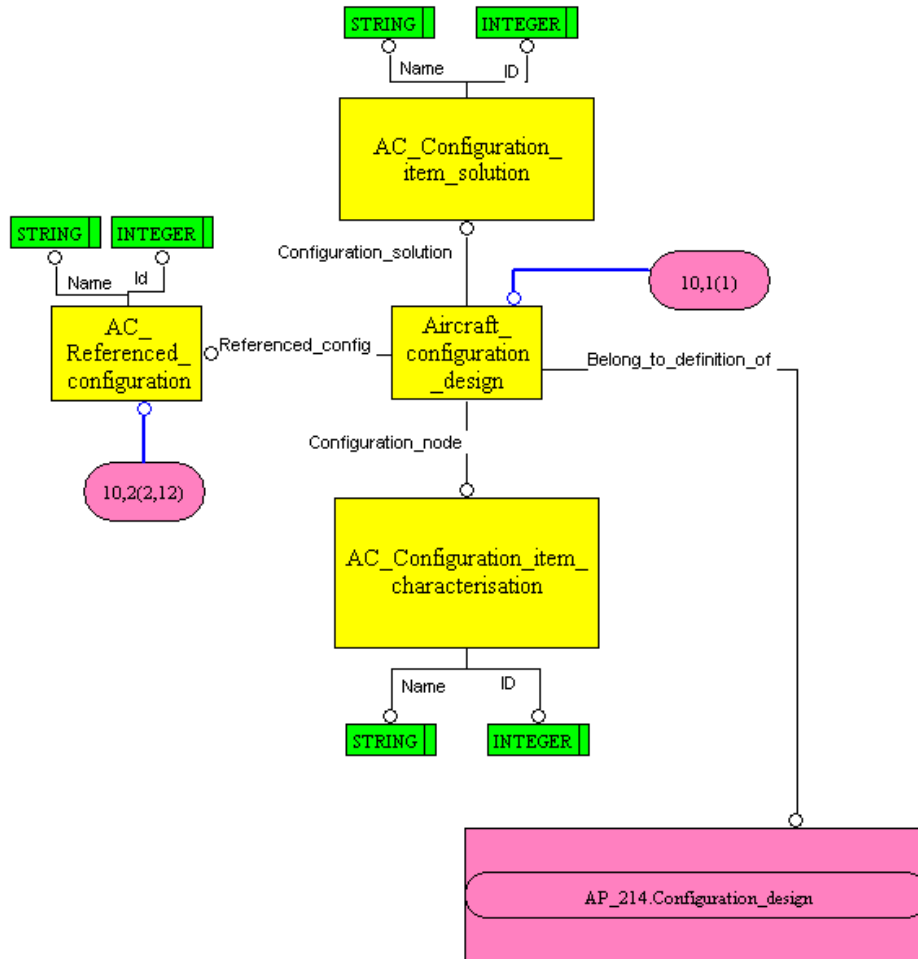


Figure 70 - Configuration Management

IV. Context implementation – Implementation of the information model

A. Link to process and project

This model details the notion of **aircraft process definition**. The central concept is the definition of the **AC process element** (Figure 71). Defined using the AP239 entity of **process property assignment**, it is characterised in 4 different ways:

- ❖ Integration of **AC project context** that defines the project in which the process is performed. The **AC project context** is characterised through the AP239 entity of project, and the notion of lifecycle.

- ❖ The decomposition of the process in **AC elementary activities** characterised by the notion of activity in the AP214 and using the notion of **activity method** from the AP239 that defines the purpose and characterises the objective of the activity
- ❖ The **AC workflow object** characterises the transition elements involved in the process sequence. They are derived from two entities: the **AC workflow events** characterising the incoming of information (as for the integration of models) and the **AC workflow decision** that characterises the push of process using conditions (for example validation of concepts launching design activities)
- ❖ The **AC contractual objects** characterising the objects inputs and outputs from a process. They are the necessary conditions to launch and end the process. They are defined by the **workflow transaction entity** that is an executive request. These requests are related to the **AC collaborative method** defining the action on the data and the adapted contractual objects to ensure the process definition.

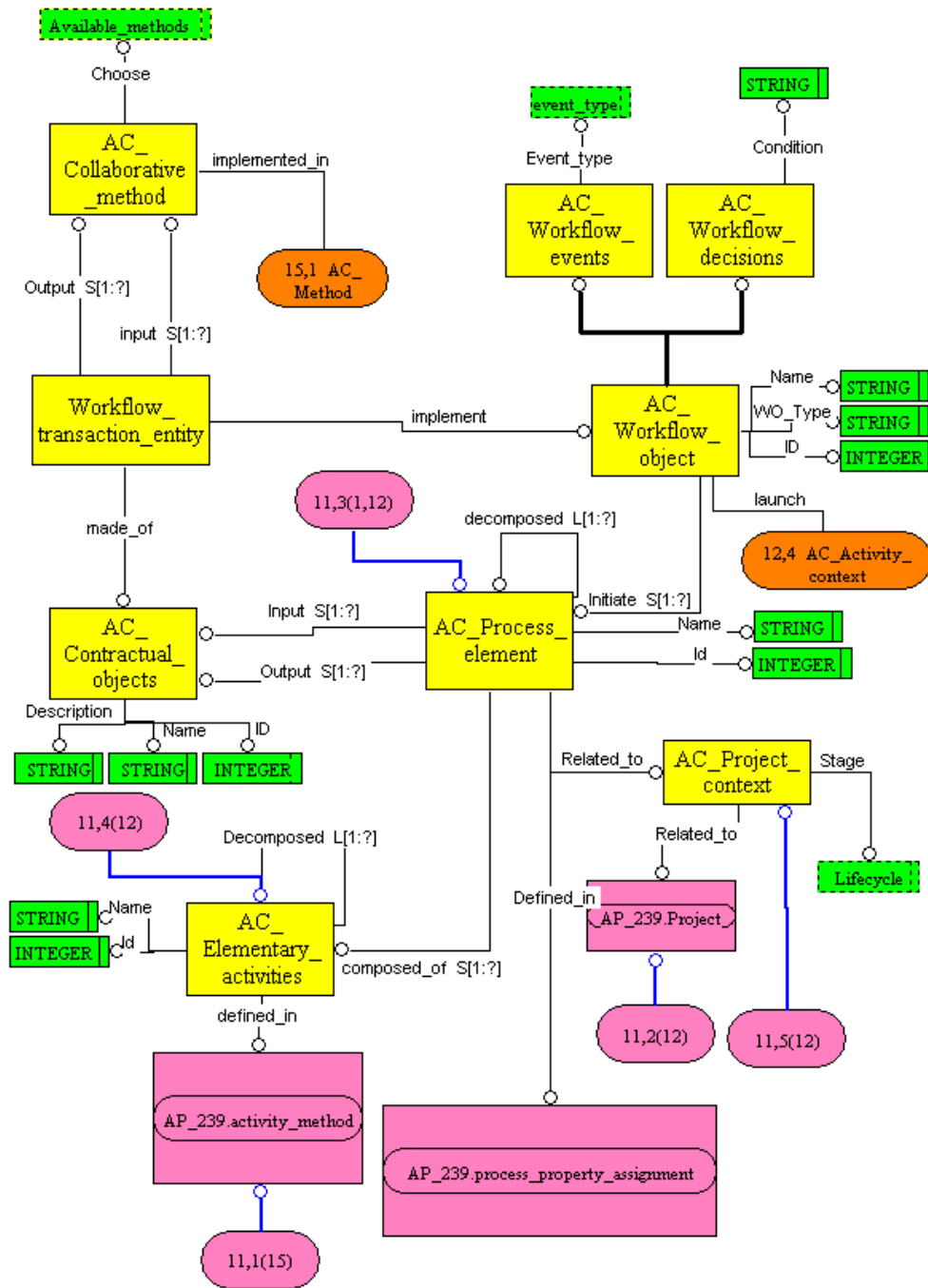


Figure 71 - Description of Process

B. The context definition in EXPRESS

This model details the **AC context definition** (Figure 72). This is the main entity (built on the notion of the AP214 **application context** and AC **project context**) that defines the context. It also defines the **AC context relationships**. This implements the notion of translating the contexts between design and simulation:

- ❖ **AC product context** defines two major objects: the first one is the **AC product root** (necessary reference to get all product data), and the second is the

data structure. The selection provides the capability to generate the context of the product regarding the **AC referenced configuration** or regarding a navigation using the **aircraft structure**. This leads also to the definition of the **AC data structure context** defining the data structure for the engineering domain required.

- ❖ The **AC activity context** defines the activity environment associated to the development of the product. The selection of the main environment is made according to the project, **process element**, or **elementary activities**.
- ❖ The **user context** is based on the description of the AP239 person **in organisation** for the characteristics. For the implication of the framework, the **framework role** takes into account the log-in of the user on the collaborative environment to set up the associated context.

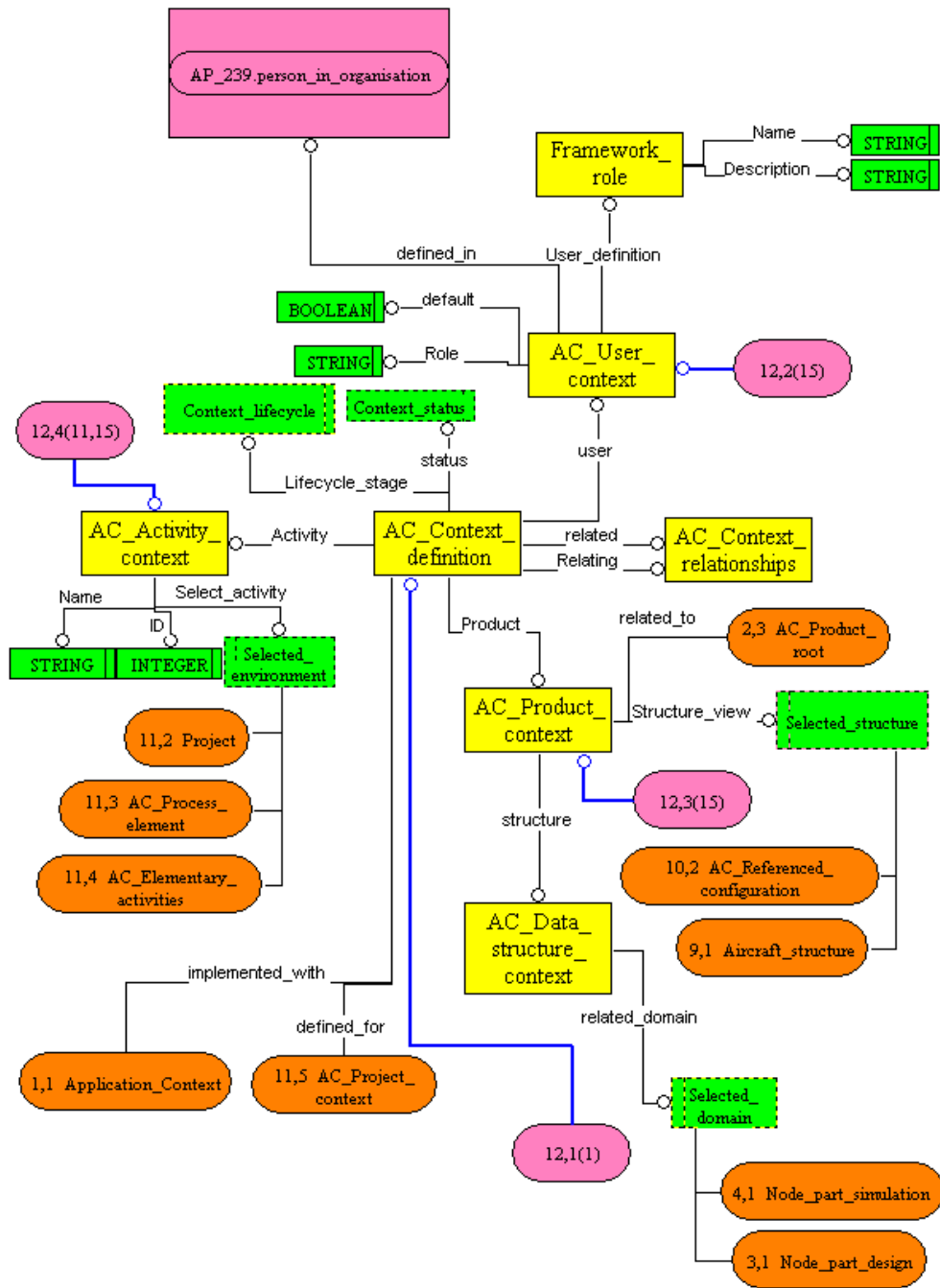


Figure 72 - Context definition

The context is also a dynamic environment. For this reason to allow a user to identify more precisely the current context, we have associated notions of lifecycle. This determines the current environment and the status of this context.

V. Resources implementation

A. Definition of tools structure

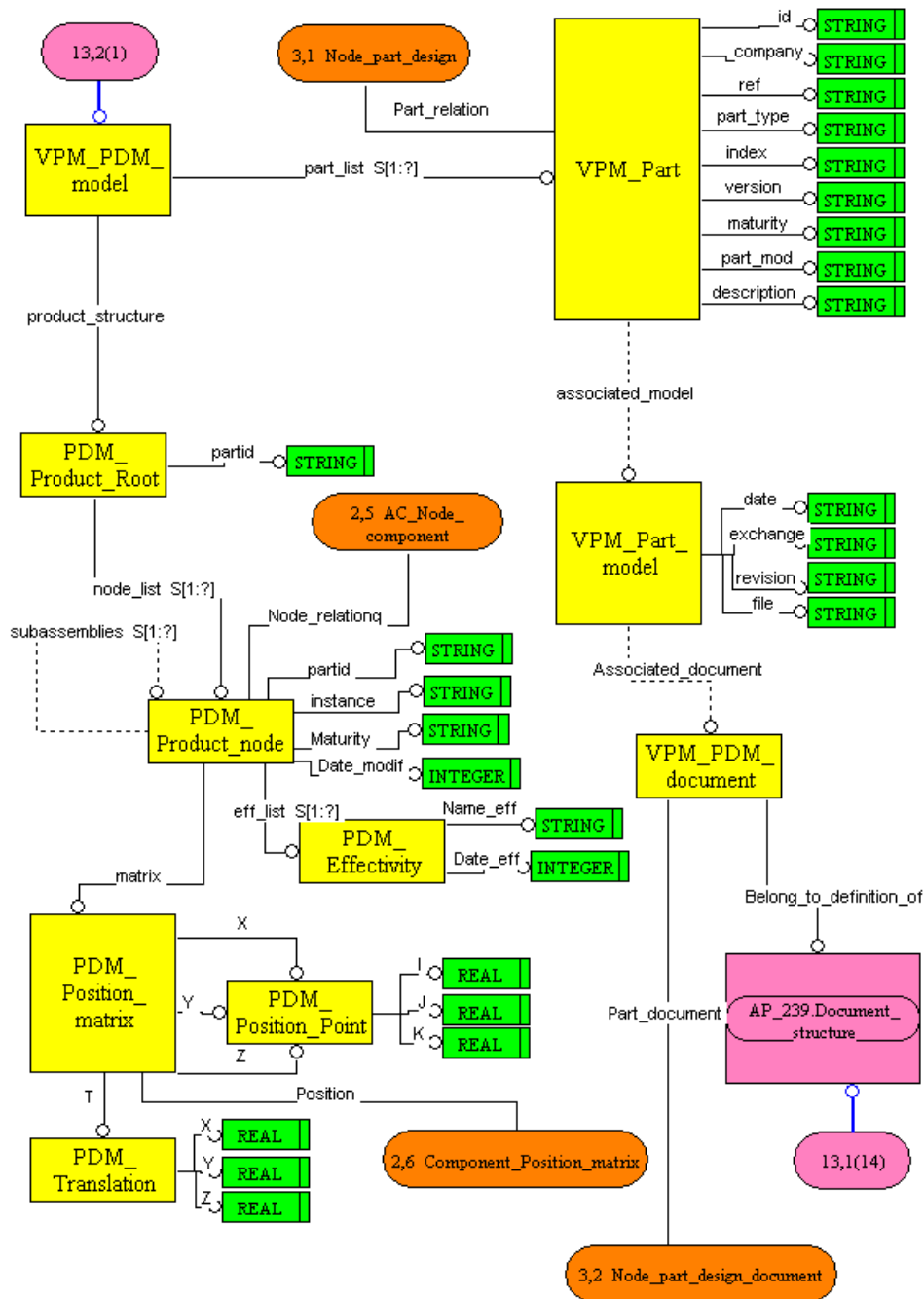


Figure 73 - VPM tool representation

Figure 73 details the structure of the product data management tool called VPM. It describes the different elements that compose the data management in this system. The structure is associates Part (**VPM Part**), models (**VPM Part model**) and documents (**VPM PDM document**). The management is made using nodes

the information exported from reference data management structure. As for the PDM, this structure represents tools; the links are made with the referenced data structure.

B. Services structure

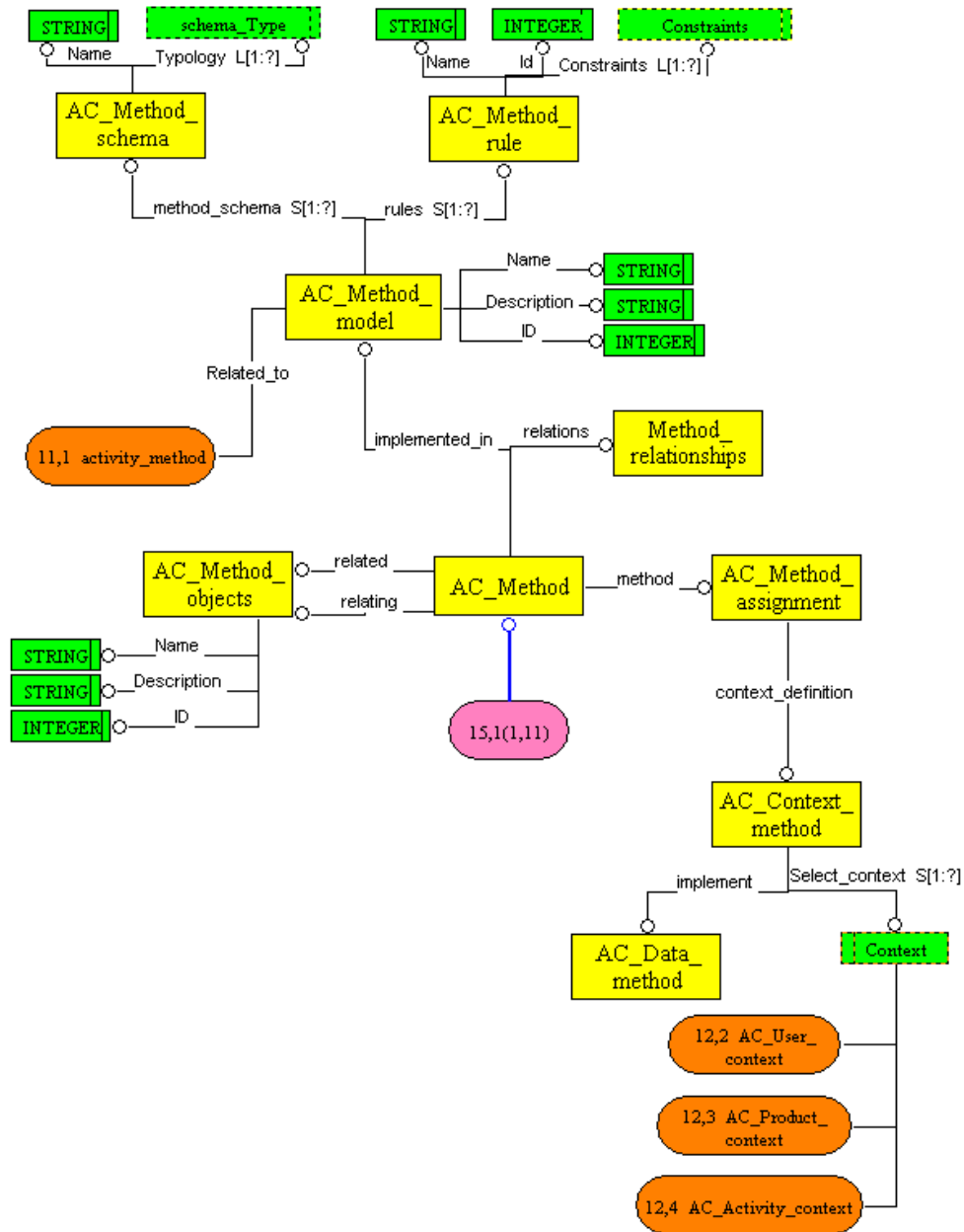


Figure 75 - Services definition

Figure 75 represents the notion of services offered by the platform definition. Here, **AC method** is the central concept. The **AC method** is composed of three major relations.

- ❖ **AC method objects** defines the objects related to the method application.
- ❖ The notion of **AC method assignment** defines in which context the method may be applied. It distinguishes the domain methods applied to a specific context

defined by the **AC user context**, the **AC product context** or the **AC activity context**

It defines also the relationship between methods related to different domains. This is defined by **AC method model**. Related to the notion of **activity method** from AP239, it defines **AC method schema** to define their action and **AC method rule** to define the limit of their action.

VI. Conclusion

In this chapter, we have defined the building elements of our overall architecture. Based mainly on the need to ensure simulation and design domains interoperability and transactions, and trans-enterprises exchanges, we have defined how the components must fit together. Particularly, we have defined three major innovative schemas:

- ❖ The first concerns the definition of the context. This schema is fundamental to provide the consistency between the product, the processes and the resources involved in the product development
- ❖ The second concerns the definition of data management principle. This schema is fundamental to provide a simple way to make data consistent.
- ❖ The third is the management of resources. This schema is fundamental to reach the interoperability in the context of tools heterogeneity

In the models building, we have particularly paid attention of the flexibility and implementation capability of models. The main objective is to be able to modify or to add new activities components and to generate their interoperability. The use of such method of fragmentation of data and semantics is a major step towards the unification platform for activities.

Chapter VIII: Prototype implementation and preliminary results on industrial scenario

I. Introduction

In this chapter, we describe a scenario based on a design and simulation optimisation loop to validate our proposed framework structure and functioning principles. This scenario is based on classical exchanges and involves the notion of partnership design and exchanges and the notion of multi-activities data migration.

We propose a review of industrial constraints involved in this kind of partnership and exchanges. We finally propose a testing process based on five main stages and using the user interface of the product context management which is the user interface of the collaborative architecture.

II. Definition and Characterisation of the Use Case

A. High level definition of the scenario

The use case targets two objectives:

- ❖ To provide a framework and methodologies to support the WEM (Whole Engine Model) activities during preliminary design and concept definition stages.
- ❖ To develop methodologies and tools to quickly derive the whole engine finite element models suitable for a preliminary design assessment from a heterogeneous assembly of CAD components (called preliminary design DMU).

Figure 76 describes the different roles necessary and implemented in our scenario. Figure 77 describes the stages of the scenario and the actions that are performed during these stages.

Short	Role name	Description and Tasks
CADP	CAD Engineer (Partner)	Specify and validate CAD environment (partner view), perform design and PDM implementation, validate its part of design
CAEP	CAE Engineer (Partner)	Specify and validate CAE environment (partner view), perform meshing and SDM implementation, validate its part of design
CADI	CAD Integrator (Integrator)	Specify and validate CAD environment (integrator view), perform assembly of models, validate integration, validate design, perform simulation (clash analysis...)
CAEI	CAE Integrator (Integrator)	Specify and validate CAE environment (integrator view), migrate CAD environment, perform

		assembly of meshed models, validate integration, validate meshing, perform simulation
--	--	---

Figure 76 - Roles and actors for the scenario

Roles involved	Macro process	Detailed Activity		Component involved	
CADI	Initiate environment	Request specifications Publish CAD environment Publish PLM environment		Requirement environment Collaborative environment CAD/PLM environment	
CADE	Perform design	Validate request Retrieve 3D context Perform 3D design Publish 3D design	Validate request Retrieve PLM context Perform PLM activity Publish PLM data	Collaborative environment CAD environment Information Model (request history + 3D design publication)	Collaborative environment PLM environment Information Model (request history + PLM data)
CADI	Validate design	3D design integration (CAD assembly, DMU perform) Clash detection – Interface and design specifications respect		CAD/PLM environment Information Model (request history + 3D design publication + PLM data)	
CAEI CAEP	Perform meshing	Validate request Retrieve FEM context Perform meshing Publish FEM	Validate request Retrieve SDM context Perform SDM activity Publish SDM data	Collaborative environment CAD/CAE environment Information Model (request history + 3D design/FEM publication)	Collaborative environment PLM/SDM environment Information Model (request history + PLM/SDM data)
CAEI	Perform overall analysis	Assemble meshing Perform calculation Analysis		SDM environment Information Model (request history + result publication)	
CAEI CADI	Request for change	Manage request Send request		Collaborative environment (request edition) Information Model (request history)	

Figure 77 - General definition of the scenario

B. General requirements

Requirements related to high level process view: These requirements are related to the overall interfacing at high level. They concern major requirements on standard format implementation, interoperability and associativity:

- ❖ **Standard format definition:** the main issue related is to allow the communication between heterogeneous tools from CAD environment but also from CAE environment. The neutral/standard format is also required to permit the communication from these two environments. The major way to manage it, is to extend our vision to STEP standard compliance within all the tools defined in the architecture. This issue is mainly related to the interoperability that can be provided through the AP203, AP209, AP 214, Part 21 and 28.
- ❖ **Interoperability:** the need of interoperability relies on the issue of standardised connectors that can be developed for tools communication. Interoperability means that each tool working separately has at least one major way to communicate with the other using the architecture. The major way to manage it is to define function principles to develop the connectors that can be implemented.
- ❖ **Associativity:** the need of Associativity relies on the semantic link between the different kinds of data created. It relies for example on the associativity between CAD definition and FEM definition, and between PLM and SDM data. It means that if one of the data has changed, the linked data are going to be impacted and will have to be updated also.

Requirements related to a specific operation in the use case:

- ❖ Requirements on data derived from PDM/PLM: As the use case starts from design environment and as in several projects this environment remains as the reference, it is addressed that each relevant data from PLM must to be addressed in the SDM. It means for example to be able to position the different elements in the same way both for 3D design models and FEM. For example, some geometrical elements could be in a local coordinate system (the same as the one used for the CAD model) and positioned in the absolute coordinate system using a position matrix derived from the CAD assembly. The derived positioning in SDM environment defines the FEM location in the simulation environment.
- ❖ Configuration Management: The configuration management is addressed to ensure the migration from design environment to simulation environment. It is addressed that a mechanism must exist to verify that the module FEM is compliant with the assembly module. It means that there is a coherency between 3D assembled models and FEM assembly.
- ❖ Interface management: The interface management is addressed to ensure the maximum of pertinence of the model. It must enable for example to link the parts

together; this could be done quite automatically using information of mechanical behaviour on the interface definition. It is also addressed that the CAD interfaces defined in the design environment can be derived in FEM interfaces.

III. Proposal of the testing Process

The Mechanical scenario is a sequential high-level process steering several operational sub-processes:

- ❖ **PCM1, PCM2 & PCM3:** Analysis and management tasks that realise the transitions between the operational sub-processes.
- ❖ **PDM:** Design sub-process, that performs a design modification
- ❖ **SDM:** Simulation sub-process that performs a mechanical analysis on collaborative data

A. PCM1

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 (PCM3 closes the loop). The sub-process PCM1, in the scope of collaborative framework, demonstrates:

- ❖ The contextualisation tool: capability to define an overall context of process and capability to control the interoperability and associativity of information.
- ❖ The interoperability with COTS or legacy: Web services call and integration with tools (Enovia, SimManager...)

The different activities realised in this step are: Access User workspace, select an activity, visualise related design and simulation information together, close analysis task, prepare and send a change request.

The change request is send to the next activity: the PDM sub-process.

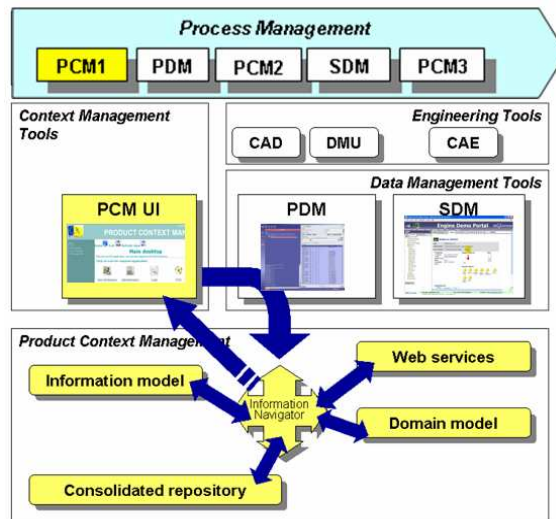


Figure 78 – First Step: Product Context Management

B. PDM

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:

- ❖ Design and changes in context: Use of consolidated data for the collaboration and management of access authorisations
- ❖ 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
- ❖ Interface management for collaborative design task.

The different activities realised in this step are: Consult a change request, perform a design modification in the CAD environment, and close the activity.

The Change Request at the end of sub-process PDM is send back to the next activity PCM2.

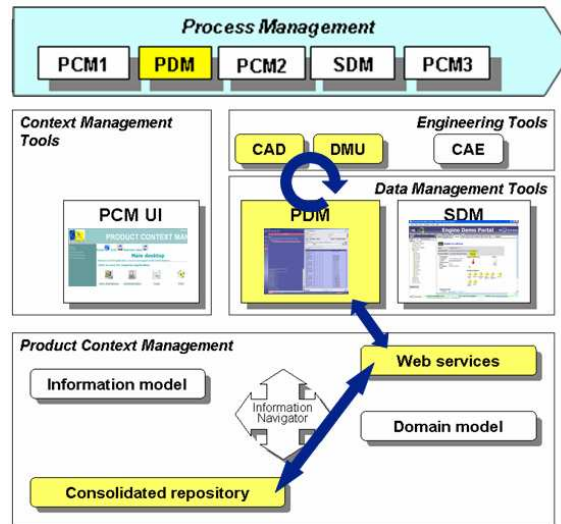


Figure 79 – Second Step: Product Data Management

C. PCM2

The PCM2 (Product Context Management 2) shows first a design analysis. Then after the validation from a multi-partner design point of view (respect of interface between component), a simulation request is launched to ask a validation from analysis point of view. This part demonstrates in the context of collaborative framework:

- ❖ How Product Context Management loads pertinent data: Use of Domain model that gets the relevant information.
- ❖ How Product Context Management facilitates the Navigation: Capability to enter context with product structure or process definition, capabilities of Web services queries.

The different activities realised in this step are: The validation of the new design, the simulation launch request.

The simulation request is send to the next activity: SDM.

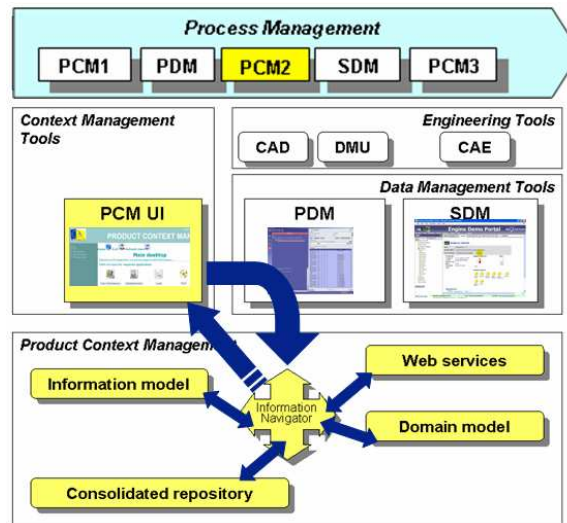


Figure 80 – Third Step: Product Context Management

D. SDM

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:

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

The different activities realised in this step are: Consult a workflow notification (the simulation request), perform a complete analysis and close the simulation request (with the publication of a simulation report).

The Simulation Request at the end of sub-process SDM is send back to the next activity PCM3.

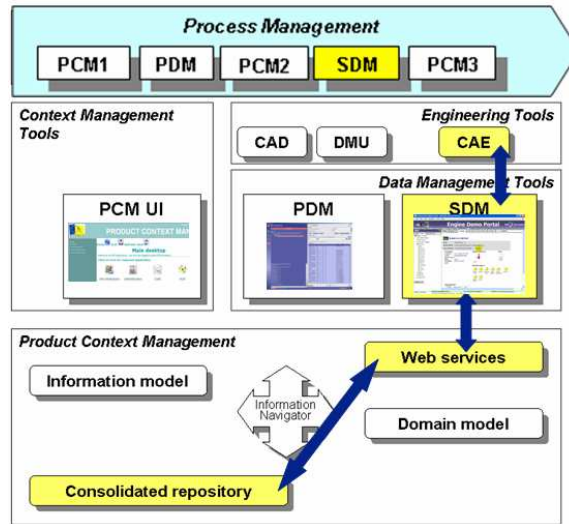


Figure 81 – Fourth Step: Simulation Data Management

E. PCM3

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:

- ❖ How the Product Context keeps a link between design and simulation information.
- ❖ 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.

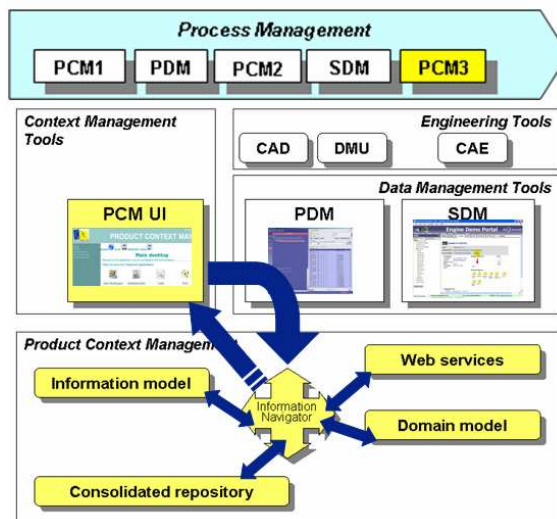


Figure 82 – Fifth Step: Product Context Management

IV. The prototype “show”

A. Description of what is invisible for “public”

Here, we illustrate the transformation of the PDM file extracted from VPM into the SDM file for SimManager.

The original file extracted from VPM is in XML format. This file describes the product structure and the different attributes (related to the different part of the product structure) contained in the PDM systems. This file is presented in Figure 83.

```

13 <PARTS_LIST>
14
15
16 <PART ID="SNE_VIV000000_T_0_000" COMPANY="SNE" REFERENCE="VIV000000" DATEMOD="20060202.162628.591885918859188" TYPE="T" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="PRODUCT-ROOT" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
17 <PART ID="AMC_VIV540000_T_0_000" COMPANY="AMC" REFERENCE="VIV540000" DATEMOD="20060207.112344.410244102441024" TYPE="T" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="AIRCRAF" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
18 <PART ID="AMC_VIV545000_P_0_000" COMPANY="AMC" REFERENCE="VIV545000" DATEMOD="20060207.110839.401194011940119" TYPE="P" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="PYLON" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
19 <MODEL NOM_ECHANGE="VIV545000" REVISION="001" DATEMOD="20060208.163802.598825988259882" MATURITY="CONVERS" FILE=
"AMC_VIV545000-----P_0_001.CATPart" />
20 </PART>
21 <PART ID="SNE_VIV710000_T_0_000" COMPANY="SNE" REFERENCE="VIV710000" DATEMOD="20060202.165008.606086060860608" TYPE="T" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="POWERPLANT" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
22 <PART ID="SNE_VIVXX0000_T_0_000" COMPANY="SNE" REFERENCE="VIVXX0000" DATEMOD="20060207.112814.412944129441294" TYPE="T" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="SYSTEMS" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
23 <PART ID="RRD_VIV780000_P_0_000" COMPANY="RRD" REFERENCE="VIV780000" DATEMOD="20060207.110745.400654006540065" TYPE="P" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="EXHAUST" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
24 <MODEL NOM_ECHANGE="VIV780000-----P0000" REVISION="000" DATEMOD="20060208.163947.599875998759987" MATURITY="GELE" FILE=
"RRD_VIV780000-----P_0_000.CATPart" />
25 </PART>
26 <PART ID="SNE_VIV790000_P_0_000" COMPANY="SNE" REFERENCE="VIV790000" DATEMOD="20060207.110917.401574015740157" TYPE="P" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="LUB CELLS" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
27 <MODEL NOM_ECHANGE="VIV790000-----P0000" REVISION="000" DATEMOD="20060208.174323.638036380363803" MATURITY="GELE" FILE=
"SNE_VIV790000-----P_0_000.CATPart" />
28 </PART>
29 <PART ID="SNE_VIV720000_T_0_000" COMPANY="SNE" REFERENCE="VIV720000" DATEMOD="20060202.165614.609746097460974" TYPE="T" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="ENGINE ASSY" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
30 <PART ID="SNE_VIV720001_T_0_000" COMPANY="SNE" REFERENCE="VIV720001" DATEMOD="20060207.113435.416754167541675" TYPE="T" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="FAN ASSY" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
31 <PART ID="SNE_VIV723500_P_0_000" COMPANY="SNE" REFERENCE="VIV723500" DATEMOD="20060207.111125.402854028540285" TYPE="P" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="SUPT BRG" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
32 <MODEL NOM_ECHANGE="VIV723500-----P0000" REVISION="000" DATEMOD="20060208.174303.637836378363783" MATURITY="GELE" FILE=
"SNE_VIV723500-----P_0_000.CATPart" />
33 </PART>
34 <PART ID="RRD_VIV722400_P_0_000" COMPANY="RRD" REFERENCE="VIV722400" DATEMOD="20060207.110957.401974019740197" TYPE="P" INDEX=
"0" VERSION="000" MATURITY="S" DESCRIPTION="FAN FRAME" UIN="" ECH="ALL" DATEEPI="" DATEMTAD="" />
35 <MODEL NOM_ECHANGE="VIV722400-----P0000" REVISION="000" DATEMOD="20060208.163902.599425994259942" MATURITY="GELE" FILE=
"RRD_VIV722400-----P_0_000.CATPart" />
36 </PART>

```

Figure 83 - XML file defining the content of the PDM called VPM

The original file is processed using the principles and schema described in the “collaborative schema”. The result is presented in Figure 84.

```

107 <xsl:template match="PRODUCT_STRUCTURE">
108 <xsl:comment>
109 =====
110 The followings are the assembly management
111 =====
112 </xsl:comment>
113 <xsl:apply-templates select="ROOT"/>
114 </xsl:template>
115
116 <xsl:template match="ROOT">
117 <xsl:apply-templates select="NODE"/>
118 </xsl:template>
119
120 <xsl:template match="NODE">
121 <xsl:variable name="ref_node" select="//PARTS_LIST/PART[@ID=current()/@PARTID]"/>
122 <xsl:choose>
123 <xsl:when test="$ref_node/@TYPE = 'T' ">
124 <xsl:comment>
125 =====
126 FEAssembly: <xsl:value-of select="@PARTID"/>
127 =====
128 </xsl:comment>
129 <xsl:text>
130 </xsl:text>
131 <xsl:element name="SimManagerAssembly">
132 <xsl:attribute name="itemType">Item</xsl:attribute>
133 <xsl:attribute name="itemId">
134 <xsl:value-of select="$ref_node/@DESCRIPTION" />
135 </xsl:attribute>
136 <xsl:attribute name="itemRev">
137 <xsl:value-of select="$ref_node/@VERSION" />
138 </xsl:attribute>
139 <xsl:attribute name="modelType">CADAssembly</xsl:attribute>
140 <xsl:attribute name="modelName">
141 <xsl:value-of select="$ref_node/@DESCRIPTION"/><xsl:text>-</xsl:text><xsl:value-of select="$ref_node/@ID"/>
142 </xsl:attribute>
143 <xsl:for-each select="NODE">
144 <xsl:variable name="ref_node" select="//PARTS_LIST/PART[@ID=current()/@PARTID]"/>
145 <xsl:choose>
146 <!--*****-->
147 <!-- Test wether it is a subassembly or a CADpart -->
148

```

Figure 84 - Transformation file with instances of VPM using the transformation of the collaborative schema

Finally the post-processing of the file is performed (always using the description and models of the “collaborative schema”) into an “Abom” file readable by SimManager.

```

13 <SimManagerItem itemType="Part" itemId="AIRCRAF" itemRev="000">
14 <SimManagerProperty key="description" value="AIRCRAF" />
15 <SimManagerProperty key="label" value="AMC_VIV540000_T_0_000" />
16 </SimManagerItem>
17 <SimManagerItem itemType="Part" itemId="PYLON" itemRev="000">
18 <SimManagerProperty key="description" value="PYLON" />
19 <SimManagerProperty key="label" value="AMC_VIV545000_P_0_000" />
20 </SimManagerItem>
21 <SimManagerItem itemType="Part" itemId="POWERPLANT" itemRev="000">
22 <SimManagerProperty key="description" value="POWERPLANT" />
23 <SimManagerProperty key="label" value="SNE_VIV710000_T_0_000" />
24 </SimManagerItem>
25 <SimManagerItem itemType="Part" itemId="SYSTEMS" itemRev="000">
26 <SimManagerProperty key="description" value="SYSTEMS" />
27 <SimManagerProperty key="label" value="SNE_VIVXX0000_T_0_000" />
28 </SimManagerItem>
29 <SimManagerItem itemType="Part" itemId="EXHAUST" itemRev="000">
30 <SimManagerProperty key="description" value="EXHAUST" />
31 <SimManagerProperty key="label" value="RRD_VIV780000_P_0_000" />
32 </SimManagerItem>
33 <SimManagerItem itemType="Part" itemId="LUB CELLS" itemRev="000">
34 <SimManagerProperty key="description" value="LUB CELLS" />
35 <SimManagerProperty key="label" value="SNE_VIV790000_P_0_000" />
36 </SimManagerItem>
37 <SimManagerItem itemType="Part" itemId="ENGINE ASSY" itemRev="000">
38 <SimManagerProperty key="description" value="ENGINE ASSY" />
39 <SimManagerProperty key="label" value="SNE_VIV720000_T_0_000" />
40 </SimManagerItem>
41 <SimManagerItem itemType="Part" itemId="FAN ASSY" itemRev="000">
42 <SimManagerProperty key="description" value="FAN ASSY" />
43 <SimManagerProperty key="label" value="SNE_VIV720001_T_0_000" />
44 </SimManagerItem>
45 <SimManagerItem itemType="Part" itemId="SUPT BRG" itemRev="000">
46 <SimManagerProperty key="description" value="SUPT BRG" />
47 <SimManagerProperty key="label" value="SNE_VIV723500_P_0_000" />
48 </SimManagerItem>
49 <SimManagerItem itemType="Part" itemId="FAN FRAME" itemRev="000">
50 <SimManagerProperty key="description" value="FAN FRAME" />
51 <SimManagerProperty key="label" value="RRD_VIV722400_P_0_000" />
52 </SimManagerItem>

```

Figure 85 - "Abom" file generated for use in SimManager

B. The user interface and the scenario

1. PCM1

We start opening the PCM UI with login aspects and go to main desktop (see Figure 89). We then follow the navigator main page



Figure 86 - Main desktop page of the Product context management interface

We then go to “user workspace” (see Figure 87) using shortcut on the left bar.

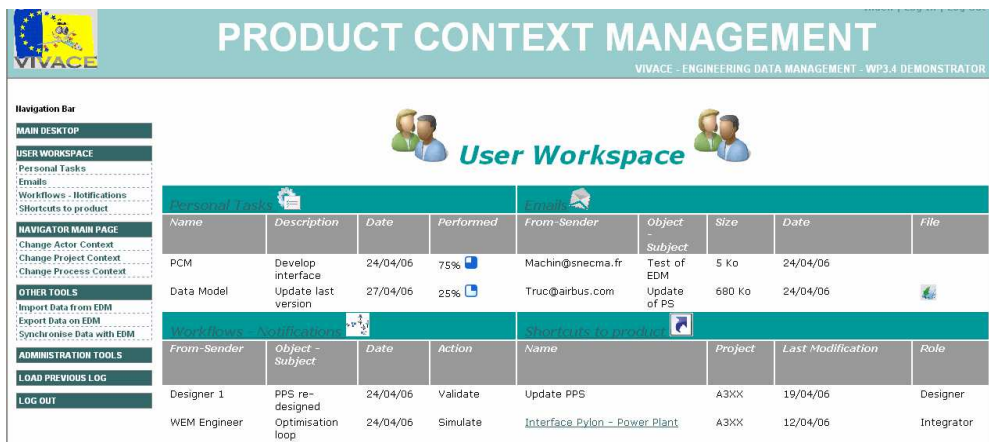


Figure 87 – User workspace page of the Product context management interface

Then we click on “interface pylon –Powerplant” identified in “shortcut to product” bar. We arrive on the page showing design and simulation views (see Figure 88 upside of panel and Figure 89 downside of panel).



Figure 88 – Interface pylon-powerplant page of the Product context management interface

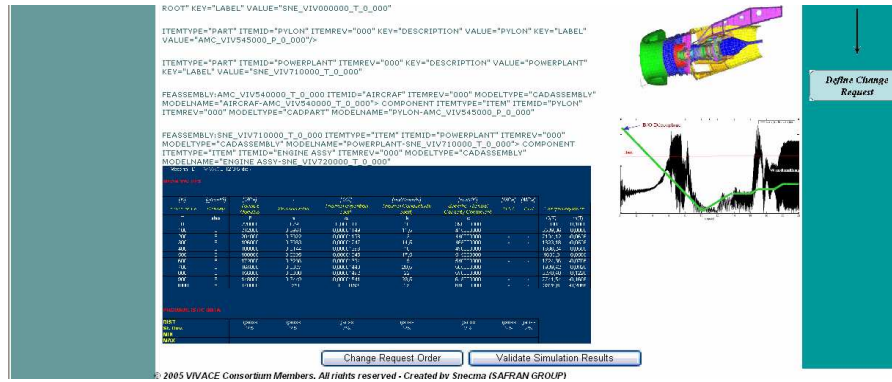


Figure 89 – Interface pylon-powerplant page of the Product context management interface

Here we identify that there is a problem with previous simulation. Then we will click on “change request order” to launch modification loop for design (see Figure 90).



Figure 90 – Confirmation pop-up of the Product context management interface

Here we decide to change pylon, and this open the page with pylon design and a field for the design change request (see Figure 91).

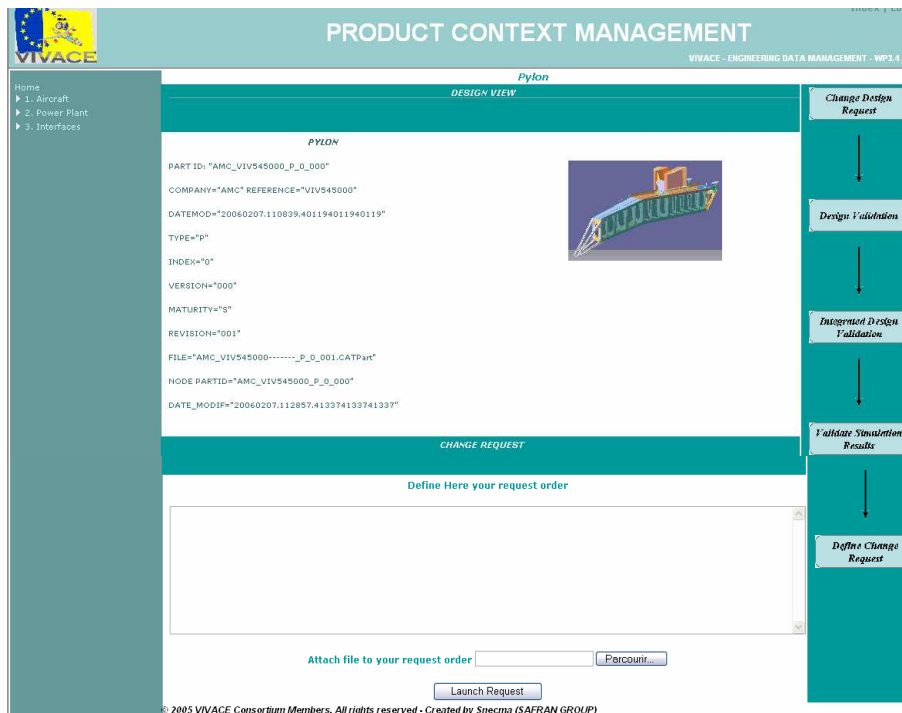


Figure 91 – Design change request page of the Product context management interface

In this panel we define the change request, add files and finally launch request. Once launched, a notification of confirmation appears and then we can go back to “user workspace”.

2. PCM2

This scenario takes place first in the user workspace. Here we identify that the partner Airbus has modified Pylon design and has sent a notification (see Figure 92).

PRODUCT CONTEXT MANAGEMENT
VIVACE - ENGINEERING DATA MANAGEMENT - WP3.4 DEMONSTRATOR

Navigation Bar

- MAIN DESKTOP
- USER WORKSPACE
 - Personal Tasks
 - Emails
 - Workflows - Notifications
 - Shortcuts to product
- NAVIGATOR MAIN PAGE
 - Change Actor Context
 - Change Project Context
 - Change Process Context
- OTHER TOOLS
 - Import Data from EDM
 - Export Data on EDM
 - Synchronise Data with EDM
- ADMINISTRATION TOOLS
- LOAD PREVIOUS LOG
- LOG OUT

User Workspace

Name	Description	Date	Performed	From-Sender	Object-Subject	Size	Date	File
PCM	Develop interface	24/04/06	75%	Machin@sneema.fr	Test of EDM	5 Ko	24/04/06	
Data Model	Update last version	27/04/06	25%	Truc@airbus.com	Update of PS	680 Ko	24/04/06	

From-Sender	Object-Subject	Date	Action	Name	Project	Last Modification	Role
Partner: Airbus	Pylon with new specifications	24/04/06	Validate	Update PPS	A3XX	19/04/06	Designer
WEM Engineer	Optimisation loop	24/04/06	Simulate	Interface Pylon - Power Plant	A3XX	12/04/06	Integrator

Figure 92 - Main desktop page of the Product context management interface

We then click on the link to open page with modified pylon (see Figure 93).

PRODUCT CONTEXT MANAGEMENT
VIVACE - ENGINEERING DATA MANAGEMENT - WP3.4 DEMONSTRATOR

Pylon
DESIGN VIEW

PYLON

PART ID: "AMC_VIV545000_P_0_000"
 COMPANY="AMC" REFERENCE="VIV545000"
 DATEMOD="20060207.110839.401194011940119"
 TYPE="p"
 INDEX="0"
 VERSION="000"
 MATURITY="S"
 REVISION="001"
 FILE="AMC_VIV545000-....._P_0_001.CATPart"
 NODE PARTID="AMC_VIV545000_P_0_000"
 DATE_MODIF="20060207.112857.413374133741337"

CHANGE REQUEST

Reload here informations about design change request

© 2005 VIVACE Consortium Members. All rights reserved - Created by Sneema (SAFRAN GROUP)

Change Design Request

Design Validation

Integrated Design Validation

Validate Simulation Results

Define Change Request

Figure 93 – New pylon page of the Product context management interface

In this page, the new information about the design of the pylon is presented. The change request also appears as it was sent by the integrator during PCM1. Then we choose to validate (see Figure 94).

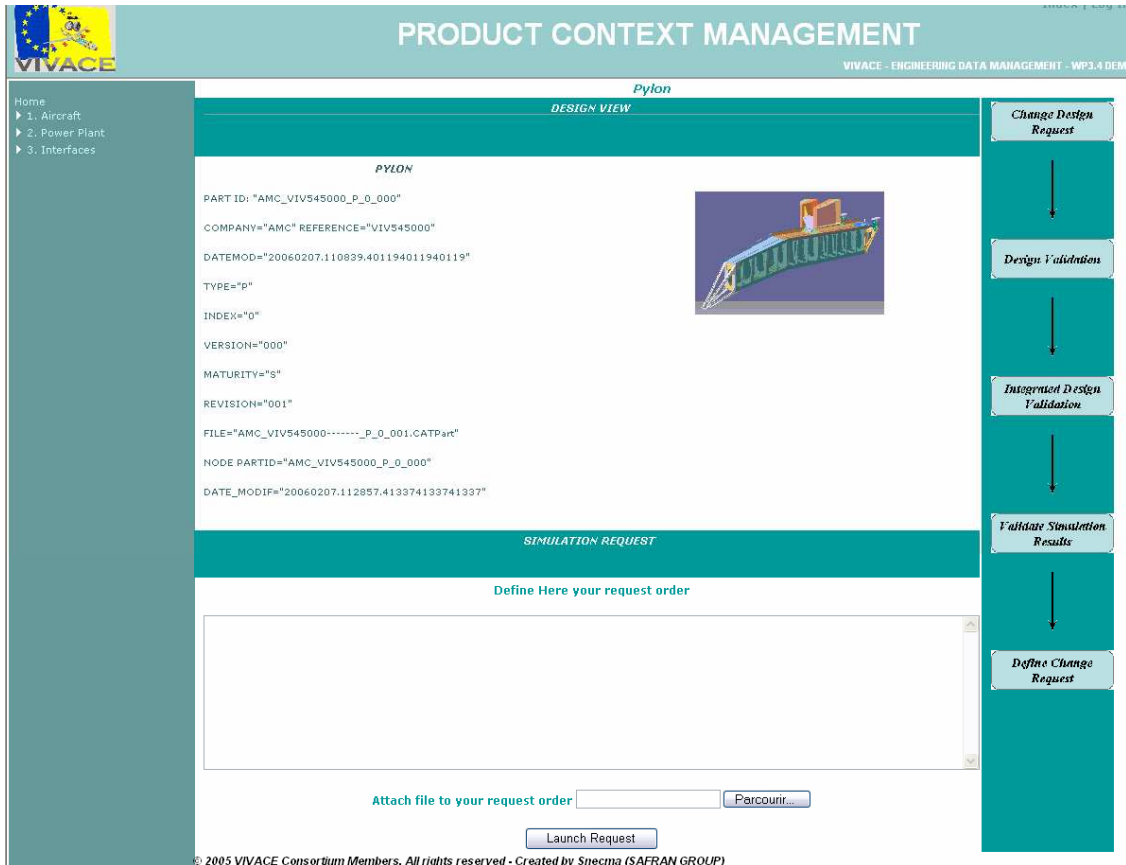


Figure 94 – Simulation request page of the Product context management interface

We then arrive on simulation request page. Here we load pylon data and open the simulation request. We then define this simulation and launch it after. A notification then appears that the message for simulation will be launched. We validate it and then we are redirected to the user workspace again.

3. PCM3

We start from “user workspace” and see that a notification from the WEM integrator appears for validation (see Figure 95).

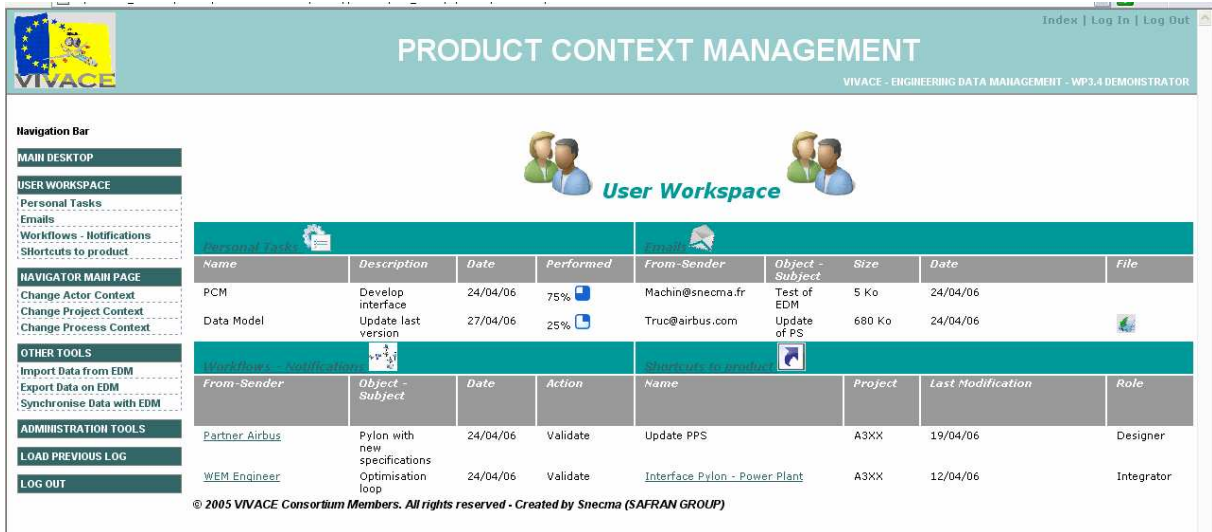


Figure 95 - Main desktop page of the Product context management interface

We click on the link and a similar page than the one of PCM 1 appears and shows both design and simulation (see Figure 96).

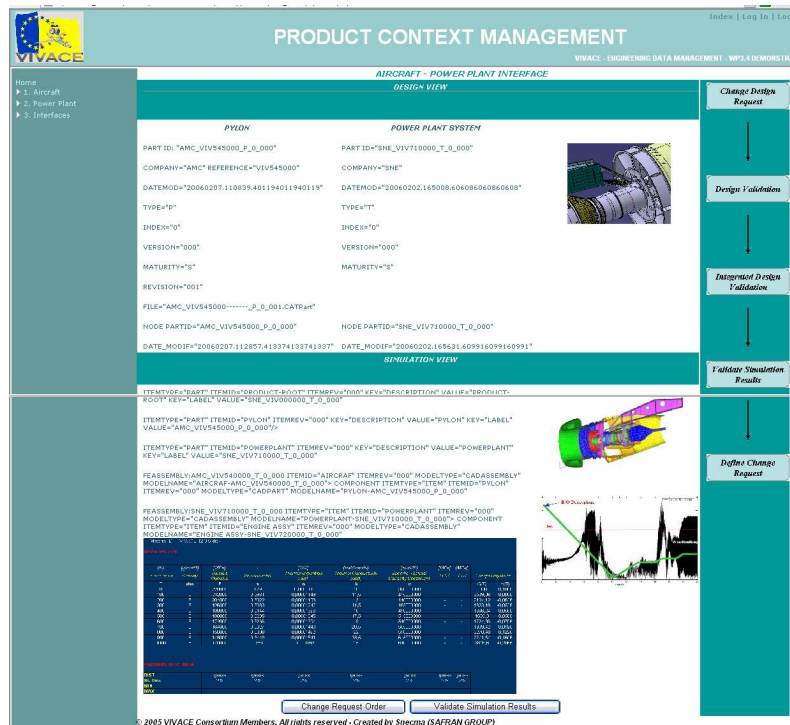


Figure 96 – New design and simulation page of the Product context management interface

V. Results and validation

A. General remarks

This scenario and the development of the prototype are tasks of the VIVACE Project. Indeed, this architecture has not already been set up in an operational

environment, gains and benefits are then more difficult to evaluate. Due to the difficulty to set up such an environment, the prototype is not entirely set up. For the overall development, the current status is the following:

- ❖ The development of the **interoperability for design and simulation** is now fully performed. The connections and data migration between the PDM and SDM are ensured. Moreover, the associativity managed by the links of the “collaborative schema” already provides associativity of data between design and simulation.
- ❖ The development of the **interoperability for product and process** is still in development. For the moment, this interoperability has been set up between design and simulation but remains not applicable to other cases.
- ❖ The development of the **interoperability for partners** is also in development. For instance, workspaces have been created and can be automatically set up using the definition of person in organisation.
- ❖ The development of **services** is nearly to be ready. The development of the services already provide functions that permit the design and simulation connection. Indeed, the most difficult task is to access the resources of legacy tools to implement such functionalities.

B. Potential benefits of the architecture

The architecture is set up and developed using three major requirements:

- ❖ **Not to influence the internal engineering systems of partners.** With such development the potential benefit of such architecture rely on the fact that only few developments are necessary. Any partner can connect the different capabilities of the architecture. Moreover, as they have been standardised using STEP, there is only one way to access these capabilities. Moreover, the flexibility of the “collaborative schema” provides the capability to add new items and engineering domains.
- ❖ **The architecture is standard based.** This is an important aspect to further develop the capabilities and connect new domains. The semantic and the description is already made, the only “new task” to do is to add the schemas of an activity and connect them to the standardised schema of the product.
- ❖ The **definition of the contexts** is also important. The architecture provides an efficient filter for environments based on the development of the connections between product, process and resources. As a consequence, the benefit is to

allow users and partners to define their working environment independently each other. This ensures to keep and respect the enterprise practices and methods.

C. Potential benefits for enterprise partnership

In this part, we are detailing the potential benefits for the enterprise partnership. This discussion and evaluation of potential gains can be described by three major topics:

- ❖ **The reduction of time in preliminary stages** due to the set up of collaborative environments. Indeed, this architecture is an add-on on current partners systems. This provides the capability to ensure different partners to connect it with less effort than developing connector for the other partners systems. This also provides the efficiency of an only one format exchanged. This should reduce practices protocols that currently need to be set up at the beginning of each project.
- ❖ The definition of a **shared environment for exchanges in early stages**. Indeed, such solution and capability currently doesn't exist. This should allow the partners to collaborate earlier in the project and should result in better decisions. The development of the shared environment and collaborative architecture has also advantages on data management and processes time-out. Indeed, for data management, it proposes a solution for exchanges and migration of data between the different systems which finally are more complete in preliminary stages. Concerning the processes, the definition and early implementation of trans-organisation requests should provide the capability to decrease time-out between processes. Indeed, once an environment has finished to be created and once it contains all necessary data, a request can be sent to partners so as to initiate new activities.
- ❖ Although, it also provides advantages on the **definition of environments**. The partners then get their own workspace to perform their activities. The building and rebuilding of environment for the different engineering domains provides more efficiency on data creation, use and update.

D. Potential benefits for activities collaboration

The potential benefits for activities collaboration rely on three major topics:

- ❖ The **definition of domains links**. Indeed, there is currently little connection between activities. The data are disseminated among these activities and are not related each other. The notion of context implemented in the "collaborative

schema” provides the capability to define the different environments and also define the necessary input for these environments. This also initiates the principle of data associativity which finally can create the impact chain between design and simulation domains.

- ❖ Also, the definition of an **integrated environment should limit redundancies and duplication of data in the different systems**. As the collaborative environment is based on the reference frame of the product development, it should allow the engineers to get the most recent data with the most relevant information for their activity.
- ❖ Finally, the interoperability created between design and simulation should allow **reducing iteration cycles on these activities**. Currently, simulation is performed independently from PDM data. The link between design and simulation should avoid the problem of working with unsuited data.

VI. Synthesis

The scenario presented in this chapter is based on exchanges processes and multi-activities integration. Based on main requirements for a collaborative environment, it covers the data management, context implementation and activities sequences during design iteration stages. Moreover, it is based on constraints identified in preliminary stages of projects and implements the concept of shared workspaces in “continuous data flows” (it doesn’t take into account set up stages that are difficult to exploit).

Here, the scenario is also based on data that could be exchanged during preliminary stages. This scenario gathers the entire test that can be made on such a prototype exploiting the major capabilities of collaboration and the specific developments such as the interfaces and the context management.

The development of the prototype for the architecture is not currently finished. Some implementations are still in progress for the inter-enterprise and inter-activities processes.

Conclusion

I. Contribution review

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. This PhD has been oriented to fit with this dimension and with the application in engineering product development.

From our analysis of the present engineering collaboration, issues are related to:

- ❖ 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, our contribution 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. This is implemented in our PhD following four main ideas:
 - ❖ The development of interoperability between engineering systems. Nowadays, tools and systems implemented in industry generate a saturation regarding communication developments. The definition of a non-invasive platform presented in our PhD ensures the plug-in and out of tools considering each system independently. This limits the integration and communication development at enterprise level.

- ❖ The development of interoperability product/process. Our PhD is mainly oriented on the definition of the context. This is the context that is in charge to implement this interoperability considering that engineering activities can be performed using either processes or product aspects.
- ❖ The development of interoperability between engineering data. The development of migration and data integration between tools is essential to ensure engineering quality on development. Our PhD has considered this problem defining the development of data management principles that consider the implementation of tools data models with the definition of a reference data model for multiple domains.
- ❖ 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. Our PhD shows these relations and the formalisation and definition of frontiers between these domains. Finally, they are interfaced using both product definition and product development processes.
 - ❖ 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. Nowadays, several domains use the concept of 3D models and their related PDM data (mechanical, thermal, aerodynamical...). As we are still in the logic of design driven activities, the common support must also be the reference for these domains.
 - ❖ 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. The definition of contexts is a central key point reached by our PhD to determine associated product and resources necessary for the quality of engineering.

II. Discussion about collaborative architectures

Our work is then mainly oriented to the definition of an intermediary and integrated environment for multi-engineering in the context of extended enterprise. The present diagnostic of the platform shows us that the development of such a collaborative environment is a necessary approach.

The development of extended enterprise and collaborative engineering around the virtual representation of the product has seen a limit during past years. Companies are now subject to the development and implementation of collaborative workspaces in their systems so as to reach new performance gains in terms of data availability and coherency. These collaborative workspaces lead us naturally to consider first a shared database. But in this bulk environment of 3D modelling software solutions and data management systems, such collaborative environments has to be extended to the definition of an interoperable and associative environment for data. This study highlights the importance of collaborative aspects for product development and puts forward the different objects that have to be implemented to provide a complete integrated environment for partnership design. The advantage of such architecture is to provide interoperability layers for partnerships and opens perspective on multi-view data use. This multi-layer approach allows a definition of a progressive differentiation of data that can be implemented step by step if we have to consider more partners and/or activities to integrate.

This study illustrates the major issues related to interoperability in aeronautic industry for both partnership and multi-activity purpose. It also underlines the strategy which can be used with the STEP standard and the application that is supported. The classical approaches such as database strategies or instantiation of semantic objects have to be implemented while considering the complexity of the software environment that have to be supported in aeronautic activities. The advantage of the combination we propose is to provide a more flexible environment by the use of different model layers.

This study also illustrates the necessary processes of data analysis in order to provide a reference object identified as the DMU. In our literature and with the industrial viewpoint no other architecture proposes such referencing using the concept of 3D models and especially the DMU, and provides an efficient link between CAX and data management systems. The advantage of such an approach stands on the capability of enabling the creation of referenced objects. It also highlights the need to integrate them in a framework that is able to parse data structure from the different tools so as to ensure each actor of a project to rely on the same reference frame.

III.Perspectives and related works

This PhD has raised multiple issues concerning communication and integration principles. The development and implementation of such technologies must be considered as a crucial step toward the globalisation of exchanges in the extended

enterprise. Moreover it must also be considered as a new element to control the collaborative capabilities. Then open perspectives concern three major objects that can be implemented in this way:

- ❖ The development of interoperability principles to the enterprises resources and integration of more complex environments for collaboration. This means to enlarge the scope of STEP use for engineering and the capability of application of resources to the development of the collaborative framework. This also means to integrate at higher level the concepts of enterprise resource planning (ERP) that are nowadays “standardised” in enterprise contexts.
- ❖ The development of the principle of process patterns for the development of collaborative activities. As processes can be considered as interdependent elements for the building of the sequences of development, the definition and application of process patterns to define and characterise the overall activities of the enterprise can be defined. This should provide control on intra and extra-enterprises activities using standardised transactions.
- ❖ The enlargement of the exploitation of interoperability to the different activities of the enterprise and development of the reference frame notion to analyse the impact on design processes and collaborative decisions. If we consider that there is a unique reference frame for the entire product development, this should simplify the engineering reviews and should allow the evaluation of the development considering the status of all the activities performed.

The development and informatics implementation of such a collaborative architecture is not easy. As the development was provided using STEP ISO 10303 norm, the development must be compliant with STEP structure. Also, our contributions can be integrated following three steps:

- ❖ The first step is short term related. Our major contribution is related to the implementation and development of Simulation Data Management principles. The development of the MSC Software tool called SimManager can already have a significant impact on the decrease and automation of simulation processes. Moreover, the definition of the data migration between the Snecma PDM and SimManager is already working. This should ensure a better consistency of data conveyed between design and simulation departments.
- ❖ The second step is middle term related. The definition of the “integrated infrastructure for multi-activity” is not so far. Our architecture and a part of the “collaborative schema” can be developed and applied. It should provide main

capabilities on defining a “collaborative view of product development” between activities.

- ❖ The third step is long term related. The collaboration is a widespread partnership such as in the aeronautic industry cannot be easily developed. The definition of shared but protected environments is still an issue.

The development of a collaborative environment for multi-partnership and multi-activity can be defined as the future of collaborative engineering. But, the creation of such environment is currently not easy. The use of the STEP norm is also in our opinion a necessary constraint to consider in order to reach a full interoperability of systems.

Bibliography

A.-P. Hameri, P. N. (2001). "Engineering data management through different breakdown structures in a large-scale project."

Aguilar-Saven, R. S. R. S. (2004). "Business process modelling: Review and framework." *International Journal of Production Economics* 90(2): 129-149.

Alm, I. (2003). "Designing interactive interfaces: theoretical consideration of the complexity of standards and guidelines, and the difference between evolving and formalised systems." *Interacting with Computers* 15(1): 109-119.

Altmann, U., A. G. Tafazzoli, et al. (2002). "XML-based application interface services--a method to enhance integrability of disease specific systems." *International Journal of Medical Informatics* 68(1-3): 27-37.

An, D., H. R. Leep, et al. (1995). "A product data exchange integration structure using PDES/STEP for automated manufacturing applications." *Computers & Industrial Engineering* 29(1-4): 711-715.

Ashworth, M., M. S. Bloor, et al. (1996). "Adopting STEP for in-service configuration control." *Computers in Industry* 31(3): 235-253.

Augenbroe, G., P. d. Wilde, et al. (2004). "An interoperability workbench for design analysis integration." *Energy and Buildings* 36(8): 737-748.

Aungst, S., R. Barton, et al. (2003). "The Virtual Integrated Design Method."

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.

Bacha R., Yannou B., (2002), A Methodology for Modeling Process Information - Towards a Process Technical Data Management, in *Integrated Design and Manufacturing in Mechanical Engineering*, Chedmail P., Cognet G., Fortin C., Mascle C., Pegna J. Editors, Kluwer Academic Publishers: Dordrecht/Boston/London.

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

Berden, T. P. J., A. C. Brombacher, et al. (2000). "The building bricks of product quality: An overview of some basic concepts and principles." *International Journal of Production Economics* 67(1): 3-15.

Bergman, R. and J. D. Baker (2000). "Enabling collaborative engineering and science at JPL." *Advances in Engineering Software* 31(8-9): 661-668.

Bernus, P. (2003). "Enterprise models for enterprise architecture and ISO9000:2000." *Annual Reviews in Control* 27(2): 211-220.

Bernus, P. and L. Nemes (1996). "A framework to define a generic enterprise reference architecture and methodology." *Computer Integrated Manufacturing Systems* 9(3): 179-191.

Beynon-Davies, P. and M. D. Williams (2003). "The diffusion of information systems development methods." *The Journal of Strategic Information Systems* 12(1): 29-46.

Bhandarkar, M. P. and R. Nagi (2000). "STEP-based feature extraction from STEP geometry for Agile Manufacturing." *Computers in Industry* 41(1): 3-24.

j. Billington, A. T. (1997). "Petri Nets and Traders : enabling technologies for virtual enterprises."

Bloor, M. S. and J. Owen (1996). "Product Data Exchange." *Computer-Aided Design* 28(8): 665.

Boujut, J.-F. and P. Laureillard (2002). "A co-operation framework for product-process integration in engineering design." *Design Studies* 23(6): 497-513.

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.

Browne, J., P. J. Sackett, et al. (1995). "Future manufacturing systems--Towards the extended enterprise." *Computers in Industry* 25(3): 235-254.

Burkett, W. C. (2001). "Product data markup language: a new paradigm for product data exchange and integration." *Computer-Aided Design* 33(7): 489-500.

Butrimenko, A. (1977). "International system for scientific data exchange." *Telecommunications Policy* 1(2): 163-164.

Camarinha-Matos, L. M. (2001). "Execution system for distributed business processes in a virtual enterprise." *Future Generation Computer Systems* 17(8): 1009-1021.

Case, M. P. and S. C.-Y. Lu (1996). "Discourse model for collaborative design." *Computer-Aided Design* 28(5): 333-345.

Chalmeta, R., C. Campos, et al. (2001). "References architectures for enterprise integration." *Journal of Systems and Software* 57(3): 175-191.

Chan, M. F. S. and W. W. C. Chung (2002). "A framework to develop an enterprise information portal for contract manufacturing." *International Journal of Production Economics* 75(1-2): 113-126.

Chang, Y.-S., M.-H. Ho, et al. (2001). "A unified interface for integrating information retrieval." *Computer Standards & Interfaces* 23(4): 325-340.

Charles S (2005) – PhD – « Gestion Intégrée des données CAO et EF - Contribution à la liaison entre conception mécanique et calcul de structures »

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.

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.

Chen, Y.-M. and T.-H. Tsao (1998). "A structured methodology for implementing engineering data management." *Robotics and Computer-Integrated Manufacturing* 14(4): 275-296.

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

Costa, C. A., J. A. Harding, et al. (2001). "The application of UML and an open distributed process framework to information system design." *Computers in Industry* 46(1): 33-48.

Ruben Costa, Oscar Garcia, Maria José Nuñez, Pedro Maló, Ricardo Goncalves – "e-Proc: A TO BE scenario for business interoperability" - - Interoperability for Enterprise Software and Applications Conference I-ESA'06 - March 22nd - 24th, 2006

Cullen, P.-A. (2000). "Contracting, co-operative relations and extended enterprises." *Technovation* 20(7): 363-372.

A.F. Cutting-Decelle, R.I.Young, J.P. Bourey, J.J. Michel, L.C. Pouchard - A Standardised Data Model for Process and Flow Management: ISO 15531-43 – A Step Towards CE in Manufacturing – Conference on Collaborative Engineering – September 2006

D'Adderio, L. (2001). "Crafting the virtual prototype: how firms integrate knowledge and capabilities across organisational boundaries." *Research Policy* 30(9): 1409-1424.

Davis, L., R. F. Gamble, et al. (2002). "The impact of component architectures on interoperability." *Journal of Systems and Software* 61(1): 31-45.

De Martino, T., B. Falcidieno, et al. (1998). "Design and engineering process integration through a multiple view intermediate modeller in a distributed object-oriented system environment." *Computer-Aided Design* 30(6): 437-452.

Delen, D. and P. C. Benjamin (2003). "Towards a truly integrated enterprise modeling and analysis environment." *Computers in Industry* 51(3): 257-268.

F. Demouzon, P.A. chevrin (1998) "Cost reduction through digital mock-up" ASME-98-GT-262 – Stock-holm 1998

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

Eben-Chaime, M., N. Pliskin, et al. (2004). "An integrated architecture for simulation." *Computers & Industrial Engineering* 46(1): 159-170.

Eder, W. E. (1995). "Viewpoint Engineering design -- art, science and relationships." *Design Studies* 16(1): 117-127.

Egelhoff, W. G. (1999). "Organizational equilibrium and organizational change: two different perspectives of the multinational enterprise." *Journal of International Management* 5(1): 15-33.

Estrem, W. A. (2003). "An evaluation framework for deploying Web Services in the next generation manufacturing enterprise." *Robotics and Computer-Integrated Manufacturing* 19(6): 509-519.

Eynard, B., T. Gallet, et al. (2004). "UML based specifications of PDM product structure and workflow." *Computers in Industry* 55(3): 301-316.

Eynard, B., T. Gallet, et al. (2006). "PDM system implementation based on UML." *Mathematics and Computers in Simulation* 70(5-6): 330-342.

Faux, I., E. Radeke, et al. (1998). "Intelligent access, publishing and collaboration in global engineering networking." *Computer Networks and ISDN Systems* 30(13): 1249-1262.

Favela, J., K. Imai, et al. (1994). "Hypermedia support for collaborative design." *Design Studies* 15(1): 45-58.

FERU Frederic, GUELLEC Pascal, NGUYEN VAN Thomas, BALATON Erik, TABASTE Olivier, COUTEAU Beatrice - An Engineering Data Management framework for Virtual Aircraft, - International Conference: Virtual Product Development 2006 - July 17-19, 2006

Fowler, S. and R. Karinthi (1996). "Remote access to CAD databases using an information sharing system." *Computers in Industry* 29(1-2): 117-122.

Fox, M. S., M. Barbuceanu, et al. (1996). "An organisation ontology for enterprise modeling: Preliminary concepts for linking structure and behaviour." *Computers in Industry* 29(1-2): 123-134.

Frederix, F. (2001). "An extended enterprise planning methodology for the discrete manufacturing industry." *European Journal of Operational Research* 129(2): 317-325.

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.

G. Loureiro, P. G. L. (2003). "A systems and concurrent engineering framework for the integrated development of space products." *Acta Astronautica* 53 (2003) 945 – 961.

George Toye, M. R. C., Larry J. Leifer, J. Marty Tenenbaum and Jay Glicksman (1993). "SHARE : A methodology and environment for collaborative product development."

Giannini, F., M. Monti, et al. (2002). "A modelling tool for the management of product data in a co-design environment." *Computer-Aided Design* 34(14): 1063-1073.

Goranson, H. T. (2003). "Architectural support for the advanced virtual enterprise." *Computers in Industry* 51(2): 113-125.

Gou, H., B. Huang, et al. (2003). "A framework for virtual enterprise operation management." *Computers in Industry* 50(3): 333-352.

Grandl, R. (2001). "Virtual process week in the experimental vehicle build at BMW AG." *Robotics and Computer-Integrated Manufacturing* 17(1-2): 65-71.

Gu, P. and K. Chan (1995). "Product modelling using." *Computer-Aided Design* 27(3): 163-179.

Hamdi A., Yannou B., Landel E., (2006), Exploration in the preliminary mechanical design of trade-offs between automotive architecture constraints and aggregate noise performances. in *IDETC/DAC: ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conferences / Design Automation Conference*, Sept. 10-13, Philadelphia, PA, USA.

Hameri, A.-P. and J. Nihtila (1998). "Product data management--exploratory study on state-of-the-art in one-of-a-kind industry." *Computers in Industry* 35(3 SU -): 195-206.

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

Herbert Praehofer, J. S., Alois Stritzinger (2001). "Concepts and architecture of a simulation framework based on the JavaBeans component model." *Future Generation Computer Systems* 17 (2001) 539–559.

Andreas Hoffmann, Tim Kogel, Heinrich Meyr (2001). "A Framework for Fast Hardware-Software Co-simulation."

Holland, C. P. (1995). "Cooperative supply chain management: the impact of interorganizational information systems." *The Journal of Strategic Information Systems* 4(2): 117-133.

Walter Hower, W. H. G. (1996). "A bibliographical survey of constraint-based approaches to CAD, graphics, layout, visualization, and related topics." *Knowledge-Based Systems* 9 (1996) 449 464.

Huang, G. Q. (2002). "Web-based support for collaborative product design review." *Computers in Industry* 48(1): 71-88.

Huat Lim, S., N. Juster, et al. (1997). "Enterprise modelling and integration: a taxonomy of seven key aspects." *Computers in Industry* 34(3): 339-359.

Hubel, H. and G. J. Colquhoun (2001). "A reference architecture for Engineering Data Control (EDC) in capital plant manufacture." *Computers in Industry* 46(2): 149-165.

Huifen, W., Z. Youliang, et al. (2003). "Feature-based collaborative design." *Journal of Materials Processing Technology* 139(1-3): 613-618.

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

Step Application Handbook - ISO 10303 - Version 3 - 30 June 2006

ISO 10303-1 IS 1994 - « Overview and fundamental principles »

ISO 10303-11 IS 1994- Description methods : « EXPRESS language reference manual »

ISO 10303-21 IS 1994- Implementation methods : « Clear Text Encoding of the Exchange Structure »

ISO 10303-22 IS 1998- Implementation methods : « Standard Data Access Interface »

ISO/IS 10303-104 2000- Integrated application resource: « Finite element analysis »

ISO/DIS 10303-203 (1994) – “Industrial automation systems and integration — Product data representation and exchange — Part 203: Application Protocol: Configuration controlled 3D designs of mechanical Parts and assemblies”

ISO/DIS 10303-209. “Industrial automation systems and integration — Product data representation and exchange — Part 209: Application protocol: Composite and metallic structural analysis and related design »

ISO/FDIS 10303-214:2000(E). “Industrial automation systems and integration — Product data representation and exchange — Part 214: Application protocol: Core data for automotive mechanical design processes”

ISO/DIS 10303-239:2004. “Industrial automation systems and integration — Product data representation and exchange — Part 239: Application protocol: Product life cycle support”

Jaafari, A. and K. Manivong (1998). "Towards a smart project management information system." *International Journal of Project Management* 16(4): 249-265.

Jasnoch, U. and S. Haas (1996). "A collaborative environment based on distributed object-oriented databases." *Computers in Industry* 29(1-2): 51-61.

Jeng, T.-S. and C. M. Eastman (1998). "A database architecture for design collaboration." *Automation in Construction* 7(6): 475-483.

Johansson, H., P. Astrom, et al. (2004). "A system for information management in simulation of manufacturing processes." *Advances in Engineering Software* 35(10-11): 725-733.

Kim, C.-H., R. H. Weston, et al. (2003). "The complementary use of IDEF and UML modelling approaches." *Computers in Industry* 50(1): 35-56.

Kim, S.-H. and K.-J. Jang (2002). "Designing performance analysis and IDEF0 for enterprise modelling in BPR." *International Journal of Production Economics* 76(2): 121-133.

Kirikova, M. (2000). "Explanatory capability of enterprise models." *Data & Knowledge Engineering* 33(2): 119-136.

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.

Kornblum, J. A., D. Raz, et al. (2001). "The active process interaction with its environment." *Computer Networks* 36(1): 21-34.

Kosanke, K., F. Vernadat, et al. (1999). "CIMOSA: enterprise engineering and integration." *Computers in Industry* 40(2-3): 83-97.

Kosanke, K. and M. Zelm (1999). "CIMOSA modelling processes." *Computers in Industry* 40(2-3): 141-153.

Kovacs, G. L. and P. Paganelli (2003). "A planning and management infrastructure for large, complex, distributed projects--beyond ERP and SCM." *Computers in Industry* 51(2): 165-183.

Kovacs, Z., J.-M. Le Goff, et al. (1998). "Support for product data from design to production." *Computer Integrated Manufacturing Systems* 11(4): 285-290.

Lau, H. Y. K., K. L. Mak, et al. (2003). "A virtual design platform for interactive product design and visualization." *Journal of Materials Processing Technology* 139(1-3): 402-407.

Lee, C. K. M., H. C. W. Lau, et al. (2004). "Development of a dynamic data interchange scheme to support product design in agile manufacturing." *International Journal of Production Economics* 87(3): 295-308.

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.

Leong, K. K., K. M. Yu, et al. (2002). "Product data allocation for distributed product data management system." *Computers in Industry* 47(3): 289-298.

Levi, M. H. and M. P. Klapsis (1999). "FirstSTEP process modeler -- a CIMOSA-compliant modeling tool." *Computers in Industry* 40(2-3): 267-277.

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

Liang, J., J. J. Shah, et al. (1999). "Synthesis of consolidated data schema for engineering analysis from multiple STEP application protocols." *Computer-Aided Design* 31(7): 429-447.

Lim, E.-P. and R. H. L. Chiang (2000). "The integration of relationship instances from heterogeneous databases." *Decision Support Systems* 29(2): 153-167.

Lim, E.-P. and R. H. L. Chiang (2004). "Accommodating instance heterogeneities in database integration." *Decision Support Systems* In Press, Corrected Proof.

Lin, Y. and W. J. Zhang (2004). "Towards a novel interface design framework: function-behavior-state paradigm." *International Journal of Human-Computer Studies* In Press, Corrected Proof.

Lindheim, C., T. Totland, et al. (1996). "Enterprise modeling a new task for process systems engineers?" *Computers & Chemical Engineering* 20(Supplement 2): S1527-S1532.

Liu, D.-R. and M. Shen (2003). "Workflow modeling for virtual processes: an order-preserving process-view approach*1." *Information Systems* 28(6): 505-532.

Loureiro, M. G. and D. P. G. Leaney (1999). "A systems engineering environment for integrated satellite development." *Acta Astronautica* 44(7-12): 425-435.

Madhavaram, M., D. L. Ali, et al. (1996). "Integrating heterogeneous distributed database system." *Computers & Industrial Engineering* 31(1-2): 315-318.

B. Maille, Chedmail, P, et al. (2002). "Etat de l'art sur l'accessibilite et l'etude de l'ergonomie en realite virtuelle: Accessibility and ergonomics study with virtual reality, a state of the art." *Mecanique & Industries* 3(2): 147-152.

Mark S. Shephard, M. B., Robert M. O Bara, Bruce E. Webster (2003). "Toward simulation-based design." *Finite Elements in Analysis and Design*.

Martinez, M. T., P. Fouletier, et al. (2001). "Virtual enterprise - organisation, evolution and control." *International Journal of Production Economics* 74(1-3): 225-238.

McClellan, S., B. Scotney, et al. (2000). "Using background knowledge in the aggregation of imprecise evidence in databases." *Data & Knowledge Engineering* 32(2): 131-143.

McFadden, E. T., F. LoPresti, et al. (1995). "Approaches to data management." *Controlled Clinical Trials* 16(2, Supplement 1): 30-65.

Mehli-Qaissi, J., A. Coulibaly, et al. (1999). "Product data model for production management and logistics." *Computers & Industrial Engineering* 37(1-2): 27-30.

Meinadier, J. P. (1998). "Ingénierie et intégration des systèmes " Paris : Hermès 2-86601-720-X.

Jie Meng, Stanley Y. W. Su, Herman Lam and Abdelsalam Helal. "Achieving Dynamic Inter-Organizational Workflow Management by Integrating Business Processes, Events and Rules." *Proceedings of the 35th Hawaii International Conference on System Sciences - 2002*

Merlo, C. and P. Girard (2004). "Information system modelling for engineering design co-ordination." *Computers in Industry* 55(3): 317-334.

Mertins, K., R. Jochem, et al. (1997). "A tool for object-oriented modelling and analysis of business processes." *Computers in Industry* 33(2-3): 345-356.

Mertins, K., M. Rabe, et al. (1998). "Designing a computer-aided manufacturing systems engineering process." *Journal of Materials Processing Technology* 76(1-3): 82-87.

Mervyn, F., A. Senthil Kumar, et al. (2003). "Developing distributed applications for integrated product and process design." *Computer-Aided Design In Press, Corrected Proof*.

Metais, E. (2002). "Enhancing information systems management with natural language processing techniques." *Data & Knowledge Engineering* 41(2-3): 247-272.

Ming, X. G., K. L. Mak, et al. (1998). "A PDES/STEP-based information model for computer-aided process planning." *Robotics and Computer-Integrated Manufacturing* 14(5-6): 347-361.

Mitschang, B. (2003). "Data propagation as an enabling technology for collaboration and cooperative information systems." *Computers in Industry* 52(1): 59-69.

Mo, J. P. T. and M. Zhou (2003). "Tools and methods for managing intangible assets of virtual enterprise." *Computers in Industry* 51(2): 197-210.

Monell, D. W. and W. M. Piland (2000). "Aerospace systems design in NASA's collaborative engineering environment." *Acta Astronautica* 47(2-9): 255-264.

Montreuil, B., J.-M. Frayret, et al. (2000). "A strategic framework for networked manufacturing." *Computers in Industry* 42(2-3): 299-317.

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.*

Moss Kanter, R. (1999). "Change is everyone's job: Managing the extended enterprise in a globally connected world." *Organizational Dynamics* 28(1): 7-23.

Murphy, C. A. and T. Perera (2002). "The definition of simulation and its role within an aerospace company." *Simulation Practice and Theory* 9(6-8): 273-291.

Naka, Y., M. Hirao, et al. (2000). "Technological information infrastructure for product lifecycle engineering." *Computers & Chemical Engineering* 24(2-7): 665-670.

Thomas Nguyen Van, Bruno Maille, Bernard Yannou, Jean-Pierre Bourey - Going through software and data interoperability: VIVACE Project vision - Interoperability for Enterprise Software and Applications Conference I-ESA'06 - March 22nd - 24th, 2006

Nguyen Van, Bruno Maille, Bernard Yannou - Digital Mock-Up – Capabilities and implementation in the PLM field - Thomas International Conference on Product Lifecycle Management PLM06 - July 10th – 12th, 2006

Thomas Nguyen Van, Frédéric Féru, Pascal Guellec, Bernard Yannou - Engineering Data Management for extended enterprise - Context of the European VIVACE Project International - Conference on Product Lifecycle Management PLM06 - July 10th – 12th, 2006

T. Nguyen Van, B. Maillé, B. Yannou, J.P. Bourey - System engineering applied to specification and characterisation of data management systems for collaborative engineering – Conference on Collaborative Engineering – September 2006

Noran, O. (2003). "An analysis of the Zachman framework for enterprise architecture from the GERAM perspective." *Annual Reviews in Control* 27(2): 163-183.

Nurcan, S. and C. Rolland (2003). "A multi-method for defining the organizational change." *Information and Software Technology* 45(2): 61-82.

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.

Osborn, S. (2002). "Integrating role graphs: a tool for security integration." *Data & Knowledge Engineering* 43(3): 317-333.

R.F. Paige, J.S. Ostroffa, P.J. Brooke (2000). "Principles for modeling language design." *Information and Software Technology* 42 (2000) 665–675

Pant, S., R. Sethi, et al. (2003). "Making sense of the e-supply chain landscape: an implementation framework." *International Journal of Information Management* 23(3): 201-221.

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

Park Kyung Hye and Favrel Joel (1999). "Virtual enterprise -- Information system and networking solution." *Computers & Industrial Engineering* 37(1-2): 441-444.

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.

Paulo Leitão, J. B., Luís Camarinha-Matos, Raymond Boissier (2001). "TRENDS IN AGILE AND CO-OPERATIVE MANUFACTURING."

Peltonen, H. (1993). "Engineering Data Management System : Architecture and Concepts."

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.

Peng, J., D. Liu, et al. (2003). "An engineering data access system for a finite element program." Advances in Engineering Software 34(3): 163-181.

Peng, T.-k. and A. J. C. Trappey (1996). "CAD-integrated engineering-data-management system for spring design." Robotics and Computer-Integrated Manufacturing 12(3): 271-281.

Peng, T.-K. and A. J. C. Trappey (1998). "A step toward STEP-compatible engineering data management: the data models of product structure and engineering changes." Robotics and Computer-Integrated Manufacturing 14(2): 89-109.

Pernul, G. (1995). "Information systems security: Scope, state-of-the-art, and evaluation of techniques." International Journal of Information Management 15(3 SU -): 165-180.

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

Pierra, G. (2000). "Representation et echange de donnees techniques." Mecanique & Industries 1(4): 397-414.

Pratt, M. J. and B. D. Anderson (2001). "A shape modelling applications programming interface for the STEP standard." Computer-Aided Design 33(7): 531-543.

Presley, A. R. (1997). "A MULTI-VIEW ENTERPRISE MODELING SCHEME."

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.

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.

Rob Armstrong, D. G., Al Geist, Katarzyna Keahey, Scott Kohn, Lois McInnes, Steve Parker, Brent Smolinski (1999). "Toward a Common Component Architecture for High-Performance Scientific Computing."

Rosenman, M. and F. Wang (2001). "A component agent based open CAD system for collaborative design." *Automation in Construction* 10(4): 383-397.

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

Rousset, M.-C. and C. Reynaud (2004). "Knowledge representation for information integration." *Information Systems* 29(1): 3-22.

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.

Sadoul, D. (2000). "La maquette numerique : Un exemple de developpement industriel." *Mecanique & Industries* 1(4): 373-382.

Sage, A. P. (1995). "Systems engineering and systems management for reengineering." *Journal of Systems and Software* 30(1-2): 3-25.

Ulf Sellgren and Cecilia Hakelius (1996). "A Survey of PDM Implementation Projects In Selected Swedish Industries."

Shaharoun, A. M., J. A. Razak, et al. (1998). "A STEP-based geometrical representation as part of product data model of a plastics part." *Journal of Materials Processing Technology* 76(1-3): 115-119.

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.

Sudarsan Rachuri, Eswaran Subrahmanian, Abdelaziz Bouras, Steven J. Fennes, Sebti Foufou and Ram D. Sriram (2006) - "The Role of Standards in Product Lifecycle Management Support" - Conference on Product Lifecycle Management PLM06 - July 10th – 12th, 2006

Sudarsan, R., S. J. Fennes, et al. (2005). "A product information modeling framework for product lifecycle management." *Computer-Aided Design In Press, Corrected Proof*.

Tang, M. X. and J. Frazer (2001). "A representation of context for computer supported collaborative design." *Automation in Construction* 10(6): 715-729.

Teeuw, W. B., J. R. Liefing, et al. (1996). "Experiences with product data interchange: On product models, integration, and standardisation." *Computers in Industry* 31(3): 205-221.

THONG, J. Y. L., W. HONG, et al. (2002). "Understanding user acceptance of digital libraries: what are the roles of interface characteristics, organizational context, and individual differences?" *International Journal of Human-Computer Studies* 57(3): 215-242.

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.

Vanderhaeghen D., Werth D., Kahl T., and Loos P. (2006). "Service- and Process-Matching – An Approach towards Interoperability Design and Implementation of Business Networks."

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

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.

Wang, F., J. J. Mills, et al. (2002). "A conceptual approach managing design resource." *Computers in Industry* 47(2): 169-183.

Wang, H.-F. and Y.-L. Zhang (2002). "CAD/CAM integrated system in collaborative development environment." *Robotics and Computer-Integrated Manufacturing* 18(2): 135-145.

Wang, L., W. Shen, et al. (2002). "Collaborative conceptual design--state of the art and future trends." *Computer-Aided Design* 34(13): 981-996.

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.

Werner Werner, C., R. Weidlich, et al. (2004). "Engineers' CAX education--it's not only CAD." *Computer-Aided Design* 36(14): 1439-1450.

Westfechtel B., Reidar C (1998). "Software configuration management and engineering data management: differences and similarities."

Willaert, S. S. A., R. de Graaf, et al. (1998). "Collaborative engineering: A case study of Concurrent Engineering in a wider context." *Journal of Engineering and Technology Management* 15(1): 87-109.

Wognum, P. M., J. J. Krabbendam, et al. (2004). "Improving enterprise system support--a case-based approach." *Advanced Engineering Informatics* 18(4): 241-253.

Xiong, D. and J. Sperling "Semiautomated matching for network database integration." *ISPRS Journal of Photogrammetry and Remote Sensing* In Press, Corrected Proof.

Xu, Z. G., J. H. Frazer, et al. (2002). "Novel design methodology supporting product life-cycle design." *Computers in Industry* 49(3): 253-265.

Yeomans, S. G., N. M. Bouchlaghem, et al. (2006). "An evaluation of current collaborative prototyping practices within the AEC industry." *Automation in Construction* 15(2): 139-149.

Yoo, S. B. and Y. Kim (2002). "Web-based knowledge management for sharing product data in virtual enterprises." *International Journal of Production Economics* 75(1-2): 173-183.

Zanella, M. and P. Gubian (1996). "A conceptual model for design management." *Computer-Aided Design* 28(1): 33-49.

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.

Zhao, H. and S. Ram (2003). "Entity identification for heterogeneous database integration--a multiple classifier system approach and empirical evaluation." *Information Systems* In Press, Corrected Proof.

Zhu, B. and H. Chen (2004). "Using 3D interfaces to facilitate the spatial knowledge retrieval: a geo-referenced knowledge repository system." *Decision Support Systems* In Press, Corrected Proof.

Annexe 1: VIVACE Project

I. Description of the VIVACE Project (From VIVACE Document of Work)

VIVACE (Value Improvement through a Virtual Aeronautical Collaborative Enterprise) is a project set-up in the framework of AECMA addressing aeronautics' Vision 2020 objectives to contribute significantly to fulfilling 3 specific targets of the Aeronautics Industry Strategic Research Agenda:

- ❖ Halve the time to market for new products with the help of advanced electronic analytical, design, manufacturing and maintenance tools, methods and processes.
- ❖ Increase the integration of the supply chain into the network.
- ❖ Maintain a steady and continuous fall in travel charges through substantial cuts in operating costs.

VIVACE will develop advanced capabilities (Knowledge Enabled Engineering, Multidisciplinary Design and Optimisation, Design to Decision Objectives, Engineering Data Management, Distributed Information Systems Infrastructure for Large Enterprise and Collaboration Hub for Heterogeneous Enterprises) applied on real case engineering and business scenarios from the aircraft and engine sectors.

The main result of VIVACE will be an Aeronautical Collaborative Design Environment and associated Processes, Models and Methods. This environment will help to design an aircraft and its engines as a whole, providing to the aeronautics supply chain in an extended enterprise, virtual products with all requested functionality and components in each stage of the product engineering life cycle.

II. VIVACE Project Objective

The document “Towards the GoP 2020 Vision” has led to the generation of The Strategic Research Agenda (SRA) prepared by the Advisory Council for Aeronautics Research in Europe (ACARE). This has set out the directions and targets for European research in the next decades towards fulfilling the ambition for the future of aeronautics established in the Report “European Aeronautics – a Vision for 2020” (see <http://europa.eu.int/comm/research/growth/aeronautics2020/en/>). This is supported by market forces impacting the aerospace industry and also the strong customer requirements that continue to place a strong emphasis on driving new technology to improve aircraft product performance. The industry is also faced with a rapidly changing landscape driven by increasingly demanding environmental rules,

consolidation in the whole industry, customer and competitive pressures and consequently the emergence of new forms of doing business. This leads to a new business model where existing operator alliances will increasingly integrate not only horizontally (e.g. through joint purchasing of services and aircraft) but vertically as well to compete with the American Aeronautical industry and to address rapidly evolving market requirements. The role of the aeronautics network of enterprise will therefore change becoming much more integrated. This change will influence the future product design process, requiring a quicker, cheaper and a more customised process. This in turn requires increased use of simulation and innovation in a Virtual Aeronautical Collaborative enterprise. If European industry is to remain at the forefront and retain its current world market share of approximately 50% then it must evolve its business and design simulation capabilities to allow for the more responsive supply of products that are tailored to the customer requirements/needs and optimised for the best business return. Europe has to focus more of its efforts on developing improved state-of-the-art processes if it is to remain competitive. The increased use of simulation in a distributed design environment will allow for an increased responsiveness and agility to new situations and issues thus reducing the time required to produce a customer solution. This is the purpose of VIVACE: to provide the Virtual Product in a collaborative environment (the Virtual Enterprise) through design, simulation and integration, starting from the first stages of aircraft conception. This Virtual Product groups all components that make an aircraft – the structure, the systems and the engines.

III. Project organisation

The project has been designed recognising the different technology needs of the key sectors in the European Aeronautics Industry – the aircraft sector and the propulsion sector. Both sectors have differing customer requirements and needs for their particular products from which differing supplier base can and have resulted. Overall market conditions are dictating increased integration and risk/revenue sharing between such sectors, which VIVACE recognises and takes measures to address. Therefore, a central theme flowing through VIVACE will be the exploration of those common areas of research where integration will generate and release synergy in the form of cost or lead time reduction. This has already been demonstrated through the construction stages of this proposal and has released ‘spin-off’ activities that both sectors are now exploring. Taking the above into account, the work carried out within VIVACE is separated into three general Sub- Projects which are defined as follows (see :

- ❖ **Virtual Aircraft:** a specific global product work area that develops the different elements of the aircraft, (airplanes and helicopters) and works around the products for design, modelling, interfacing and testing.
- ❖ **Virtual Engine:** a specific global product work area that develops the different engine modules of the aircraft propulsion system and key areas of multidisciplinary optimisation, knowledge management and collaborative enterprises.
- ❖ **Advanced Capabilities:** a key integrating work area that develops common tools, methodologies and guidelines to be shared among the developments in the previous two work areas and provides for further integration of these two.

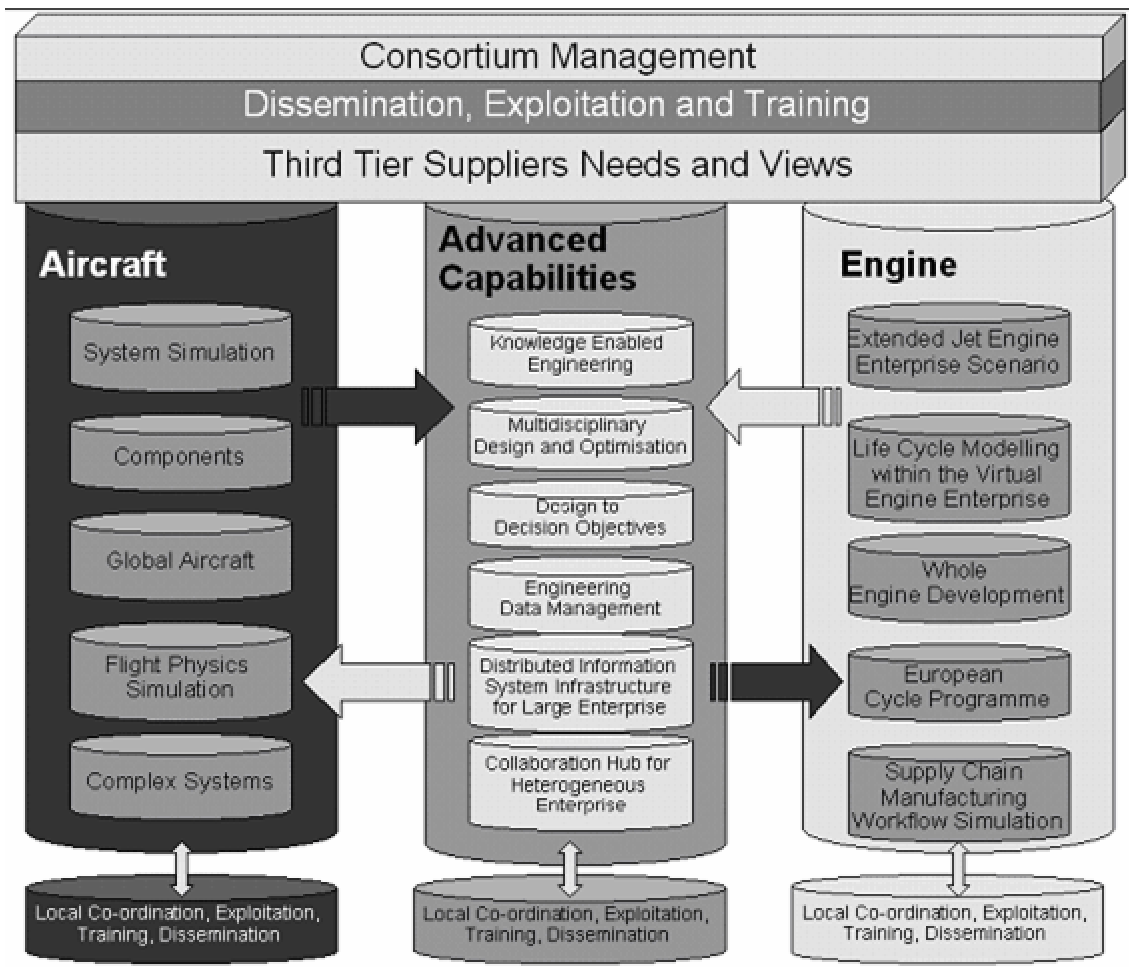


Figure 97 - VIVACE Project breakdown

Annexe 2: EXPRESS - G

From Information modelling – Getting started with EXPRESS–G by Siemens AG - Industrial Projects and Technical Services

Getting started with EXPRESS–G

1 Introduction

EXPRESS G is a diagrammatic modeling notation for the purpose of an object-oriented information modeling. This notation is based on the standardized EXPRESS-G notation, which is itself described in ISO 10303-11: Industrial Automation Systems – Product Data Representation and Exchange – Part 11: Description Methods: The EXPRESS Language Reference Manual.

2 Modeling and notation

2.1 Entity

An **entity** (or **entity type**) is a class (or set) objects. Each object in the class is somehow similar to every other object in that class, in that the object exhibit the same characteristics and/or behaviour.

In the diagrammatic modelling notation, an entity is shown as a rectangular box; the entity name is written inside the box. Examples of entities are: the set of all employees in a company (entity name **employee**); The set of all work projects (entity name **project**). The notation for these examples is shown in figure A.1



Figure A.1 – Entity

2.2 Attributes

An entity has attributes, describing the characteristics of this object. Attributes are defined as simple data types , e.g. integer, string, logical, boolean or more complex types as, e.g. other entities. Lines are used to connect the entity with its attributes. The names of the attributes are written above and along the lines. If the name of an attribute is composed of several terms, the underscore sign (_) is used as intermediate character. A solid line shows a mandatory attribute, whereas a dashed line shows an optional attribute.

Figure A.2 presents a person defined by certain characteristics including a first name, a last name, a birth date, a description of their hair and including the possibility to have a nickname.

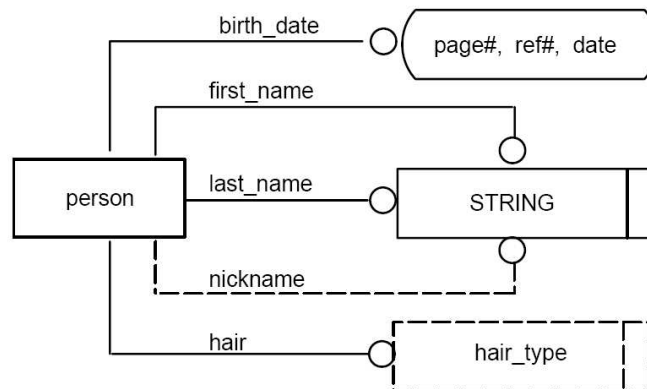


Figure A.2 – Attribute and data type

2.3 Enumeration

A data type **enumeration** expresses the existence of a range of values belonging to this attribute. Only one of the enumerated values may be chosen. Figure A.2 depicts the symbol, by which the data type is written within the

dashed box having a vertical bar at the right end of the box.

In figure A.2 different hair types are defined and exactly one value of the range has to be associated with the person.

2.4 *Select*

A data type **select** allows the possibility to choose exactly one of a defined variety of, e.g. other data types or entities. The name of the data type is written within a dashed box having a vertical bar at the left end of the box.

Figure A.3 shows an example for the use of the select data type. The data type has to be selected as one of the data types INTEGER, STRING, BOOLEAN, LOGICAL, REAL, BINARY or the entity measure_with_unit.

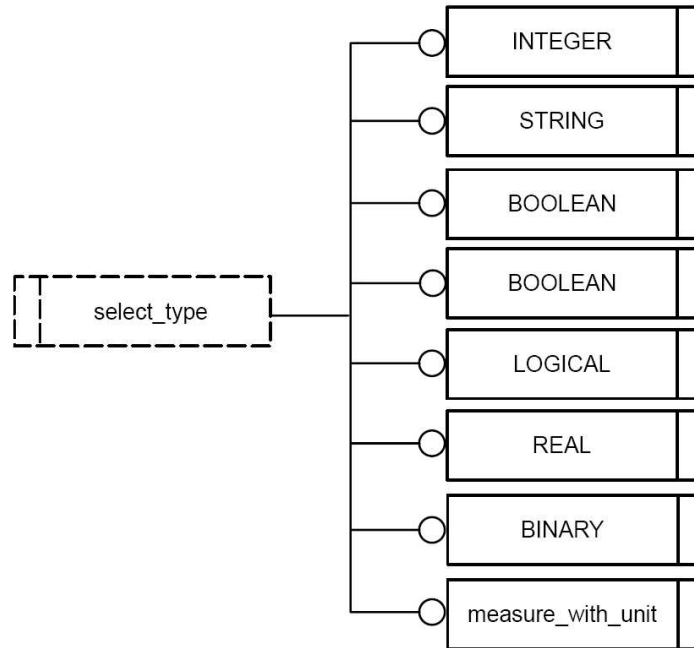


Figure A.3 – Select data type

2.5 *Cross references*

Graphical representations can span more than one page. If a relationship occurs between definitions on separate pages, the relationship line on each of the two pages is ended by a rounded box. It contains the page number, the reference number and the name of the entity referred to (see figure A.2). The page number is the number of that page where the referenced entity occurs. A reference number is used to distinguish between multiple references on a page.

2.6 *Relationship*

A **relationship** expresses a dependency or interaction between two entities. A relationship has cardinality, which indicates the number of objects in each of the entities at either end of the relationship that may be involved in a particular instance of that relationship. A relationship also has a name.

In the diagrammatic modelling notation, a relationship is shown as a solid or dashed thin line which is terminated by a circular arrowhead; the relationship name is written next to the line. The direction of a relationship is towards the arrowhead, which is important in that the name of the relationship must reflect its direction. A solid line indicates a compulsory relationship, whereas a dashed line indicates optional relationship (see figure A.4).

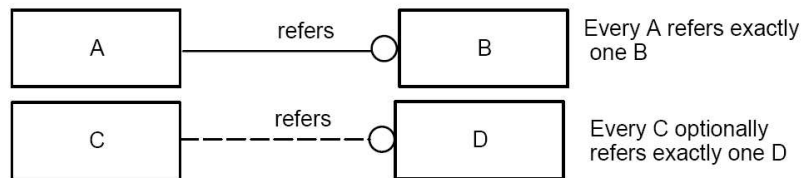


Figure A.4 – Relationships (1)

A cardinality string may be added to the relationship name, in which case it can take one of a number of forms (see below). If a relationship has no cardinality string included in its name, then the cardinality is assumed to be exactly one.

Cardinalities can be expressed in terms of sets, bags, lists and arrays. A **set** is an unordered variable length collection of unique items. A **bag** is an unordered variable length collection of not necessarily unique items. A **list** is an ordered variable length collection of not necessarily unique items; however there are possibilities to constraint the list to a list of unique items. An **array** is a fixed size collection of not necessarily unique items which can be accessed by an index.

A cardinality can be shown as a string in the form **C[m:n]**, where **C** is one of **S**(set), **B**(bag), **L**(list) or **A**(array), where **m** is the lowest number of items allowed in the aggregation, and where **n** is the highest number of items allowed. If the cardinality is shown as **m:n**, then it is assumed to be as set (S). If the cardinality is shown as a single number (n), then it is assumed to be **S[n:n]**. See figure A.5. Note that an upper limit of ? indicates "unbounded".

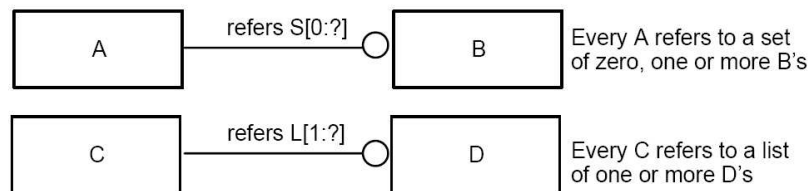


Figure A.5 – Relationships (2)

Examples of relationships are: *manages* (a **manager** manages exactly one **project**); *is_worked_on_by* (a **project** is worked on by zero, one or more **non-managers**). The notation for these examples is shown in figure A.6.

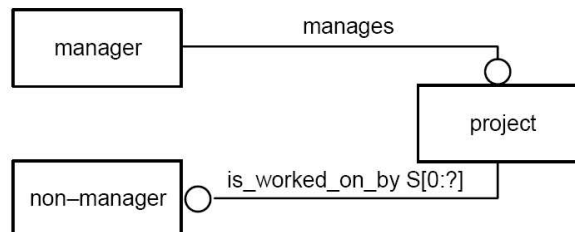


Figure A.6 – Relationship (3)

NOTE – A containment relationship is a special type of relationship (also called ownership). Such relationships are specially identified because of the special behaviour of the relationship.

Consider the two entities A and B, which have a containment relationship *owns* between them (every A owns zero, one or more B's). See figure A.7.

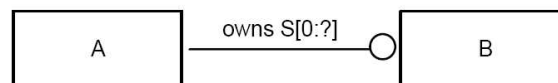


Figure A.7 – Relationships (4)

NOTE – If a database were constructed according to this model, then the database would not be in a valid state unless every instance of B were owned by one and only one instance of A. If an instance of A were deleted, then all the instances of B owned by that A would need to be deleted as well. (This is the special nature of the containment relationship). Note that this type of relationship is not specially indicated in the diagrammatic notation; however, any relationship like that shown in figure A.7 must be considered to be a possible containment relationship.

2.6.1 Inverse Relationship

In most cases, an inverse relationship can be inferred directly from the original relationship. For example, in figure A.7, the relationship *owns* has inferred inverse relationship *is_owned_by*, directed from entity B towards entity A, and having in this example the cardinality "exactly one". Strictly speaking, this should be shown as in figure A.8. (An inverse relationship is indicated by writing (INV) at the front of the name of the relationship).

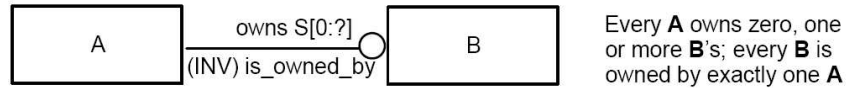


Figure A.8 – Relationships (5)

2.6.2 Entity generalisation and specialisation

Two or more entities which have some (but not all) characteristics and/or behaviour in common may be generalised into a **supertype**. Each of the entities that the supertype generalises is known as a **subtype**.

Another interpretation of subtypes and supertypes is to consider that a supertype entity is specialised into a series of subtypes. Each of the subtypes is a specialisation inherits all of the characteristics and behaviour of the supertype, but adds new characteristics and/or behaviour of its own.

Note that it is possible for an entity to be both a supertype and a subtype simultaneously.

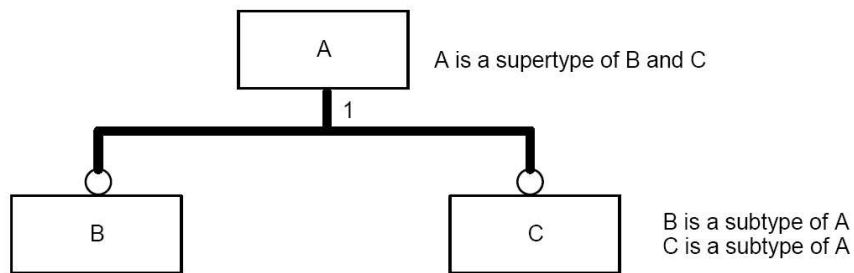


Figure A.9 – Subtypes and supertypes (1)

In the diagrammatic modelling notation, the relationship between a supertype and its subtypes is shown by using a series of thick lines, each one terminated at the subtype end with an open circle. See figure A.9.

Figure A.9 shows the notation for a **non-abstract supertype**. A non-abstract supertype is an entity which can be instantiated as an object, but which is nonetheless a supertype. An **abstract supertype**, on the other hand, is a supertype entity which cannot be instantiated. Figure A.10 shows the notation for an abstract supertype.

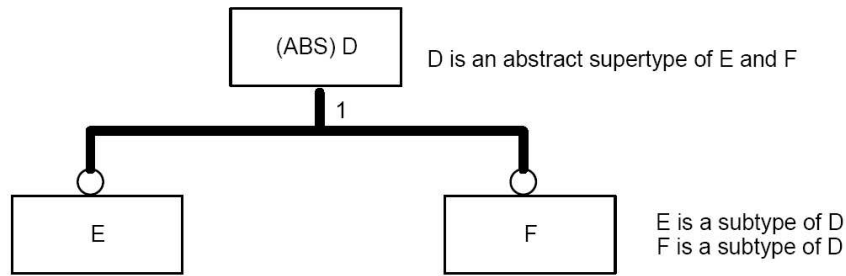


Figure A.10 – Subtypes and supertypes (2)

In figure A.9, an object of type **A** exhibits the characteristics and behaviour of the entity **A**, whereas an object of type **B** exhibits the characteristics and behaviour of entities **A** and **B** simultaneously. In figure A.10, it is not possible for objects of type **D** to be instantiated. In other words, figure A.10 shows a "complete specialisation" (all objects of type **D** are also of type **E** or type **F**), whereas figure A.9 shows an "incomplete specialisation" (there are some objects of type **A** which are neither of type **B** nor of type **C**).

An entity can be a subtype of more than one supertype simultaneously. This is known as "multiple inheritance". Figure A.11 shows the notation for an entity **A** which is a subtype of two other entities **B** and **C**.

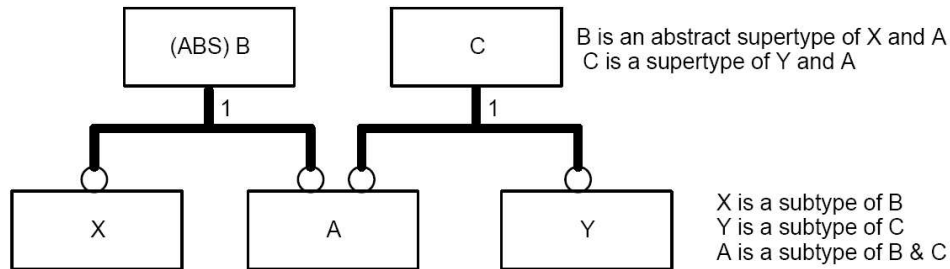


Figure A.11 – Subtypes and supertypes (3)

An example of a generalisation is that which generalises the **manager** and **non-manager** entities into the **employee** supertype (see figure A.12). The objects in the **manager** entity are slightly different to those in the **non-manager** entity (managers manage a project and supervise other employees, whereas non-managers share a lot of common characteristics (e.g. both appear on a payroll)), there are occasions when it is convenient to refer to such objects as simply an employee. This implies the existence of an abstract supertype called **employee**.

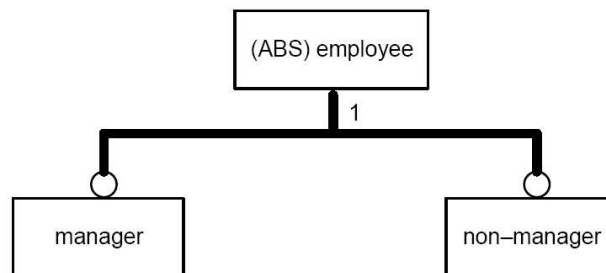


Figure A.12 – Subtypes and supertypes (4)

An example of a specialisation is that which specialises the **mammal** entity into the **cat** and **dog** subtypes (see figure A.13). All objects in the **mammal** entity share common characteristics and behaviour (e.g. they all have hair, and suckle their young). The objects in the **cat** entity are slightly different to those in the **dog** entity in a number of respects. Note that this is an example of an incomplete specialisation, because there are objects in the **mammal** entity which neither cats nor dogs (e.g. whales, horses, humans).

As a general rule, the process of generalisation will often lead to the definition of an abstract supertype, whereas the process of specialisation will often lead to the conversion of an existing entity into a non-abstract supertype (unless the specialisation is complete, in which case the entity is converted into an abstract supertype).

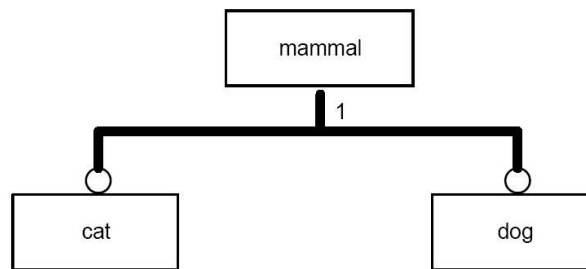


Figure A.13 – Subtypes and supertypes (5)

Many models specifically avoid the use of non-abstract supertypes. This is achieved by the addition of an extra catch-all subtype. For example, in the specialisation of the **mammal** entity shown in figure A.13, a new subtype entity called **other_mammal** could be added (as in figure A.14), making the specialisation complete, and converting the **mammal** entity into an abstract supertype.

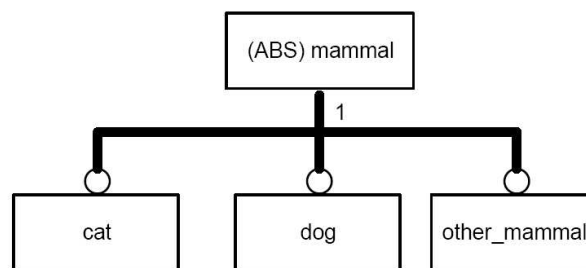


Figure A.14 – Subtypes and supertypes (6)

3 Example Model

Figure A.15 shows a simple example of an information model drawn using the notation described above.

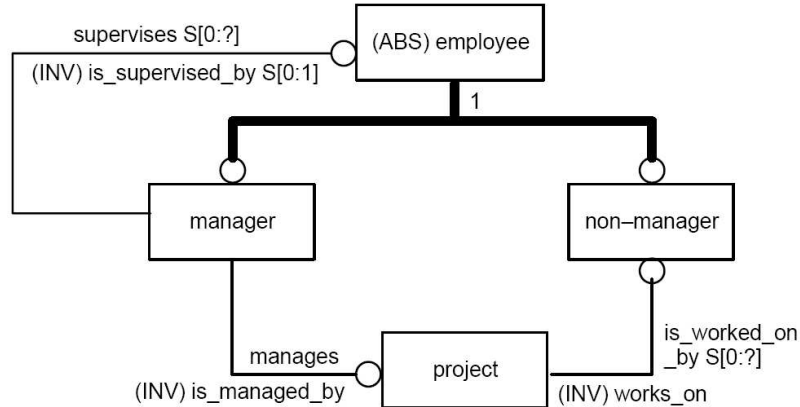


Figure A.15 – Example Model

The model should be read as follows:

1. Every manager and non-manager is also an employee; every employee is either a manager or a non-manager.

2. Each manager manages exactly one project; each project is managed by exactly one manager.
3. Each manager supervises zero, one or more employees; each employee is supervised by zero or one manager. Note that there may be some employees who are not supervised (e.g. the General manager).
4. Each project is worked on by zero, one or more non-managers; each non-manager works on exactly one project.

Note that each entity describes a complete set of objects in a particular domain of interest. A relationship is between one group of objects and another group of objects; each group of objects involved in such a relationship may be a subset of the whole group like objects in the domain of interest. A common mistake that is made while reading an information model is to assume that a relationship is between all of the objects in one entity and all of the objects in another entity.

ANNEXE 3: Data EXchange set (DEX)

Information extracted from STEP APPLICATION HANDBOOK
ISO 10303
VERSION 3
30 June 2006

I. Overview

Product Life Cycle Support (PLCS) is an ISO STEP standard (ISO 10303-239) that enables the creation and management through time of an Assured set of Product and Support Information (APSI) which can be used to specify and control required support activities throughout a complex product's life.

ISO 10303-239 provides an application-specific, but flexible, information model as part of the ISO STEP series of standards. The information model can be tailored by industry and organizations through the use of Reference Data Libraries (RDL). The role of RDL is to complete the semantics of the PLCS model necessary for deployment in industry.

The benefit of ISO 10303-239 (PLCS) is its integrated view. However this means that it has a large and generic information model that is larger in scope than most business processes require or most IT applications can manage. This problem is addressed by defining "Data EXchange Set (DEX)"

II. Data EXchange Sets (DEX)

A DEX is a way of dividing up the ISO 10303-239 (PLCS) information model into sections suited for a particular business process. A DEX provides a subset of the PLCS information model and usage guidance. A DEX can be used to contract against or for setting conformance but AP239 implementations do not have to use DEXs.

ISO 10303-239 (PLCS) has been published as an ISO standard. The DEXs are initially being standardised by publishing the subset of ISO 10303-239 (PLCS) and associated usage guidance material as OASIS standards. Once they have been used extensively, they will be included as conformance classes of ISO 10303-239.

III. The contents of a DEX

Each DEX is comprised of

- ❖ Introduction;
- ❖ Business process;

- ❖ A description of the business process that the DEX is supporting;
- ❖ Identification of the process in the AP239 activity model supported;
- ❖ Usage guidance for the model;
- ❖ DEX specific Reference Data;
- ❖ The subset of the Information model supported by the DEX;
- ❖ EXPRESS information model;
- ❖ XML Schema (derived from the EXPRESS);

There are a number of parts of the PLCS model that will be common to many DEXs. (e.g. date and time). Rather than each DEX replicating the usage guidance for these, they are packaged into chapters called "Capabilities" that are reused across different DEXs.

IV. Current set of DEXs

A. D001 - Product Breakdown for support

Exchange of the relationship of the parts assembly structure, derived from a PDM system, to an LSI/LCN structure used to manage support, and the links to relevant documents

B. D002 - Faults related to product structures

Exchanges the output from Fault Analysis programs in a form that can be used to identify required diagnostic and maintenance tasks, and to provide coherent fault reporting

C. D003 - Task Set

Exchange of a set of task descriptions, to support a work plan, or for use in multiple support solution definition.

D. D004 - Work Package Definition

Exchange and negotiation of a work package for a specific support opportunity including the list of required tasks, location, dates, products and resources.

E. D005 - Maintenance plan

Exchange for defining and communicating the work required to sustain a product over time including the results of any Logistic Support Analysis.

F. D007 - Operational Feedback

The exchange of the observed configuration, location, state or properties of an actual product, and the communication of work requests to resolve issues arising from feedback on its usage

G. D008 - Product as Individual

Exchange and collation of manufacturing and serialised part information and its relationship to the product assembly structure from which it derived.

H. D009 - Work Package Report

The exchange to support the reporting of work completion against a work package definition.

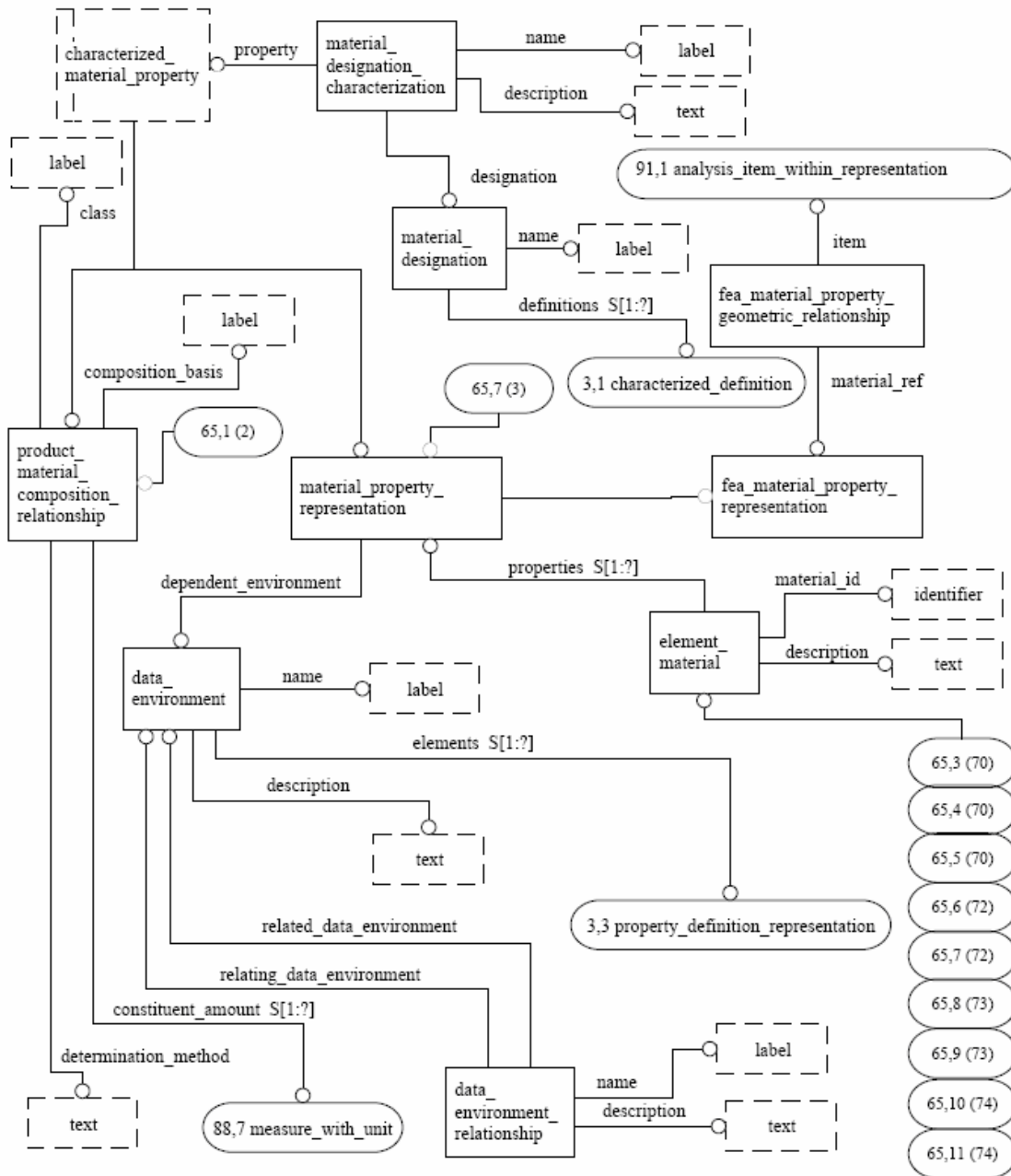
I. D010 - System requirements

The exchange of requirements information related to a system.

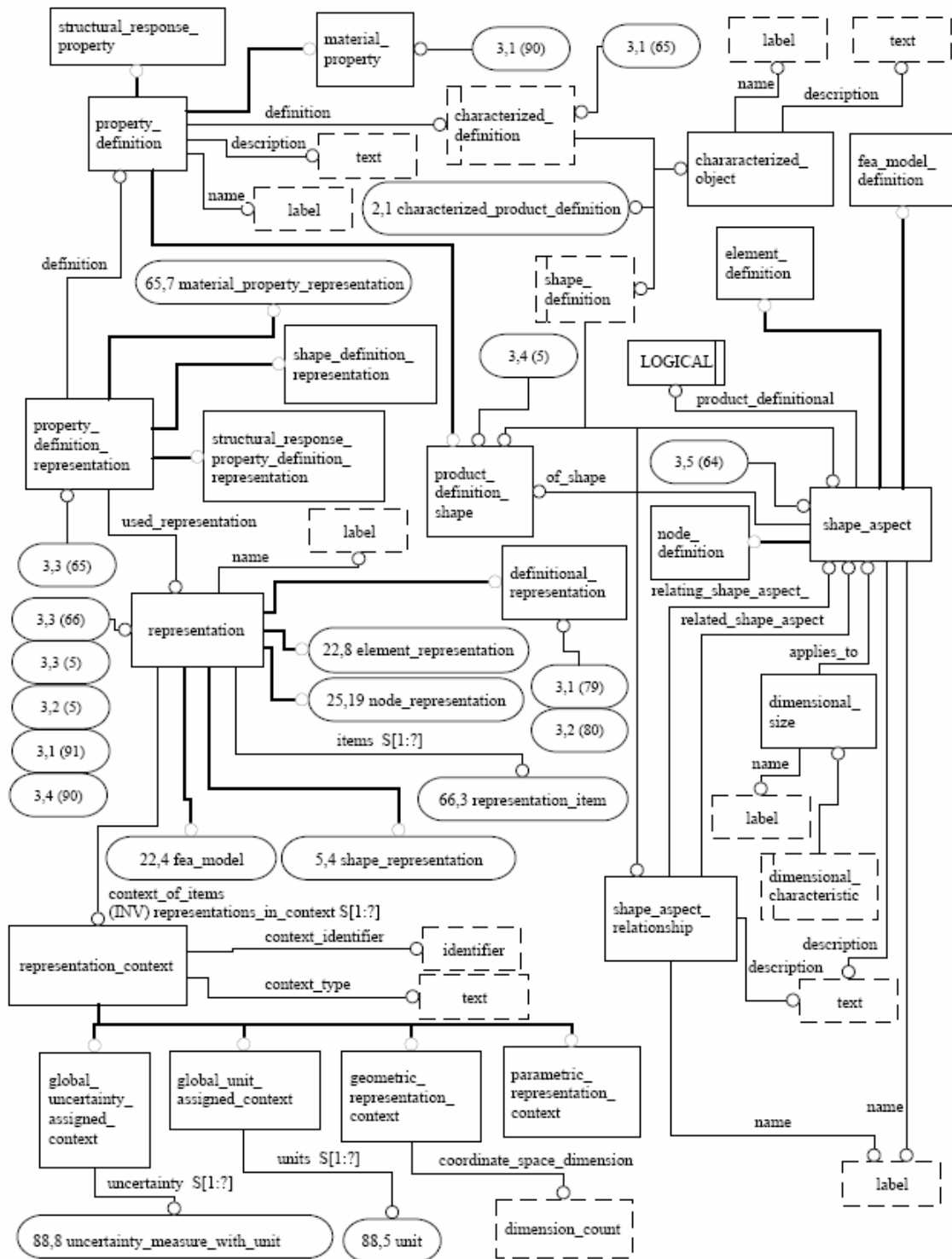
**ANNEXE 4: Application protocols
references in the “collaborative
schema”**

I. Application Protocol 209 – AIM diagrams details

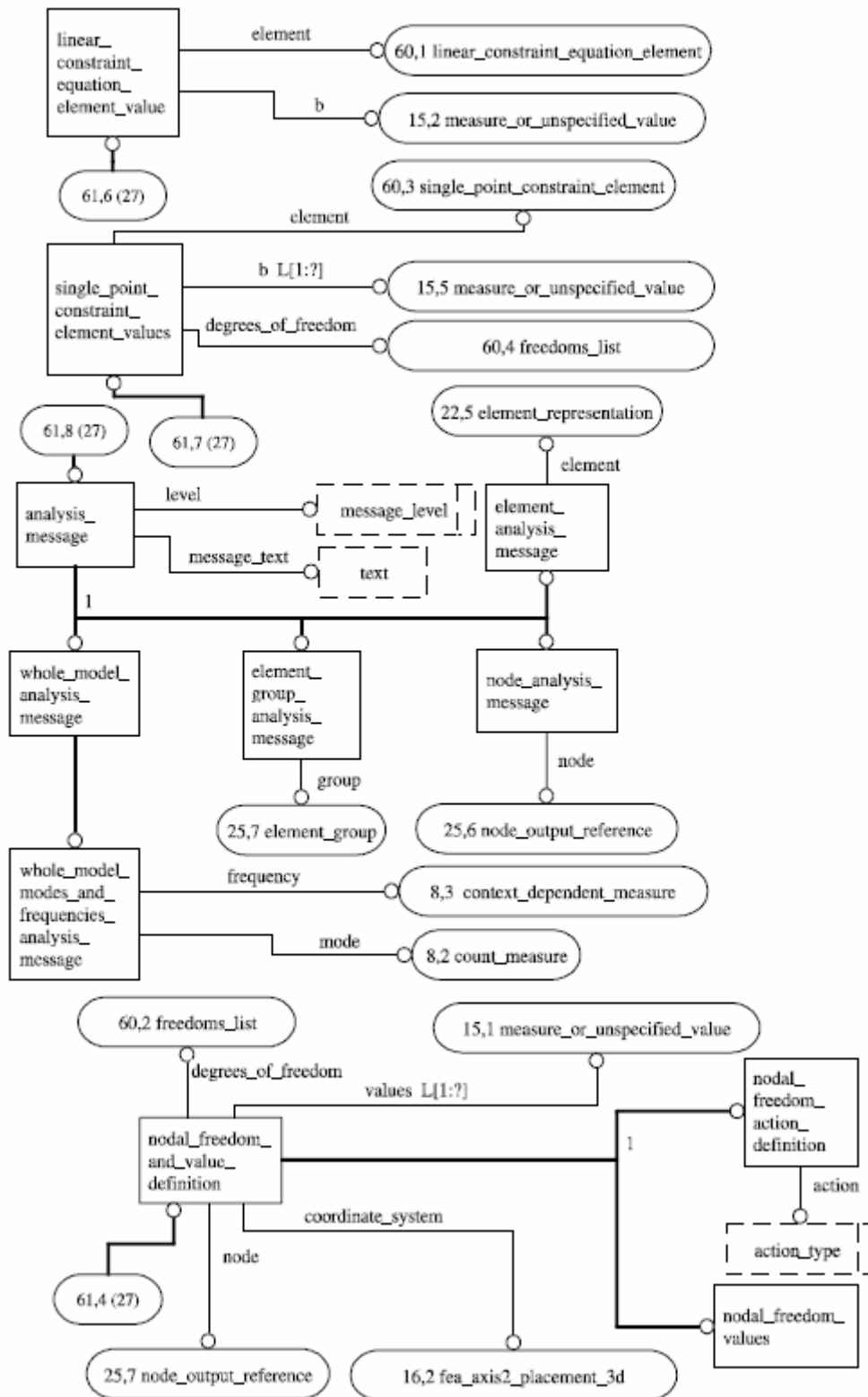
A. Material_designation



B. Material_property and Shape_definition / fea_model_definition relationships

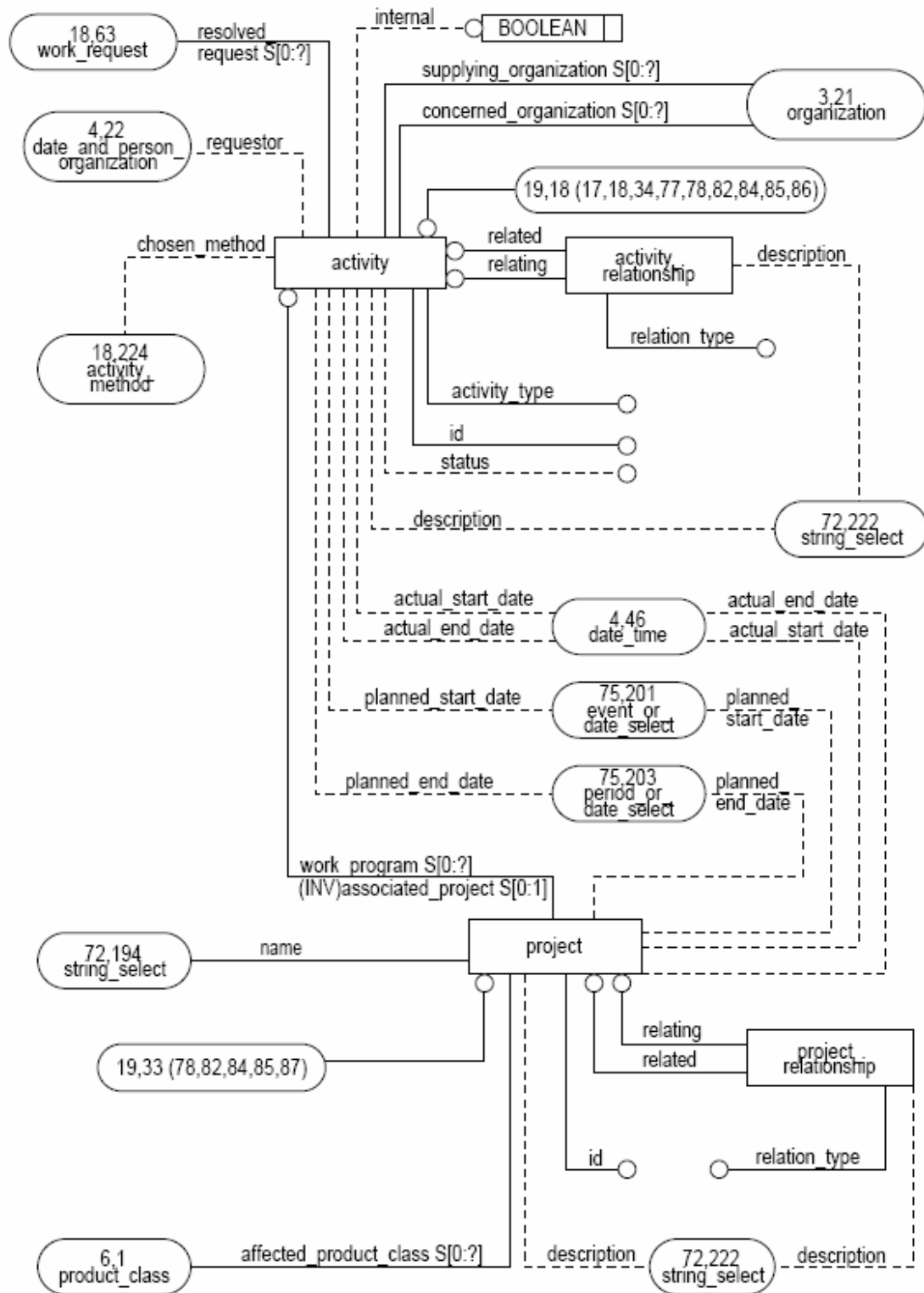


D. Nodal_freedom_and_value_definition

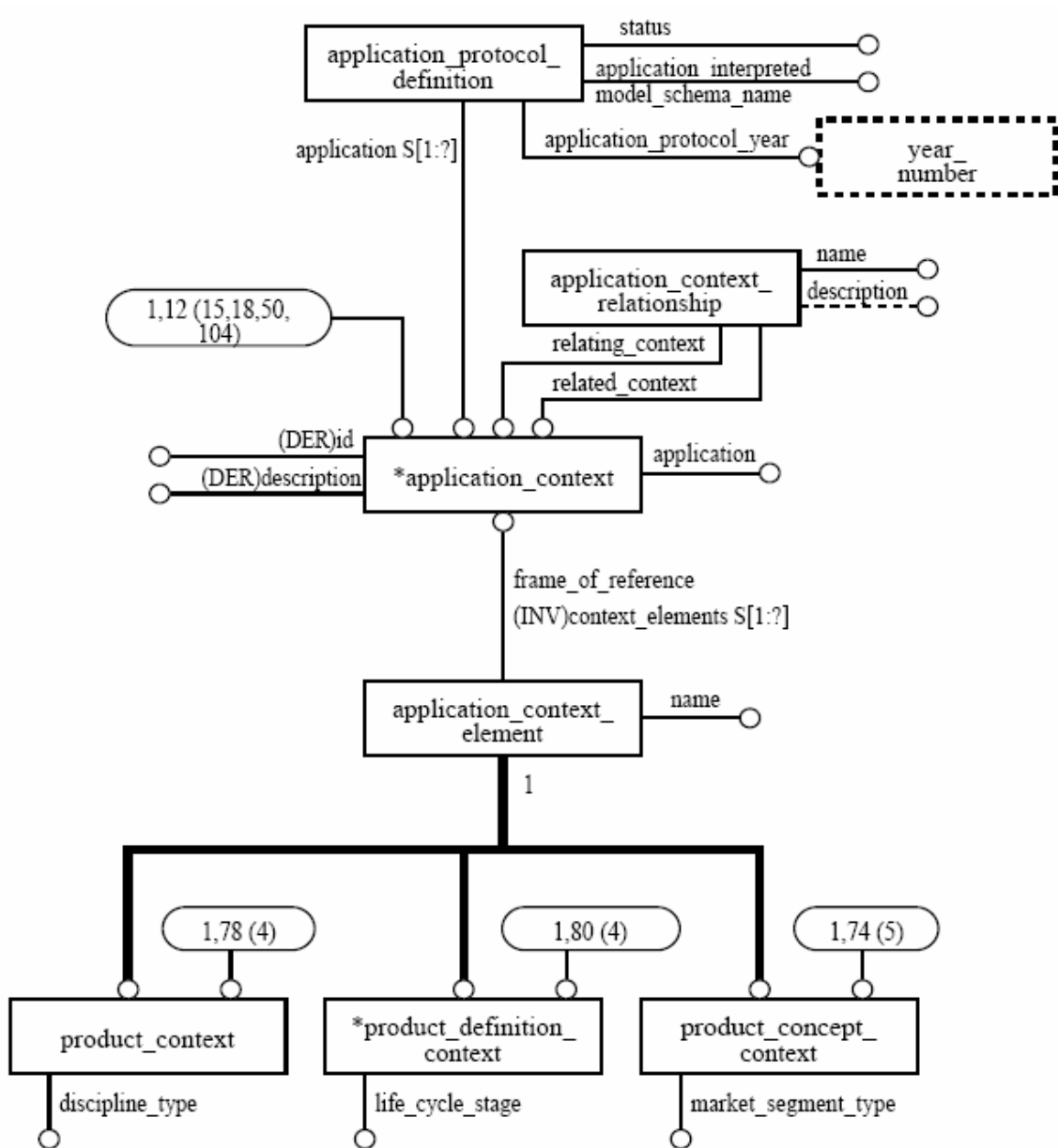


II. Application Protocol 214 – diagrams details

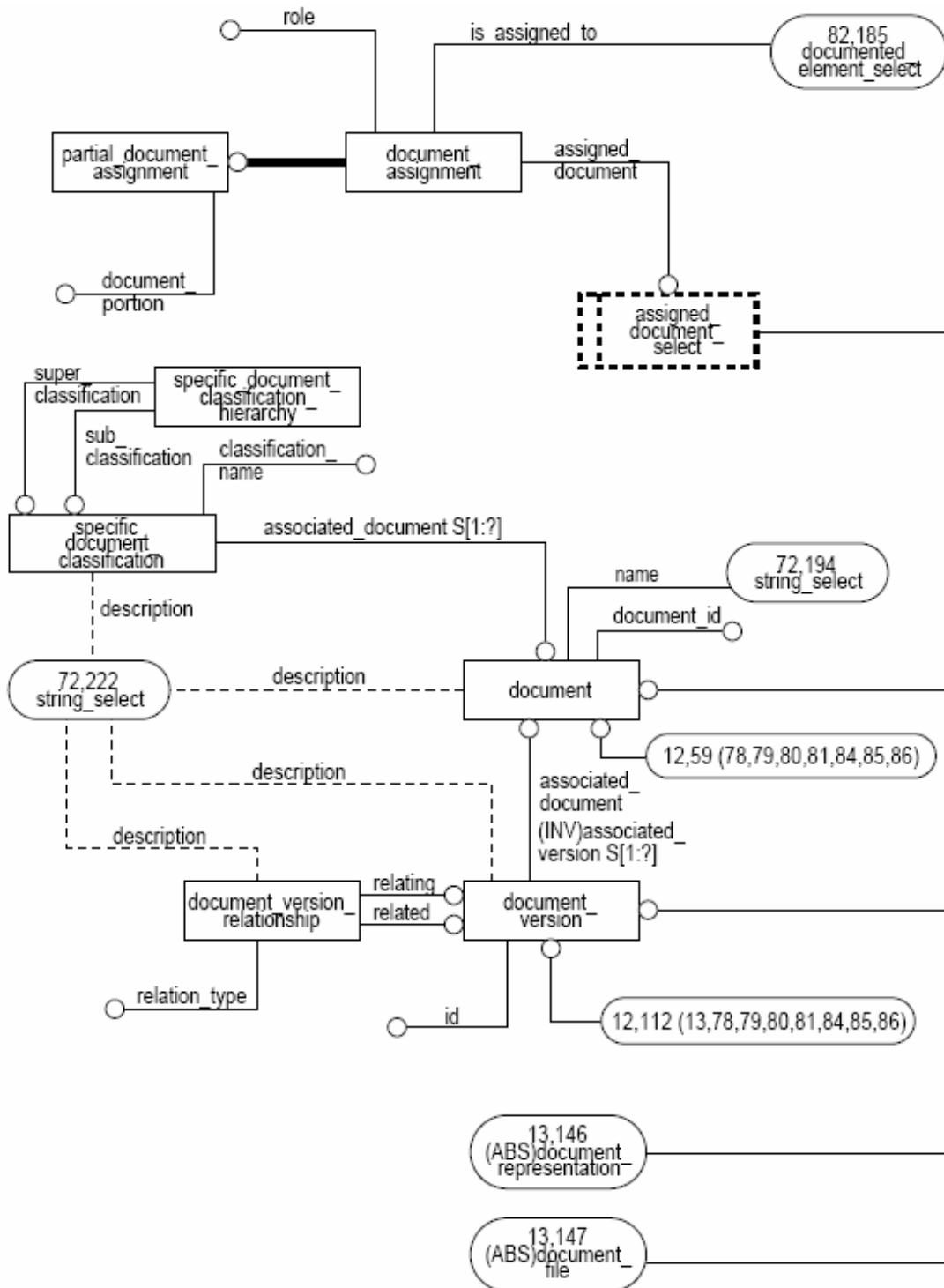
A. Activity (ARM diagram)



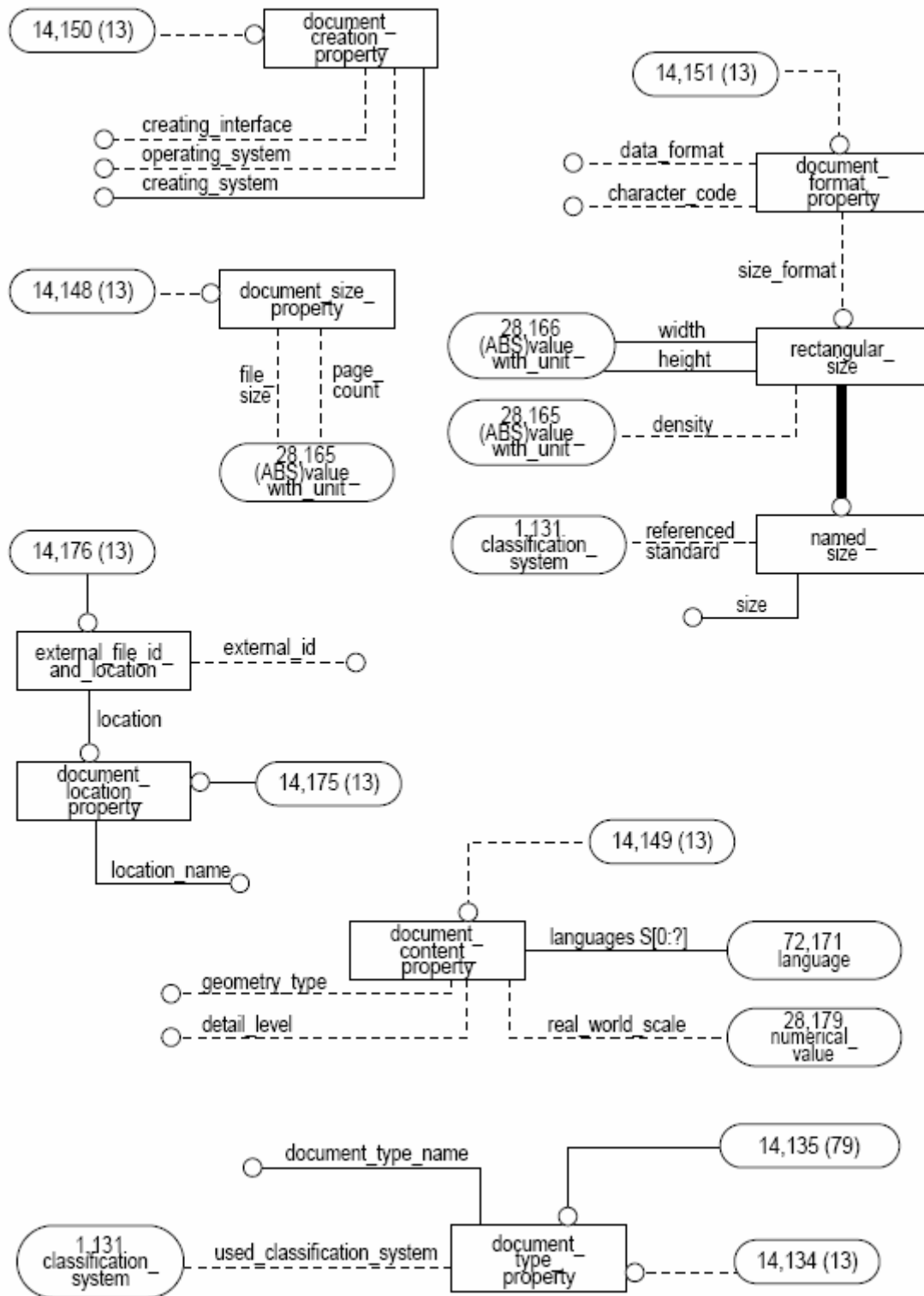
B. Application_context (AIM diagram)



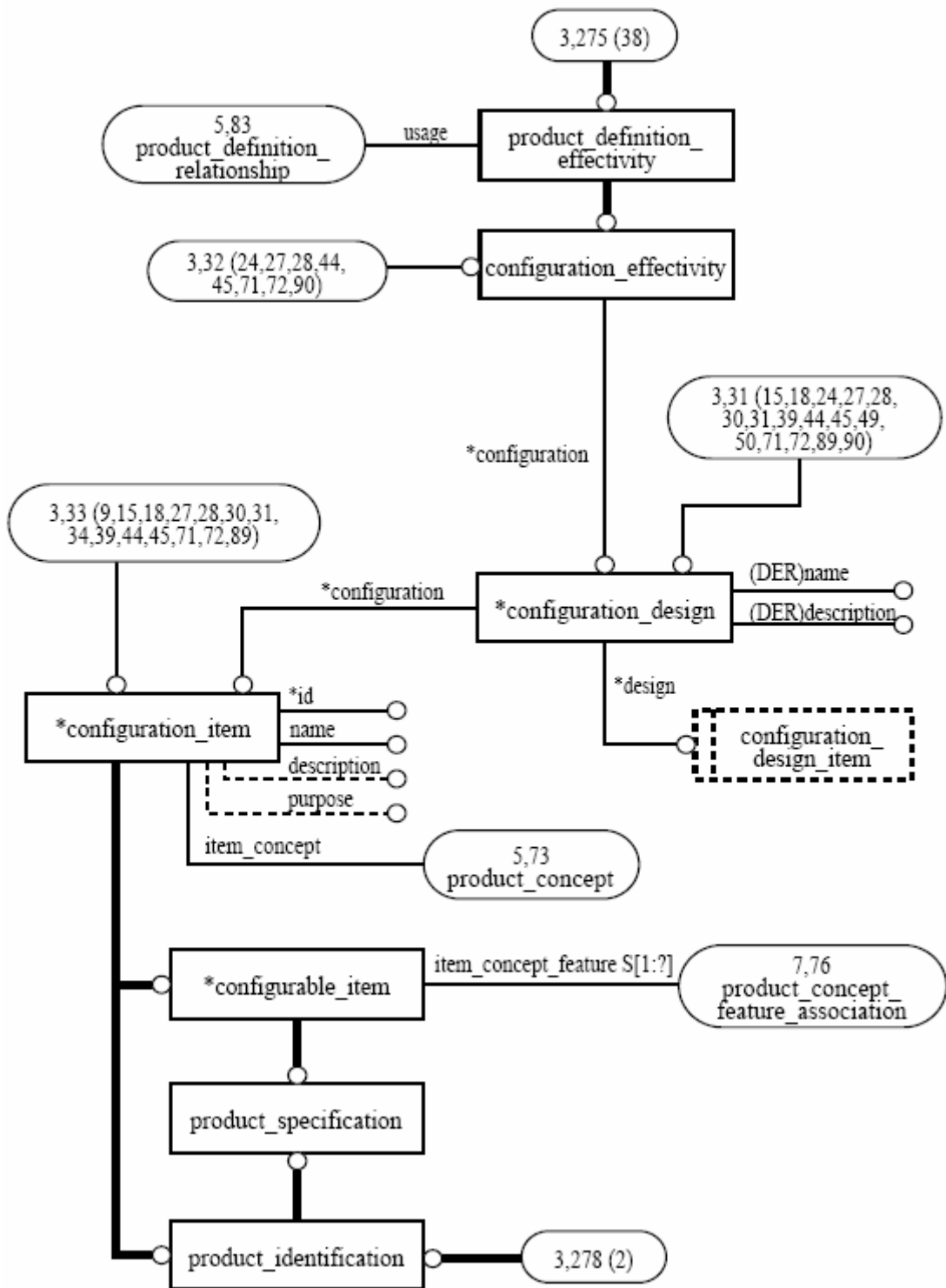
C. Document (ARM diagram)



D. External_file_id_and_location (ARM diagram)

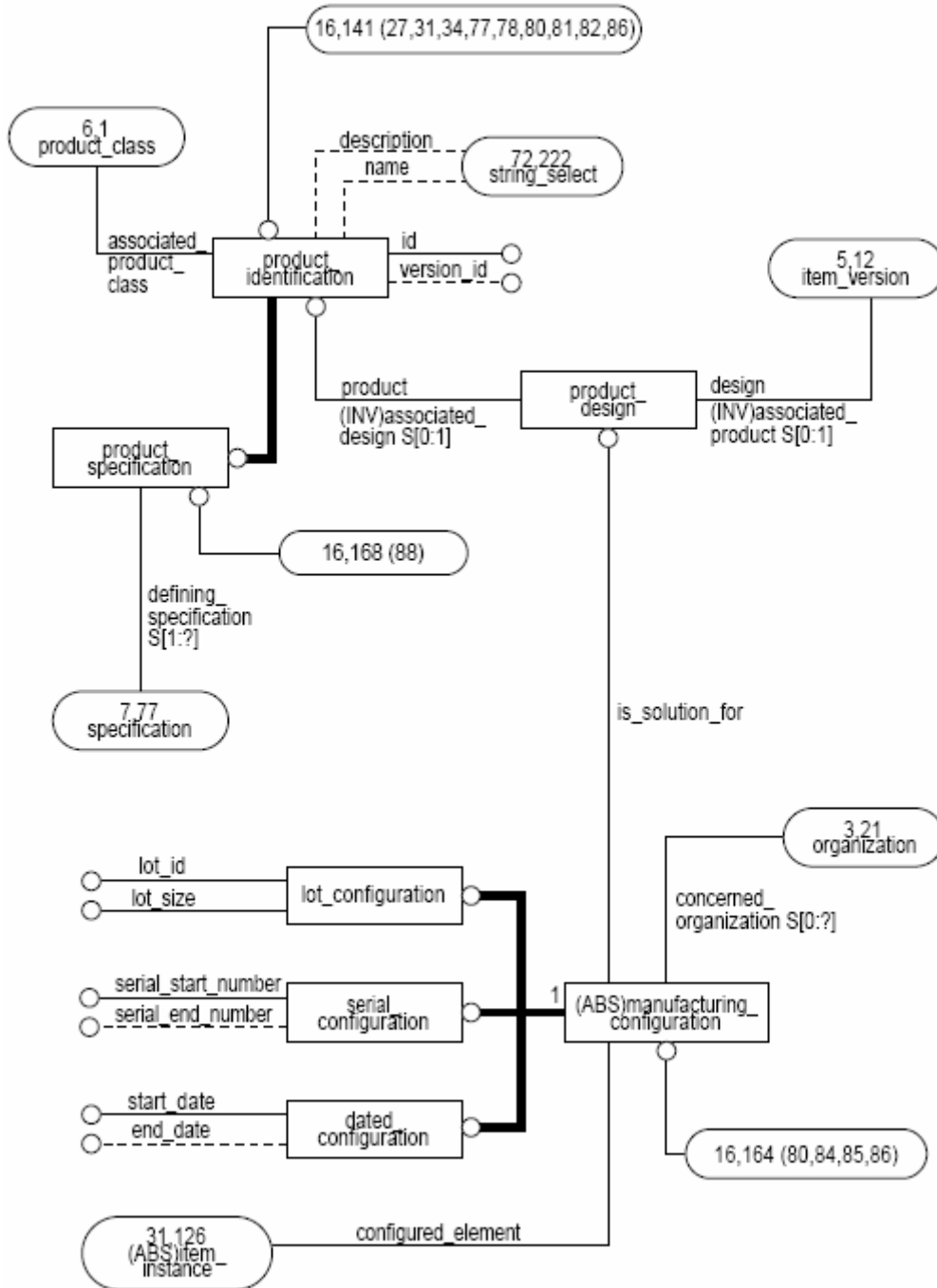


E. Configuration_design (AIM diagram)

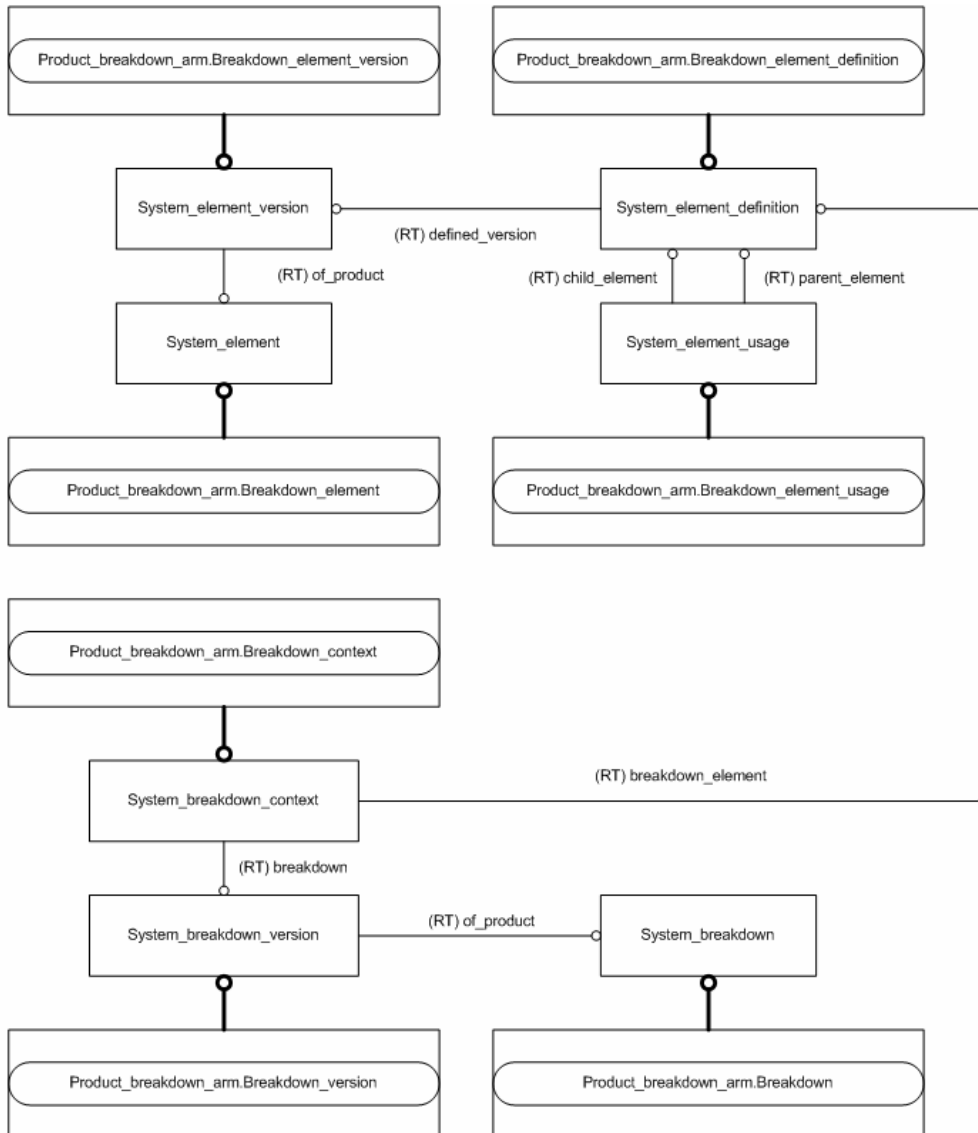


III.Application Protocol 239 – ARM diagrams details

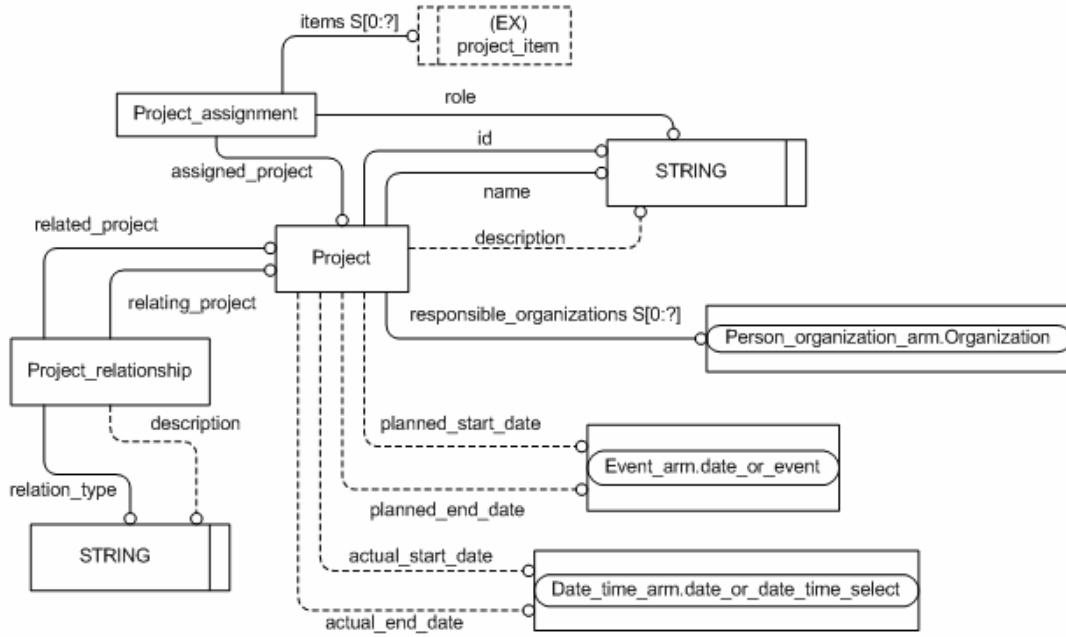
A. Product_identification



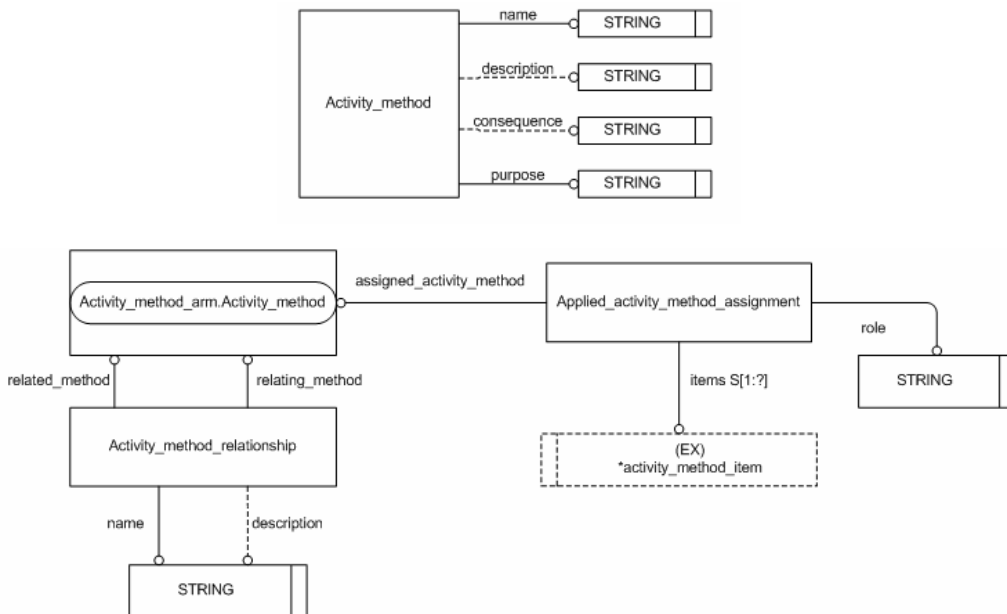
C. System_breakdown



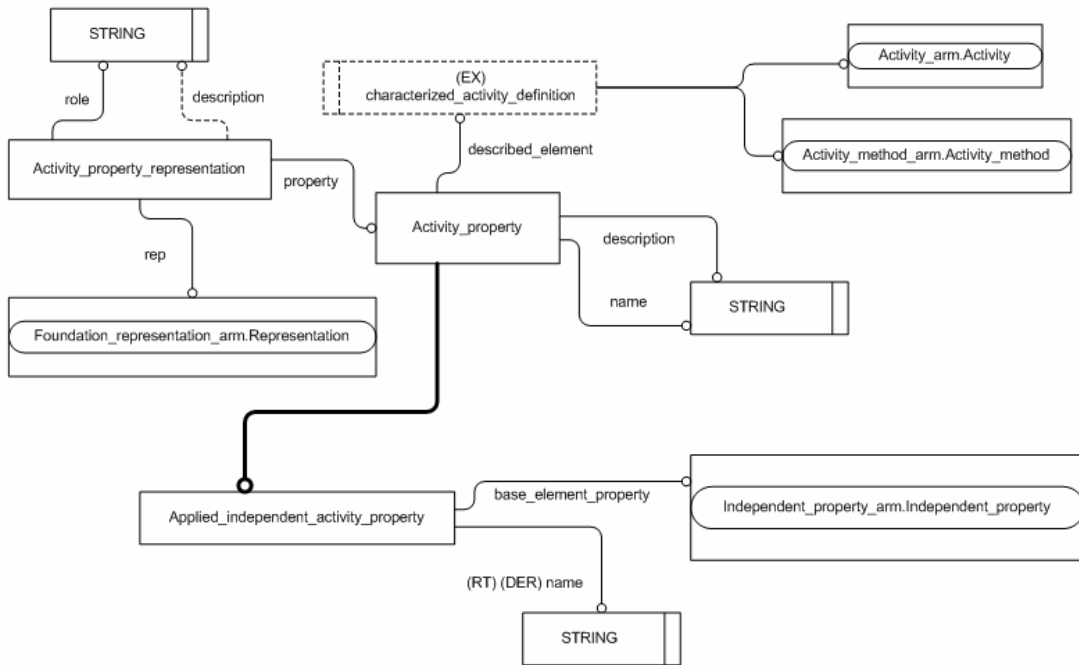
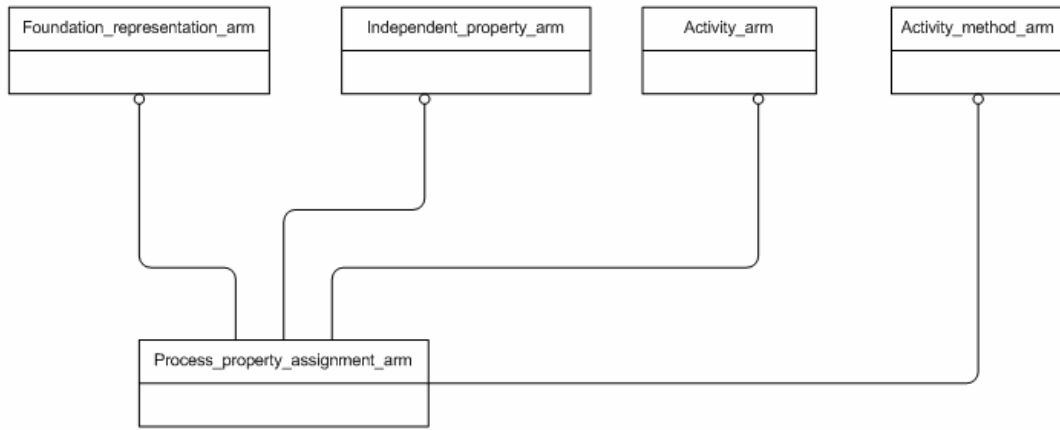
D. Project



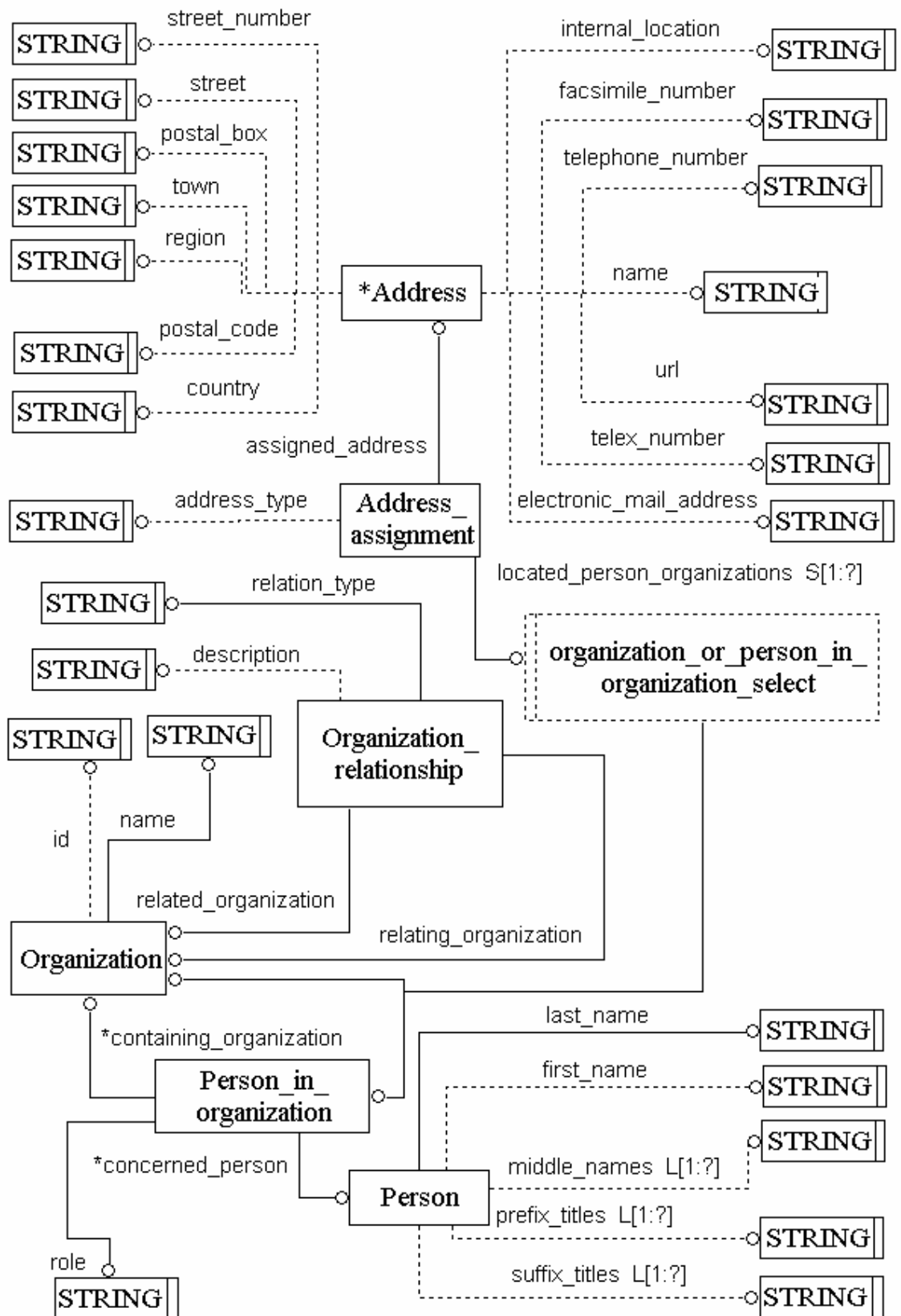
E. Activity_method



F. Process_property_assignment



G. Person_in_organisation



H. Document structure



Résumé

Cette thèse analyse l'évolution des technologies numériques et des méthodes applicables aux systèmes de gestion des données dans le contexte de l'entreprise étendue. De nos jours, dans le contexte de développement de produit en aéronautique, les projets évoluent vers des partenariats à grande échelle. Une grande quantité de données créées pour ce développement de produit doit alors être traitée et contrôlée d'une manière logique entre les différents partenaires. En effet, le besoin de permettre l'échange et l'intégration de données a évolué vers la gestion de contextes intégrés de conception. De cette façon, une intégration plus étroite entre les données créées et contrôlées est identifiée comme facteur permettant de raccourcir le cycle de développement. Cette intégration définit aussi bien l'intégration des données créées tout au long du projet que la structure informatique qui soutient ces processus. Cette thèse présente l'étude d'une architecture centralisée pour assurer le multi-partenariat et l'utilisation multi-activité des données en fournissant un référentiel pour celles-ci. L'utilisation des normes dans cette étude est également une condition pour résoudre la problématique d'interopérabilité en termes d'uniformisation et de migration des données et informations entre les différents systèmes d'ingénierie et les activités.

Mots –clés: ingénierie collaborative, plate-formes collaboratives, gestion des données, standardisation, maquette numérique

Abstract

This doctoral thesis analyses the evolution of digital technologies and methods for data management systems applied to the context of extended enterprise. Nowadays, in the aeronautics' product development context, projects are evolving towards large-scale partnerships. A large amount of data created for this product development has then to be processed and managed in a coherent way between the different partners. Indeed, the need of enabling data exchange and integration has evolved from the simple file to the management of an integrated design context. This way, closer integration between data created and managed is identified as a factor to shorten the development cycle. This integration involves as well the integration of the overall data created all along the project as the structure to support processes. This PhD dissertation then presents the study of a centralised architecture to ensure the multi-partnership and multi-view engineering by providing a common referential for data semantic. The use of standards in this study also complies with a high level of interoperability issues in terms of consistency and easy migration of data between engineering systems and activities.

Keywords: Collaborative engineering, collaborative platforms, Data management systems, Standardisation, Digital Mockup