How to Improve PLM Approach Efficiency Based on Knowledge Engineering, Knowledge Management and Semantic Web Technologies Domains?

Bernard Chabot, Philippe Gautreau, Brice Sommacal

How to improve PLM approach efficiency based on knowledge engineering, knowledge management and semantic web technologies domains?

Bernard Chabot[1*], Philippe Gautreau[2*], Brice Sommacal[3*]

[1] Business Consultant   [2] Consulting Director   [3] R&D Engineer
[*] <firstname>.<lastname>@pco-innovation.com
PCO Innovation, Europarc, 1 place Berthe Morisot, 68900 Saint-Priest, France

Abstract : The objective of this article is to show that SEAMLESS, a designed and developed method from PCO Innovation, allows improving PLM approach efficiency by proposing to set up a business/Information System referential. PLM, as a PLM solution, is a software which basically supports product conception and product industrialization business processes (and potentially other upstream and/or downstream business processes). The functional scope may be really wide and the various amounts of subjects are strongly connected to each other. A PLM approach (which seeks to establish a PLM solution in the organization) is basically composed of a program with several projects. Among all PLM approach activities; upstream activities (and specifically specification activities) are absolutely crucial. This kind of approach is, generally, lengthy and complex. Specification activities raise many challenges. They come from a functional positioning (whereas a business one is expected) of user's requirements and also from the deliverables documentary nature. Looking at what is a business/Information System referential; we observe that it has all the ingredients to avoid misunderstanding of user's needs and the nature of the document. SEAMLESS is an assisted method to modelize, capture and restitute (graphically and/or textually) the underlying knowledge domain delimitating a set of activities. SEAMLESS allows implementing any knowledge referential by using a generic approach as well as modular and reusable components. By applying SEAMLESS approach on PLM specification activities, we improve PLM approach efficiency.

Foreword :   In every section of this article, the PLM term used alone designates a PLM solution by default, meaning a technical solution in the field of computing, otherwise known as one (or many) software program(s) / software package(s) interacting together and acting as a single software solution.

1. Introduction

As business consultants, we are solicited by our clients to help them set up a PLM solution in their company. Alongside this progress, most of the time, we encounter some behaviors (independently of their sectors) which prevent to run successfully our mission. That's why; we want to show how our SEAMLESS methodology allows improving PLM approach efficiency.

This article is divided into 4 complementary sections. The first section describes the general nature of the PLM approach (i.e. as a PLM solution) and the specificities of a PLM project. The second section describes how activities which contribute to the specification of a PLM are traditionally apprehended, as well as the pitfalls which are usually encountered and the factors which are at the origin of these pitfalls. The third section describes what a business/information system referential is and how it could solve pitfalls mentioned in the previous section. The fourth section describes SEAMLESS, an assisted semantic methodological approach which allows implementing this kind of referential.

2. The PLM and the PLM approach

This section will be used to define exactly what a PLM is by identifying what the PLM is used for, what PLM speak about, how PLM works and finally what are the reasons for a company to acquire a PLM.

This prerequisite will strengthen the understanding of what a PLM project is and what the main activities in it are. It will also describe the recurring features of a PLM project and show how a PLM project is not really like other projects.

2.1 The PLM at a glance

A PLM, as defined by a "PLM solution", is a software solution which generally comprises a PLM software package, is to an extent configured and/or "customized" and serves as the main component, as well as many other software applications which are more or less integrated with the main component. The ultimate purpose of a PLM is to support and coordinate every activity which is linked to every step in the lifecycles of products and/or services in a company: product innovation needs definition, design, industrialization, production, distribution and others, and in any other case (relative to the nature of the product): operation, maintenance and support, dismantling or even recycling. Depending on certain criteria (product nature, business model type, company strategy, etc.), this end goal will also encompass activities located upstream or downstream of the product lifecycle. A PLM solution will allow a company to design more products and/or design them quicker and/or design less costly products (Miller 2003).

Due to the nature of activities supported by PLM solutions, we may deduct that the major topics (in terms of area of interest) of a PLM are core business concepts used during design and industrialization activities. Other « Core » subjects can be found, such as products informations, but also other subjects which are more or less related to products such as customers, product component suppliers, product requirements, control, verification and product validation methods. These subjects have traditionally been portrayed, described and/or characterized in documents or office software format (Word, Excel, PowerPoint, etc.) such as briefs, functional and technical specifications, assembly nomenclature or mixing recipes, audit files, procedural file control plans, …

It can easily be said that the primary function of a PLM solution is to organize and to bind together the various concepts which are handled by the trades relative to differing views of the "Product". The most known views are:

- "As required" view (describing the uses and requirements for the future product),
- "As designed" view (describing the product's technical reference solution)
- "As to be built" view (describing a local industrial solution for the product)

- "As built" view (describing a real product which has already been built)
- "As maintained" view (describing a real product which has been modified)

It also covers configuration mechanisms and upholds the coherence of these different views, modification tracing, data access control (access rights) and the orchestration of the various tasks which handle these (responsibilities, workflow).

2.2 A PLM project in short

Since a PLM project must ultimately deliver a software application, this type of project is too often considered as an ordinary IT project. The apparent software nature of a PLM project is a trap which must not be fallen into. Even if the advent of a PLM solution in the application landscape represents an opportunity to rehabilitate and simplify (which is not trivial) by eliminating many established legacy programs, it is not an end in itself but rather an opportunity to restructure the company's technical craft. This is why a PLM project must be considered as being a project which is essentially related to the company's core business activity but with a software component to it.

As with any project, a PLM project potentially requires pure project management activities (like planning, resource allocation, budget fixing), project steering activities (like meetings with various committees) and project quality assurance activities (like project quality implementation plan, analysis and project risk analysis activities). As with any project which must ultimately deliver a computer-related solution, a PLM project may potentially require:

- Application specification activities (identification of project issues, determining the project perimeter, formalizing the needs of users, functional and technical specifications, etc.),
- Application construction activities (PLM software package parameterization, specific development and screen customization, associated application interfacing, etc.),
- Application deployment activities (existing software landscape integration, data recovery and interface preparation and execution, etc.),
- Activities regarding the migration of existing application data to the PLM (migrating data identification, preparation, consolidation, migration, etc.),
- Activities surrounding application decommissioning (in order to guarantee that the PLM will be used efficiently and effectively and that "legacy programs" will eventually be removed from the application landscape),
- Change management activities (adapted communication to several groups: project deciders and sponsors, project team members, key and end users, user training, administrator and user support activities, etc.).

Each one of these activities does not weigh on the success of the PLM project and its overall efficiency in the same manner (generally apprehended via the Scope / Schedule / Cost (Newell and Grashina 2004).

A PLM has several recurring features which are almost systematically found in every PLM project. The first recurring feature of a PLM project is its *transversality*, which is to say that the processes which are sustained by the future software application cross over many of the company's functional areas (e.g. marketing, R&D, engineering and design, purchasing, production, logistics) and also many technical subjects (whose list closely depends on the nature of the product which is offered by the company and covers a very large spectrum of components: mechanical, electrical, electronic, pneumatic, software, chemical, organic, living beings, etc.). This implies that the different core business partici-

pants in the company are capable of understanding each other and that the project team has the ability to make sure that they are coherent. The reason is that it is essential to identify and define the core business concepts which are handled through the activities in project perimeter. The second recurring feature of a PLM project is its major impact on the existing application landscape. This implies being able to precisely define which applications store and/or handle data within the project perimeter. The third recurring feature of a PLM project is its *timeframe*. It is generally acknowledged that a PLM project is one which is set over a length of time: many months or even many years are often required to deliver a complete PLM project (which is often organized into several projects). And of course, over this type of period, many kinds of changes are likely to appear (like changes in the company's organization, technological developments, and changes in the project's participants). This implies that the ability to express which needs must be fulfilled within the scope of a project by relying on invariable descriptions of the company's field of expertise is required. The fourth recurring feature of a PLM project is linked to the target application's nature as a software package. Nearly every PLM solution in the market is based on software packages which load a certain number of generic structures and/or mechanisms which must be configured in order to adapt them to the company's context. This implies ensuring that the selected generic elements (structural and/or functional) implementing each core business need are relevant. The reason is that this approach can be used to limit the immediate cost of specific development and the future costs of application maintenance.

Every previously cited feature confirms that a PLM project is first and foremost a company-related project. It is an "in depth" transformation project that affects how the company is run, and even if this transformation strongly affects the application landscape in the company, it's a transformation that must be steered by business and for business. It is also acknowledged that this type of project may become a structuring agent for an organization since, bearing the right tools, it forces the company to follow the rules and processes it has set out for itself.

As a result, a PLM approach is complex because it depends on many factors. In the next section, we will see that this complexity is also integrated into PLM specification activities.

3. The traditional approach and its pitfalls

This section will describe, within the context of a PLM project, how PLM specification activities are traditionally apprehended, which classic pitfalls result from them and what their root causes are. In this section and later in the article, we focus exclusively on *PLM specifications activities* because all others activities depend on the produced informations from this step.

3.1 PLM specifications activities positioning

The root cause comes from the fact that this kind of project is addressed from a *computing perspective*. In many cases, PLM software is selected in the earlier phase of the PLM project (even before user's needs are specified). It may also be justified by the fact that the organization already has a PLM solution in production and would like to change their solution (from minor version upgrade to editor's change). Evidence shows that more and more companies wish to drive PLM projects with an "Agile" method (of the "Scrum"

type) as is the case for the development of functional components such as internet browsers.

Another root cause is when the starting point to capture user's need consists in describing the "as is" situation by focusing on the current practice on the one hand and the evolutions of these practices on the other hand (said "to be" situation). Therefore, the business processes description (which should describe what the business has to do) become the application processes description (e.g. how the business makes things). In others words, instead of capture what the business does (independently of the manner), we capture the way things are realized (or will be), whom realize them, and how they are realized (like resources, tools). It means that user's need capture is done at a procedure level (describe the way to make an activity) instead of at a process level (describe what has to be done). This doesn't allow the organization to take a step back regarding theirs original and genuine needs.

Capturing information at a level which is highly functional and settling for "as is" and "to be" practical descriptions often results in implementing current practices to the upcoming PLM solution, since the description of needs was never done in a way which was focused on what needed to be done (independently to the motivations for doing so and the means to do so). In the end, the company will not have benefitted from the new opportunity to lay flat its core business processes and to reflect on a new way of putting them into practice, in a way closer to best industrial practices and more compatible with the standards in the PLM software package, all of which is provided by the PLM project.

We may also encounter that the various business stakeholders may have trouble to understand each other, because from a functional level, the real profession has been erased. The capture is made from an applicative way whereas it should only be focused on the needs (push from PLM approach and/or selected PLM solution).

3.2 PLM specifications activities deliverables

Specifications activities produce a huge amount of informations which describe business organization in the company, PLM program organization, potential business scope and the one covered by PLM program, existing practices (using legacies system), future practices (using PLM solution), business process "to be", business concepts semantic model, logical data model and applicative cartography. This list is not exhaustive, but, what is important is that these deliverables are produced in corpus of document.

The necessary information to complete activities to specify the PLM solution (define the trade perimeter and the parceling of the project, formalizing user needs, depicting core business processes, specifying system functions, etc.) is almost exclusively managed with office document-type support tools (Word, Excel, PPT, Visio, etc.). Unfortunately, this approach implies that a large corpus of documents is created, with many different documents relating to the same topics (business usage cases, artefacts and business concepts, business roles, application, etc.), viewed under different perspectives (process, responsibility, access to information, etc.) and created by different participants in the project (sponsors, business experts, solution experts, etc.) who often use different jargons.

The fact remains that this body of documents rapidly swells in the project development phase and becomes increasingly difficult to maintain, always requiring greater effort to preserve overall coherence. The resulting pitfall is that an important burden is dragged throughout every phase in the project, which consequentially reduces the productiveness of the project team and increases the risk for error.

Table 1: Criteria to avoid pitfalls

Criteria	Description	Example
Accurate semantic	Don't have the possibility to associate several terms to one concept (Roche 2011)	The same business concept may exist in the corpus of documents but named in different manners.
Translation	For each information, dispose of the translation in one or several languages	Each document has to be translated.
Coherence	Dispose of a consistent corpus of document	The same business concept could be implemented by 2 different applicative objects in the context of 2 projects
Completeness	Can list a specific kind of information	Difficult to list all business roles involved in all projects
Inference	Can easily deduce informations from a corpus of documents	Not possible to deduce inter project dependence from inter processes dependence

4. Business/information system referential

This section allows showing that a business/information system referential approach is well-suited to support PLM specification activities. Essentially by the fact that it allows to avoid a large majority of traditional effects linked to deliverables produced form.

4.1 What's a business/information system referential?

A business/information system referential is a computer application which would allow storing all required elements representing deliverables contents (traditionally produced by PLM specification activities). It allows capturing elements from different levels such as business, functional and/or technical and the entire set of relations that we have to create in the aim of reconstitute the globality of information retrieved in traditional deliverables.

The following table shows examples of *the type of elements* we could retrieved in a business/information system referential:

Table 2: Example of business/information system referential

Referential	Level	Type of element	Example
Business	Business	Business process	Design a product, Industrialize a product, Buy a component, etc.
		Business concept	Need, "As designed" product, "As to be built" product, etc.
		Business role	Designer, Industrializer, Buyer
		Semantic conceptual model	N/A

Information System	Functional	Logical object	Technical specification, purchase order, etc.
		Functional role	Technical specification writer, purchase order validator, etc.
		Logical data model	N/A
	Technical	Application	Team Center, SAP, MySQL, etc.
		Applicative object	TC.OT_DOC, TC.OT_PART, etc.
		Workflows	N/A

Note that what it is named here *a business/information system referential* (in the context of this article about PLM) could be renamed *enterprise referential* in the context of another approach such as Information system urbanization or enterprise architecture. In fact, if instead of taking in account PLM subsection of information system, we could considerate all the information systems of the company, we could execute a similar approach.

4.2 What are the main functionalities of a business/information system referential?

First and foremost, a business/information system referential has a *model*. It means that elements can only be captured in accordance with this model. In other words, it's model duty to allow defining what type of element to capture in the model and what type of relationship to capture between elements. It is also due to this model that user's need specification could be more oriented to the business level

Then, a business/information system referential allows ensuring element *unicity* (independently of its type). In this manner, even if an element is used at several places in the referential, it's stored like a unique occurrence. For example, the same business role may occur in several business processes. The unicity of the information allows identifying that the business role is engaged into several business processes.

Finally, a business/information system referential allows allocating to a particular element *one or several labels* in one or several languages. In this manner, we are able to capture several designations for a same thing (as it is often observed in the same enterprise) and also in a multilingual way. For example, the same business concept is named differently by R&D and marketing team.

Furthermore, we are able to determine new informations (from existing informations) using *inference mechanism*. For example, interaction between projects is induced by interactions between processes

Also, knowledge referential can be considered like a structured database which can be requested to only extract some elements (depending on the context). For example, we may consider these extractions:

- Compact or detailed representation of business process (with or without interims informations)
- Provide adapted representation for each kind of actors in the different projects of a PLM program
- Publish representations for all actors (synthesis)

4.3   The ins and outs of business/information system referential approach

This kind of approach allows facilitating the communication (less ambiguity or misunderstanding) between various team's actors (program, projects, business), respecting the terminology of each party and find a common term for each thing, ensuring each specification element unicity, global consistency and informations exhaustivity, helping actors to apprehend – without minimizing it – the complexity of a PLM demarch and providing to all a view of the PLM specification (with not more, not fewer informations).   However, this approach is time consuming and should be started as soon as the PLM approach is launched.

5. The ins and outs of SEAMLESS

5.1 What's SEAMLESS?

In a few words, *SEAMLESS* is *an assisted methodological semantic approach* to *modelize*, *capture* and *restitute* (graphical and/or textual formats) the knowledge domain supporting a range of bordering activities.
What it is called *knowledge domain* is all of the concepts and relationships underlying activities (independently of the type of these activities).
SEAMLESS is an *approach* in the sense that it suggests a specific course of action to support its specific activities, in the instance of PLM specification activities within the context of a PLM project.
SEAMLESS is *methodological* in the sense that this approach is rooted in best practices which are stable, invariable and repeatable, which adds formality and rigor to each step within this approach.
SEAMLESS is *semantic*, in the sense that this methodological approach gives fundamental importance to specifying what it covers but also because one of the key components of this approach is the use of ontology (e.g. a formal semantic model with concepts).
SEAMLESS is *assisted*, in the sense that this methodological semantic approach could theoretically be implemented without any particular tool, but in practice relies on a software solution which is specifically adapted to sustain it.
It is a key point to note that SEAMLESS has been initially designed to be applied on a knowledge domain (even far away from PLM); and that's why, in the scope of this section, the primary interest will appear to be focused on SEAMLESS approach and then on the particular *application field*: "SEAMLESS applied to PLM specification activities".

5.2 SEAMLESS fundamentals

SEAMLESS mission is to propose solutions which allow effectively equipping any controlled activity by PCO Innovation at an industrial level, quickly apprehending a new knowledge domain, capitalizing self-supporting semantic data (not about tools) and restituting all or some parts of knowledge domain by graphical and/or textual manner.
In the context of the missions (consulting or integration) of PCO Innovation, SEAMLESS allows properly understood communication by having the assurance to speak

about the same thing (by the use of a semantic model), guarantying the coherence and description homogeneity of the scope (unicity informations), converging to the same terminology (differentiation between terms and concepts) (Roche 2011), describing as much relationships as necessary between different elements - semantic network (Sowa 1991) - and identifying all susceptible elements to have an interest for the current application field (exhaustively).

In order to achieve this, SEAMLESS designs, develops and deploys solutions which allow equipping activities of increasing knowledge, management and necessaries informations restitution to execute a mission. These solutions are respectful of knowledge management and engineering best practices, adaptable to context (because they are comprised of generic software components, re-usables), consistent to web semantic technologies, norms and standards and based on open-source tools (like Jena API and Protégé software).

SEAMLESS approach is an iterative approach which allow specifying knowledge domain model, building knowledge model-driven referential (eg. populate the knowledge base with respect to the specified model) and, then, exploiting them using pre-cabled requests which allow representing and visualizing all or some parts of the knowledge referential.

Methodologies used by SEAMLESS directly report to *knowledge engineering* and *management domains*. They allow specifying semantic model of knowledge domain (textually or graphically expressed as a semi-formal mode) and then *ontologies* (expressed as a formal mode – eg. OWL)

5.4 SEAMLESS technologies

Tools used by SEAMLESS directly report to *semantic web technologies norms and standards domain* (Hebler, Fisher, Blace, Perez-Lopez and Dean 2009).
Note that, in the matter of equipment, SEAMLESS may be considerate as Excel. It means that it is a generic tool which allows supporting a huge amount of various *application fields*.
Globally, this is a computer solution only constituted of generic standard components (eg. no links with PLM, PLM project, PLM specification activities or even with enterprise architecture) which are customizable by using *ontology*. This kind of application is named *ontology driven application* (Kühne and Atkinson 2003).
The principle of this kind of application is that the ontology is used to drive knowledge base population with individuals (and links between individuals), which allow to guarantee the fact that the knowledge base respect as a whole all the rules (typing rules, optionality, cardinality, etc.) defined in the ontology.
The generic components allow for example:

- Storing ontology, storing individuals, add, read, modify and delete individuals from the knowledge base.
- Requesting the knowledge base using SPARQL, infering new knowledge from existing knowledge using reasoner like Pellet (Sirin and Parsia 2004) or by creating new data with SPARQL, extracting informations about individuals
- Putting these informations in a good shape (graphical and/or textual reports). Note that we basically use MS Office Excel to visualize tabular data, and yEd for graphical data.
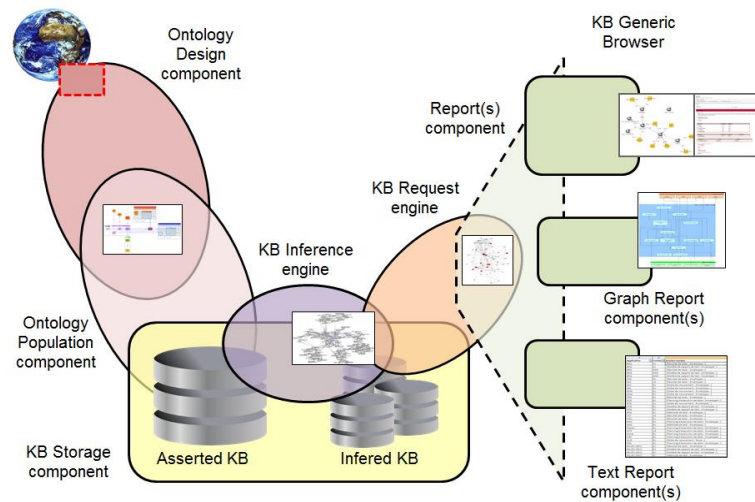- Maintaining the knowledge base (save, export, import, back-up)

Figure 1: SEAMLESS functional architecture

These different components are mainly used with an open-source license, but some of these are commercials. All remaining components are developed by PCO Innovation (like web applications, Protégé plug-ins).

6. Conclusion and perspectives

We have shown that PLM projects are complex and business / information system referential could improve their efficiencies. The key component for reaching this goal is the ontology. Our current ontology is mainly focus on business concerns and information system concerns are just defined by a small set of core concepts. In the future, the ontology should be completed with more accurate concepts in order to cover a whole enterprise architecture scope : business, functional, logical & technical aspects.

Our SEAMLESS approach allows the knowledge base population thanks to an ontology-driven knowledge acquisition process. This is really an added value which allows to easily check the knowledge base coherency and/or completeness and also to produces text and/or graphical reports with a high quality level. But our acquisition module based on Protégé is not really suitable for end-user and not useful for collaborative knowledge acquisition. In the future, we could avoid those pitfalls thanks to a web interface for the knowledge acquisition process.

In short, one complete and relevant enterprise architecture ontology and an ontology-driven web interface for an ontology-driven acquisition process are the 2 mains perspectives for improving our global approach.

References

Allemang D and Hendler J (2008), Semantic Web for the Working Ontologist

Chappelet J.-L and Glassey O (2002), Comparaison de trois techniques de modélisation de processus: ADONIS, OSSAD et UML

Charbonnel G and Dumas P (1990), La Méthode OSSAD pour maîtriser les méthodologies de l'information, Tome 1 : principes

Guarino N (1992), Concepts, Attributes, and Arbitrary Relations

Hebler J, Fisher M, Blace R, Perez-Lopez A, Dean M, (2009), Ed. Wiley, Semantic Web Programming

Kühne T and Atkinson C (2003), Model-driven development: a metamodeling foundation

Le Duigou J (2010), Cadre de modélisation pour les systèmes PLM en entreprise étendue

Miller E (2003), State of the PLM Industry, Proceedings of the CIMdata PLM – Conference, Dearborn, USA

Newell M and Grashina N (2004), Ed. Amacom, The Project Management Question and Answer Book

Partridge C (2005) (2nd ed.), Businesss Object: Re-Engineering For Re-Use

Praxème Institute. http://www.praxeme.org/index.php?n=Main.HomePage?userlang=en

Roche C (2011), Terminologie conceptuelle versus Terminologie textuelle

Sirin E and Parsia B (2004), Pellet : An owl dl reasoner

Sommacal B (2013), Outillage support à une méthodologie de cartographie sémantique

Sowa J (1991), Principles of Semantic Networks

Tricot C (2006), Cartographie Sémantique : Des connaissances à la carte

Vandenbussche P-Y and Charlet J (2009), Méta-modèle général de description de ressources terminologiques et ontologiques