Actionable Meta Models to Support Inter Organizational Business Processes Modeling for e-Services

P. Radha Krishna[1] and Murali Mohan Narasipuram[2, *]

[1]SET Labs, Infosys Technologies Limited, Hyderabad, India.
[2]Department of Information Systems, City University of Hong Kong,
83 Tat Chee Avenue, Hong Kong.
radhakrishna_p@infosys.com, ismohan@cityu.edu.hk

Abstract. E-services usually span multiple organizations and evolve in time-domain due to the changes in the environment pertaining to each organization. Enacting composite e-services over evolving inter-organizational business processes is a challenging task. In this paper, we highlight modeling of business processes for e-services and present enactment of the e-services with the help of meta models in order to cope with the changing needs. We also describe high-level framework to enact e-services through actionable meta models.

Keywords: Meta models, Business process modeling, Run-time environment

1 Introduction

Modeling e-services is an important task in business processes specifications and execution. A single model may not be enough to design e-services when it spans multiple organizations. At the same time, it is hard to describe a generic model for any kind of e-services. This necessitates an abstract level of modeling, that is meta-modeling, which helps in instantiating context-specific model instances to suit the needs of e-services. It also provides required facilities to support functionality for adapting the model to new requirements.

In the domain of business process modeling, meta-models take a leadership role in defining how the data and corresponding application/business logic is handled right from the modeling to all the way to enactment of business processes. E-services involve inter-organizational business processes. Enactment of such business processes involves multiple parties and is modeled, using workflows by introducing e-services in business process models [1].

[*] currently visiting International Institute of Information Technology, Hyderabad (IIITH), INDIA

J. Barjis, M.M. Narasipuram, G. Rabadi, J. Ralyté, and P. Plebani (Eds.):
CAiSE 2010 Workshop EOMAS'10, Hammamet, Tunisia, pp. 77-83, 2010.

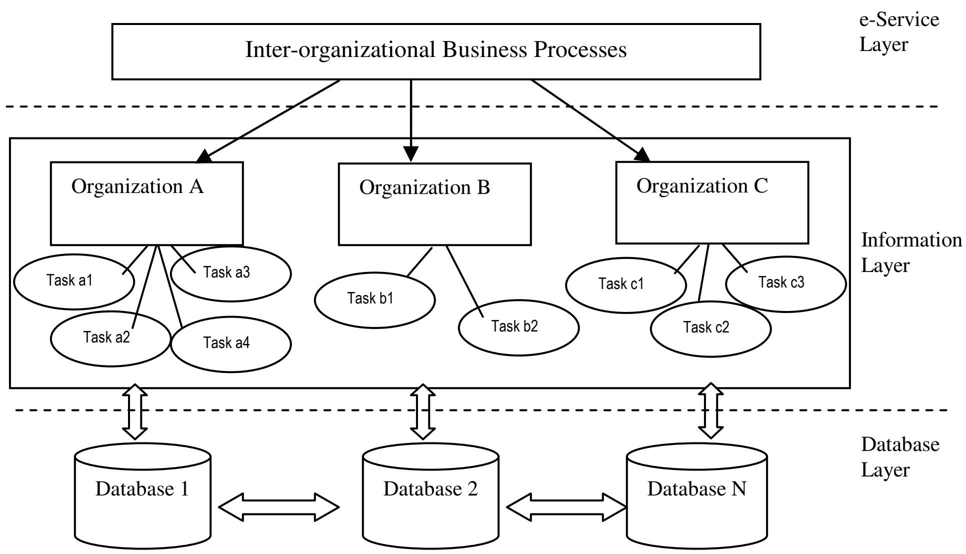

Fig. 1. An example scenario of inter-organizational business processes

Figure 1 shows a layered diagram to illustrate an example of e-services involving multiple organizations. The top layer is concerned to e-service layer, where an e-service spans multiple business organizations and carried out using inter-organizational business processes. The middle layer is information layer, which outlines information about the tasks of each organization and the e-service will involve some or all of these tasks. The bottom layer is a database layer where all the data pertaining to respective organizations are stored for further processing. Consider an example scenario where a particular e-service may need task a1, task a4, task b2 and task c1. So there is a need for collaborative business processes from different organizations. Web service applications integrate inter-organizational and heterogeneous services on the Web. Due to the changes in application/business logics at run-time, Web services are created and updated dynamically.

Several researchers have developed meta-model for different domains. Karagiannis and Höfferer [5] presented a survey on meta-models and developed taxonomy for classifying application scenarios of meta modeling according to domain, design, and integration. In [2], a formalism-independent meta-model is described to model a business process and presented the interactions among the different aspects in order to capture the dynamic behaviour of a business process model. The ER^{EC} meta-model proposed in [6] is useful to instantiate appropriate data model for electronic contracts. Chiu et al., [3] presented a meta-model approach to develop web services based e-Negotiation framework. Rosemann and Muehlen [7] presented a meta model approach to evaluate and compare different workflow management systems. They used extended entity relationship models to design the meta data models and provided a generic organizational reference meta model for enacting inter-organizational workflows. Design of all these meta models are specific to an application and assumes complete specifications of it is known in advance.

In this work, we consider services as workflows and present how meta-models are useful to facilitate the changes in specifications during run-time.

Actionable meta modeling enables upkeep of meta model and the corresponding mapping all the way to run-time caused by the changes in organizational environment at run time, or sometimes changes initiated by users due to external factors. The main feature of this framework is that it facilitates to systematically address the problem of proactively responding to changes driven from environment.

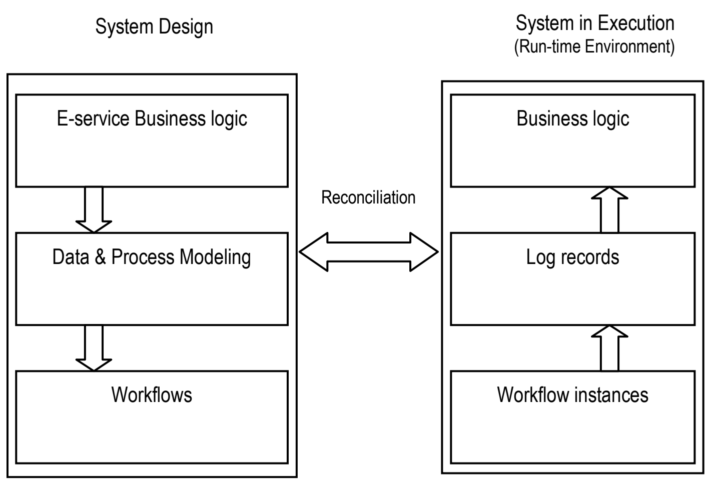

Fig. 2 Reconciliation of Meta Model with Run-Time Environment Changes

During e-service system design, the requirements are collected and the e-service elements such as activities and payments are extracted to model an e-service. Workflows will be identified to execute the e-service activities carried out by organizations. The log records keep track of the e-service execution. However, changes in organizational environment during e-service enactment need modification at conceptual level as well as at logical level. In order to propagate changes that occur during enactment to conceptual and logical level, there is a need to obtain business logic from log records and should be reconciled with the actual business logic of the e-service (Fig. 2).

In the next section, we present the framework for enactment of e-services for evolving inter-organizational business processes.

2 Architecture for Actionable Meta-modeling

E-service execution requires the modification of model schema definitions and in-turn changes in workflow instances, as a remedy. Moreover, additional constructs are needed in the system during work in progress. The present approach offers a practical solution from modeling and enactment of e-services, driven by meta-models. We maintain business policies that refer to adaptable instances of workflow schema while the service is being executed. Workflow patterns [4, 8] are useful to describe how the changes will be specified, implemented and perceived in e-services.

Figure 3 shows architecture of a system to model business processes and its instances. Web Service Server provides the services and transport of inter-organizational communications among business partners involved in the service. The *Run-time environment* (RTE) details such as workflows, rules, etc. are maintained in the database. *Dynamic Workflow Instance Generation (DWIG)* generates workflows on-the-fly and rules. It also allows the administrators to customize and edit them. Workflow definitions created or specified are executed by the *Meta-workflow Driven Workflow execution* engine. That is, the workflow engine enacts the workflows specified by the dynamic workflow instance Generation. The *Event handler* manages the events occurring during the execution of workflows. It handles events in a unified manner for both normal and exception parts of a business process workflow. The Event-Condition-Action (*ECA) Rule Manager* initiates appropriate ECA rules based on the input from Event Handler. It also keeps track of generated rules with their corresponding actions and allows users to define additional rules if necessary.

The workflow engine and the ECA rule manager works in a synchronized manner. Thus, the ECA rules control the workflow execution and the events that occur during the workflow execution result in appropriate actions. The changes in the design-time update the corresponding database. The *Knowledge Base* maintains the currency of information such as application policies, versions, new requirements etc. that governs execution of the workflow. It also stores the vocabulary of an e-service domain. Knowledge Base captures the updates that take place in the run-time as well as in the design-time. These updates, in turn, become input to the *Run-time Business Processes evaluator (RTBPE),* which generates candidate meta-model(s) and add/modify the meta-model repository. *Meta-Model repository* contains the meta-models that are specific to a service under consideration. Meta-models are added or modified based on the requirements collected from run-time environment changes as well as design-time changes.

Business Processes logic and events are captured by execution support of meta-models. *Control and Monitoring Unit* (CMU) generates/modifies the *workflow patterns* and also specific elements namely *Events, States, Processes and Rules (ESPR)* in order to keep track of changes and their progression with respect to business processes under consideration. CMU receives events and results from the run-time environment and service specific components and policies. The modeling of changes during service

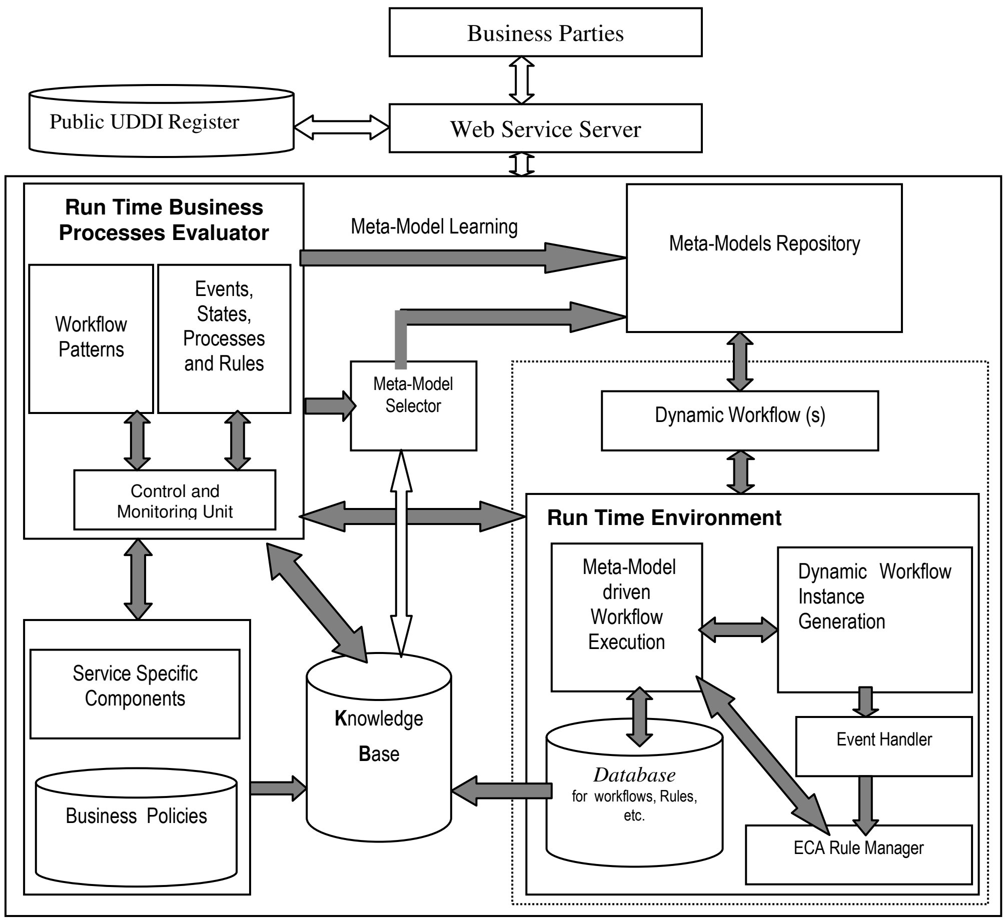

Fig. 3. Architecture for actionable meta-modeling to support evolving businesses processes

enactment can be seen as a different kind of meta-processes (tasks). Further, *Service Specific Components* are required for encapsulation and realization of the domain-specific logic for the service. The three components namely CMU, workflow patterns and ESPR will serve as RTBPE. Meta-model selector selects the appropriate meta-model(s) from the meta-model repository, which in turn drives the dynamic workflows to carry out business processes execution according to the changed context.

3 Discussion on the Proposed Meta-modeling Architecture

Proposed architecture facilitates learning from Business Activity Monitoring (BAM) to feed into the meta model updates. In the current Business Process Management domain, there is a lack of suitable architectures for BAM to support learning. The RTBPE proposed in this architecture provides the BAM metrics and their respective evaluations. These evaluations are the primary drivers for the learning required to update the meta models. The meta model learning is fed back into RTE through the dynamic work flow link with meta model repository. The benefits derivable from the proposed architecture depend on the e-service BAM design in terms of the RTBPE. Another advantage is that the dynamic workflow link between the meta model repository and RTE makes this architecture actionable.

Further work is needed to enhance the proposed architecture by explicitly specifying the service design issues such as granularity, complexity, flexibility, interoperability, autonomy, contract specifications, etc. in the RTBPE. Such enhancements will help the Business Process Management Systems (BPMS) software developers to bench mark their products and the BPMS users to assess and evaluate the BPMS products in a more comprehensive manner.

4 Conclusion

Modeling of e-services facilitates effective specification and execution of business processes. In the case of inter-organizational business processes, development of a generic model is a complex task. In this paper, we presented a meta modeling approach which helps to create context based model instances according to the changes that occur during run-time. Moreover, the meta-models learn from the services environment and update the models as necessary and hence making the meta-models actionable.

References

1. L. Aversano and G. Canfora, Introducing eservices in business process models, Proceedings of the 14th international conference on Software engineering and knowledge engineering, 2002.
2. B. Axenath, E. Kindler and V. Rubin, AMFIBIA: a meta-model for integrating business process modelling aspects, International Journal of Business Process Integration and Management, 2 (2), pp. 120 – 131, 2007.
3. D. K. W. Chiu, S. C. Cheung, P. C. K. Hung, S. Y. Y. Chiu and A. K. K. Chung, Developing e-Negotiation support with a meta-modeling approach in a web services environment, Decision Support Systems, 40 (1), pp. 51-69, 2005.

4. G. Greco, A. Guzzo, G. Manco, and D. Sacca`, Mining and Reasoning on Workflows, IEEE Transactions on Knowledge and Data Engineering, 17 (4), pp. 519-534, 2005.
5. D. Karagiannis and P. Höfferer, Metamodels in Action: An Overview. In J. Filipe, B. Shishkov, and M. Helfert, eds. ICSOFT 2006 - First International Conference on Software and Data Technologies, Insticc Press, Setúbal, 2006.
6. P. Radha Krishna, Kamalakar Karlapalem and D. K. W. Chiu, An EREC Framework for E-Contract Modeling, Enactment and Monitoring, Data and Knowledge Engineering, 51 (1), pp 31-58, 2004.
7. M. Rosemann, and M. zur Muehlen, Evaluation of Workflow Management Systems: A Meta-Model Approach, Australian J. of Infor. Systems, 6(1), p.103-116, 1998.
8. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros, Advanced Workflow Patterns, In Proc. of CoopIS 2000, pp. 18-29, 2000.