

Comparison of Two Distinctive Model Driven Web Service Orchestration Proposals

Konrad Pfadenhauer¹, Schahram Dustdar², Burkhard Kittl¹

¹ Vienna University of Technology, Institute for Production Engineering, Karlsplatz 13, 1040 Vienna, Austria

{pfadenhauer, kittl}@mail.ift.tuwien.ac.at

² Vienna University of Technology, Information Systems Institute, Distributed Systems Group, Argentinierstraße 8/184-1, 1040 Vienna, Austria

dustdar@infosys.tuwien.ac.at

Abstract

The concept of MDA (model driven architecture) aims at the goal of longevity (through business logic and architecture migration between different platforms) and quality improvement (through raising the level of abstraction). In the context of business process modeling and execution a frictionless mapping between high level, platform independent business process definition and platform specific, executable code has to be achieved. In this paper we compare two existing real world mapping scenarios of IBM and Microsoft concerning Web Service Orchestration. One “top-down” approach starting with high-level UML models down to executable code, the other as a “bottom-up” approach, with proprietary modeling integrated within the IDE. We try to answer the question how far both proposals implemented the MDA concept yet and discuss how they could support our current project of a service oriented architecture for the manufacturing shop floor domain.

1. Introduction

Modeling in the context of information systems lifecycle management has today one major driving force: knowledge alignment. If one’s aim is real process flexibility, then the model has to support the process user in choosing between alternative flows or changing control and information flows. The users should interact with the system at the same level of abstraction as the domain analysts, who initially set the static and dynamic structure. Both roles demand for simplicity [11]. The models should also absorb the system and IT knowledge about platforms, and finally the programmer knowledge about development practices.

Concerning dynamic system behaviour, which is the focus of this paper, the stakeholders’ diverse backgrounds reflect a major challenge. For one group, process modeling is an essential part of overall system modeling, for the next it is just a graphical vehicle within the IDE and for the third it is a pure visualization of their business logic without any IS context.

The strong focus on architecture dynamic and adjustability, together with an ever growing analogy between business- and IS-processes, that is to say the IS has frictionless control over the process, make a common view necessary. To achieve this, generally speaking, two main model driven process design and execution approaches exist.

The first one postulates the concept of model enrichment through extensions combined with elaborate transformation and mapping mechanisms, most prominent represented by means of OMG’s MDA (Model Driven Architecture) [17]. This ambitious concept demands for a strict separation of concerns, expressed through a layer stack consisting of different levels of abstraction. To achieve this, the models have to cover a broad set of requirements. We mention as an example the complex task of PIM (Platform Independent Model) to PSM (Platform Specific Model) mapping, extended by an additional executable UML layer, introducing mark-ups and languages like ASL (Action Specific Language). This intermediate step makes the model executable, especially valuable for testing before mapping. The benefit is a stringent process from abstract domain models down to executable code. Therefore, we call it a “top-down” approach.

The second is best described as “bottom-up”, because not the portable architecture model is the starting point, but the existing process development

and execution possibilities within a certain vendor environment. The vendor independent visualization is just an extension of graphical drag & drop interactions offered by the IDE. In this group we classify MS Biz Talk 2004/Visual Studio 2003, IBM WebSphere Studio [14] or the SAP NetWeaver Platform [21]. In addition to platform specific (PSM), the IDE models have to be called vendor specific (VSM). Only very recently within this group portability is not only achieved through the mapping to process mark-up languages like BPEL4WS, but also at model level through e.g. UML-XMI mapping [5]. Both paradigms have the above mentioned knowledge alignment in mind, but the difference in the realization shall be described by means of two examples for model driven Web Service Orchestration.

The structure of this paper is as follows. We first introduce in Section 2 our current project shortly and fix the actual state. For the realization of a Web Service oriented architecture, we need to make our platform independent process models executable, hence we continue with a more detailed discussion of model driven Web Service Orchestration in Section 3. We then choose one representative from each group mentioned above, each has to fulfil the requirement of real-world proceedings for model driven WS Orchestration with tool support. For the “top-down group” notably an IBM-BPWS4J approach, which is at a rather early testing stage, and the MS-BizTalk approach, representing the “bottom-up group”. Both proceedings, together with a theoretical comparison, are described in Section 4. Having the evaluation results of both approaches in mind, we define in Section 5 MDA and SOA requirements for future model-driven WS Orchestration proposals. With a conclusion and an outlook at the work to come we will finish this paper.

2. Problem description and project state

In our current project “Modeling the shop floor for a service oriented architecture” [19] we use the shop floor domain to implement a methodology for platform independent, service oriented model generation. To overcome a situation of vertical, interrupted processes and partly unavailable, partly static accessible functionalities we introduce our concept of a MDSA (Model Driven Service Architecture) for the shop floor, together with a top-down methodology and a tool for user friendly model creation, which in addition is the basis for automated flow execution. Therefore, we introduced three levels of model granularity and completeness which allow for static and dynamic modeling, starting with generic constructs and ending with a model of a given shop floor domain.

Fast and easy initial, computation independent modeling of a given shop floor system has to be supported, focusing on functionality and connectivity of the system as a whole. We achieve this by a generic high level model called “Shop Floor Tool-Box”. The sophisticated platform independent model (PIM) does not include detailed information enabling automated flow execution, but takes into account the socio-technical structure of the domain. Furthermore, we are developing a cascade of control loops to specify platform requirements in a platform specific model (PSM) and otherwise to maintain the system model and the service flow definitions (Figure 1).

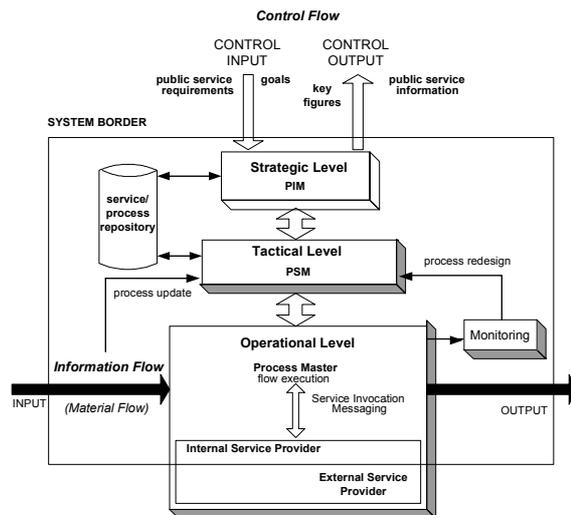


Figure 1. Process life cycle oriented domain modeling

It has to support long term platform, infrastructure and service provider decisions through PIM as-is and to-be comparisons. This high level model has to interact with the PSM concerning process definition. The latter serves at a tactical level for the (re)design of service flow definitions which are semantically rich enough for executable code generation. Both levels interact with the service repository. It provides patterns specifying typical system processes, each on the one hand in a more generic (computation independent) model form for the use within the PIM and on the other hand particular process models for the interaction with the PSM. Alternative service flows, different versions or diverse technological specifications of one and the same process are, therefore, separated from the actual valid PSM. Templates allow for fast process creation, either directly within the PSM or within the repository.

This proceeding takes into consideration the necessity of business- and IT-view alignment, allowing e.g. architectural views or process definition at distinct levels of abstraction.

The visualized aim made UML the first choice, because UML is the de facto standard for IT professionals and, as we demonstrate with our “Shop Floor Tool-Box” methodology, is well suited for abstract domain analysis too.

One of the main platforms of interest is WS/SOAP/XML, a promising set of technologies for service oriented architecture implementation. Business process management within this environment has to cover the complete range from high-level, graphical process design down to low-level, coded process execution [15], [13]. Having a platform independent UML model of the shop floor with typical process designs on the one hand and a repository of Web Service interfaces realized by typical shop floor service providers on the other hand raises the question how to develop and implement executable service flows.

3. Model driven WS Orchestration

Figure 2 illustrates the three main areas which have to be combined for model driven service orchestration. Alonso et al. [3] mention three main elements for WS composition middleware: the modeling environment, the development environment (IDE) and the run-time environment. The latter one has to execute the coded composition specification. How this specification is achieved, i.e. how and within which environment the necessary specifications and tasks from high level system model down to code execution are fulfilled, is a matter of ongoing discussion. Tasks like syntactical and semantic verifications have to be considered, just as service discovery and binding. Which type of composition specification, the composition model, the model representation language or the executable composition language, is best suited for which kind of task, can not be answered definitely yet. The dependencies between these composition specifications on the one hand and their integration within the three environments on the other determine the demand for mapping and testing functionality.

As one can see in Figure 2, the dependencies between a single composition specification and an overall system model are in our opinion a vital part of a methodology which supports life cycle oriented service composition.

The challenging factors depicted in Figure 2 establish our collection of criteria for evaluation of existing approaches. The number of real-world approaches especially for “top-down” model driven WS Orchestration with an appropriate tool support is limited.

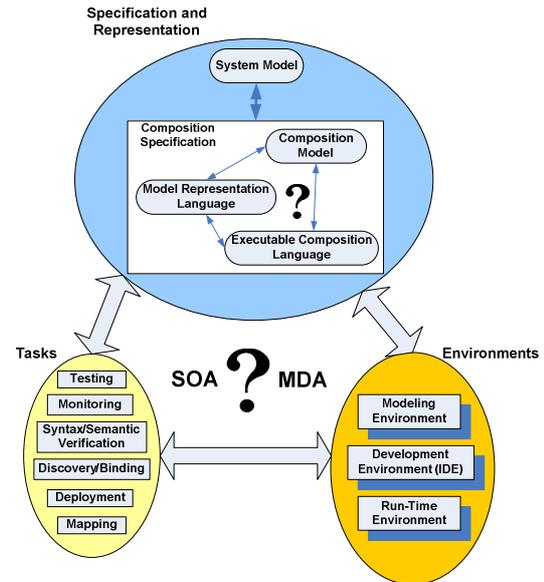


Figure 2. Composition challenges

The reasons, therefore, are amongst others:

- Still a number of process mark-up languages, the “missing link” between model and execution, like BPEL4WS or WSCI/BPML exist. None of them can be called a standard yet. Worse, the execution engines availability is limited (one is IBM’s BPWS4J for BPEL4WS).
- Model mapping and transformation is still limited to static constructs like classes and components, the complex task of dynamic information transcription is just at the beginning.
- The transformations have to be completely bi-directional, a requirement which is hard to achieve.
- Model and code have to support equally certain control patterns, the evaluation of this is still in progress (see [6], [1]). By means of the emergence of second generation Web Service technologies in conjunction with the service oriented architecture [8] it is likely these patterns have to be further expanded.
- Intelligent validation and testing mechanism between the mapping steps hardly exist yet.

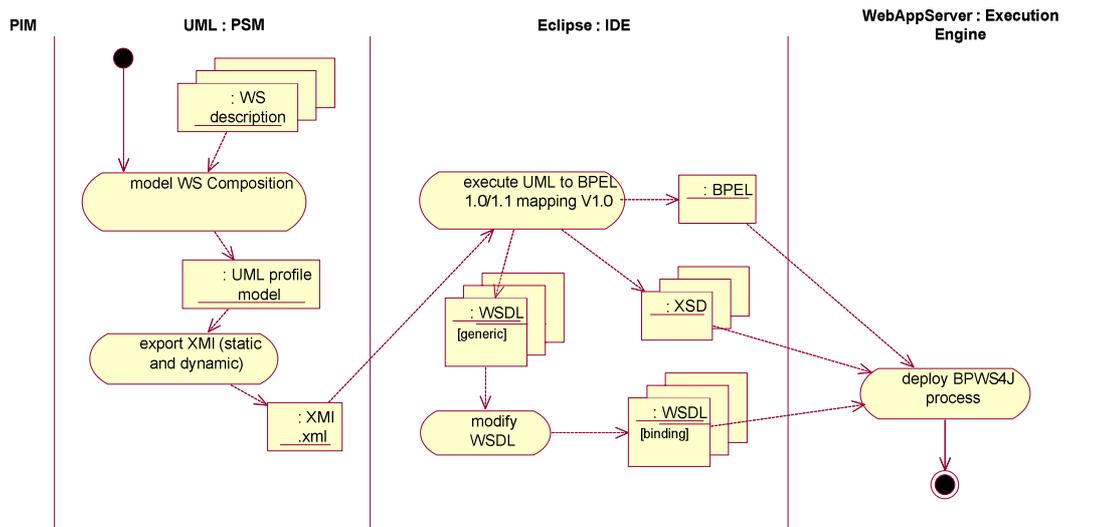


Figure 3. IBM UML-BPEL-BPWS4J proposal

To sum it up, the proceeding from WS composition graphs to executable service flows is in the state of lively theoretical discussion, although first implementations exist. Skogan, Gronmo and Solheim ([20] and [10]) present a very interesting approach. They introduce not only a technique how to import WSDL service descriptions into UML, but also how to use UML activity graphs enriched by means of certain UML extensions to define WS compositions. In addition they use XMI in combination with corresponding XSLTs to achieve a transformation towards several execution languages, at the moment BPEL4WS and WorkSCo are supported. In the future, more theoretical oriented evaluations will have to consider this and similar approaches.

4. Two model driven Web Service Orchestration proposals

At this time two rather distinctive approaches can be observed, one with model environment focus, the other with IDE focus. Not only due to the need for publicly available tools have we focused on the two MS and IBM scenarios. Moreover, they represent two completely different attempts how the model driven concept can be interpreted.

Note that the explanation of the two proceedings is simplified to illustrate the main steps and tasks which we experienced as determining for usability. Thus these steps are especially important for our comparison and strongly influence our overall review.

4.1 PSM “top-down” approach

The IBM approach is illustrated in Figure 3 [see also [16)]. The starting point is an UML model, which is specific for the BPEL4WS platform through extensions following the “UML 1.4 Profile for Automated Business Processes with a mapping to BPEL 1.0” [4]. Modeling according to the profile utilizes the information within the WS descriptions (porttypes, operations, messages) of the involved services, but no WSDL import support is available. Hence, this information has to be extracted and included into the class diagrams (static view of data types/messages, protocols and roles) and activity diagram (dynamic view) by hand.

After finishing the model gets exported into XMI format, which then is imported into an IDE java project. There an add-in performs the mapping to BPEL4WS v1.0 or v1.1, generating the WSDL, XSD and BEPL files. Bindings, location paths and service links are added by means of a WSDL modify tool. Deployment is separated from the IDE, afterwards the BPEL process can be executed via the BPWS4J 2.0 runtime engine.

4.2 VSM “bottom-up” approach

Within the Visio Orchestration Designer a proprietary modeling syntax prevails. It enables the design of basic flow pattern, allowing fork, join, group, decide and loop constructs.

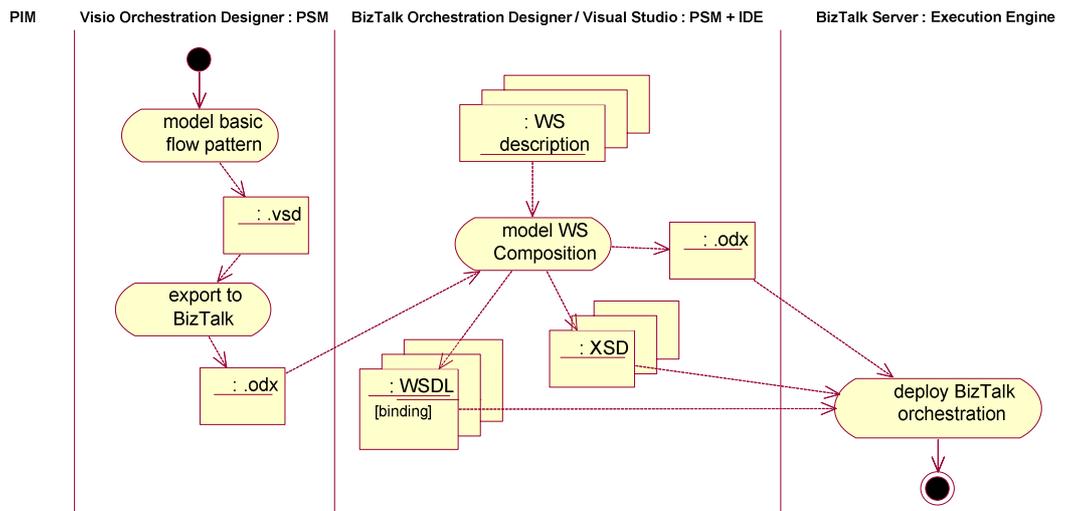


Figure 4. MS Visio-XLANG/s-BizTalk Server proposal

An add-in exports this activity graph into a BizTalk Orchestration XLANG/s .odx format, which can be imported into the BizTalk Orchestration Designer, which again runs within the Visual Studio IDE. The rudimentary flow is now supplemented with ports and messages, which are created by means of wizards utilizing the port types and message types retrieved from the added web references. All WSDL, XSD and the .odx files are generated and updated with strong graphical support. The process can be deployed within the IDE using an administration console (Figure 4).

4.3 Comparison and evaluation

As we mentioned above, both approaches are model driven Web Service Orchestration implementations, but with very diverse emphases. In chapter three we emphasized current challenges on WS Orchestration. This discussion led to a collection of criteria which in the following shall be used for comparison purposes (table 1).

Generally speaking, IBM tries to design the WS composition at a higher level of abstraction, using an UML profile for nearly complete process definition. Therefore, the mapping afford is rather high, unfortunately no syntax validation takes place before XMI export, which makes the failure risk during the BPEL4WS mapping very high. The IBM approach offers at no level WS discovery or reference mechanisms. The manual WS description import, i.e. the WSDL file transformation into different types of stereotyped classes within UML is tedious. On the other hand the XMI format containing the whole business logic and platform specific information offers

theoretically a certain flexibility concerning model reuse and transformation. In practice the reimport of created XMI files into Rational Rose, our UML tool, was not complete, manual adjustments were required. The necessary support within the IDE is reduced to WSDL modifications. The XMI mapping to BPEL4WS itself takes place automatically, but again without XMI validation. If an error occurs, the mapping stops without error handling. A BPWS4J Editor plug-in for the Eclipse IDE offers enhanced verification functionality and more important rudimentary graphical support for process editing. Nevertheless the hierarchical visualization is much less expressive and functional than the one within the MS Orchestration Designer.

Table 1. Evaluation results

	IBM "top-down"	MS "bottom-up"
Environment		
Modeling	substantial	rudimentary
Development	code oriented	graphical/wizards
Run-time	BPWS4J	BizTalk Server
Specification		
System model	no	no
Composition model	enriched UML	poor + proprietary
Model representation language	standardized XMI	proprietary
Executable composition language	standardized BPEL4WS v1.0/v1.1	proprietary XLANG/s
Tasks		
Testing WS links	no	within IDE
Testing orchestration	run-time	run-time
Monitoring	poor	strong
Syntax Verification	limited	strong
Semantic Verification	no	no
Discovery	not supported	search functionality
Binding	static	static
Deployment	isolated	integrated
Mapping	isolated	integrated

Also concerning deployment, monitoring and testing the MS approach is superior, but that is what we expected. It should be mentioned here the pointlessness of a comprehensive IDE comparison. In that case IBM WebSphere Studio rather than Eclipse would have to be the challenger, resulting in a comparison between two IDE centric approaches.

The second proposal places the Web Service Orchestration at a lower level of abstraction, within the IDE and, therefore, establishing a tight coupling between visualization and coding. The initial Visio modeling in the MS proposal offers very limited additional value, since beside the control flow no additional information can be included. Moreover, for control flow modeling the IDE functionality is as easy to use. The diagram is not integrated within other stencils, therefore, the integration in an architecture model is missing. After the import into the IDE the model is platform specific twofold (WS/XML and .NET). Once the basic service flow is set up, the service binding is very comfortable by means of drag&drop and wizard functionality. At this level code and graph are updated real time, a benefit which is only possible by means of a tight integration into the IDE. Although the underlying XLANG/s language is proprietary, restricted BPEL4WS 1.1 import and export functionality is offered which allows for a certain degree of portability.

Both approaches are concerned with single process definition and do not take into account a system model from which the process may be derived. That is to say, the UML allows for further extensions referring to this issue, whereas the Visio model doesn't offer this flexibility. Reactive semantic verification mechanisms should be implemented twofold, firstly concerning the control logic (e.g. the detection of deadlocks) and secondly concerning inconsistencies regarding the semantic at different levels of abstraction after manual intervention. Both ignore the first requirement, and only the MS approach inhibits inconsistencies between code and graphical representation due to the tight integration within the IDE (but not between the Visio model and the IDE graph).

5. MDA and SOA considerations for the future

5.1. Computation and platform independent modeling

Following the MDA notation, IBM's UML-BPEL proposal starts with a UML 1.4 compliant definition of a PSM (the platform is WS/XML) and transforms (via

separated mapping rules) it into a model specification language by means of XMI, a standardized XML language. The core MDA concepts, different levels of abstraction, separation of concerns and the model transformation paradigm are, therefore, fulfilled. This is not true for the MS proposal, where both, the initial and the BizTalk Orchestration model, are proprietary and not transformable in a standardized format like XMI. Both approaches do not take into account the mappings from computation or even platform independent process descriptions, in our opinion their main weakness regarding MDA principles. Enterprises will not make the same mistake twice like it happened regarding proprietary ERP (Enterprise Resource Planning) systems in the past. Often the business logic was modeled and defined tightly coupled to a certain platform. The WS/XML platform will be one integration technology among others in service oriented architectures, thus platform independent modeling is crucial.

5.2. Integration of modeling techniques

It is also a fact that concerning process modeling the use of a unique syntax is last but not least within the area of WS Orchestration far from becoming true. Historically, two more or less separated paradigms have evolved, the business and the information system view, resulting in two groups of modeling techniques, one for BPM (Business Process Modeling) containing IDEF0, Petri Nets, EPC (Event-Driven Process Chain), Flowcharting, etc., the other for ISM (Information System Modeling) containing Data flow diagramming, ER diagramming or UML. Integrated design strategies rarely have been the case in practice [9]. With the emergence of Web Service-based process design and execution an alignment of these two views is more than ever vital for successful BPM (Business Process Management). Therefore, and because of the reuse of process repositories already existing within the organizations, the integration possibilities of proprietary business process models within the above mentioned techniques should play a more important role.

In the context of a "top-down" approach, mappings from XML representations of widespread used and well-defined modeling techniques (EPC Markup Language in the case of EPC, or PNML Petri Net Markup Language in the case of Petri Nets) to platform independent MDA-UML models seem promising. OMG's Business Process Definition Metamodel [18], an UML 2.0 profile, aims at this goal. This proposal supports the mapping to a common metamodel and thus facilitates the communication among a variety of process models. We mention in this context ArcStyler, which is in the first place a classic MDA tool vendor

for software engineering purposes, but with the MDA-Business Transformer for ARIS [12], an eEPC (extended Event-Driven Process Chain) to UML mapper, they offer an interesting approach of business and IT view alignment. Unfortunately they do not support UML to BPEL4WS mapping. Van der Aalst [2] has successfully undertaken a similar task, that is to say a mapping between EPC and Petri Nets. Modeling techniques alignment can also be achieved at a lower level of abstraction, neglecting platform independent representation through direct composition language mappings. This would be a similar, platform specific solution like the UML-BPEL4WS example from IBM described above.

For a “bottom-up” approach, mappings to proprietary modeling syntaxes, embodied e.g. in MS BizTalk Orchestration Designer or IBM WebSphere Business Integration Modeler, would be needed.

5.3. System Modeling

As we already mentioned, the dependencies between a single process model and an overall architectural model are not considered within the two approaches. To see the process isolated from its environment has many disadvantages:

- Inter process dependencies like synchronization or process hierarchies (nested sub-processes) are missing. Hence, appropriate business rules and constraints have to be coded. Monitoring by model is no longer possible.
- Service repositories are missing at initial modeling level. The design of the activities takes place without knowledge about their availability according to necessary QoSs (Quality of Service).
- Within architectures different paths exist and new evolve to achieve a certain process aim. Without an architecture model and the knowledge about the given connectivity, one has to design a single process model for every control flow, not knowing whether the needed interoperability exists or not.
- The consequences of a change in the architecture or the introduction of new service providers for the process designs can not be monitored accordingly.

Regarding the MDA approach, OMG’s Business Process Definition Metamodel [18] again seems

noteworthy, because the final specification is expected to achieve a metamodel that complements existing UML metamodels so that business processes specifications can be part of complete system specifications to assure consistency and completeness.

5.4. Platform specific model completeness

Recent second generation Web Service Technologies like WS-Reliable Messaging or WS-Policy [8] are not embodied within the platform specific modeling environments, although they are fundamental for transaction and context implementation regarding service-oriented inter-organizational integration [7]. Their absence in both modeling approaches means these concepts have to be included belatedly, an advancement that jeopardizes model and implementation congruency. In our opinion a process lifecycle management without a complete model covering all relevant interaction aspects is not possible.

6. Conclusion and outlook

Within this paper we compared two general approaches of process design and execution, one with strong focus on the abstract, platform and vendor independent model, which is semantically rich enough so that an IDE is exclusively needed for (complex) mapping tasks. The other with the focus on “applied” modeling within the IDE, supported by a rudimentary and abstract graph describing the control flow.

The latter “bottom-up” approach, represented by MS BizTalk Server, has its main advantage concerning easy integration of existing Web Service definitions, which is not possible within the “top-down” approach. On the other hand the integration of the Visio or the Biz Talk Orchestration models in an overall, syntactical homogeneous architecture model is not possible, thus these directed graphs have to be built from the scratch. Here the UML-BPEL4WS approach offers much more possibilities of integration in an enclosing MDA concept.

Furthermore, we discussed challenges for future model driven process engineering proposals. We demand for a platform independent model as the starting point, which supports not only isolated process design, but architecture modeling. The behavioural views within these models have to be the basis for initial Web Service Orchestration design.

Next steps to come are investigations, whether these approaches are suited for shop floor domain usage. Therefore, we plan to implement some of our processes in a use case scenario, allowing for tests regarding applicable mechanism for process model and process specification synchronization, with BPEL4WS as the main specification of interest. Still another open

question is how the process models have to be embedded in the overall architecture model e.g. how mechanisms, which synchronize between static architecture structure (“the sum of all possible processes”) and single process lifecycle management, can be established.

7. References

- [1] van der Aalst, W.M.P.: ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P., Workflow Patterns, BETA Working Paper Series, WP 47, Eindhoven University of Technology, Eindhoven, 2000
- [2] van der Aalst, W.M.P.: Formalization and Verification of Event-driven Process Chains. Information and Software Technology, 41(10), 639-650, 1999
- [3] Alonso, G. et al.: Web Services – Concepts, Architectures and Applications, Springer Verlag, Berlin Heidelberg, 2004
- [4] Amsden, J., Gardner, T. et al.: Draft UML 1.4 Profile for Automated Business Processes with a mapping to BPEL 1.0, Version 1.1, IBM, June 9th 2003
- [5] Beck, K., Joseph, J., Goldszmidt, G.: Learn business process modeling basics for the analyst, IBM developerworks, Februar 22nd 2005, <http://www-128.ibm.com/developerworks/webservices/library/ws-bpm4analyst/index.html>
- [6] Dumas, M., ter Hofstede, A.H.M.: UML Activity Diagrams as a Workflow Specification Language, In: M. Gogolla, C. Kobryn, editors, UML 2001 – The Unified Modeling Language, Proc. Of the Int. Conference in Toronto, Canada, October, Springer Verlag, Berlin Heidelberg, 2001, pp76-90.
- [7] Dustdar, S.: Web Services Workflows - Composition, Coordination, and Transactions in Service-Oriented Computing, Concurrent Engineering: Research and Applications, Sage Publications, September 2004, p. 237-246
- [8] Erl, T.: Service-Oriented Architecture, Prentice Hall PTR, New Jersey, 2004
- [9] Giaglis, G.M.: A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques, International Journal of Flexible Manufacturing Systems, Volume 13, Issue 2, April 2001, 209 – 228
- [10] Grønmo, R., Solheim, I.: Towards Modeling Web Service Composition in UML, presented at The 2nd International Workshop on Web Services: Modeling, Architecture and Infrastructure (WSMAI-2004), Porto, Portugal, 2004
- [11] Haeckel, S.H.: Leading on demand business-Executives as architects, IBM Systems Journal, Volume 42, Issue 3, 2003, pp405-413.
- [12] Interactive Objects Software GmbH: ArcStyler MDA-Business Transformer Modeling Style Guide For ARIS, For ArcStyler Version 3.x, November 2002
- [13] Kalogeras, A.P. et al.: Vertical Integration of Enterprise Industrial Systems Utilizing Web Services, in Sauter, T., Vasques, F.: Proceedings of the 2004 IEEE International Workshop on Factory Communication Systems, Vienna, September 2004
- [14] Kloppmann, M., König, D. et al.: Business process choreography in Websphere: combining the power of BPEL and J2EE, IBM Systems Journal, Volume 43, Issue 2, 2004, pp. 270 - 296
- [15] Leymann, F., Roller, D., Schmidt, M.-T.: Web services and business process management, IBM Systems Journal, Volume 41, Issue 2, 2002, pp198 – 211
- [16] Mantell, K.: From UML to BPEL, IBM developerworks, Sept 9th 2003, <http://www-106.ibm.com/developerworks/webservices/library/ws-uml2bpel/>
- [17] Miller, J., Mukerji, J. (Editors): MDA Guide Version 1.0.1, OMG, June 12th 2003
- [18] OMG: Business Process Definition Metamodel, Request for Proposal, OMG document, 2003-01-06, <http://www.omg.org/docs/bei/03-01-06.pdf>
- [19] Pfadenhauer, K., Kittl, B.: With a system approach towards a model driven service architecture for the shop floor, in: Proceedings of the 2004 International Research Conference on Innovations in Information Technology (IIT2004), College of Information Technology, UAE University, Dubai, 2004
- [20] Skogan, D., Grønmo, R., Solheim, I.: Web Service Composition in UML, The 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC), Monterey, California, 2004
- [21] Woods, D. and Word, J.: SAP NetWeaver for dummies, Wiley Publishing, Indianapolis, Indiana, USA, 2004