A FRAMEWORK FOR ADAPTIVE DATA INTEGRATION IN DIGITAL PRODUCTION

Tobias Meisen[1], Rudolf Reinhard[1], Daniel Schilberg[1], Sabina Jeschke[1]
[1]Information Management in Mechanical Engineering of the RWTH Aachen, Dennewartstraße 27, Aachen, Germany

Abstract
Modern production processes' complexity increases steadily. Therefore, virtual planning has been prevailed as a method used to evaluate risks and costs before the concrete realization of production processes. In doing so, virtual planning uses a number of numerical simulation tools that differ in the simulated production techniques as well as in the considered problem domains. Users may choose between tailor-made, thus costly, simulation tools delivering accurate results and off-the-shelf, thus less costly, simulation tools causing post-processing efforts. Thereby, simulating a whole production process is often hardly realizable due to insufficient prediction accuracy or the missing support of a production technique. The supposed solution of interconnecting different simulation tools to solve such problems is hardly applicable as incompatible file formats, mark-up languages and models describing simulated objects cause an inconsistency of data and interfaces. This paper presents the architecture of a framework for adaptive data integration that enables the interconnection of such numerical simulation tools of a specific domain.

Keywords:
Semantic Information Integration, Ontology, Service-Oriented-Architecture, Semantic Services

1 INTRODUCTION

The increasing complexity of modern production processes requires their simulation before they can be implemented into a real environment. Besides, the requirement of a lasting optimization in production processes is best achieved by making use of simulations [36][38]. Because of high costs and time needs, it is hardly possible to realize such simulations or optimizations by experiments representing the complete production process. Hence, the usual approach to realize these with minimized experimental effort consists in the use of computational simulation tools. Unfortunately, because of the highly specific types of these simulation tools, the underlying models are heterogeneous and therefore incompatible in most cases. Furthermore, these tools mostly regard single aspects of the production process, like heat treatment or welding aspects, and do not consider information gained from simulation results of previous steps. Consequently, interactions of effects on different scales are oversimplified or conditionally included in the used simulation models (e. g. material models). Thus, it is necessary to manually link the simulations by creating translation tools and providing the needed infrastructure. In the field of simulations, the process development time using manually linked simulations incurs an overhead of 60% solely caused by the manual operation of the individual simulation components and translation tools of a simulation chain [30]. Hence, to benefit from the simulation, it is necessary to realize the interconnection in a semi-automatic way.

Facilitating such an interconnection of highly interdependent models firstly requires the identification and assessment of the character of interdependencies between the models. Secondly, translators considering these interdependencies and facilitating the interconnection between these models have to be created and thirdly, the degree of data resolution necessary for adequately representing the interdependencies has to be adjusted. Because the simulation model and its outcome highly depend on the concrete data, such a translator has to consider the data structure as well as its instantiation at runtime.

In this paper, a framework is presented that uses adaptive data integration to provide base implementations of methods and structures that facilitates the interconnection of heterogeneous simulation tools. In contrast to existing solutions, the framework and its translators use domain-specific knowledge for realizing the interconnection. It has been successfully adopted to implement the AixViPMaP® (Aachen (Aix) Virtual Platform for Materials Processing) [7].

The paper is structured as follows: In section 2, the current state of the art will be outlined in order to provide a foundation for the following sections. Subsequently, an overview of a concrete case study of the framework will be described in section 3 that will be used to explain the framework as well as the underlying technologies and methods. Section 4 gives an insight into the architecture of the framework, whereas section 5 focuses on the adaptive data integration. In section 6, a conclusion and an outlook will be drawn from the insights generated in this paper.

2 STATE OF THE ART

Integration problems belong to the most frequented topics with reference to finding answers to questions which are raised across application boundaries [17][24]. The complexity of such integration problems, in particular in the domain of simulation tools, arises by reason of the many topics that have to be regarded to provide a solution: Besides application interconnection on the technical level, the data has to be propagated and consolidated. Furthermore, user interfaces are required to model the underlying process and to visualize the data for the purpose of analysis. In addition, the integration of data requires the knowledge and thus the comprehension of the underlying processes on a domain expert's level. Because of those reasons, integration solutions are often specialized and highly adapted to the specific field of application. One example of such a solution is the Cyber-Infrastructure for Integrated Computational Material Engineering (ICME) [2] concerning the interconnection of MATLAB applications. Other examples are solutions that require the making of adjustments on the source level of the application, like CHEOPS [29] or the FlowVR toolkit [19]. Yet others require the implementation of standards like SimVis [15]. Realizing a flexible solution, the technical, the data and the semantic level have to be taken into account.

On the technical level, simulation tools are special software components that run on corresponding hardware resources. Starting with remote procedure calls (RPC) [27], the Common Object Request Broker Architecture (CORBA®) [11], web services and concepts like service-oriented architectures (SOA) [31] or ones specific to grid computing such as the open grid services architecture (OGSA) [25], there are currently a number of different concepts at various levels of abstraction for creating a distributed software architecture. Hence, the whole area of "simulation coupling" presents a heterogeneous landscape of concepts and different views as to what is meant by "simulation coupling".

On the data and on the semantic level, different research areas are working on the raised problems and promising contributions have been provided in the last decades. Application integration is concerned with the development and evaluation of methods and algorithms that allow application functionalities to be integrated along processes [18][32]. Information integration, in turn, deals with the evaluation of methods and algorithms that can be used to merge information from different sources [17][28]. Data warehouses are a popular example of the use of information integration methods. Another example is the information integration of meta-search engines, which gather and display information from numerous search engines. Both application and information integration, have in common that they can only be successful if the heterogeneity between the pieces of information or applications that are shared can be overcome. A variety of preliminary studies have identified different heterogeneity conflicts [12][39] that can be generally classified as syntactic, structural or semantic. In the past, a variety of methods and algorithms have been developed to overcome these conflicts. In particular, the definition of data exchange standards is often proposed as one possible solution. In the field of production technology, numerous standards have been introduced including the *Initial Graphics Exchange Specification (IGES)* developed in the United States, the French standard for data exchange and transfer "*Standard d'Exchange et de Transfer (SET)"* and the German neutral file format for the exchange of surface geometry within the automobile industry "Verband Deutscher Automobilhersteller – Flächen-Schnittstellen (VDAFS)" [23]. Such standards are usually limited to specific disciplines and this inhibits all-embracing, cross-disciplinary integration. In turn, the Standard for the Exchange of Product Model Data (STEP) aims at the definition of a standard that is not limited to specific disciplines [35]. Such standards are characterized by complex specifications and by a slow realization of necessary adjustments [18]. Although the complexity is often not needed by specific disciplines, it is essential for the realisation of the interconnection as the standard needs to be supported by each tool.

More promising solutions developed within the last years have been results of research projects involving semantic technologies. The already mentioned Service-Oriented-Architecture (SOA), which is an essential model for developing software out of reusable and distributed services [13], might serve as an example. SOA has been commonly lauded as a silver bullet for application integration problems [4][5]. In traditional scenarios, where workflow and business processes rely on syntactically specified and fix processes, services have been a way to facilitate a loose coupling between interacting services using adaptors or mediators as translators between different models and formats. Thereby, as already described, it requires substantial manual effort to define such adaptors and mediators. Semantic Web Services have been a proposal to provide formal declarative definitions of the semantics of services and to facilitate a higher level of automation in using services [5]. However, such approaches are in need of semi-automated and automated concepts to search and locate services as well as to select and compose them to handle a given task (e.g. the translation of data, so that it can be used by another simulation). Thereby, semi-automatic or automatic service composition requires information about the service's semantics and the used data structures. Several conceptual models [33] like SAWSDL [14], WSDL-S, WSMO and OWL-S to describe the semantics of a service and frameworks like WSMX [22] and METEOR-S [21] to provide the base functions for discovery, selection, ranking, composition, orchestration and invocation of services have been proposed. However, many scenarios (cf. chapter 3) require the consideration of conditions during service composition that are only knowable at runtime. Hence, common solutions for service selection cannot be employed. Instead, the framework uses a similar approach as presented in [1] facilitating the replacement or the selection of services that fit in a current context. Therefore, the traditional approach, which comprises the modifying of process models with branches mapping all possible contexts, becomes unnecessary.

3 CASE STUDY

As described in chapter 1, the aim of the framework is to provide a generic, flexible solution for interconnecting heterogeneous numerical simulations so that the simulation of whole production processes becomes possible. In this section, an example scenario is described that will be used in the following sections to explain the architecture as well as the underlying technologies and methods. The example is an extract of the scenarios implemented in the AixViPMaP®.

In the test case, the production process of a line pipe focusing on the material models using different heterogeneous simulation tools has been the object of consideration. The simulation has to consider the macro- and the micro-level. The process and the involved simulation tools are depicted in Figure 1.

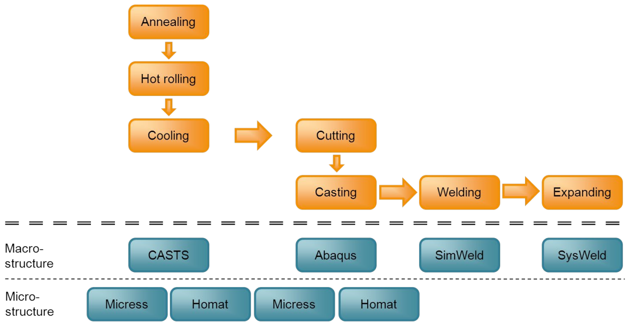

Figure 1 : Simulation process and simulation tools

The simulation process starts with the simulation of the annealing, the hot rolling as well as the controlled cooling using the software tool CASTS. In the next step, the cutting and the casting of the line pipe are simulated with the aid of Abaqus. The welding and the expanding of the line-pipe are simulated via SimWeld, a tool which has been developed by the Welding and Joining Institute (ISF) of the RWTH Aachen University, and via SysWeld, a software product contrived by the ESI-Group [16]. Furthermore, the simulation of modifications in the microstructure of the assembly will be realized by making use of Micress [37] and Homat [3], which were both developed by Access e. V.

All simulation tools that are employed on the macro level are based upon the finite element method. Hence, each tool requires a finite element model as input enriched by tool-specific configuration data. In the following, it is described how the translation process between the models of the simulation tools Abaqus and SimWeld can be realized and how it depends on the instantiation of the finite element model. Two example translation processes are visualized in Figure 2.

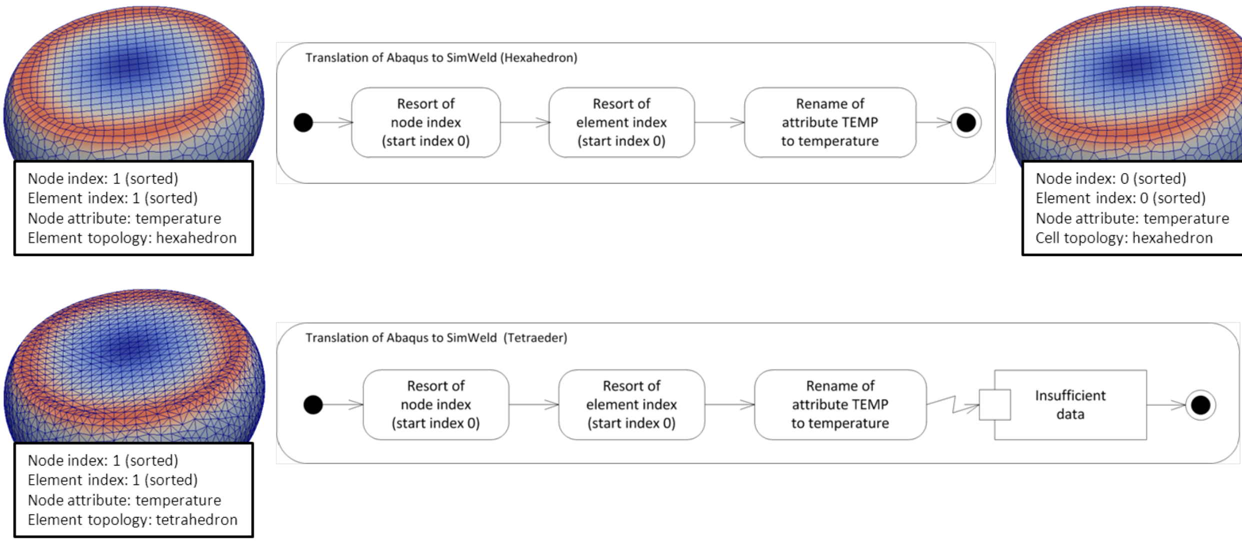

Figure 2: Translation process for different cell topologies

Both Abaqus and SimWeld use different start indexes to uniquely indicate the nodes and elements of the finite element model. In addition, the labelling of spatially resolved properties differs (e.g. temperature). Hence, the translation process has to resort the indices of the nodes and the elements and it has to rename the labelling of relevant properties. More complex is the handling of the element topology. Because SimWeld does only support hexahedrons as element topology, the given finite element model has to consist of elements of this type or it has to be translated by the translation process. In the case of tetrahedrons, such a translation cannot be realized without making use of remeshing methods. For this example, it is assumed that such a remeshing is not available. Hence, the translation cannot be realised for such an initial state.

In traditional approaches, a translation process for each possible initial and target state would be deployed. Unknown states could not be handled, even if the needed transformation would be available. Hence, instead of using the traditional approach, the framework uses semantic technologies to identify the initial and the target state and to identify the required translation process.

4 ARCHITECTURE

The architecture of the framework is based upon a message-oriented approach. Thereby, messages are employed to facilitate the communication between the services by making use of a message bus. A message contains all information that is necessary to execute a service. This includes the message header containing technical data like information about the sender and the receiver as well as the message body that involves data semantically related to the simulation tool that has created the data. Following the example (cf. section 3), the input and the output data of the simulation tools are written into different data files. Hence, the message body has to contain information about the location of such files.

Messages are exchanged by services following a defined process, whereby the process does not determine which service has to be executed. Instead, the framework executes a generic translation process each time a simulation tool produces data that have to be used by another simulation tool. The translation process focuses on the translation of the different kinds of data by considering the target format and, optionally, the storage of data for a continuous analysis after the simulation process has been finished completely. Hence, a central functionality of the framework is to provide and to monitor such generic translation processes. A translation process has to be able to resolve differences between the supported data formats. If numerous applications had to communicate with each other, such a generic translation process would have to consider each possible pair of applications, which would in turn result in a high complexity of such a process. The complexity can be reduced by dividing the required functionality into different services and by introducing a so-called Canonical Data Model [26]. Such a model provides an additional level of indirection between applications' individual data formats. Following the service-oriented approach, a translation process can then be described by composing three types of services: a service for integrating data into the Canonical Data Model, which is called Integrator, a service for extracting data from the Canonical Data Model, which is called Extractor, and a service to transform the data in such a way that it can be extracted. Figure 3 summarizes this approach.

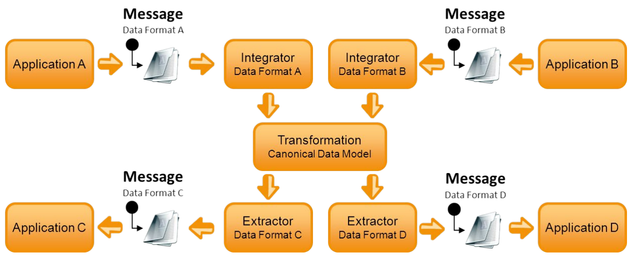

Figure 3: Service-oriented translation process

The concrete Integrator and Extractor have to be specified during runtime as the concrete simulation process is not known previously. Hence, the Integrator and the Extractor can be defined as so-called service templates [1]. The transform service is more complex than the Integrator and the Extractor and cannot be described by a service template. Instead, the transformation of data itself requires the composition of concrete services, whereby, unlike the generic translation process, the required service templates are unknown to the greatest possible extent. The determination of the service composition and the execution is the main functionality of the adaptive data integration, which is outlined in detail in section 5. The described functionality is summed up by the integration layer of the framework.

Besides the translation of data, the framework comprises another functionality required to facilitate the linking and the execution of the simulation tools on the technical level. As described in section 2, a wide range of solutions has been focused by the research. Instead of selecting one solution, the framework facilitates the usage of different solutions considering the domain-specific requirements. Therefore, a gateway [26] is used that connects the selected middleware to the integration layer. The gateway extends the middleware functionality, namely the interconnection of different simulation resources, by functions that facilitate a message-oriented communication. Within the AixViPMaP®, a gateway for the application-oriented middleware Condor [10] has been implemented and integrated into the framework. Summarized, the architecture of the framework is depicted in Figure 4.

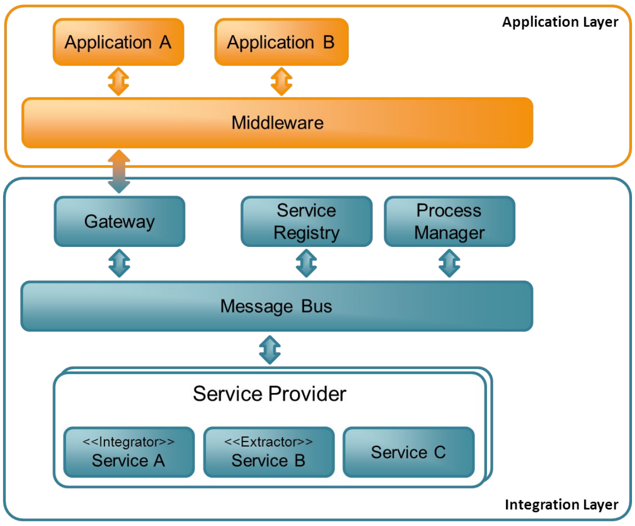

Figure 4: Architecture of the framework

The figure extends the previously described layers and components by a service registry, a service provider and a process manager. The service registry is a central service used to find, register and unregister service providers and their corresponding services. The service provider is a collection of different services and facilitates the overcoming of the technical heterogeneity between the message bus and concrete services. It contains service activators [26] for different service protocols like the web protocols SOAP and WSDL and implements required functionality like the registration in the service registry. Hence, concrete service implementations do not have to implement any specific interfaces to work within the framework. The process manager provides the functionality to monitor and to execute translation processes. It triggers the Integrator and the Extractor services as well as the adaptive data integration.

5 ADAPTIVE DATA INTEGRATION

The adaptive data integration focuses on the automated determination of a service composition considering the current context of data. The main goal is to overcome the structural and semantic heterogeneity by considering domain-specific knowledge so that the data can be handled by an extraction service. Thereby, an extraction service does not contain any validation or transformation steps, with the exception of syntactical transformations (e.g. number format). Instead, the adaptive data integration supports the extraction service with regard to the extraction of data and the loading of it into the desired format without being dependent on a complex rule-based translation process. The determination of the required service composition is realized within three steps (cf. Figure 5).

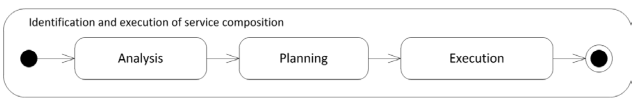

Figure 5: Process for service composition and execution

First, the existing data is analysed. The goal of the analysis is the determination of so-called features that are fulfilled by the data. A feature is domain-specific and expresses a structural or semantic property that is satisfied by the data. In addition, the analysis step determines features that have to be fulfilled by the data to satisfy the requirements of the specific output format. Second, following the analysis step, planning algorithms are used to find a data translation process that transforms and enriches data in such a way that the transformed data fulfils the requested features. After the planning is finished, the determined service composition is executed in a third step. The transformation algorithms are realized as services. Hence, they are loosely coupled and can be run in a distributed environment. In the following, the different process steps are described in detail.

The analysis step serves the purpose of evaluating a given set of data and of identifying domain specific features fulfilled by this set. It is implemented into a so-called Analyser service that represents, similar to the Integrator and Extractor service, a template for services with such functionalities. Therefore, first of all, the structure and the semantics of the data have to be specified in a formal, explicit specification so that this information can be consulted by the analysis. In addition, the features that have to be evaluated have to be specified in a similar way. Following the example presented in section 3, possible features are the cell topology of the finite element model, properties of the node and the element index, like the starting number, or the dimensionality of the data. The presented approach makes use of the Web Ontology Language (OWL) [8] to specify the features and the data structure with the help of ontologies. Figure 6 illustrates an excerpt of the data structure ontology.

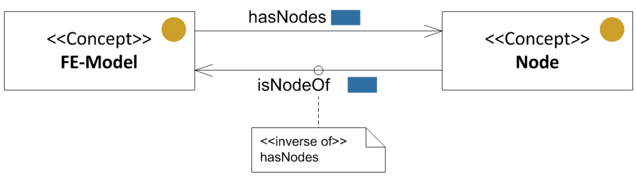

Figure 6: Example of data structure conceptualisation

By making use of ontologies, the fulfilment of a feature can be determined by reasoning. Because of the huge amounts of data that have to be analysed, the framework contains a so-called ontology-based analyser. Similar to the KAON2 reasoner [20], it supports the definition of mappings between the data structures specified in the ontology and a relational data model. However, instead of defining the mapping in a separated configuration file, this analyser supports ontology-annotations. Hence, instead of being limited to the KAON2 reasoner, different reasoners can be used. Figure 6 shows the example depicted in Figure 7 extended by annotations.

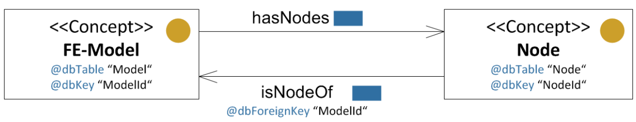

Figure 7: Annotations for data structure conceptualisation

The ontology-based analyser employs this information to evaluate the given data. Until now, the analyser only supports relational database and the query language SQL. Figure 8 depicts the concept of the ontology-based analyser.

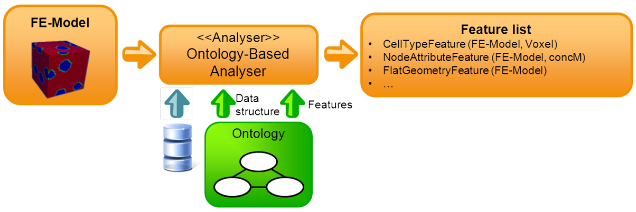

Figure 8: Example of the ontology-based analyser service

Subsequent to the analysis, the planning step is executed. The aim of this step is to identify a composition of available transformation services that transforms and enriches the data in such a way that the requirements of the subsequent Extractor service, defined by a feature list, are fulfilled by the data. The approach makes use of SAWSDL (Semantic Annotations for WSDL and XML Schema) [14] to specify the services' semantics. The preconditions and effects of the different transformation services are formally specified in OWL ontologies. The result of the planning is the required translation process that is finally executed by the process manager. The process is exemplarily visualized in Figure 9.

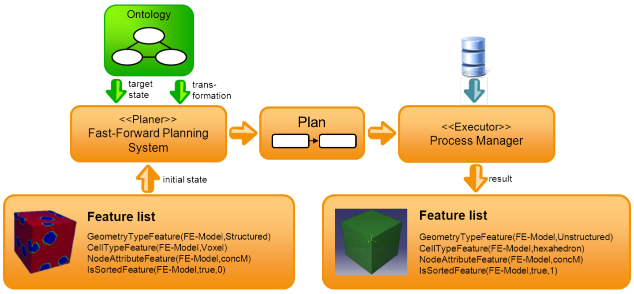

Figure 9: Example of planning and execution

For the realisation of the planning, the fast-forward planning system [34], which has been integrated into the framework, is used. In the example formerly presented within this paper, the initial mesh is structured and consists of voxel elements. In addition, it has a node property named 'concM'. The indexing of the nodes and elements is closed, which means there are no gaps in the numbering of the index, starting with zero. For extracting the data into the desired target format, the mesh has to fulfil a set of features defined in the target state. These features determine that the mesh has to be unstructured and that it consists of hexahedrons. Besides, the indexing of nodes and elements has to be closed and started by one. Hence, the planning algorithm would determine a plan that contains three steps. First, the structured mesh would be transformed into an unstructured one. Second, the cell topology would be translated from voxel to hexahedron and at least, the node and element indices would be resorted so that they start by one.

6 CONCLUSION

In this paper, the architecture of a framework which facilitates the interconnection of heterogeneous numerical simulations has been presented. In the process, adaptive data integration assures data consistency at the syntactical, structural and semantic level. The framework has been successfully used to implement the AixViPMaP®, which allows new ways of exploration and investigation of simulation data at an inter-simulation tool level.

As presented, the complexity of the researched case study is determined by the complex data structures and semantics that have to be analysed to find appropriated services. The framework handles this complexity by providing the concept of adaptive data integration. This facilitates the analysis of data and the determination of features that are used to define the requirements of each simulation tool. By describing the set of features that are changed by a transformation service, the framework supplies analysis, planning and execution services to identify if a translation process exists and how it has to be executed in case of its existence.

Currently, the framework focuses on the problem of data heterogeneity and does not consider additional information like the Quality of Service (QoS) to evaluate a determined translation process. In the future, the framework could be extended by such aspects. Promising solutions for QoS can be found in [6][9]. However, because the framework is currently not to be used in open environments, the consideration of quality aspects is not the main focus.

Instead, the framework will be extended by an analysis layer to provide a complete solution for simulation interconnection and integrated analysis - a so-called Virtual Production Intelligence solution.

7 ACKNOWLEDGMENTS

The approaches presented in this paper are supported by the German Research Association (DFG) within the Cluster of Excellence "Integrative Production Technology for High-Wage Countries".

8 REFERENCES

[1] Pedrinaci, C., et al., 2010, Adaptive Service Binding with Lightweight Semantic Web Services, Service Engineering: European Research Results, pp. 233–260.

[2] Haupt T., 2010, Cyberinfrastructure for Integrated Computational Material Engineering, Proceedings of Magnesium Technology 2010, pp. 77-82.

[3] Laschet G. and Apel M., 2010, Thermo-elastic homogenization of 3-D steel microstructure simulated by phase-field method, Steel Resarch int., vol. 81, n. 8, pp. 637-643.

[4] Krafzig, D., 2010, Serviceorientierte Architekturen (SOA), Informationsverarbeitung in Versicherungsunternehmen, Springer Berlin Heidelberg (Springer-Lehrbuch Masterclass), pp. 163–174.

[5] Pedrinaci, C., Domingue, J. and Sheth, A., 2010, Semantic Web Services, Handbook of Semantic Web Technologies, ed. 2, Springer.

[6] Xin, Y., Weise, T., Ke, T., Bleul, S., 2010, QoS-aware semantic web service composition for SOAs, IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp. 1–4.

[7] Schmitz G. and Prahl U., 2009, Toward a virtual platform for materials processing, JOM Journal of the Minerals, Metals and Materials Society, vol. 61, pp. 19-23.

[8] W3C OWL Working Group, 2009, OWL 2 Web Ontology Language, Hg. v. W3C.

[9] Aiello, M., et al., 2009, Optimal QoS-Aware Web Service Composition, Commerce and Enterprise Computing, CEC '09, pp. 491–494.

[10] Cerfontaine P., Beer T., Kuhlen T., Bischof C., 2008, Towards a flexible and distributed simulation platform, ICCSA '08: Proceedings of the international conference on Computational Science and Its Applications, Part I, pp. 867-882.

[11] Henning M., 2008, The rise and fall of CORBA, Communications of the ACM, vol. 51, no. 8, pp. 52-57.

[12] Gagnon M. 2008, Ontology-based integration of data sources, Information Fusion, 10th International Conference on, pp. 1-8.

[13] Erl T., 2008, SOA. Principles of Service Design, Upper Saddle River, N.J, Prentice-Hall.

[14] Kopecky J., Vitvar T., Bournez C., Farrell J., 2007, SAWSDL: Semantic Annotations for WSDL and XML Schema, IEEE Internet Computing 11, pp. 60–67.

[15] Kalden O., Fritzen P. and Kranz S., 2007, SimVis - A Concurrent Engineering Tool for Rapid Simulation Development, Proceedings of Recent Advances in Space Technologies, RAST '07, pp. 417-422.

[16] Rossiter E. and Mokrov, O., Integration des Simulationspaketes SimWeld in FEM Analyseprogramme zur Modellierung von Schweißprozessen, Sysweld Forum.

[17] Halevy A., Rajaraman A. and Ordille J., 2006, Data integration: the teenage years, Proceedings of the 32nd international conference on Very large data bases (VLDB), pp. 9-16.

[18] Conrad S., 2006, Enterprise Application Integration: Grundlagen, Konzepte, Entwurfsmuster, Praxisbeispiele, 1. Aufl. München, Heidelberg: Elsevier Spektrum Akad. Verl.

[19] Allard J. and Raffin B., 2006, Distributed Physical Based Simulations for Large VR Applications, Proceedings of the IEEE Virtual Reality Conference, Alexandria, USA, VR ‘06, pp. 98-96.

[20] Motik B., Sattler U., 2006, A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes, Logic for Programming, Artificial Intelligence and Reasoning, vol. 4246, Springer Berlin / Heidelberg, pp. 227-241.

[21] Verma K., et al., 2005, METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services, Journal of Information Technology and Management, Special Issue on Universal Global Integration, vol. 6, no. 1, pp. 17-39.

[22] Haller, A., et al., 2005: WSMX - A Semantic Service-Oriented Architecture, Proceedings of the IEEE International Conference on Web Services, pp. 321-328.

[23] Eversheim W. and Schuh G., 2005, Integrierte Produkt- und Prozessgestaltung. Berlin, Springer.

[24] White C., 2005, Data Integration: Using ETL, EAI and EII Tools to Create an Integrated Enterprise, tech. rep., The Data Warehousing Institute.

[25] Foster I., et al., 2005, The Open Grid Services Architecture, Version 1.0, Technical report, Global Grid Forum (GGF).

[26] Hohpe, G., 2004, Enterprise integration patterns, 14. print. Boston, Mass: Addison-Wesley.

[27] Reynolds J. and Ginoza S., 2004, Internet Official Protocol Standards, Internet Official Protocol Standard RFC, http://tools.ietf.org/rfc/rfc3700.txt, last access on 22.02.2011.

[28] Visser U., 2004, Intelligent information integration for the Semantic Web, Berlin, New York: Springer.

[29] Schopfer G., et al., 2004, CHEOPS: a tool-integration platform for chemical process modelling and simulation, International Journal on Software Tools for Technology Transfer, vol. 6, iss. 3, pp. 186-202.

[30] Lang U., 2003, Process Chain Integration for Simulation and Data Processing. Space Grid Workshop.

[31] Natis Y., 2003, Service-Oriented Architecture Scenario, Gartner Research Note AB-19-6751.

[32] Gorton I., Thurman D. and Thomson J., 2003, Next Generation Application Integration: Challenges and New Approaches, COMPSAC '03: Proceedings of the 27th Annual International Conference on Computer Software and Applications, p. 576.

[33] van der Aalst, W. M. P., 2003, Don’t go with the flow: web services composition standards exposed, IEEE Intelligent Systems 18, vol. 1, pp. 72-76.

[34] Hoffmann J., 2001, FF: The Fast-Forward Planning System, AI magazine 22, pp. 57-62.

[35] Anderl R. and Trippener D., 2000, STEP: standard for the exchange of product model data eine Einführung in die Entwicklung, Implementierung und industrielle Nutzung der Normenreihe ISO 10303 (STEP), Stuttgart, Leipzig, Teubner.

[36] Belytschko T. and Black T., 1999, Elastic Crack Growth in Finite Elements with Minimal Remeshing. International Journal for Numerical Methods in Engineering, vol. 45, iss. 5, pp. 601–620.

[37] Laschet G., Neises J. and Steinbach I., 1998, Micro-Macrosimulation of casting processes, 4iéme écoled' été de ‘Modelisation numerique en thermique’, vol. C8, pp. 1-42.

[38] Zacharia T., et al., 1995, Modeling of Fundamental Phenomena in Welds, Modelling and Simulation in Materials Science and Engineering, vol. 3, iss. 2, pp. 265–288.

[39] García-Solaco M., Saltor F. and Castellanos M., 1995, A Structure Based Schema Integration Methodology, Proceedings of the 11th International Conference on Data Engineering, pp. 505-512