An SOA Based Approach to Improve Business Processes Flexibility in PLM

Safa Hachani[1], Lilia Gzara[1], Hervé Verjus[2],

[1] G-SCOP Laboratory, Grenoble Institute of Technology, France
[2] LISTIC Laboratory, University of Savoie, France
{safa.Hachani, lilia.Gzara}@grenoble-inp.fr
herve.verjus@univ-savoie.fr

Abstract. Companies collaborating to develop new products need to implement an effective management of their design processes (DPs). Unfortunately, PLM systems dedicated to support design activities are not efficient as it might be expected. DPs are changing, emergent and non deterministic whereas PLM systems based on workflow technology don't support process flexibility. So, needs in terms of flexibility are necessary to facilitate the coupling with the environment reality. Furthermore, service oriented approaches (SOA) ensure a certain flexibility and adaptability of composed solutions. Systems based on SOA have the ability to inherently being evolvable. This paper proposes an SOA based approach to deal with DP flexibility in PLM systems. To achieve this flexibility, the proposed approach contains three stages. In this paper we focus on the first stage "identification".

Keywords: PLM system; Business processes; flexibility; SOA.

1 Introduction

To stay competitive, companies are adopting IT solutions to facilitate collaborations and improve their product development. Among these IT solutions, Product Lifecycle Management (PLM) systems play an essential role by managing product data. Furthermore, one of the goals of PLM is to foster collaboration among different actors involved in product development processes [1]. Thus, each PLM system provides (1) a database to store product information and the functions necessary to the management of this stored data and (2) integrate a tool to model, execute and control business processes (BPs) associated to product design. Most PLM systems are adopting workflow management solutions to cope with BPs. Nevertheless, brakes analysis of design processes (DPs) support in PLM systems points out a lack of flexibility for change; justified by the stiffness of formal workflow models and workflow systems used in PLM [2]. However, in a context where the organizations are in a constant seek of balance facing up a highly volatile environments; work methods (BPs) cannot be fixed definitively, especially, when dealing with DPs which are emergent. Furthermore, various hazards intervene during DPs due to external constraints (such as customer requirements evolution, supplier constraints, etc.) and/or internal constraints (such as technical feasibility problems, staff absence, etc.) Thus, DPs are characterized by their instability, relative inconsistencies of the rules that govern them and their incompleteness [3]. As a result, companies face several obstacles, including the limited implementation of new work methods. So, needs in terms of flexibility are necessary to facilitate the coupling with the business reality. Following these two findings: i) instability of DPs models and ii) rigidity of workflow technology, we identify a critical need to deal with DPs flexibility.

Furthermore, software engineering evolved towards new paradigms such as services oriented approaches (SOA) based on services composition. Composition approaches ensure a certain flexibility and adaptability of composed solutions [4]. Systems based on SOA have the ability to inherently being evolvable [5]. Therefore, some standards development organizations have been involved in the development of standards for PLM with SOA [6, 7]. Much attention has been directed towards the use of SOA with PLM [8, 9, 10]. However, the main objective of current studies on SOA in PLM systems is to enable the online integration of heterogeneous PLM systems in order to enhance partner's collaboration (when many companies collaborate in product design and industrialisation). Otherwise, in PLM domain, there is no work that focused on BP agility using SOA. Although much work has been done to date, more studies need to be conducted to deal with BP agility using SOA.

To deal with BPs flexibility in PLM, we propose an approach that makes profiles from SOA. The objective is to specify, design and implement DPs in a flexible way so they can rapidly adapt to changing conditions. In this paper, we present the approach and their three stages and then we focus on the first stage; "service identification".

The remainder of this paper is organized as follows. Section 2 presents our contribution to Sustainability. Section 3 illustrates how dynamic process change happens in PLM systems with a motivation example. Section 4 present existing approaches dealing with PLM and SOA. Section 5 presents the SOA based approach for business process flexibility. Section 6 presents the functional PLM services identification approach. Conclusion remarks and future work are given in Section 7.

2. Contribution to Sustainability

Changing BPs can be considered both at design time and at runtime. Evolvable BPs stands for processes that may be adapted on-the-fly (i.e. while they are enacted and executed). So, we only focus on approaches being able to apply changes without the need to redefine the whole process model. Recent investigations deals with service orientation that promotes light-coupling, services reuse and dynamic composition that cope with sustainability. Loosely coupled services may be organized and composed according to needs and expectations. Dynamic services composition stands for assembling and re-assembling services while the process is executing. So, by proposing evolvable business process based on services technologies, the change can be done without need to redefine whole process model. Moreover, changes will be always done on the unique version of process model with reusing deployed services. It means the system doesn't need to archive unused version. Furthermore, as change is done by reusing deployed services, we may reduce the consumed energy of developing new functionalities and process models. Services expose functionalities as operations independently of their real implementation and can be reused even if the implementation is changing (as consequence, some changes do not affect the services composition).

3 Motivation Example

We propose to use an example of process managed in PLM systems to illustrate the implementation and dynamicity issues of DPs on these systems. This example describes the process of assigning production orders to production workshops according to illustration of figure 1 (a). It includes several activities: visualizing of commands' catalog, creating production order, consulting workshops timetable, assigning production order. In order to be managed on PLM system, this process should be automated. The first step consists on creating the whole functionalities necessary to the fulfillment of the process and storing them on the PLM database. Ones the functionalities stored, workflow template should be defined. Finally, the workflow can be instantiated from the template and executed till its completion [11].

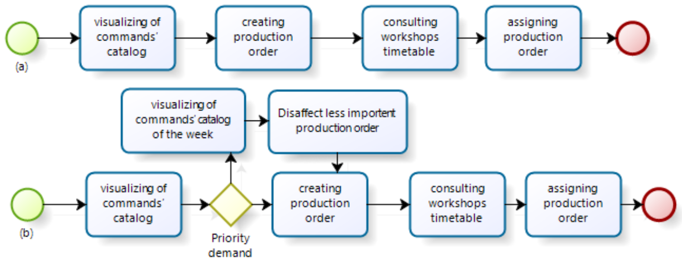


Fig. 1. Change in the assigning production orders process.

As this BP is executed several times, the process model may change due to practical situations. In fact, the actual change is driven by many factors, such as technology shift, internal changes or external changes, etc., and they cannot be anticipated at design time. Some of these factors, for example, external changes (i.e. in order to satisfy suppliers or partners needs), may only cause temporary changes of a business process. While, some factors, such as regulation change, may cause permanent changes. Nevertheless, the BPs in PLM systems are in permanent change due to the dynamic environment justified by the diversity of suppliers, project's partners and requirement changes (for instance, time and cost constraints). For example, some production orders may be urgent; the customer requires a time constraint. The workflow process will evolve to the one shown in Figure 1 (b). Ones process changes occur, new workflows templates are defined and workflows instances are initiated accordingly. Defining new workflows templates requires creating and storing the new functionalities in the PLM database. Thus, the deployment of new functionalities can take much time. In addition it's necessary to handle the previous workflows instances which are initiated from old workflow templates. Most workflow systems, used in PLM consider two steps; before applying the new workflow template, the old instance has to be stopped and restarted according to the new workflow template. Indeed, after restarting the workflow instance some completed tasks have to be carried out unnecessarily (for example, creating production order). To address the above problem, dynamic business process management in PLM system might be brought in as a potential solution. We need a method which facilitates change on business process by (1) reducing the time lost on the deployment of necessary functionality and (2) eliminating repetitive execution of completed tasks.

4 Related Works

Currently there are two initiatives in terms of PLM services, OMG PLM services and OASIS PLCS web services. We studied these proposed standards in order to decline a SOA vision applied in PLM domain and to list existing PLM services. PLM Services is an OMG standard specification [6]. It was developed to implement, operate and support online access and batch operation using several international standards. It's based on PDTnet standard [12] which defines mechanisms to query and traverse instances of the data schema defined for the PLM services specification. This standard proposes a set of services to launch the execution of PDTnet queries which provide the necessary computational functionalities to create, read, update, and delete instances of data. The second standard, OASIS PLCS PLM Web Services [7] is an ISO STEP standard. PLCS is an information exchange standard that provides a number of functional modules. Each one targets a specific area of PLM information. These modules provide services such as searching for PLM business objects or loading information objects. Each one addresses a set of business objects (nouns) and a number of services (verbs) that operate on them. The standard verbs are: Create, Remove and Update. A first analysis of these two standards enables us to conclude that their main focus is related to the problem of online integration of heterogeneous PLM systems to implement their collaborative processes. Furthermore, this study has highlighted a first track for services we target to use to achieve workflow's activities.

Few researches have used these two standards to handle PLM system issues. Erkan Gunpinar [8] have used OMG PLM service standard to interface two PLM systems. His objective is to use PLM Services standard for PLM data exchange via internet in order to speed up collaborative business process done between two heterogeneous PLM systems. With the same objective, Dag Bergsjö [9] proposed a framework to support ECM along with two developed KBE applications that simulate effects of a change in real time, as the product is updated in the PLM system. To do that, he separates the uppermost layers (applications used in the business process to create and process information) and the bottom layers (legacy systems which store information, such as PLM systems) with a connector based on PLM services. Kim [10] used OMG PLM services to introduce MEMPHIS a data exchange middleware that provide common interfaces and enable a centralized integration of multiple PLM systems (server) from any point (clients). All above researches focused on the integration of heterogeneous PLM systems to allow collaboration and did not address the problem of dynamic change of business processes in PLM. So, dynamic business process change remains an unsolved problem in PLM system.

5 Adopted Method

BPs flexibility can be perceived differently. The meaning it takes determines the way to handle the flexibility topic. From our part, BPs flexibility is the fast reactivity to internal and external changes and the easiness to modify BPs models and to set up the new business activity. This perception of process flexibility arises the need to get a method which allows composing evolvable BP models. We propose to enrich the expression of BP models and open the way for modeling by dynamic service orchestration. This supposes that once change happens, we can add to, delete from or replace an activity with another. The challenge here is to address the mechanism needed for a solid implementation of dynamic BP change on a real PLM system.

To fulfil these expectations, we resort to service orientation [4, 5, 13]. SOA is a promising solution answering these expectations since it allows building agile and interoperable enterprise systems. So, if services represent a good answer to technical level issue, they can be used to handle the business level issues: service can be directly invoked by business users and executed as basic steps of BPs. Services can be combined and reused quickly to meet business needs. SOA promotes light coupling between services which can be dynamically combined in order to support agility. The services are defined as providers of reusable business functions in an implementation independent function that is loosely coupled to other business functions [12]. Indeed, SOA organizes the basic functionality contained in the systems on a set of services, which can be combined and reused to meet business needs. This vision therefore allows the construction of new systems by reusing existing services; which called services composition [14]. Thus, the concept of service as previously described as a loosely coupled piece of functionality can be composed and reused to quickly respond to BP change and to achieve the new model without needing to replace it completely and to re-execute completed tasks. So, we resort to service oriented approach in order to propose reusable activities as business services and evolvable business processes as business services composition. A business PLM service exposes a business activity needed to support a business need of a product DPs. It specifies corresponding features (functional PLM service) necessaries to the development of this activity. Afterward, we dynamically compose identified business PLM services in order to implement on a flexible way the articulations of business. Moreover, to insure the alignment between technical level and business level they should be a mechanism that allows execution of identified business PLM services with the same language chosen for the business level. Thus, we propose a set of functional PLM services that represent the whole possible features of PLM system. So, once a new functionality is needed to perform change, operations of functional PLM services can be solicited from the database (service repository) to do it.

In order to achieve this solution, we have to complete three stages. Propose an approach for service identification: steps, techniques and criteria necessary to the identification of business and functional services catalog. Propose services based modeling paradigm for dynamic BP definition. Propose alignment technique that allows moving from business level to technical level. This technique ensures a continuum of transformation from specification to implementation of BPs. In this paper we concentrate only on the first stage; service identification stage.

6 Service Identification Stage

Our objective is to offer two catalogs of services; business PLM service catalog expressing the business needs of product DP and functional PLM services catalog enabling the execution of business needs. To define the services catalogs, we defined the appropriate techniques and necessary steps to achieve service identification stage. Thus, we conducted an initial pass of SOA development approaches.

6.1 Service Identification Approach

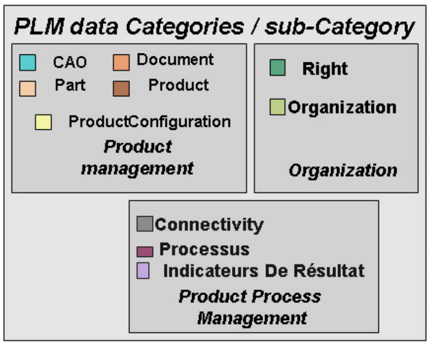
Several approaches are interested on the development of SOA [15-18]. All proposed approaches are based on Service Oriented Modeling Architecture (SOMA) [15]. SOMA is an analysis and design method used for the design and construction of SOA. It focuses on techniques for the identification, specification, and realization of services. All studied work present what activities should be carried out to develop an SOA and how each activity should be conducted. Especially, we focused on service identification activity and studied the proposed service identification techniques. We have identified two complementary approaches used to identify candidate services: Top-Down approach and Bottom-up approach. A Top-Down approach offers a mapping of business use cases, which means to separate the business domain into major functional areas and subsystems. Then functional areas are decomposed on sub-processes, and high-level services. This technique is called domain decomposition. While a Bottom-Up approach analyzes the existing systems to identify low-level services. This approach is called Existing asset analysis.

So, we propose an hybrid approach to identify the two catalogs. We propose to deal with a top-down approach to define business services needed to achieve business process and a bottom-up approach to identify functional PLM services. In this paper we concentrate on the functional PLM service identification method.

6.2 Functional PLM Services Identification Method and Catalog

A functional PLM service is a collection of PLM operations which reflects functions expected by PLM system users. Each operation implements the concept of automated business task and exposes a function of PLM. We propose a bottom-up approach based on three steps for Functional PLM Service identification: (i) Identifying PLM data categories, (ii) Identifying operations of each category and (iii) Grouping identified operations on functional PLM services. Therefore, some criteria are needed to help decide which operations can be grouped together; functional dependencies and process dependencies. Below we detail this approach throw Functional PLM service identification.

To identify the functional PLM services operations, two kinds of information sources were used following the proposed identification approach. On the one hand, two PLM systems were examined to identify the product data categories and their related operations. Thus we have identified a first truck of operations offered on PLM system. For instance, Display product structure, Compare BOMs, etc. On the other hand, we have organized meeting with business expert to validate and enrich the list of identified operations. The results of this two first step are shown on Fig. 2.



Nomenclature	
LinkComponentToProduct	BrowsingUpDocument
LinkDocumentToProduct	MoveComponentFromProduct
LinkDocumentToComponent	MoveComponentFromComponent
UpdateLinkCPQuantity	MoveDocumentFromProduct
UpdateLinkDPQuantity	MoveDocumentFromComponent
UpdateLinkDCQuantity	CompareProductsStructure
DisplayComponentStructure	CompareComponentsStructure
DisplayProductStructure	CompareProductStructure
LinkProductToProject	CompareComponentStructure
BrowsingUpComponent	CopyProductStructure
GetComponentTopParent	CopyComponentstructure

Fig. 2. On the left: Cartography of PLM data categories. On the right: An excerpt from Functional PLM Services operations list.

The last step of the identification approach is to classify the identified operations on functional PLM services. This classification is done by testing functional dependencies and process dependencies criteria between the identified operations by using dependencies matrix in which column headers are the same as the corresponding row header and they correspond to the identified operations. Functional dependencies are those between operations have a common global purpose. In this case each matrix entry *(aij)* correspond to the couple (OperationPurpose i, OperationPurpose j). While, processing dependencies are those between services that are choreographed together to make up a high level service; used together frequently. In this case each matrix entry *(aij)* corresponds to the couple (OperationHighLevelGoal i, OperationHighLevelGoal j). Thus, if an entry has the same couple's element it means that there is dependency between it corresponding row and column operations. To group identified operations, the final decision is done by the superposition of two dependency matrix. After all we have grouped Validate parts, Build Combinations and Develop BOM operations, etc. on a same functional PLM service named Manage Product Configuration. Moreover, we have grouped ReviewDatapack, DistributeReviewDatapack, and NotifyReviewRequestor operations on another functional PLM service named Management of Design Review.

6 Conclusion

In this paper we discussed the problem of BPs flexibility on PLM system. To allow dynamic BP change in PLM system, an approach based on service technology is introduced. This paper proposes to deal with BP model as a business PLM services composition. The challenge here is to react quickly to changes either by replacing some services by other ones or by adding new services to the composition. In order to deal with alignment issues between technical and business level, we propose a service type for each level (functional and business). Business PLM services reflect the business needs of different stakeholders and will be executed by a composition of functional PLM services which meet the functionalities of PLM system. In this paper we concentrated on the identification stage for functional PLM Service. At this stage, a reflexive posture is going to be conducted to define the business service concept, its identification method in order to propose the business service catalog. Moreover, we will address techniques that allow moving from one level to another level to ensure a continuum of transformation from specification to implementation of BPs.

References

1. A. Sääksvuori, A. Immonen, Product lifecycle management, Birkhäuser, 2005.
2. J. Bowers, G. Button, W. Sharrock, "Workflow from within and without," In Proceedings of the Fourth European Conference on CSCW, Kluwer, Dordrecht: 1995, pp. 51-66.
3. P. Green, M. Rosemann, "Integrated process modelling: an ontological evaluation, information systems," Information Systems, 2000, pp. 73-87.
4. M. P. Papazoglou. "Service-oriented computing: Concepts, characteristics and directions," In I. CS, editor, WISE'03, pages 3–12, 2003.
5. K. Kontogiannis, G. A. Lewis, D. B. Smith, "The landscape of service-oriented systems: A research perspective," In Proc.s of Int. Workshop on Systems Development in SOA Environments, 2006.
6. L. Lämmer, R. Bugow, "PLM Services in Practice," The Future of Product Development, pp. 503–512.
7. V. Srinivasan, "An integration framework for product lifecycle management," Computer-Aided Design, vol. In Press, Corrected Proof.
8. E. Gunpinar, S. Han, "Interfacing heterogeneous PDM systems using the PLM Services," Advanced Engineering Informatics, vol. 22, Jul. 2008, pp. 307-316.
9. D. Bergsjo, A. Catic, J. Malmqvist, "Implementing a service-oriented PLM architecture focusing on support for engineering change management," International Journal of PLM, vol. 3, 2008, pp. 335-355.
10. S.R. Kim, D. Weissmann, "Middleware-based Integration of Multiple CAD and PDM Systems into Virtual Reality Environment," Computer-Aided Design & Applications, vol. 3, 2006, pp.547-556.
11. S. Khoshafian, M. Buckiewicz, "Groupware et workflow", InterÉditions, 1998.
12. R. Credle, M. Bader, K.Brikler, M.Harris, M.Holt, Y. Hayakuna, "SOA Approach to entreprise integration for product lifecycle management," Oct. 2008, pp. 66-80.
13. T. Erl, "Service-Oriented Architecture (SOA): Concepts, Technology, and Design," Prentice Hall, 2005
14 B. Manouvrier, L. Ménard, Intégration applicative EAI, B2B, BPM et SOA, Hermès Science, 2007.
15. A. Arsanjani, A. Allam "Service oriented modeling and architecture for Realization of an SOA,"On the proceding of IEEE int. conf. on service computing (SCC'06), IEEE, 2006.
16. M. P. Papazoglou, W.-J. Heuvel, "Service-oriented design and development methodology," Int. J. of Web Engineering and Technology (IJWET), 2, 2006, pp. 412-442.
17. O. Zimmermann, P. Krogdahl, C. Gee, "Elements of Service-Oriented Analysis and Design," 2004, available at: http://www.ibm.com/developerworks/library/ws-soad1/.
18. S. H. Chang, S. D. Kim, "A Service-Oriented Analysis and Design Approach to Developing Adaptable Services," In the IEEE International Conference on Services Computing (SCC 2007), pp. 204-211.