UDK 621.91

J. MINGUEZ, S. SILCHER, B. MITSCHANG,
E. WESTKAMPER, Stuttgart, Germany

TOWARDS INTELLIGENT MANUFACTURING: EQUIPPING SOA-BASED ARCHITECTURES WITH ADVANCED SLM SERVICES

ДЖ. МІНГЕЗ, С. СІЛЧЕР, Б. МІТШАНГ, Э. ВЕСТКЕМПЕР
НАЗУСТРІЧ ІНТЕЛЕКТУАЛЬНОМУ ВИРОБНИЦТВУ SOA-БАЗОВАНЕ УСТАТКУВАННЯ ІЗ ТИМЧАСОВИМИ СИСТЕМАМИ SLM

Передбачення заснованих на знаннях і штучному інтелекті виробничих систем викликає розвиток системних структур, які можуть гнучко управляти інформаційними потоками в паралелях спільно працюючих виробничих систем і забезпечувати необхідне обслуговування для підтримки виконання виробничих процесів. Ділова кон'юнктура, що постійно змінюється, і нестабільні сценарії дії виробничих компаній спонукують їх до постійної готовності адаптувати виробничі процеси й виробничі системи до нових умов. У такому контексті необхідна гнучка інфраструктура, що підтримує повну інтеграцію процесів і пристосована до їхнього обслуговування. Стаття представляє семантичні основи інноваційного підходу до обслуговування, що надає можливість адаптивного керування експлуатаційними ресурсами в інтегрованих рамках обслуговуючих структур.
Ключові слова: виробництво, інформація, структура обслуговування

Предвосхищение основанных на знаниях и искусственном интеллекте производственных систем вызывает развитие системных структур, которые могут гибко управлять информационными потоками в параллелях совместно работающих производственных систем и обеспечивать необходимое обслуживание для поддержки выполнения производственных процессов. Постоянно изменяющаяся деловая конъюнктура и нестабильные сценарии действия производственных компаний побуждают их к постоянной готовности адаптировать производственные процессы и производственные системы к новым условиям. В таком контексте необходима гибкая инфраструктура, которая поддерживает полную интеграцию процессов и приспособлена к их обслуживанию. Статья представляет семантические основы инновационного подхода к обслуживанию, предоставляющего возможность адаптивного управления эксплуатационными ресурсами в интегрированных рамках обслуживающих структур.
Ключевые слова: производство, информация, обслуживающая структура

The vision of knowledge-based and intelligent manufacturing systems is driving the development of system architectures, which can seamlessly manage information flows across multiple heterogeneous manufacturing systems and provide the necessary services to support the execution of production processes. Constantly changing business conditions and turbulent scenarios force manufacturing companies to continuously adapt their business processes and manufacturing systems. In such a context, a flexible infrastructure that supports the full integration of processes and adapts its services is needed. This paper presents an innovative se-

mantic service framework that enables the adoption of service lifecycle management (SLM) in an SOAbased integration framework.
Keywords: Manufacturing, Information, Service-oriented Architecture

1 INTRODUCTION

Current manufacturing faces constantly changing business conditions and turbulent scenarios that require business processes to be continuously adapted. The vision of adaptive, knowledge-based and intelligent manufacturing focuses on agility and anticipation to permit flexible production through the integration of intelligent systems and processes [1]. Under these circumstances of continuous change and adaption, the implementation of flexible IT infrastructures that enable the full integration of processes becomes a fundamental requirement in current adaptive manufacturing. The usage of standards and modular system architectures is a key aspect to transform the principles of adaptability and provide adaptive information management in manufacturing environments.

The principles of reusability and loosely-coupled services have made Service-oriented Architecture (SOA) the most used paradigm for software design at the business level. The penetration of SOA in different manufacturing domains can be best observed at the current servicebased solutions for Enterprise Resource Planning (ERP) and Supply Chain Management (SCM). In addition to this, the presence of SOA in Business Process Management (BPM) and integration is rapidly growing. For companies that have focused on internal SOA deployments, the leading investment has been application, process and data integration [2], also known as Enterprise Application Integration (EAI). Nowadays, the Enterprise Service Bus (ESB) is one of the leading concepts in EAI. An ESB acts as the backbone of SOA in event-driven enterprises by providing the foundation of a loosely coupled, highly distributed integration network [3]. However, this technology alone is not enough to support the adaptive information services needed in responsive manufacturing environments because of the constant need for adaptation of services. In order to accommodate an EAI infrastructure to a continuous adaptation of the business processes, enterprises need to integrate appropriate Service Lifecycle Management (SLM).

EAI process models following a SOA-based approach, as in current BPM modeling tools, are executed on an ESB, which acts as integration middleware. The configuration of the service bus must be continuously adapted to the changes of EAI processes, or EAI process fragments. In a SOA, process fragments are referred to as services. An integration middleware for adaptive manufacturing needs to incorporate techniques to dynamically reconfigure processes, discover and select suitable services by automated means. These aspects are essential to support EAI process modelers and are among the major research challenges in SLM and SOA governance [4].

The Manufacturing Service Bus (MSB) is a SOA-based approach that extends the Enterprise Service Bus capabilities in three areas: event management, factory context and change propagation workflows. The MSB enables loose coupling between service requesters and providers by brokering requests between them. However, a service bus is not aware of the semantics of information. This fact hinders this integration platform from adopting self-reconfiguration processes, automated discovery of services and dynamic service compositions, which can support business process modelers.

In this paper, we present a semantic service framework as an MSB enhancement that enables the automation of service discovery and dynamic process reconfiguration through semantic annotations. This extension will permit to support EAI process modeling tools, thus optimizing the reconfiguration of processes. Moreover, the presented MSB semantic extension provides the means to adapt the MSB execution infrastructure to changing EAI process models. Our contribution enables the adoption of SLM in an SOA-based integration framework, which will increase the level of automation that is needed in adaptive manufacturing.

In the next section, the service-oriented architecture paradigm is explained along with the current research challenges in SLM. In Section 3, the MSB architecture is described. The semantic service framework and its adoption in the MSB as SLM enhancement are presented in Section 4 and 5, respectively. Related Work is detailed in Section 6 and finally, our conclusions and outlook are given in Section 7.

2 SERVICE LIFECYCLE MANAGEMENT

2.1 Service-oriented architecture

SOA is a paradigm of designing business applications by using – or reusing –self-contained, independent and discoverable services. Two of the distinguishing principles of SOA are reusability of existing assets and loose coupling of services.

SOA can empower a business with a flexible infrastructure and processing environment by provisioning independent, reusable automated business processes as services [4]. The challenge of integration, the cost of managing IT and the inflexibility to respond to changing requirements are the decisive reasons why most organizations adopt SOA.

Web services represent a common implementation of SOA and part of their success of their adoption is due to the standardization efforts made in describing service interfaces and messaging. These standards support SOAs to follow the “find, bind, invoke” paradigm, where a service provider publishes its service description in a service registry in order for a service consumer to find it, and then invoke the service through a request/reply mechanism. This communication keystone for web services is known as web service discovery.

Despite the success of web services, identifying reusable services and integrating them as process fragments in executable business processes is not trivial. Here, it is crucial for industries with a high level of service reusability, self-reconfiguration and automation to have a service lifecycle vision and a service management in their integration platforms.

2.2 Service lifecycle

A service lifecycle is defined by a series of stages through which an IT service passes during its lifetime. In SOAbased architectures, there is typically a loop that represents the reuse of services in different business processes and its adaption to different business requirements (see Figure 1). In the pre-design phases, services are planned. In the planning phase, an analysis of the requirements takes place, followed by the identification of possible reusable services. In the design phase, the appropriate service granularity is determined, as well as security, performance and quality of service (QoS) aspects.

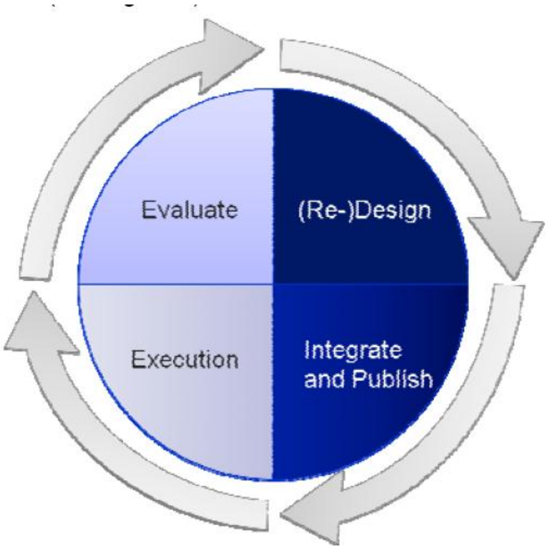

Figure 1 – Service Lifecycle

A service repository with possible reusable, process fragments as well as BPM tools may be used to support the design process. After the design phase, the service is encapsulated and the service configuration is set in the so-called integrate and publish phase. In this phase, the service is also published on a service registry for other services to discover. Additionally, it may be marked as reusable process fragment, depending on its occurrence metrics, business domain and business function. After the service is published, then it is deployed and possibly integrated in other processes. Once a service is deployed, the service begins the execution phase, where a service is operative. Then, in the evaluation phase, service interactions and performance metrics are logged for further analysis in a future re-design phase.

2.3 Enabling SOA governance

In order to address these challenges and ensure the success of a SOA-based approach, services are managed according to specific service lifecycle guidelines and implementation methods, which define the SOA governance of a company.

The adoption of SOA in a company involves not only the operational aspects of the services lifecycle management, but also, the management of service design policies, reusability guidelines, service change policies, Service Level Agreements (SLA) and, most important, effectiveness measurement methods. These play a very important role and represent the feedback loop within the SOA governance mechanism of a company. With the proper policy enforcement infrastructure and monitoring system in place, service performance metrics can be contrasted with the specified SLAs. Monitoring is the opportunity for a business to refine its services, start a new service cycle, guarantee SLAs and truly ensure the effectiveness of SOA.

3 SERVICE LIFECYCLE IN MANUFACTURING

3.1 SOA in event-driven manufacturing

Manufacturing environments present an extremely heterogeneous landscape of equipment and production systems. Applications that exchange production data, communicate with each other following an event-driven pattern. Most interchanged messages are based on some kind of event, alarm or notification, which is due to the nature of production processes. Events are associated with an event emitter and one or more event consumers. In the event emitter, a significant change of state takes place, which generates an event. This generated event is then propagated to the event consumers, which react to the event according to a predetermined procedure or internal, fixed rules. The architecture paradigm for this type of event-centralized communication and integration of systems is known as Event-driven Architecture (EDA) and it is widely adopted in manufacturing environments. Typically an EDA infrastructure enables the detection, propagation and processing of events.

At the business level, manufacturing companies have adapted their business processes to service-oriented paradigms in order to gain flexibility. It's Mostly in the areas of ERP and SCM where SOA has gained more presence. However, the strong penetration of this approach has created a gap between event-driven manufacturing environments and SOA-based business processes. In an event-driven manufacturing environment, businessrelevant events can alter the normal course of business processes entailing turbulent scenarios. Manufacturing companies need an event-driven SOA to have the agility to react to constantly changing business requirements and adapt their business processes. In order to achieve this, the gap between event-driven manufacturing processes and service-oriented business processes needs to be bridged.

3.2 The Manufacturing Service Bus

The backbone of SOA in event-driven enterprises is the Enterprise Service Bus (ESB), which combines messaging, data transformation and intelligent routing services to connect distributed applications across an enterprise while assuring reliability and transactional integrity. An ESB infrastructure contains the right mechanisms to enable the required flexible business environment, such as a workflow engine, mediation and content-based routing services. The ESB integration pattern retains centralized control over configuration while allowing for bus infrastructure services, such as message routing or addressing, to be physically distributed. This pattern is especially relevant from the perspective of extending ESB capabilities by deploying new services without affecting the existing infrastructure. Based on this concept, we defined the Manufacturing Service Bus [5] as an ESB with domain-specific services for manufacturing, which aims to fill the gap between EDA-based manufacturing environments and SOA-based business processes.

The MSB enhances the functionalities of an ESB by integrating event management services needed in a manufacturing environment. In this architectural model, five abstraction layers across a manufacturing environment can be distinguished, where the MSB acts as the integration layer (Figure 2).

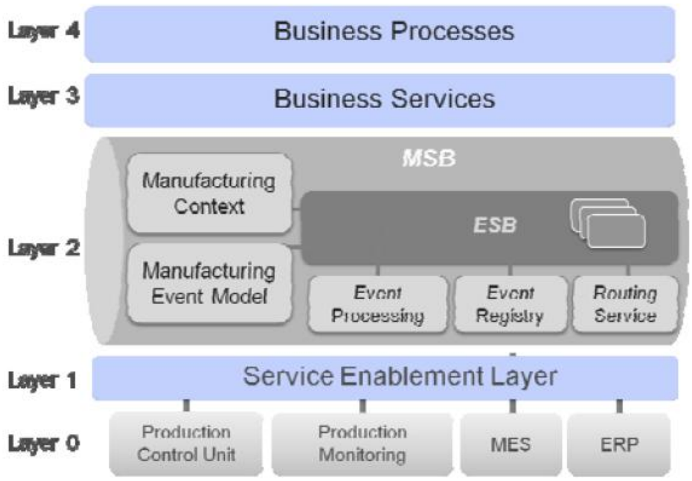

Figure 2 – The Manufacturing Service Bus

Manufacturing systems and digital factory information systems, such as MES, or ERP, are grouped in Layer 0, which forms the source of manufacturing information flows. Layer 1 defines a service enablement layer for all source systems, by providing systems with a service interface. In this layer, service adapters enable manufacturing systems and applications to provide data as services which can be connected to the MSB. Layer 2 facilitates the integration of data provisioning services (Layer 1) into complex business services (Layer 3). In the MSB, event processing and routing components are used to propagate events to the appropriate event consumers.

The MSB uses an XML-based canonical format for eventmessages, which facilitates event-processing and routing tasks. Event attributes contain information about the nature of the event, the current state of the event as well as routing parameters, such as origin and destination. Extended Schemas are used to extend the event model for different manufacturing sub-domains like maintenance, customer relationship or supply chain. This approach enables the adoption of event processing techniques in a service-oriented computing (SOC) environment, which is one of the most important requirements in order to fill the gap between the SOAbased business processes and the event-driven manufacturing environments. A Workflow Management System enables the orchestration of different business services in Layer 3. Such business services are executable parts of business processes (Layer 4) and can be dynamically adapted, depending on the incoming events.

Nevertheless, an integration framework for adaptive manufacturing, such as the MSB, needs to incorporate techniques to (I) discover and select suitable services by automated means; (II) dynamically analyze and reconfigure business processes; (III) detect problems in service interactions and transparently upgrade and version services without affecting normal operation.

3.3 SLM Challenges in Manufacturing

For manufacturing environments that pursue an adaptive manufacturing approach, a high level of service reusability, automation and self-reconfiguration represent the three major requirements. In order for an integration platform to provide the necessary services that fulfill these requirements, a service lifecycle strategy is needed. The implementation of a lifecycle strategy in SOC environments with a high level of automation presents a number of challenges that need to be met:

- In the early phases of design and development of services, business process modeling tools must be aware of existing assets and services. One of the key issues in reusing services is the ability to discover them effectively. For this purpose, only the inherent interface data of a service may not be enough.
- As services are adapted to different business requirements and used – or reused – in composite applications, the need to document changes, updates, versions increases considerably. The reuse of process fragments may become impracticable if services cannot be distinguished from each other or from former versions.
- In addition to this, once services are deployed, reconfiguration of business processes must be aware of the different service dependencies. Upgrading a business process cannot be done blindly, without knowledge about the dependencies between the different process fragments.

These challenges are addressed by the Semantic Service Framework, which is described ahead.

4 SEMANTIC SERVICE LIFECYCLE FRAMEWORK

4.1 Need for Semantics

Semantics is the study of meaning and usually focuses on the relation between different representations of concepts, or content of data. The use of semantics has been already successful in other fields like linguistics, knowledge representation, and artificial intelligence. Semantic annotations can also be helpful in integration platforms for adaptive manufacturing, where the fully interoperability of systems represent the biggest barrier to achieve the desired degree of automation.

There are four types of information heterogeneity [6]: (I) system heterogeneity, which considers the storage of data in different platforms and operating systems; (II) syntactic heterogeneity, where information sources use different representations and encodings for data; (III) structural or schematic heterogeneity, which considers the storage of data in different formats, data models, structures or schemas; and (IV) semantic heterogeneity, which considers the content of information and its intended meaning. In a highly heterogeneous landscape of information systems, such as manufacturing environments, semantic heterogeneity can be encountered at the data level, where the meaning of data is expressed differently, depending on the residing system. For instance, a customer relationship management CRM) application using an XML dialect to represent customer orders doesn't necessarily understand the XML dialect of an order management system. A possible solution to the problem of interoperability is to semantically describe the meaning of the terminology of each distributed data using shared concepts. Usually, a shared ontology is used to make clear the relationships and differences between concepts.

However, the semantic interoperability problem can also be encountered at the business process level. For services to interact properly with each other as part of composite applications, which perform more complex functions by orchestrating numerous services and pieces of information, the requester and provider entities must agree on both the service description and semantics that will govern the interaction between them [7]. This implies an agreement between requester and provider. The aspect of semantic interoperability between service requesters and providers can be exploited to solve the challenges exposed in the last section and will be the focus in this section.

4.2 A manufacturing service semantic framework

As it was mentioned in Section 3, service discovery is one of the most challenging aspects in the reuse of services. Business process modeling tools must be aware of existing assets and services. Here, the use of semantics to describe existing services provides advanced features that simplify business process modeling, such as automatic semantic-based discovery of services, autocompletion, and edition of new process and data mediators [8]. These tasks lead to more effective modeling and reduce the time spent in the design phase [9].

The challenge is now how to describe services correctly so that these can be discovered. The method we use is based on service semantic annotations, which is based on associating semantic metadata with resources.

A semantic annotation application enables service modelers to describe services. The description process is very important for later service discovery. The more accurate is the description of a service, the easier are the appropriate services discovered in a semantic search. The description of a service needs to address four types of semantics [10]:

1. Data semantics: description of input and output messages of a service.

2. Functional semantics: definition of the capabilities of a service, that is, a description of its operations.

3. Non-functional semantics: definition of quantitative or non-quantitative constraints related to QoS or policy requirements, such as message encryption.

4. Execution semantics: definition of the execution flow of operations within a service or of services in a process.

The definition of all these aspects of a service gives a detailed description of what a service can do, how it can be done, the conditions under which it can run and its interaction or dependency with other services. When services are semantically described, there are three steps that need to be done: (i) the service data and service operations are semantically annotated, (ii) an inference engine checks for inconsistencies with existing concepts and relationships, and (iii) then the annotated services need to be stored in a semantic service repository. These three steps are very important for later service discovery and reuse.

The presented service semantic annotation framework permits the annotation of services in a manufacturing environment. Event-data service providers describe the input and output data that the service manages as well as the functionality of the service, i.e. the operations of the service, by linking the data and operations in the service interface to the concepts expressed in the domainspecific production ontology. This ontology contains all relevant concepts and relationships of a concrete manufacturing environment. For instance, the meaning of a ‘customer order’, as well as all its attributes, such as the order status, and relationships with other concepts, like customer request, are described. This ontology is populated with instances of the ontology classes, establishing a knowledge base for a concrete manufacturing environment, so that services from different domains, such as ERP, CRM, SCM, can overcome semantic heterogeneity problems.

The annotation and discovery of services is shown in Figure 3. Once the service provider annotates a service, like the MES event-data provider does (1) in the example, the semantic service engine starts a reasoning process about new knowledge (2), i.e. new service interactions, and service dependencies. Then, the semantic service engine updates the service ontology and feeds the semantic service repository with the new semantic service. Once a service is registered, a service requester can formulate its requirements in a semantic template by using the same terms as those expressed in the production ontology and service ontology. In the example shown in Figure 3, a CRM event consumer service sends this event provider search request (3). The reasoning techniques of the semantic service engine are then used to compare the requirements of the template with the capabilities of the service available in the semantic service repository, allowing services to be discovered by requesters (4). The value of semantics is to provide a much richer description of services as purely syntactical thus providing powerful support for service discovery.

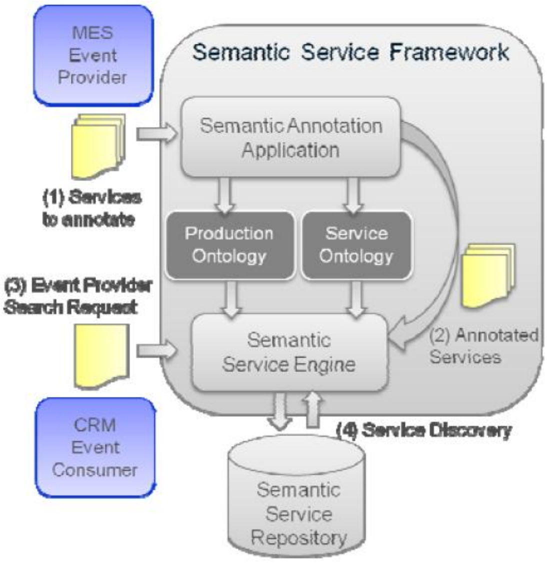

Figure 3 – Semantic Service Framework

Additionally, service providers can include optional descriptions that define preconditions and effects of its operations. Preconditions must be met before the service is invoked and effects are the expected results of invoking an operation. Moreover, a service provider may consider adding a description about non-functional aspects of a service, like quality, reliability or security. These nonfunctional characteristics are the basis for SLA stipulations that allow services to agree on QoS aspects. Finally, service providers can also include execution semantics when describing a service which provides knowledge about service interactions and the dependencies with other services or processes. These are mapped to the concepts expressed in a service ontology, which is populated with instances of the service ontology classes, establishing a knowledge base for service dependencies. This allows, for instance, dynamically establishing new SLAs when a service updates its security policy by discovering service consumers whose semantic description of their requirements match the stated policy. This is the basis for the implementation of a SOA governance strategy.

The benefits of adding semantics is pervasive in the entire lifecycle of a service [10]. The service semantic annotation framework can be used to implement a service lifecycle management system. As it was mentioned inSection 3, one of the major challenges in SLM is the need to document changes, updates and versions of services. In dynamic environments, such as in adaptive manufacturing, services are adapted continuously, i.e. incorporate new functions, update or delete existing functions, etc. Under these circumstances, the needed level of automation cannot be achieved, unless the mechanism for service discovery and process reconfiguration is supported by an adequate service lifecycle management. The architecture described in the next section uses the service semantic annotation framework to implement an SLM component that can be incorporated into service-based integration platforms for manufacturing, like the Manufacturing Service Bus.

5 SEMANTICS-ASSISSTED SERVICE LIFECYCLE IN THE MANUFACTURING SERVICE BUS

In order to gain the required responsiveness and adaptability in current manufacturing, we must provide the right support for self-reconfiguration of processes and service management in order to increase the level of automation in integration infrastructures. The MSB is a service-based integration platform for manufacturing environments, but with no SLM. The adoption of the framework presented in the previous section provides a SLM implementation that enables service management and versioning, service dependency management and contributes to the self-reconfiguration of processes and automated service discovery, which are considered key technological aspects in integration infrastructures for adaptive manufacturing.

The adoption of the framework in the MSB is based on an SLM strategy. All connected services to the MSB are considered event sources or destinations. As, described in the reference architecture of the MSB, a content-based routing service routes events to their appropriate destinations. This is done thanks to an XML-based canonical event format which services use to send their events to the MSB. Through this event model, our factory integration platform can keep track of events, route messages to the appropriate destinations and perform mediation tasks on messages. The MSB routing serviceis based on a fixed set of XPath expressions, which evaluate certain nodes in incoming event messages. An example is shown ahead:

```
/*[@eventIdRegistered="true" and
@eventFlowIdRegistered="true" and
not(@eventId="") and
not(@eventFlowId="") and @eventType="85"]
```

The routing service looks up the corresponding destination in its routing tables and determines where to route the message by evaluating the event data. Each event type is mapped to one or more destinations, which process the event messages of that type. Nevertheless, these routing tables have to be edited manually when new service dependencies are established. In addition to this, this person has to be an IT specialist due to the complexity of XPath expressions. These aspects avoid the MSB to provide the desired flexibility when adding new services or when the operations of a service are updated. Therefore, a certain degree of automation needs to be incorporated that supports service management. For these reasons, we propose a semantic approach to enable SLM in our integration platform, and provide the desired degree of automation. This way, when a service is updated, a new version will be registered in the semantic service repository by using the presented semantic service framework. As a consequence, once service dependencies are updated, the routing tables will be automatically updated as well with the corresponding updated routing information for incoming events.

In Figure 4, a scenario to reflect the automatic updates in the routing tables is shown. In the example, a CRM system uploads a description of the new service version (1). Then the routing tables are updated with its new destination (2), s_CRM_v2. In this manner, events of type 85 from the MES can be routed to the new destination (see Figure 3). Additionally, all service compositions that contained version 1 of this CRM event consumer need to update the service endpoint as well. Such service dependencies are managed by the semantic service framework, and stored in the service ontology knowledge base.

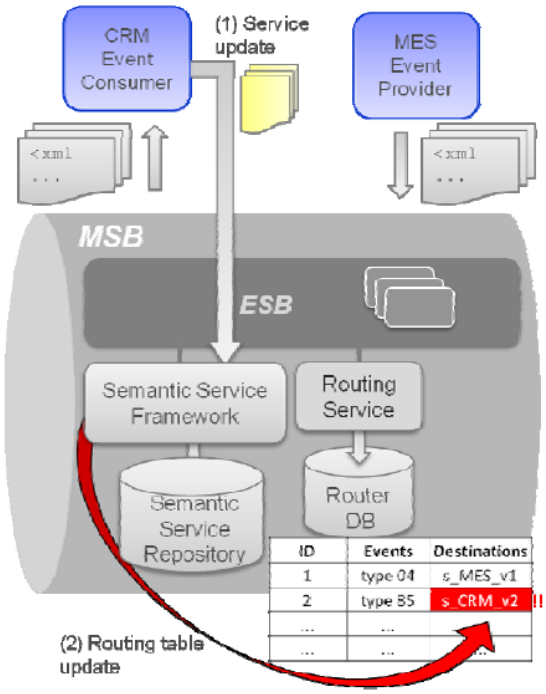

Figure 4 – Semantics-enhanced MSB

6 RELATED WORK

The concept of a manufacturing service bus was first introduced by Biennier et al. [11]. This work extends the concept of ESB, adapting the architecture to the manufacturing context. Special attention is paid to the monitoring of QoS parameters. Our work is oriented on the adoption of SLM in the integration middleware in order to achieve a higher degree of automation in serviceoriented computing environments for adaptive manufacturing. In this direction, current research activities focus on different areas, namely (i) semantic BPM, (ii) dynamic routing and (iii) event-based SOA.

Semantic Business Process Modeling (SBPM) aims to achieve a higher degree of automation in BPM by using semantic technologies. The functional requirements for each phase of the BPM lifecycle and the benefits of adopting semantic technologies are explored in [9]. The major benefits are automated service discovery and enabling dynamic binding of services to process tasks during process execution. A reference architecture [8] and implementation of a SBPM system has been carried out within the SUPER project [12]. The integration layer is based on a semantic service bus. This contribution is a conceptual architecture and focus on service orchestrations more than routing technologies.

Dynamic routing in service-oriented architectures is another area of research that matches the agility requirements for integration platforms in adaptive manufacturing environments. In this area, dynamic routing processes can be implemented as processes, like in [13], where SOAP message routing logic is expressed in terms of processes, which enables routing by SOAP message processing. In [14], a review of current efforts to adopt content-based routing in SOA is made. These efforts focus on incorporating publish/subscribe technologies, such as the WS-notification standard. However, in most approaches, semantics are not used or only used for service discovery.

A combination of both architecture styles SOA and EDA is introduced in [15]. The result is a model that uses Event-driven Process Chains representing a standardized, event-centric business process notation for modeling the initial processes, which are then transformed into a BPEL process, the web service standard language for executable business processes. This approach uses events to trigger execution of individual business activities, whereas our approach routes events by means of a mediation service bus. The incorporation of semantics to the MSB enables a higher degree of automation, which is one of the fundamental requirements in adaptive manufacturing.

7 CONCLUSIONS

The vision of adaptive and knowledge-based manufacturing can only be implemented if the integration infrastructure in a manufacturing environment includes the right SLM, adaptability and automation mechanisms.

The benefits of adding semantics is pervasive in the entire lifecycle of a service. We propose a semantic service framework to implement an SLM system that allows managing services and establishes a knowledge base for service dependencies. This extension will permit to support EAI process modeling tools, thus optimizing the reconfiguration of processes. The presented MSB semantic extension provides the means to adapt the MSB execution infrastructure to changing EAI process models.

In highly dynamic environments, the desired level of automation can only be achieved if service discovery and process reconfiguration are supported by the adequate SLM. We have shown how the semantic service framework can be incorporated into an integration platform, namely the MSB, improving the adaptability and agility of the platform.

8 ACKNOWLEDGMENT

We extend our sincere thanks to the German Research Foundation for financial support of the project within the Graduate School of Excellence advanced Manufacturing Engineering and to the reviewers for the valuable hints.

REFERENCES: **1.** *Jovane, F., Westkämper, E., Williams, D.,* 2009, The ManuFuture Road, Springer Verlag, Berlin. **2.** *Austvold, E., Carter, K.,* 2005, Service-Oriented Architectures: Survey Findings on Deployment and Plans for the Future, AMR Research. **3.** *Chappell, D. A.,* 2004, Enterprise Service Bus, O'Reilly, Sebastopol, CA. **4.** *Papazoglou, M., Traverso, P., Schahram, D., Leymann, F.,* 2007, Service-Oriented Computing: State of the Art and Research Challenges, IEEE Computer. **5.** *Minguez, J., Lucke, D., Jakob, M., Constantinescu, C.; Mitschang, B., Westkämper, E.,* 2009, Introducing SOA into Production Environments: The Manufacturing Service Bus, Proceedings of the 43rd CIRP International Conference on Manufacturing Systems (ICMS), pp. 1117-1124, Vienna, Austria. **6.** *Sheth, A. P.,* 1999, Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics, Interoperating Geographic Information Systems. C. A. Kottman, Kluwer Academic Publisher: 5-29. **7.** *Papazoglou, M.,* 2008, Web Services: Principles and Technology. Pearson, Prentice Hall, Harlow. **8.** *Karastoyanova et al.,* 2008, A Reference Architecture for Semantic Business Process Management Systems, Multikonferenz Wirtschaftsinformatik. **9.** *Wetzstein, B. et al,* 2007, Semantic Business Process Management: A Lifecycle Based Requirements Analysis, Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management, ISSN 1613-0073. **10.** *Cardoso J., Sheth, A. P.,* 2006, Semantic Web Services, Processes and Applications, Springer. **11.** *Biennier, F., Ali, L., Legait, A.,* 2007, in IFIP International Federation for Information Processing, Volume 246, Advances in Production Management Systems, eds. Olhager, J., Persson, F., (Boston: Springer), pp. 87–94. **12.** SUPER – Semantics Utilized for Process Management within and between Enterprises, www.ip-super.org **13.** *Scheibler, T., Karastoyanova, D., Leymann, F.,* 2009, Dynamic Message Routing Using Processes, Proceedings of 16th Fachtagung Kommunikation in Verteilten Systemen (KiVS 09). **14.** *Cugola, G., Di Nitto, E.,* 2008, On adopting Contentbased Routing in service-oriented architectures, Information and Software Technology, Elsevier Science, ISSN: 0950-5849. **15.** *Wieland, M., Martin, D., Kopp, O., Leymann, F.,* 2009, SOEDA: A Methodology for Specification and Implementation of Applications on a Service-Oriented Event-Driven Architecture, International Conference on Business Information Systems (BIS 2009).