Leonardo-Finmeccanica – Aircraft Division needs for Integrated Systems Engineering: the CRYSTAL user experience

Bruno Di Giandomenico, Claudio Pessa, Laura Valacca, Elena Valfrè, Ivo Viglietti
Engineering Department
Leonardo-Finmeccanica - Aircraft Division
Torino, Italy

***Abstract*—In the aerospace sector the products complexity clearly highlights the need to manage a wide variety of technologies in evolving and changeable scenarios.**

A way to successfully govern this multiplicity is by designing products thoughtfully integrated and ready to operate in scenarios that grow more and more entangled by the day. Fully embracing an integrated Systems Engineering (SE) approach is one way to untie this knot. Leonardo-Finmeccanica – Aircraft Division has been involved in different innovation initiatives aimed at promoting the use of Model Based Systems Engineering (MBSE) methodologies and tools integration as standard practices, in order to define an internal structured and robust Systems Engineering Process.

The participation to the ARTEMIS CRYSTAL research project is part of these initiatives and is focused on validating innovative workflows and modeling approaches in a multi tool – multi user scenario. The definition of an appropriate Project Management environment and of a minimum set of Configuration Items to be properly managed starting from the Preliminary Design phase also represents a primary objective.

This paper explains the industrial needs, the applied solutions and the experience made in implementing the demonstration scenario in the CRYSTAL research project focused on a specific aerospace Case Study.

Keywords—Model Based Systems Engineering; Interoperability Specification; Tool Chain; OSLC; SPEM; Product Lifecycle Management.

I. Introduction

In order to maintain a competitive position in a very crowded market scenario, it is vital to reduce the development projects costs. “Best practices” collection and reuse management criteria are additional means which can help to realize this objective.

Our choice has been to fully embrace a Model Based engineering approach enabling us first and foremost to provide the capabilities required by the stakeholders requirements with a strong focus on the integration aspects of the product’s functionalities [1], [2]. To support our multidisciplinary, concurrent and integrated development process the challenge lies in harmonizing all the growing specialized ‘systems’ (processes, methods and tools) and related heterogeneous data supporting the engineering activities throughout the entire lifecycle. Often these ‘systems’ are based on the “best in class” solutions in a given discipline, that in many cases should be complemented by customized in-house ones (often proprietary).

One of the initiatives aimed to validate our choices we have been involved in is the ARTEMIS CRYSTAL research project that is mainly focused in establishing and pushing the development and adoption of an Inter-Operability Specification (IOS) as an open European standard for the development of embedded systems in the automotive, aerospace, rail and health care domains. Based on open Web technologies such as OSLC [8], IOS allows loosely coupled tools to share and interlink their data enabling interoperability among different engineering environments without closely coupled process integration.. Within the project a dedicated process description language, SPEM [6] supported the formalization of the Engineering Environment specification and of the System Engineering “best practices”.

For this research project, we have selected an industrial Case Study implemented with the support of a dedicated tool chain and focusing on the Preliminary Design Phase of the Aeronautical products Lifecycle. This paper focuses on the Case Study implementation.

II. Challenges and CRYSTAL vision

A. Industrial Needs

The market for most aeronautic companies shows a heightened sensibility on costs and is subject to many customers’ budget reductions. To keep the market share, the robustness of design and awareness of budget must be supported by adequate methodologies, virtual validation and comprehension of design choices and changes impact costs since the earliest phases of product lifecycle. The growing operational complexity of aeronautical products also fosters the need for a strong relationship between suppliers and customer who does not want anymore a simple product, but an organized set of functions to be satisfied in different operational scenarios.

The current emerging standard in the aerospace field, SAE ARP-4754A [4] suggests an approach in which development and safety assurance processes are closely coupled and puts a strong emphasis on the functional driven development which can be represented with models to deliver a structured, linked and retrievable information set. A system model may support also various degrees of simulation to assess and validate the customer's requirements already in the first phases of the project development. As a reference, we have chosen the emerging standard language in functional modeling, the System Modeling Language (SysML) [5], which also aid in accelerating the change from Document Based to Model Based Engineering and shall have a major impact in our company engineering practices, by changing the basic perception of the product design and improving knowledge reuse.

B. *Project Approach*

CRYSTAL, as an ARTEMIS Innovation Pilot Project takes up the research results of previous projects in the field of Reference Technology Platforms (RTP) and Interoperability Specification (IOS) (e.g. CESAR, MBAT) and aims at enhancing and maturing them with the clear objective of industrialization take-up.

CRYSTAL's vision is to enable new Systems Engineering methods and practices by promoting the RTP, an open tool integration platform that can be viewed as a set of formalized components (tools, methods, connectors) that can be used to set up a Systems Engineering environment in a company, and supporting an IOS which will simplify tools connection by exposing services and linked data.

RTP and IOS can enable common interoperability among various lifecycle domains, which can reduce significantly the complexity of the entire integration. CRYSTAL is strongly industry-oriented and is aiming at providing ready-to-use integrated tool chains with a mature Technology Readiness Level (up to TRL 7).

Following the ARTEMIS mission, in order to strengthen the European market for Embedded Systems, CRYSTAL fosters cross-domain reusability (Aerospace, Automotive, Health and Rail) and drives the IOS towards standardization.

The strategy for CRYSTAL technical innovation is based on 4 pillars:

1. To increase the maturity of existing concepts developed in previous projects and allowing integration into existing environments, by applying them in industrial scenarios.
2. To provide high maturity technical innovations to fill the gaps through a step by step evolution and an assisted Systems Engineering environment configuration approach.
3. To contribute to the ARTEMIS envisaged Cooperative RTP and to push the IOS towards Standardization through a validation provided by "Close-to-real-world" demonstrators.
4. To support Small and Medium Enterprise integration in the engineering ecosystem for the embedded systems development.

C. *Enabling Technologies*

The analysis of the project needs for interoperation led to the identification of innovative "enabling" technologies, the most relevant ones being OSLC [8] to support interoperability and SPEM [6] to support process and knowledge formalization and reuse.

OSLC

Open Services for Lifecycle Collaboration (OSLC) [8] is a community which is creating specifications to allow conforming independent software and product lifecycle tools to integrate their data and workflows in support of end-to-end lifecycle processes. Examples of lifecycle tools may include requirements management, change management, asset management, etc.

In OSLC, each artefact in the lifecycle – for example, a requirement, a test case, a source file, a model element and so on – is an HTTP resource that is manipulated using the standard methods of the HTTP specification (GET, PUT, POST, DELETE).

OSLC specifies a common tool protocol for Creating, Retrieving, Updating and Deleting (CRUD) lifecycle data based on internet standards like HTTP and Resource Description Framework (RDF), using a Linked Data model achieved by embedding the HTTP URL of one resource in the representation of another.

SPEM

SPEM (Software and System Process Engineering Meta-model) is a process meta-model with focus on engineering processes. It is an Object Management Group specification and it is based on the Meta-Object Facility (MOF) model [9] and its UML2 Profile. SPEM can be considered as an industrial standard that facilitates human understanding and communication of software processes to promote their reuse and improvement.

It has been selected in our context because it is widely used for process definition, becoming a de facto standard that allows companies to define highly personalized processes. It allows the representation of the basic items that compose processes: the approach is having two main branches called "Process" and "Method Content" respectively which are based on a Core specification and are developed to allows users to select the packages they need to define their processes, giving some freedom in applying the specification and avoiding useless implementation by reusing already accomplished work.

III. CASE STUDY

Our Case Study had to fulfill the CRYSTAL objective of staying strongly industry oriented, focusing on a highly relevant on-board system for future aeronautical products. Health and performance monitoring functionalities and the related integration with ground systems for the flight data analysis and the identification of the maintenance activity is undergoing a fast evolution. We have thus identified the need to specify the functionalities of an Enhanced Integrated Monitoring and Support System (EIMSS), which includes processing capabilities in a Maintenance Computer (MC) and a Multi-Functional Display (MFD) and monitors “member” systems’ health. In addition, the Ground components of the EIMSS which enables operator to interact with the Aircraft is called Ground Support System (GSS). Fig. 1 illustrates the most important relations among the EIMSS and the other systems.

Fig. 1. Interactions of EIMSS with other systems

As the main focus of the Case Study is on the flight phase’s functionalities, the design of a representative Aircraft’s system to be monitored is also part of the exercise. The Fuel System has been selected as representative on-board system, in order to verify the capability of the EIMSS in collecting and elaborating different types of data (e.g. discrete/ analogue inputs) coming from different types of sources (e.g. sensors, control units, electronic circuitries, etc..).

A. Process Management

One of the requirements in implementing a complex scenario from the Aeronautical domain was related to the formalization and the “persistence” of both domain knowledge and best practices, such as design activities and approaches. During the Case Study implementation we also learned that this requirement can be supported by the adoption and efficient use of Engineering Environments modeling services, such as the CRYSTAL Platform Builder Modeler.

The advantage in adopting Process formalization and management services is the easier configurability through well recognized process specification standards which is an efficient enabler for interoperability at process level. We applied the SPEM meta-model with a dedicated structured library for the knowledge representation and organization.

One of our achievements was a proposal for the above mentioned library, as shown in Fig. 2 that have been defined on the basis of experiences gained through the past ARTEMIS CESAR project [3]. This library has been adopted to shape and build the applied processes description.

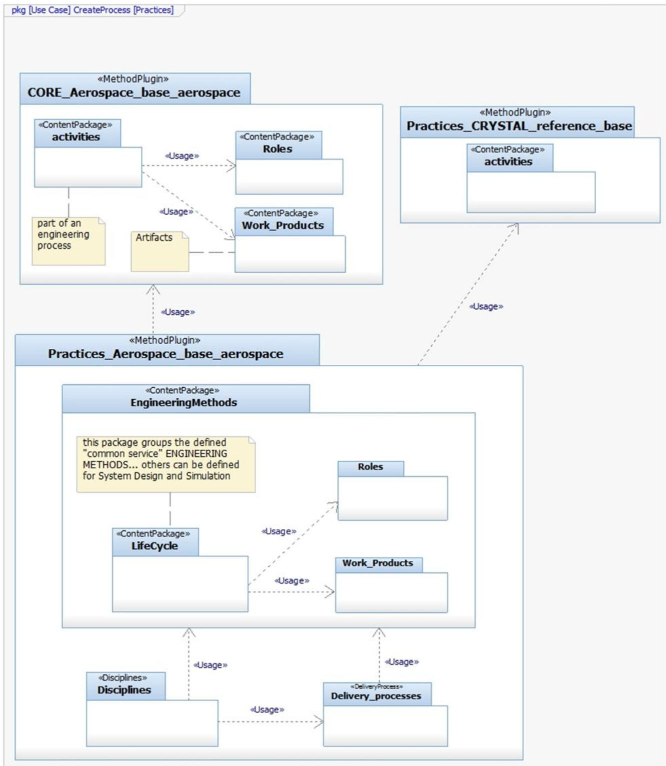

Fig. 2. Structure of the implemented Practices Library

It takes into account the need of modularity and of content re-use at both domain/cross-domain level. The knowledge was partitioned in different “method plug-ins” that can be filled and exported/imported independently. A method plug-in is defined as “generic” and stores concepts that are valid at a cross-domain level, while another is defined as “base-aerospace” and stores concepts to be considered basic for the entire Aeronautical domain.

Practices plug-ins allow the definition and storage of proper, domain specific, delivery processes through the composition of activities and tasks. In this library we defined a structure related to Aeronautical domain.

The approach expects the definition of proper “catalogues” for both activities and methods, where we can define and store all the knowledge that can be used for defining the intended processes. After having defined these items we have proceeded with the composition of the Engineering (delivery) process itself.

This process holds different levels of detail that can be navigated and filled with proper information, such as methods, applicable artefacts and roles for performing activities.
The defined process can also be represented in terms of a dedicated activity diagram that visually shows the activities sequence and relations.

The next step in supporting the Engineering processes enactment is represented by providing support to the users in performing their daily tasks through the “collaboration” and “guidance” capabilities offered by the Collaborative Lifecycle Management platforms.

In order to simplify the acceptance and implementation of new processes and practices by the stakeholders, such as System Engineers and Domain Experts, we have to provide them “context-aware” support and information about the assigned tasks. This means providing access to the right information about the system under development, including the monitoring, development status and process implementation.

We instantiated these solutions in the definition of different Usage scenario developed in our Case Study:

a) Modeling and Analysis: Functional Modeling and Reliability, Availability, Maintainability and Safety (RAMS) Analysis Processes;

b) Product Configuration in the Design Process: Design Review, Configuration Change and Requirement Traceability Processes.

With respect to the SAE ARP-4754A reference process, a simplified formal process has been considered in the RAMS Analysis scenario including only the most significant RAMS activities. The relevant workflow is reported in Fig. 3.

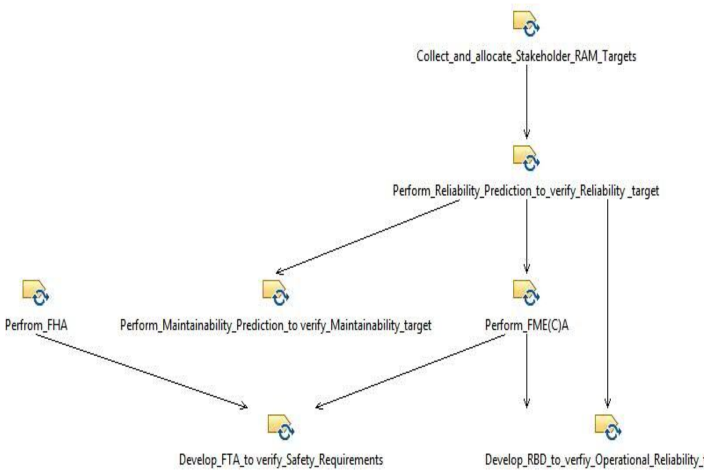

Fig. 3. Formalized RAMS Analysis Process

The Library also describes the “roles”, the “artefacts” and the guidance that are applicable in the process.

With reference to Design Review activities Fig. 4 describes the formal steps to be undertaken by Chief Engineer role while performing the Design Review Process.

These formal practices can be continuously improved and stored in the related library for reuse.

.

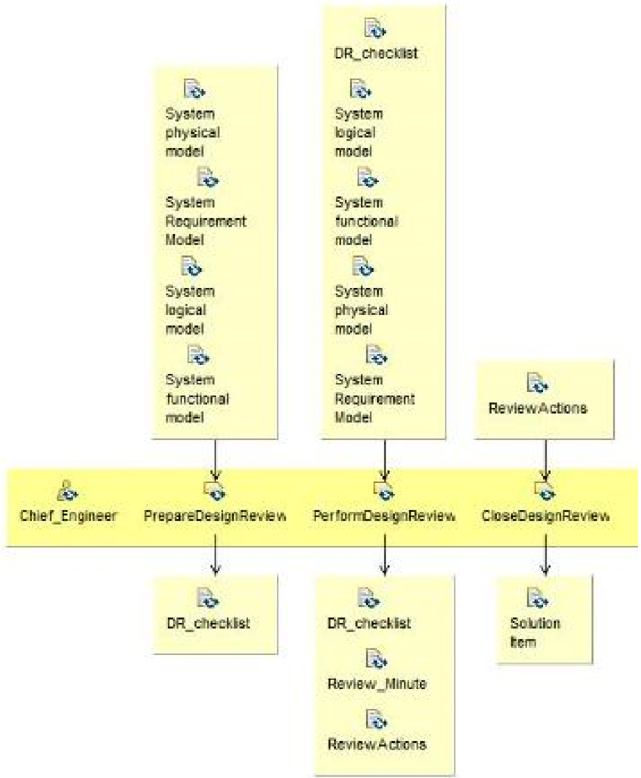

Fig. 4. Chief Engineer role in Formalized Design Review Process

B. Modeling and Analysis

As a first step in the Use Case development we performed the requirement definition and analysis at aircraft level and then at EIMSS and Fuel System level.

As a second step we then developed and animated the functional model of the involved systems in order to assess in a preliminary way the coherence and completeness of the requirements. Different methodologies have been employed for modeling the two systems, in order to assess how they could be applied in different cases:

- for the EIMSS system we have followed the Harmony[1] MBSE workflow, developed by IBM and based on SysML, because we felt it was more suitable for modeling new systems where defining the "control" logic was the priority;
- for modelling the functionalities of more "traditional" systems like the Fuel System, in which the system architecture is more consolidated, we followed a more direct in-house developed Systems Engineering approach, also using SysML.

---

[1] The Harmony workflow from IBM divides the development in 3 main basic phases, Requirement Analysis, Functional Analysis, Design Synthesis.

In both cases, we used the same modeling tool: IBM Doors Next Generation for Requirement Management and IBM Rhapsody/Design Manager for Functional Analysis and Design Modeling.

As third step we validated and refined the initial requirements referring to the output from the second step and we iterated until a reasonable design quality was obtained.

The last goal in our approach was to link the functional model to a specific disciplinary model for RAMS investigations. We used Isograph Reliability WorkBench (RWB) for RAMS Analysis and Isograph Data Link Manager for linking Doors and RWB artifacts.

The process (see Fig. 3) starts from the execution of a Functional Hazard Analysis on the system functional breakdown derived from the MBSE activities carried out during the Functional Requirement Analysis, in order to derive the Safety Requirements. With the evolution of the system design (logical architecture), RAM performance requirements, defined at system level during the preliminary requirement definition phase, are allocated to system design architectural elements (Logical Blocks). As soon as preliminary physical system architectures are developed, preliminary RAMS analysis are carried out for an early verification of relevant requirements to support Preliminary Design Review and eventual changes on preliminary system design. In our implementation, two main OSLC connections have been considered and established between System design modeling and RAMS analysis.

A connection gets data from Design Manager via OLSC and provides them to RWB. On the opposite direction, another connection transfers data from RWB versus Rhapsody to update the relevant model specification, tracking the changes.

Typically, as first step, the functional breakdown is transferred into RWB to perform the Functional Hazard Analysis for Safety Requirements identification.

In a second step, after system functions allocation to a preliminary logical architecture in the Design Environment, the relevant logical breakdown is transferred into RWB in order to allocate System RAM performance requirements (e.g. Mean Time Between Failure, Mean Time To Repair) to logical blocks.

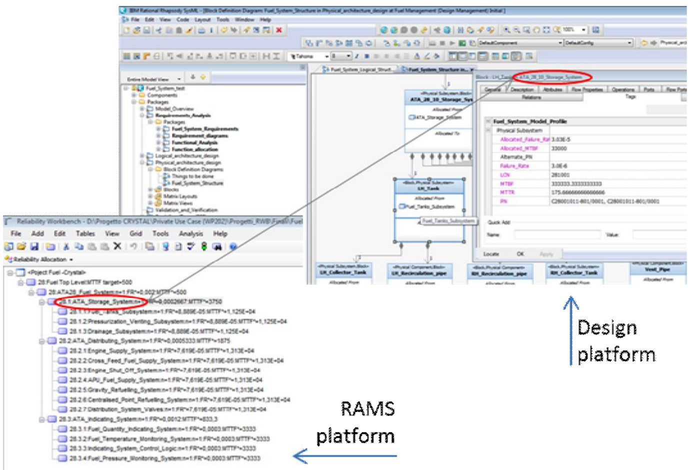

Fig. 5. Synchronization among RAMS Analysis and Design platforms

Finally, after logical architecture allocation to a physical architecture in the Design Environment, the relevant physical breakdown is transferred into RWB together with some identification information (e.g. Part Number).

Here reliability and maintainability predictions are performed and corresponding results are transferred into Rhapsody to verify data previously allocated to the corresponding logical blocks.

C. *Product Configuration in the Design Process*

Product Lifecycle Management (PLM) is a conceptualised vision of interoperable and federated tools where all data across the disciplines in the context of a project are stored and managed along a lifecycle. It may follow that items of the various disciplines can be linked together to form traceability chains and queries can be performed to get impact analyses of the linked data items; thus change management can be supported in a thru-lifecycle manner.

The current aeronautical scenario is still far from this vision; we have specific tools which have their own repository and can exchange data and interface effectively to other tools only to a reduced extent. Product Data Management (PDM) tool has the main purpose of managing Physical Product Data under Configuration Control while Application Lifecycle Management (ALM) is more oriented to the management of the day by day System Design Development Data.

In order to overcome this limitation, we implemented a PLM as an aggregation of interoperable tools, with a PDM directly connected to some design tool and a number of Application Lifecycle Management (ALM) tools integrated in a federated architecture, where the access to information can be provided via linked data and/or federated data-stores.

The OSLC standard specification is the glue which knits it all together, providing the capability of integrating the lifecycle components.

The implementation built by the CRYSTAL project is using:

- IBM Systems and Software Engineering (SSE) platform as ALM repository for managing System Design Development data (from requirements to System Models). This implementation is based on the Jazz platform that provides the basic services shared through Web Applications which deliver all the requested functionalities via OSLC links.
- Siemens Teamcenter as PDM repository for managing Physical Data under Configuration Control associated to Design Data such as Computer Aided Design data, in an OSLC compliant version.

The use of OSLC for the linked data supported the realization of the scenario shown in Fig. 6 in which artefacts in different repositories are linked together.

In addition, the PLM is an environment which can support different Product Views all along Product Lifecycle for System Integration, Configuration Management and Data Management Processes. The different Product Views of a typical Aeronautical Product are the As Required, As Conceived, As Designed, As Planned, As Built and As Maintained views, as shown in Fig. 7.

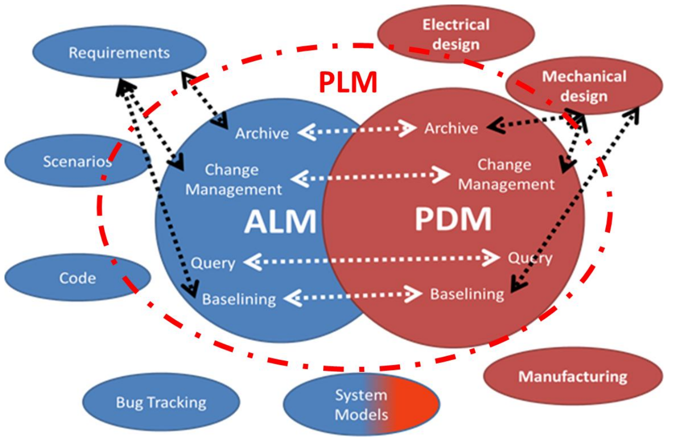

Fig. 6. PLM Environment

An objective of the CRYSTAL project is to improve the interoperability between the ALM platform and the PDM commercial solutions. To minimize data duplication, a specific data model has been defined as illustrated in Fig. 8. In our CRYSTAL scenario implementation, focusing on the Preliminary Design Phase, it has been decided that the management of the System Element artifacts as Configuration Items shall be performed in the ALM environment only. System Elements are instantiated as Blocks of Internal Block Diagrams in the SysML view of the logical/physical architecture, while System Function artifacts are instantiated as Activities in the Activity Diagrams of the SysML view and are allocated on System Elements. They are linked using OSLC to the Physical Elements of the "As Designed view" in the PDM environment.

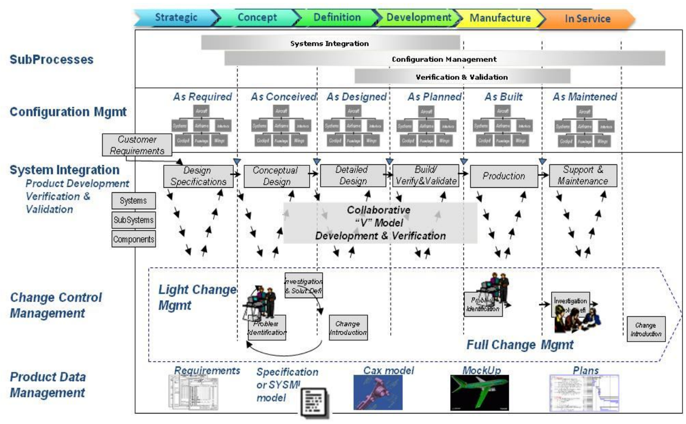

Fig. 7. PLM Product View

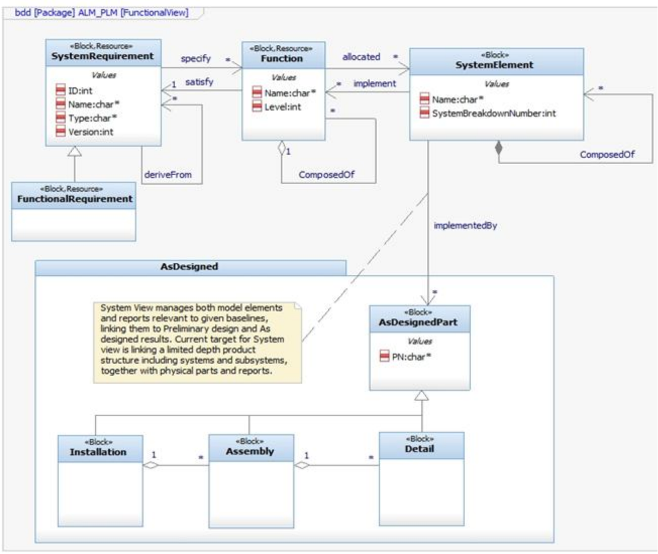

Fig. 8. Data Model for ALM/PDM interoperability

The link joining the data between the ALM and PDM environments is the "Implemented by". In order to validate the improvements, a detailed scenario related to a Change Management process during Preliminary Design Review has been defined and implemented.

D. Design Review and Basic Release Change Scenario

The interoperability between the ALM and PDM environment has been implemented in a Basic Release Change Scenario authorized following the successful accomplishment of a Preliminary Design Review (PDR) in accordance with ALM/PDM reference interactions described in Fig. 9.

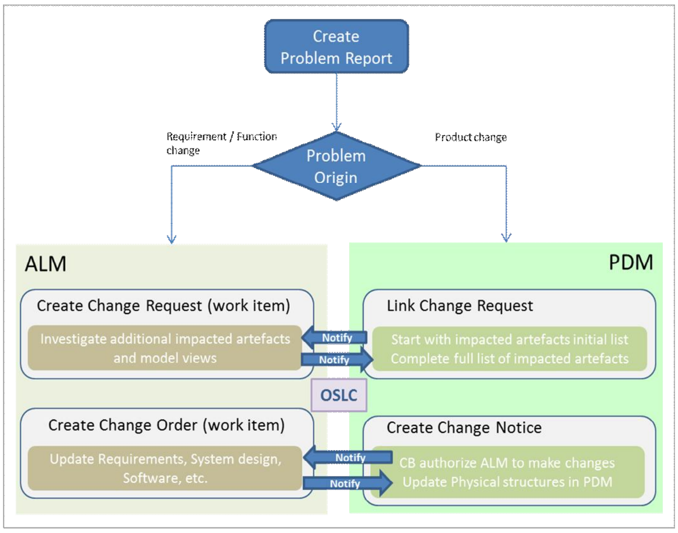

Fig. 9. ALM/PDM Change Process reference interactions

Before Preliminary Design Review, a Change Request is created in order to authorize the Preliminary design (functional and physical). A Request is created in the ALM environment asking for Requirements, Functional and Safety Analyses, while another request is created in PDM environment asking for Preliminary Physical Product definition. A link between the two Change Requests have to be created.

During the same Review preparation, Requirement Traceability and Verification is then performed.

After Review a parallel flow for Change Notice in both ALM and PDM environments is launched in order to release the Requirement Baseline, the Functional Baseline and the Preliminary Product System View Structure.

E. Requirement Traceability

By using OSLC links it is possible to connect key artifacts of the System Development Lifecycle.

Thus, navigation and reporting functionalities supporting important processes such as Change impact analysis can be easily ensured. Our implementation demonstrated how requirements artifacts can be linked via OSLC and traced to a design implementation developed using Rhapsody and MATLAB/Simulink models which were stored in the Design Manager tool.

Data are stored with configurations, called "local configuration" in both DOORS Next Generation and Design Manager. Each local configuration represents a baseline or stream and contains a set of versioned artifacts.

Global Configuration Management assembles configurations for all contributing applications in order to give an overall view of Product configuration.

For traceability and impact analysis purposes, OSLC links must be established among artifacts. In particular the link "Satisfied by an architectural element" was used to connect requirements to design implementation's blocks/functions.

Links can be navigated from requirement to block and vice versa to show connections. Once the objects are linked via OSLC, artifacts can be navigated and traceability/impact analysis reports can be generated as shown in Fig. 10.

IV. CONCLUSION

In our Case Study we had the objective of identifying the needs in terms of methods, practices, data models and tools interoperability required when setting up a structured, integrated and robust System Engineering development environment. Then we had to select those technical solutions proposed in the context of the project research activity that would be beneficial for supporting our needs.

We tested different MBSE approaches when designing the systems that have been the subject of our exercise, taking into account the peculiarities of each of these systems: one is more "software intensive" (EIMSS), the other is more traditional (Fuel). Moreover we associated other discipline analysis to the pure functional analysis. In this paper we detailed the integrated approach with RAMS analysis in order to define and validate, starting from the Preliminary Design Phases, a Logical architecture followed by a Physical one in order to be compliant with emerging standard in the aerospace field, such as SAE ARP-4754A.

Our approaches produced a significant amount of heavily interconnected outputs to be properly managed with dedicated configuration and traceability tools; the seamless traceability between requirements and final output have to be granted all along the project lifecycle through these "new" MBSE models via the "As Conceived" view that links requirements domain (As Required) to the domain of development (As Designed).

In order to enforce the required interoperability we implemented a Change Management scenario in a Preliminary Design Review exploring interactions between ALM and PDM domains and relevant tool chains.

All the processes and tool chains implemented in our Case Study have been successfully formalized by means of the extended SPEM language and the CRYSTAL Platform Builder Data Model.

The implemented solutions proved to be valid and we were able to evaluate them in a near operational context.

Many of the involved tool providers also consider the OSLC based Interoperability a promising technology to be applied. As an Aeronautical company we believe that this technology can reduce the effort in setting up tools interoperation and data duplication/exchange with the use of linked data.

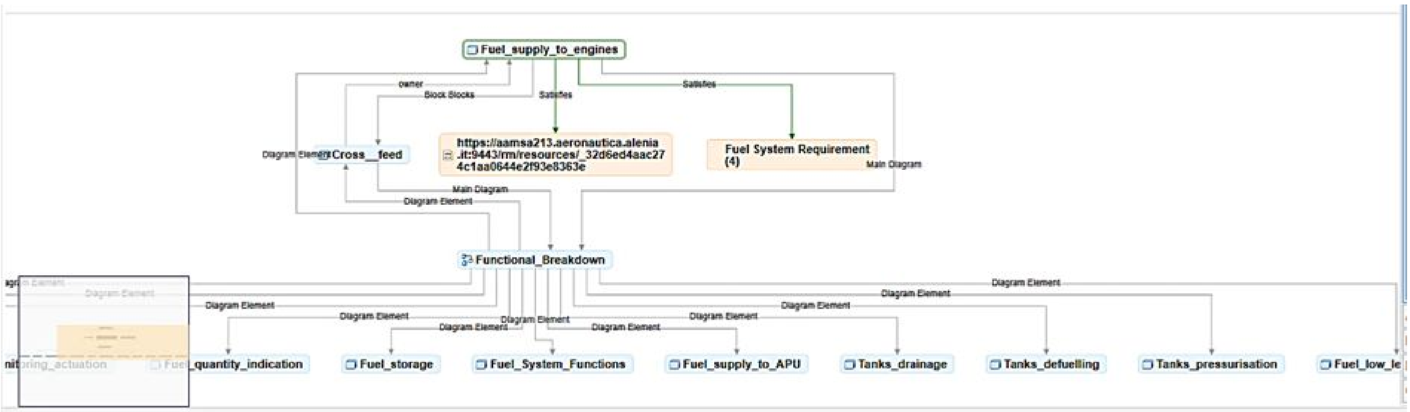

Fig. 10. Impact Analysis Visualization

Acknowledgment

The research leading to these results has received funding from research project CRYSTAL (Critical System Engineering Acceleration), funded from the ARTEMIS Joint Undertaking under grant agreement n. 332830 and from ARTEMIS member states Austria, Belgium, Czech Republic, France, Germany, Italy, Netherlands, Spain, Sweden, United Kingdom.

References

[1] M. Alemanni, E. M. Latino, A.Corallo and E. Valfrè. MBSE Feasibility Study to improve PLM Business Solution System Specification and Design. 22nd Annual INCOSE International Symposium - Rome, Italy - July 9-12, 2012

[2] L. Lo Verde, E. Valfrè, G. Pavan and M. Fioriti. An Alenia Aermacchi study and experience on MBSE application. 22nd Annual INCOSE International Symposium - Rome, Italy - July 9-12, 2012

[3] Odile Laurent, Airbus; Fabien Paganelli,SAGEM; Ivo Viglietti, Alenia SIA ; Stéphane Bonnet , CNRS ; Gérard Cristau, Thales TRT ; Nikolaos Priggouris , HAI; Philippe Baufreton, SAGEM. Customization principles of an aeronautics SLM environment and an illustration on aeronautics Use Cases, the doors management system and the flight control system. Embedded Real Time Software and Systems 2012.

[4] SAE ARP 4754A "Guidelines for Development of Civil Aircraft and Systems" – Rev. A- Revised on 12/2010.

[5] Object Management Group, System Modeling language, v.1.3, 8 June 2012, http://www.omgsysml.org/.

[6] Object Management Group, "Software & Systems Process Engineering Meta-Model Specification Version 2.0," 1 April 2008. [Online]. Available: http://www.omg.org/spec/SPEM/2.0/PDF.

[7] The Eclipse Foundation, "Eclipse Process Framework Project (EPF)" 2015. [Online]. Available: http://eclipse.org/epf/.

[8] OSLC, "Open Services for Lifecycle Collaboration" 2015. [Online]. Available: http://open-services.net/.

[9] Meta-Object Facility, version 2.5, June 2015, http://www.omg.org/spec/MOF/2.5/.