Multi-level Service Approach for Flexible Support of Design Processes

Safa Hachani, Lilia Gzara, Hervé Verjus

Multi-level service approach for flexible support of design processes

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

[1]G-SCOP Laboratory, Grenoble Institute of Technology, Grenoble, France
{safa.Hachani, lilia.Gzara}@grenoble-inp.fr

[2]LISTIC Laboratory, University of Savoie, Annecy, France
herve.verjus@univ-savoie.fr

Abstract. The need to answer quickly to new market opportunities and the high variability of consumer demands tend industrial companies to review their adopted organisation, so to improve their reactivity and to facilitate the coupling with the business enactment. Therefore, these companies require agility in their information systems to allow business needs scalability and design process flexibility. We propose in this paper, the business activities as a service based on the service paradigm and whereas a design process is made of agile services orchestrations. We discuss the interest to use a service-oriented approach and propose a layered architecture for design process enactment.

Keywords: Design process, BPM, agility, PLM, SOA, service orchestration, MDA

1 Introduction

Traditional business process management (BPM) tools with a fixed process' structure are no longer adequate to meet the continuing evolution of the market, enterprises' organization and customers' requirements. These tools tend to be inflexible and time consuming to change. In fact, in a context where organizations are in a constant seek of balance facing up to more and more constraints of the competitive environment, working methods cannot be fixed definitively. Especially in product engineering field; design processes are emergent because of the creativity aspect in engineering and constantly evolving because of the fuzzy and changing expression of customers' needs. Furthermore, unpredictable events may occur during design processes due to external constraints (such as sub-contractor or supplier constraints, etc.) and/or internal constraints (such as delay constraints, staff/resources availability, etc.). Some of these factors, such as satisfying suppliers' constraints, may only cause temporary changes of design processes. While others, such as regulation evolution may cause permanent changes. Reflecting these changes on time represents an ongoing challenge, especially for Product Lifecycle Management (PLM) systems editors since that design processes are mainly supported by these systems. As a result, companies face

some obstacles, including the limited implementation of new working methods. Needs in terms of rapid and automated support of business operations are necessary to reflect such changes. To address this issue, Service Oriented Architecture (SOA), already used as an enabler of software flexibility, is considered in this study to explore its potential as an enabler of business flexibility. SOA can be seen according different perspectives (business, architecture or implementation) [1]. Our research falls into the first perspective (business) and considers SOA as an approach that supports integrating the business as linked services. Then, services are seen as repeatable business tasks, accessible independently of implementation or transport [1].

Several researches have attempted to solve the problem of BPM tools rigidity by proposing several modeling approaches [2] [3] [4] [5]. Some of them propose to enrich the expression of business processes in order to meet flexibility needs. Some others only address implementation level. Defining an approach for enhancing flexibility at various abstraction levels with a continuum of transformation remains an issue. Based on the findings outlined in literature, we conclude that flexible business processes require specific methods for their design and implementation. Thus, the expected approach is the one in which not only the behavior of activities are not defined a priori but also the relations between activities. Like Boukadi [5], we resort to service based solution in order to map design processes to service-oriented solution. Service-oriented approach, deployed at different abstraction levels (business, functional, software), can promote a flexible support of design processes. The idea is to propose reusable activities as services and evolvable product design processes as services composition. The concept of service defined as providers of reusable functions can be composed and reused to quickly respond to design process changes. That supposes once changes occur we can add to, delete from or replace one service by another one. Indeed, the generalization of SOA to information systems (and thus, the design and requirements analysis layers) would allow the definition of design processes and their implementation by reusing existing services.

The aim of this paper is to propose a methodology specifying and implementing a design process by services composition. This paper is structured as follows. The second section presents the approach we retained, which is based on service paradigm. Then, we present the proposed methodology to identify services (section 3). Section 4 presents the approach for specifying and implementing processes by services' composition. The final chapter concludes this paper.

2 The Proposed Service-Based Approach

The aim of a design process model is to depict interactions between business partners and model their corresponding activities. In past decades, these processes could operate in relatively stable and predictable environments. Now a product design process (named PDP in the following) may not remain steady due to the business environment. That's why we need process flexibility. Process flexibility depends on the

easiness to modify design process model and to set up the new business activities. This perception of process flexibility arises from the need to have a method, which allows composing evolvable design process models. In other words, flexibility requires processes made of piece of functionalities that can work together and that can be quickly reconfigurable. The challenge here is to address the mechanisms needed for a solid implementation of dynamic design process change on a real PLM system.

In order to address the problem of design process rigidity in PLM systems, we propose to resort to SOA and to enrich the formalization of design process models and open the way for process modeling by dynamic service composition. Thus, we should have a set of product design services (named PDSs in the following) that expose the business activities of the industrial engineering domain needed to support PDPs. Afterward, we dynamically compose the necessary identified PDS in order to enact the articulations of design process. This process can be materialized by an orchestration of PDS [6]. In fact, we propose to use service as a means for describing the business operations needed to support the design processes. These loosely coupled services may be composed in order to enact in a flexible way the articulations of business. This process is called services orchestration [7]. SOA makes this possible; it allows decomposing processes and business activities into independent business services. Dynamic services orchestration stands for assembling and re-assembling these business services while the process is executing. Thus, service can be composed and reused to quickly respond to design process change and to achieve the new model without needing to replace it completely. Moreover, as services expose operations independently of their real enactment, they can be reused even if the enactment is changing (as consequence, some changes do not affect the services orchestration). This supposes that once a change happens, we can dynamically add to, delete from or replace a service operation with another one. The main characteristic of this service-based approach is that it provides a flexible process structure, which provides the necessary agility to face changing conditions and unpredictable situations.

As we have discussed above, business agility is the fundamental business requirement. So, the entire PLM system, starting by IT level, must support business agility. It's important to remember that design processes are very dependent on the information technology that supports it. So, the business also depends of the IT flexibility. So, we propose the whole system reconsideration and not only a business level reconsideration.

To insure the alignment between technical and business level, there should be a mechanism that allows execution of design process with the same language chosen for the business level. So, we propose a service type for each level of the organization. The different levels that we consider are justified by the reality of enterprise information system. On the one hand, we find the organizational IS. It consists of information (business objects) and actors whose act on this information through their work methods (business activities). On the other hand, there is the system infrastructure or Computerized IS. The computerized IS consists of an organized set of items, machines and applications. It allows the implementation of the working methods of the companies and the organization of their information. Moreover, the actors of the or-

ganizational IS use the computerized IS through the interfaces provided by its tools. Therefore, there are three levels in the enterprise IS: the business level associated with organizational IS, the technical level associated with Computerized IS and the functional level associated with the interfaces of the Computerized IS. So, by analogy to this classification, we propose three layers of services. First, we propose a catalog of PDS. A PDS is a collection of PDS operations. A PDS reflect solutions to some business needs of a product design domain. In fact, each PDS operation partly reflects a business activity typically presented in design processes. These PDS operations will be used and composed by the business process designer to build design processes. Secondly, we propose a catalog of Functional PLM Services, which (i) ensures alignment between business and technical levels and (ii) aims to be independent of any PLM system. A Functional PLM service is a collection of Functional PLM services operations. The services operations of this layer represent all functions of the PLM as seen by PLM users independently of any existing PLM tool. A set of functional PLM services operations support a PDS operation. They will be used by process performers and composed to achieve the PDS. In other words, once a new business activity is needed to perform a change at the business level (in a design process), a PDS operation is invoked and added to the orchestration and thus operations of functional PLM services can be solicited from the repository to do it. Finally, we propose a set of technical PLM services that allow the real implementation of functional PLM services. These Technical PLM services cover all technical operations carried out in a PLM system and they will be dynamically orchestrated during the enactment of design processes. They are intended to PLM systems editors. This distinction between functional and technical PLM services allow the reuse of process models, defined only in terms of business and functional PLM services, on different PLM systems. Indeed, once a process deployed on a new PLM system, we have to make correspondence between functional PLM services and technical PLM services.

To make the transition from one level to another one, our conceptual approach is based on a Model-Driven Engineering one (MDE) (Fig. 1). Starting at the top with the business level (Computer Independent Model) it is primarily concerned by the design processes that comprise the day-to-day activities of the enterprise. It contains also the business services (PDS), which allow composing the design process. Moving down a level (Platform independent Model), we see the functional PLM services and orchestration fragments that can be predefined or user-defined. Predefined orchestration fragments define the recommended functional PLM services orchestration, which allows fulfilling a given PDS operation.

Functional PLM services are non-specific to any PLM system. That's why services can be mixed and matched into meaningful combinations without concern for what systems are actually performing the work. Down this, we find the technical level (Platform Specific Model), which contains the technical services. This layer forms the API (core functions) of the PLM system used.

In order to achieve the proposed approach, first we have to propose the services catalogs. Then we have to express design processes as service orchestration. Finally we should propose alignment techniques that allow moving from business to technical

level. In the rest of this paper we concentrate on the two first steps of our approach (services identification and design processes definition as service orchestration).

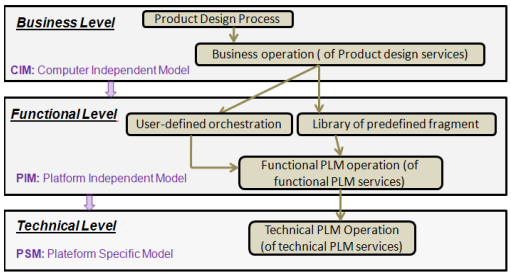

Fig. 1. Conceptual Architecture of the Proposed Approach

3 Service Cartography

As we have discussed above, we aim to offer three services catalogs. A PDS catalog, which expresses the business, needs related to design processes. Functional PLM services catalog enabling the execution of PDS through a functional PLM services orchestration. Finally, technical PLM services catalog enabling the implementation of functional PLM services. The third catalog is dependent on which PLM system we use, so we concentrate only on the business and functional service catalogs. In a previous work the proposed service identification approach is described [8]. Thanks to this approach, fifteen PDS and nine functional PLM services are identified (Fig. 2 and 3).

PrototypeRealisation
CreatePrototype
LaunchPrototypeCreation
DistributePrototype
AskForThePrototype

DesignDepartementDysfonctionnalAnalysis
ElaborateDysfonctionnalAnalysis
DistributeDysfonctionnalAnalysis
EvaluateDysfonctionnalAnalysis
Val;idateDysfonctionnalAnalysis
DysfonctionnalAnalysisElaborationRequest
ConsultDysfonctionnalAnalysis
DysfonctionnalAnalysisEvaluationRequest
DysfonctionnalAnalysisValidationRequest

TestDefinition
ElaborateTestPlan
DistributeTestPlan
EvaluateTestPlan
ValidateTestPlan
TestPlanElaborationRequest
ConsultTestPlan
TestPlanEvaluationRequest
TestPlanValidationRequest

BomManagement
CreateBOM
DistributeBOM
EvaluateBOM
ValidateBOM
BOMElaborationRequest
ConsultBOM
BOMEvaluationRequest
BOMValidationRequest

Fig. 2. An expert from product design service catalog

BOM Management	Compenent Management	Product Management	Document Management	CAD Management
LinkComponentToProduct	DefineComponent	DefineNewProduct	NewDocument (type)	RetreiveASMStructure
LinkDocumentToProduct	DefineComponentData	DefinePartData	DefineDocumentData	RetreivePartQuantity
LinkDocumentToComponent	CreateFromComponent	CreateFromProduct	AddDocumentAttachment	VisualizeCAO
UpdateLinkCPQuantity	UpdateComponentData	UpdateProductData	NewDocumentFromModel	
UpdateLinkDPQuantity	NewComponentVersion	NewProductVersion	EditDocumentAttachement	
UpdateLinkDCQuantity	NewComponentRevision	ChangeProductStatus	CompleteDocumentData	
BrowsingUpDocument	DisplayComponentEditor	DisplayProductAtachment	ExportDocumentToModel	
CompareProductStructure	DisplayComponentHistoric	DisplayProductEditor	SearchDocument	
CompareComponentStructure	DeleteComponent	DisplayProductHistoric	PrintAttachment	
CopyProductStructure	GetComponentVersion	DeleteProduct	PrintAttachmentToPDF	
CopyComponentstructure	LockComponent	GetProductVersion	ExportDocumentToModel	
GetComponentToParent	UnlockComponent	LockProduct	DisplayDocumentEditor	
		UnlockProduct	DeleteDocument	
			GetDocumentVersion	

Fig. 3. Expert from Functional PLM service catalog

4 Design process as a service orchestration

Specifying and implementing one design process consists of:

- Discovering, from PDS catalog, the set of business processes and services' operations that meet the various process modeling requirements. Services are selected based on services' operations labels which allow expressing the business process aims. A business process is seen as a set of steps that contribute to the achievement of the process' aim. Each of these steps has an elementary aim. Selection of services' operations is made regarding these steps' aims. Then, in order to identify the suitable services' operations for each step aim, ontology of product design actions and business objects (product design objects) is proposed to the user. By matching the selected design action with the selected design object, one or more service' operations are proposed to support each process' step. This assumes that user translates each step' aim in pairs "design action/design object". We should highlight that sub-processes which are already defined as orchestrations (i.e. orchestration fragments) can participate in defining the global process. More details on design actions / design objects ontology are given in [9].
- Composing the business orchestration fragments and/or the selected business services'operations, by specifying the set of control flows which link these elements. Control flows allow expressing chaining between operations.

The design process is seen as a composition of orchestrations and services' operations from the design field (PDS). The service orchestration is used to assemble services to achieve a particular goal, through primitive controls (loops, test, exception handling, etc.) [10] or control flow. Regarding the specificities of product design field, service composition approach adopted in this work should:

- be simple at conceptual level, to facilitate the use of various concepts for the process manager and provide a process description which is independent of any platform and understandable to the design actors.
- rely on a manual service composition so that the process manager retain control over the process (decide what services and routines are best suited to the business context).
- allow the dynamic evolution of the services' composition in order to automate the process but also to make changes at the structural level of the process.

The central element of the design process modeling is process decomposition unit, namely service' operations or sub-processes defined by service composition and exposed as full services full (orchestration fragments). Other concepts are needed for the modeling and implementation of the design process. Figure 4 illustrates the concepts that we use for composition approach, using a class diagram described in UML and called orchestration meta-model. This generic orchestration meta-model provides a comprehensive overview of the necessary concepts to express the services composition but also the structural relationships that may exist between these concepts. We

define in the following the various concepts included the metamodel and show how this generic metamodel can be used at all levels of our approach.

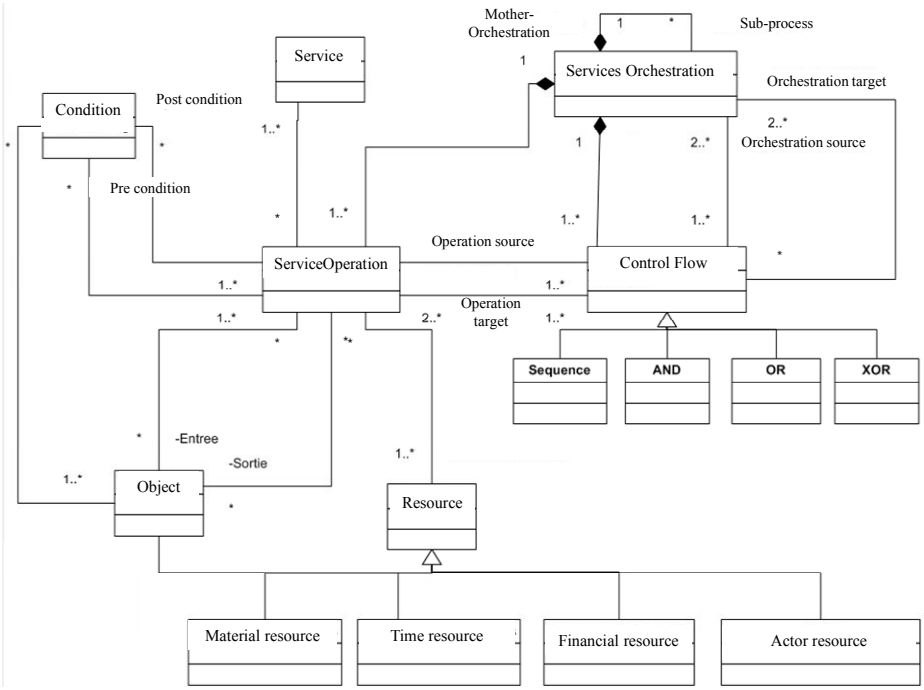

Fig. 4. Generic orchestration meta-model

The central element of this meta-model is the "Services Orchestration" class. This concept corresponds, at the business level of our approach, to the design process model expressed as a composition of services. At the functional level of our approach, it corresponds to the composition of functional PLM services which allow implementing the business level services.

As we can see in Figure 4, the concept of orchestration can be composed by other orchestrations or by instances of the "ServiceOperation" class. This means that the orchestration can call other orchestration fragments already defined. For example, at business level, the design process (i.e. Services Orchestration) may involve sub-processes which are already defined. The orchestration can also be composed by service operations belonging to the services we proposed previously. Structural relationships that we defined between instances of the "Services Orchestration" class and the "ServiceOperation" class facilitate the task of the process manager and simplify the final model. This proposal will allow defining processes (Services Orchestration) which are complex but unambiguous, because the complexity of the sub-processes is masked by the orchestration fragments composing the overall orchestration. Resources element includes resources family that may have an instance of the

"ServiceOperation" class. In fact, so they can be executed, instances of the "ServiceOperation" class may need some resources. For example, at the functional level of our approach, the PLM functional operation related to the design of a 2D model will need an instance of the "Actor resource" class who can make CAD models. Another kind of resource necessary for our approach is the "Object" class. Instances of this class represent material or information objects handled or generated by design actors during the execution of the "ServiceOperation". Instances of this class are characterized by their states (created, modified, validated, archived, etc.) which express pre / post conditions on instances of the "ServiceOperation" class in terms of input and output objects. These structural relationships between ServiceOperation, Object and Condition classes can provide flexibility in the design process. The design process ongoing can be altered by changing the states on the objects related to the service operations participating in the orchestration.
Finally, the order for executing the service operations and orchestrations involved in the composite orchestrating is controlled using a set of Control Flow. There are many kinds of control flows in the literature [11] [12] where each connection type has a distinct semantics. We consider the commonly used connections types : sequence flow (to represent the execution order of services participating in the composition of services), AND (to create and synchronize parallel flows linking service operations, business orchestrations of business or service operations with business orchestrations), XOR and OR (to specify the possibility of multiple choice between several alternatives, exclusive or not). The use of control flows allow to express the sequence of service operations and orchestration fragments. The source and target of each flow control can be instances of two classes: ServiceOperations or Services Orchestration. The structural relationships between "ServiceOperation", "Services Orchestartion" and "Control Flow" classes allow providing flexibility at the structural level of orchestration (i.e. the process). Indeed, by changing the source and target of control flows which define the process structure (i.e. orchestration), service operations or orchestrations fragments can be replaced by other service operations or orchestration fragments.

As we can see through Figure 4 and the explanations that follow, the composition approach is generic and independent of any platform or execution system. We also specializing the generic orchestration meta-model at every level of our approach.

5 Discussion and Conclusion

In this paper we discussed the problem of design processes flexibility in PLM system. Product Design processes are emergent and design actors cannot deal with existing technical solutions. Current modeling approaches dealing with business process flexibility were discussed and analyzed. The assumption made is that flexible design processes require specific methods for their design. Our contributions respond to the limitations and problems described above by providing a methodological approach that aims to provide design process flexibility by adhering to service orientation. A service-based approach was introduced to address dynamic design process

changes. This approach presents the design process model as a PDS orchestration. In this case the process refers to business behavior in which all steps are PDS operations. Thus, a PDS can be invoked to perform a given step of a design process. The challenge here is to react quickly to changes either by replacing some services by other ones or by adding new services to the orchestration. In order to deal with alignment issues between technical and business level, we first proposed a service type for each level. Then, we proposed an orchestration model in each level (business, functional and technical) and a mean that allows transforming the business orchestration model to a functional orchestration model by adhering to MDE techniques. Finally, according to the used PLM system, a mapping between the functional services of the functional orchestration model and the technical services of the PLM system should be done using the same deployment techniques. In this paper we presented the business and functional services catalog. We also defined the meta-model needed to orchestrate PDS. Then, the proposed service-based approach supports changes at both levels: process model (process model is changed by composing reusable PDS and orchestrations' fragments) and process instance (process instance is changed by composing reusable PLM technical services obtained through transformation of business services into technical services). Support of emergent process evolution has not been addressed in this paper. To allow managing "on the fly" emergent or ad-hoc processes, the composition approach should include mechanisms to allow deferred services selection. Techniques that allow moving from one level to another one have not been addressed either. To do so, we proposed a conceptual architecture based on a Model-Driven-Engineering approach [13]. We defined a mapping meta-model between business and functional levels, which we named business deployment meta-model. According to the business deployment meta-model and the business orchestration model, executing a set of mappings rules can generate the functional orchestration model. As far as, we defined a mapping meta-model between functional and technical levels that we named functional deployment meta-model. Using the same deployment techniques, the technical orchestration model can be generated based on the functional orchestration model and the functional deployment meta-model. The distinction between functional and technical PLM services shows the genericity of the proposed approach. In fact, an enterprise can change her PLM system or even have multiple PLM systems. The advantage of our approach is that the company can uses the same business orchestration model (and the same business deployment model), but execute many functional deployments model according to the used PLM systems. Indeed, once a process model deployed on a new PLM system we have just to execute the right functional deployment. In contrast, the limitation of our approach is that we define the deployment only on a top-down manner and we consider that all functional PLM services operations have a corresponding list of technical PLM service operations.

Acknowledgements: The authors would like to thanks the Region Rhône-Alpes for financial support of this research work.

References

1. Credle, R., et al.: SOA Approach to entreprise integration for product lifecycle management. IBM International Technical Support Organization (2008) 66-80.
2. Bessai, K., et al.: Context Aware Business Process Evaluation and Redesign, In: Int. Workshop on Business Process Management, Design and Support. Int. Conference on Advanced Information Systems, Montpellier, France (2008) 407-414
3. Zhao, X., Liu, C.: Version Management in the Business Change Context. Int. conference on Business Process Management, Brisbane, Australia (2007) 198-213
4. Lezoche, M., Missikof, M., Tinini, L.: Business Process Evolution: a rule-based Approach. Int. Workshop on Business Process Management, Design and support, at Int. Conference on advanced Information Systems, Montpelier, France (2008). 407-414
5. Boukadi, K., et al.: CWSC4EC : How to employ context, web service, and community in enterprise collaboration. The 8th Int. Conference on New Technologies of Distributed Systems, Lyon, France (2008)
6. Erl, T.: Service-Oriented Architecture (SOA): Concepts, Technology and Design. PTRUpper Saddle River, NJ, USA: Prentice Hall (2005)
7. Manouvrier, B. Menard, L.: Integration applicative EAI, B2B, BPM et SOA. Hermes Science – Lavoisier (2007)
8. Hachani, S. Gzara, L. Verjus, H. : Support of Business Processes Flexibility in PLM Systems Using a Services-Based Approach. Int. Conference on Industrial Engineering and Systems Management (IESM'11), Metz, France (2011)
9. Hachani, S. Verjus, H. Gzara, L.: Support of Product Design Processes Flexibility in PLM Systems using a Service-based Approach. International Journal of Services operations and Informatics (IJSOI) vol. 7 (4), ISSN 1741-5403 Ed. Indersciences (2012)
10. Kellert, P. Toumani, F.: Les services web semantiques. Revue I3 « Information-Interaction-Intelligence » (2006)
11. Russell, N. Ter Hofstede, A.H.M. Van der Aalst, W.M.P. Mulyar, N.: Workflow Control-Flow Patterns: A Revised View. BPM Center Report BPM-06-22, BPMcenter.org, Available at (http://www.workflowpatterns.com/patterns/control/) (2006)
12. Van der Aalst, W.M.P. Ter Hofstede, A.H.M. Kiepuszewski, B. Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases, vol 14(3) (2003) 5-51. Available at (http://www.workflowpatterns.com/patterns/control/)
13. Schmidt, D.C.: Guest Editor's Introduction : Model-Driven Engineering. Computer Journal, vol 39(2) (2006) 25-31