Chapter XVII
Model-Driven Integration in Complex Information Systems:
Experiences from Two Scenarios

Sven Abels
FlexaDot Information Systems, USA

Wilhelm Hasselbring
University of Oldenburg, Germany

Niels Streekmann
OFFIS – Institute for Information Systems, Germany

Mathias Uslar
OFFIS – Institute for Information Systems, Germany

ABSTRACT

This chapter introduces model-driven integration in complex information systems by giving two practical examples. It relies on the experiences the authors have made in two different research projects at the public utilities domain. The chapter starts with a short introduction of the general problem domain and it gives detailed background information about the current state of the art in model-driven integration. Afterwards, the two research projects are introduced. The purpose of the first project (MINT) was to provide an integration approach allowing interoperability among several different legacy systems. Hence, the project itself was only acting as a "bridge" between the systems. The second project (DER) was built from scratch and got the challenge of integrating several existing third party systems into the newly designed system. In this project, the main system is a core element and only needed to integrate existing legacy systems for specific tasks.

1. INTRODUCTION

Business processes today usually involve several different information systems. A study by Marx Gómez and Brehm in 2007 with 658 participating SME companies in Germany turned out that 90.3% of all companies are using more than one product for their financial business needs (i.e. ERP related tasks). More then 53% are using 4 or more products and almost 15% of all companies are using 10 or more different products, most of them being produced by different software vendors (Details can be found in Marx Gómez & Brehm, (2007). Considering those figures reveals the strong need for integrating different software systems into a coherent solution. This is usually achieved by creating interoperability between software systems. As defined by the IEEE, interoperability is „the ability of two or more systems or components to exchange information and to use the information that has been exchanged" IEEE, (1990). At the I-ESA 2007 conference, Jeusfeld argues that this topic is often neglected in the design of modern information systems (Jeusfeld, (2007)). For example, he describes that the well known Software Engineering Body of Knowledge (Swebok, (2004)) mentions interoperability only twice, once as an example for a system requirement and the second time as a title of a standard library of data models. Interoperability and the possibility to integrate different heterogeneous systems in a coherent architecture is, however, a key of the MDA strategy as defined by the OMG (Object Management Group) (OMG, (2003)). The following sections focus on this complex area and they put it in context of the model-driven software development (MDSD). This chapter demonstrates how to cope with integration and interoperability issues by explaining the intermediate results of two research projects, namely MINT and DER.

The examples, used in this chapter are settled in the public utilities domain. Within utilities, several systems have a very long lifetime compared to systems from other domains. Once a company has chosen a SCADA (supervisory control and data acquisition system), it is unlikely to ever change it again in the next years or decades. Therefore, one has to deal with a lot of legacy systems which have to be integrated in both technical and business-related systems. The critical aspect in this context is that modern software systems are expected to quickly adapt to changing business processes by considering quality and reliability issues at the same time. This requires a flexible yet robust architecture and an approach to easily connect and enhance information systems.

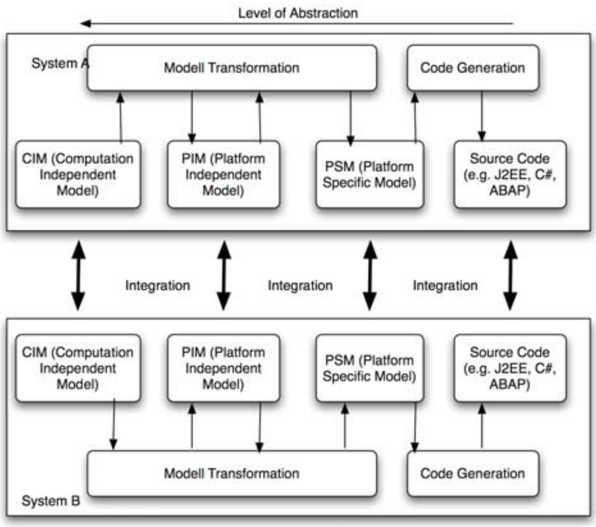
2. BACKGROUND ON THE TECHNIQUES USED

Integration of software systems may take place on different levels. The OMG defines CIM (Computation Independent Model), PIM (Platform Independent Model), PSM (Platform Specific Model) and code levels. Those are defined and described in detail earlier in this book. We will therefore focus on putting those levels into the domain of our specific problem of integrating information systems. Considering this, the following figure visualizes the current state of the art using CIM, PIM and PSM as different stages of abstraction.

The figure shows two different information systems with their levels of abstraction. Within one system, the OMG defines the following levels that can be distinguished when modelling, creating and refactoring systems in the Model-Driven Software Development (MDSD) approach:

- CIM which is an abstract description of the system, mostly created by domain experts.
- PIM that defines the "What and How" of an information system independently from the actual technology.
- PSM that describes the "What and How" in a technologic dependent model and

Figure 1. Integration on CIM, PIM, PSM and Code-Level (©2007 Sven Abel, Wilhelm Hasselbring, Niels Streekmann, and Mathias Ulsar. Used with permission)



- Code Level that contains the actual source code (e.g. C#, Oracle SQL, etc.) being the most technology dependent level of this approach.

The main idea of the overall approach is to use model transformation between CIM, PIM and PSM and automatic code generation between PSM and Code Level. This allows software engineers to perform software development on a high level of abstraction and to handle very complex information systems.

When dealing with more than one information system it is, however, necessary to connect those information systems in order to either

i. exchange information between two equal systems, integrating them into an overall system that consists of different components or to
ii. exchange information between one main system and several sub systems that are integrated into the main system seamlessly.

This integration may happen on basically all levels of abstraction. There can be an early integration on CIM level, PIM or PSM level as well as a message-based integration on Code-Level. The two major European projects in Interoperability issues, namely INTEROP[1] and ATHENA[2] explicitly distinguish between the different integration of those levels for example in (Elvesæter et al., (2005)).

Beside those different levels of integration Elvesæter et al. also suggest to distinguish between the integration of different aspects which they call "Conceptual Integration" (Elvesæter et al., (2005)). They distinguish between the integration of

1. **Information aspects**
   Information aspects are related to the messages or structures exchanged, processed and stored by software systems or software components
2. **Service aspects**
   Services are an abstraction and an encap-

sulation of the functionality provided by an autonomous entity

3. **Process (and Rules) aspects**
   Processes describe sequencing of work in terms of actions, control flows, information flows, interactions, protocols, etc.
4. **Non-functional aspects**
   Additional functional qualities that can be applied to services, information and processes

It should be mentioned that in this approach of the two projects, INTEROP and ATHENA, “Non-functional aspects” also includes issues such as quality and security. Especially in systems that need to be available 24/7, such as in the public utilities domain, the two topics Quality and Security are usually crucial issues that need to be solved when integrating information systems in order to guarantee a coherent and consistent solution.

From a technical point of view, the area of integration can be distinguished. This defines where the integration takes place (see e.g. Reussner, (2005)). Considering this, we can define the following technological areas of integration:

i. Integration at persistence level
ii. Integration at functionality level
iii. Integration at service or process level
iv. Integration at presentation level

In the past, practical approaches incorporated integration on the persistence level, e.g. different systems using the same database, or the function logic level, where applications directly called functions of other applications. Current approaches towards the integration of legacy systems focus on integration on the level of business processes using service-oriented architectures. These allow for the decoupling of functional aspects and their implementation.

3. CASE STUDIES FOR MODEL-DRIVEN INTEGRATION

In this section, we will present two case studies for performing a model-driven integration considering quality and scalability constraints, too. We introduce two research projects from the utilities domain. Afterwards, we will give a brief description on how to connect them.

MINT

The purpose of MINT (Model-Driven INTegration of Business Information Systems) was to provide an integration approach allowing interoperability among several different legacy systems. Hence, the project itself was only acting as a “bridge” between the systems.

Service Identification and Extraction

As stated in the last section, the integration in MINT will be performed at the service and business process level. The problem of this kind of integration is that the functionality provided by legacy systems was not designed and not implemented under the consideration of service-orientation. Actually, the legacy systems that needed to be integrated have been up to 10 years old and consisted of a monolithic architecture.

We use a practical approach to solve this problem by implementing adapters for legacy systems in the form of WebServices. To build these adapters and describe the orchestration of the WebServices we applied a model-driven approach. This approach based on the BALES methodology (van den Heuvel, (2000)). The BALES methodology combines forward and reverse engineering techniques in order to create models for the generation of adapters.

Integrating legacy systems and modern systems by employing service-oriented architectures and especially WebServices can be seen as a proven solution. The success of this method was shown in several case studies as in Hasselbring, (2004) and Zimmermann, (2005). The extension of the service-based integration by the generation of service adapters from architectural models is the subject of current research that has been carried out in our project. Besides the MINT project, there are some other approaches that use a similar approach as described in Winter, (2006) and Ziemann, (2006).

A challenge of the integration of legacy systems is to identify adequate services, because legacy systems were not designed to provide services for other applications. The identification and design of these services is a non-trivial problem since usually manifold requirements have to be considered. Service design should therefore follow certain rules. Hess, (2006) introduced such rules. The first rule is that the components that implement the services should be built by functional criteria. This principle is followed in MINT. Service adapters for legacy systems are derived from business process descriptions in the CIM. The challenge is to find the right granularity of a service and match the services derived from the CIM to the interfaces of legacy systems.

Business Requirements

Our approach follows the MDA standard and especially concentrates on the interdependence between CIM and PIM, because it defines the connection between business requirements and the interfaces of the legacy systems. The consideration of business requirements at CIM level allows us to add new functionality by considering new requirements. Of course this means that new requirements defined on CIM level must be transferred to the PIM level and afterwards the information needs to be used to create a service-interface that is able to provide WebServices for the legacy systems being involved in MINT. Thus, the proper transformation from CIM to PIM is one of the major aspects for the quality assurance of legacy integration. MINT thereby focuses on process descriptions on a computation independent level from which architectural models are derived. These models are combined with interface models of the legacy systems to generate service adapters and orchestrations. We propose the use of model-driven and generative techniques to achieve this. The first allows the quick and consistent change of the requirements of the system by domain experts. Generative techniques as described in Czarnecki, (2000) allow the fast implementation of a software system in a standardized engineering manner. In a model-driven environment this can be achieved by reusable transformations. These transformations encapsulate design decisions and recurring implementation details. Thus changing requirements can be implemented fast and traceable.

Considering Domain Experts

The integration of legacy systems on the process level has two important issues from the software architecture point of view. The first is the generation of adapters to fit a legacy system into a modern software architecture or modern system landscapes. The second is the orchestration of these services according to the business process.

An important quality aspect for process-based integration is the correct conversion of domain-specific concepts into the generated adapters. This requires the participation of domain experts and system experts in the integration process. The system experts need to model the interfaces of legacy systems in an interface-based model of these systems. The domain experts on the other hand model the processes that integrate legacy systems in modern applications and are needed to match the concepts of the different systems to gain a correct integrated system. The involvement of domain experts is the base of MINT to guarantee quality issues.

A related approach to include models in the development process and thereby ensure the quality of the resulting software is domain-driven design as described by Evans, (2004). The approach also puts domain models into the centre of the software development process, but does not consider the generation of code from domain models.

To support the work of domain experts and system experts and to bring their work together in a consistent way new tools are required. Model-driven development is a promising method to achieve this. Models are able to capture all necessary information for the integration of software systems. Views on the models adjusted to the corresponding stakeholders and transformations between models on different levels of abstraction make it possible to offer an adequate working environment to each expert.

Model Transformation

To realize adequate views on the same model special modelling languages for each expert have to be employed. The MDA standard (OMG, (2003)) recommends the usage of UML to describe software systems. In our point of view the UML is an appropriate language for many tasks of software architects and developers. But since the UML offers a very technical view on software systems, it is not appropriate for all tasks that need to be fulfilled by domain experts describing their view and requirements on the integrated system. For this reason, we propose the usage of domain-specific languages (DSL) for the computation independent viewpoint of the MDA standard.

Furthermore the transformation from computation independent models to architectural models is in our view an important part of the software development process. To assist this by the utilization of DSL and automated model transformations is a step towards an engineering approach to software development that incorporates all necessary steps from the definition of requirements to the generation of code.

The main influences on the software architecture are decisions of the software architect based on the requirements of domain experts. Architectural decisions therefore strongly influence the transformation from CIM to PIM. Since the CIM only includes domain knowledge and requirements, it is clear that the decision for certain architectural aspects cannot be made out of the CIM alone, but also requires the knowledge of experiences with the influences of architectural decisions. These can be encapsulated in CIM-to-PIM transformations and made configurable, which is proposed by Zimmermann, (2006). To weigh up the requirements and make a decision for a specific architecture is the task of the software architect, but the transformation can encapsulate former design decisions that proved of value in certain situations. E.g. architectural decisions can have a fixed influence on the usage and configuration of design patterns, which e.g. is proposed by Becker, (2006) to introduce an engineering approach to component adaptation.

In MINT, we have focussed on the quality of former and future architectural decisions based on both incorporating more existing knowledge from the legacy systems and the software engineers into future decisions. While this work mainly deals with quality as a non-functional requirement and takes code transformations into account to ease the development, the use case described in the next section will have a different focus.

DER

The project DER (Distributed Energy Resources) is a scientific project dealing with a lot of problems coming from the domain of renewable energies. Therefore, we are going to provide an introduction to the IEC 61970 Common Information Model CIM.

Due to the fact that the MDA also uses the word CIM as an abbreviation, we are going to change the abbreviation to IEC-CIM for this contribution. While MINT had a different focus which was driven by the software architecture rather than the domain, DER strongly takes the domain semantics into account. In the following, we are going to provide some insights to the new drivers which lead to new objects and therefore semantics to be exchanged between systems in the utility domain.

Legal Unbundling

Running an energy grid is a commerce often combined with generating energy. In the European Community, those monopolistic functions provided in certain domains like transport (railways), communications (cellular, phone) and energy are subject to a dissembly which should lead to new competitors entering the (formerly) monopolistic markets Bundesministerium für Wirtschaft (2004), European Parliament (2003). In the utility domain, there must be a dissembly of energy generation and energy distribution through grids. This applies to both electricity and gas.

Summarizing the impact of the legal unbundling, it becomes clear that the changed processes have new entry points for third-party participants which have to be satisfied using IT-technology. Unfortunately, all the hundreds of different formats used by those companies cannot be easily integrated and processed.

New processes are being developed and the whole communication structure must be changed. Databases formerly used by the now unbundled distribution and generation structures must be split and kept in sync while their data schemes must be dissembled. This imposes both a threat and a chance to the systems. The chance is that new data schemes and techniques can be incorporated which better fit the needs of the market and provide less efforts needed when developing new adaptors and interfaces for exchanging data with new systems or third-party companies Robinson, Greg (2002)a, Robinson, Greg (2002)b.

Different standards and frameworks have been developed all over the world to cover this needed communication and data exchange structure. The two most prominent ones are the NRECA MultiSpeak 2.0 (see MultiSpeak, 2003) standard and the IEC Common Information Model (IEC-CIM) standard (see IEC (2003), Podmore, Robin et al. (1999)). The IEC-CIM has proven to be the better choice (e.g. Neumann (2003)) and is being further described within this contribution.

Not only the new legal requirements impose changes, also environmental changes and increasing needs for new and different energy producers lead to changes in processes, data models and field level communication. The concept of decentralized energy producers like bio mass plants, wind power plants and fuel cells must be coordinated and their fed into the electricity power grid must be properly controlled and predicted Brand, Klaus-Peter & Buchholz, Bernd (2003). This leads to completely new function blocks and data models which have to be integrated in EMS and SCADA systems and must also interact with the commercial system like SAP.

To summarize the imposed requirements, we get to know that there is a strong need for coupling both new and old systems which have to deal with the proper semantics for the payloads. In order to use common semantics, we strongly make use of an existing domain ontology which is described in the next section.

IEC-CIM

The previous paragraph showed the two main drivers nowadays changing the IT-landscape in a utility company. Both data exchange processes and models heavily change with regard to complexity and sheer number of used standards. New standards must take this into account. As mentioned before, the IEC-CIM is the most prolific approach to deal with these problems. Due to the length of this chapter, more about the IEC-CIM and its object semantics has to be found in Uslar et al. (2005). Anyway, there are still many problems unsolved when adopting the IEC-CIM norm.

We have chosen the IEC TC 57 framework which incorporates the model as one sub-norm as the data model and communication standards framework. This lowers the amount of efforts which have to be spend on developing a domain model for the utility domain and market/substation communication. The IEC-CIM can serve as a global ontology for the utility domain and covers when converted into OWL (Web Ontology Language) about 2.500 RDF triples IEC (2004). Using the IEC-CIM ensures a high quality of the data format within the DER project since it is build on a solid base and since it has been evaluated and applied in various practical scenarios.

Currently, the IEC-CIM is mostly used for a message-based coupling and the exchange of power grid data. We are going to focus on the integration of heterogeneous system within this contribution. In order to achieve this, we created a RDF representation of the CIM format which will be transported as payload information using WebServices and SOAP, allowing all systems in the DER project to communicate and exchange information in a semantically standardized way.

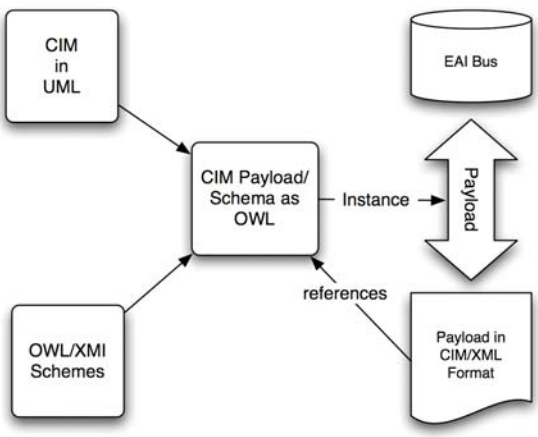
Representing IEC-CIM Using RDF and XML

Although the IEC-CIM itself is modelled as an UML diagram and provides useful insight to the important objects within the power industry, it is difficult to exchange data due to the fact that object related databases are available but not widely used. Instantiated objects must be represented via serialization formats which can be exchanged in binary or ASCII format.

The IEC proposed RDF as a proper way to exchange topology (power grid) data in a common format IEC (2004)b. The RDF schema is documented as the IEC standard 61970-501. Like any other XML based format, it has several advantages over binary formats. Due to XML based mechanisms, it is possible to extend the model with versioning mechanisms and, more important, namespaces as a mechanism which is easily extensible and supports site-specific needs.

RDF is both machine and human readable and self-describing, although it is primarily intended

Figure 2. Defining payloads for EAI and EMB in the context of CIM (©2007 Sven Abel, Wilhelm Hasselbring, Niels Streekmann, and Mathias Ulsar. Used with permission)



for programmatic access by tools which support the Document Object Model (DOM) API. Current web standards can be met when using a RDF based representation of the data.

IEC standard 61970-501 defines standards mechanisms to convert the UML model into an RDF model. The conversion of the Rational based UML file can be done manually or tool-supported by the Xpetal converter. It reads the Rational Rose .mdl file and creates RDF or XML schemes from the corresponding model. Still, one valid RDF representation can differ from other valid representations. Furthermore, RDF models often tend to become quiet complex having nested tags and a large overhead of administrative meta-data compared to the actual data used. Often, large files describing topologies have to be exchanged while e.g. only some of the breakers have changed. The IEC standard 61970-503 IEC (2004)b therefore defines a simplified RDF syntax being an actual subset of RDF but still valid RDF and an differential model providing the chance to exchange tiny subsets of changes instead of complete model status snapshots. The amount of data exchanged between energy management systems or even companies can become very large. Processing this data is most often time-critical.

Even though XML related data has proven to be useful due to its self-description capabilities (see Dag, Hasan & Urkan, Ulmut (2004), Zhou, E.Z. (2000)), nested tags instead of a simple serialized structure lead to slower process times as described in deVos, Arnold (2000). At implementation level, the IEC proposed a slightly changed syntax to a common RDF/XML representation, the so called simplified CIM/XML exchange format IEC (2004)b; deVos & Widergren & Zhu (2001). RDF provides many ways to represent the same set of data, e.g. an association between two resources can be written either with a resource attribute or by nesting one element within the other. This makes processing via XSLT tools sometimes a bit more difficult. The reduced syntax is still compatible with available RDF-de-serialization tools but provides a generally faster access to the data due to its pure simplicity. One improvement of the data structure is useful for the exchange of partial or full model data exchange. Another improvement in processing speed can be achieved by optimizing the amount of data exchanged

Figure 3. A simplified extract of a CIM/XML file modeling idle power (q) and effective power (p) (in RDF) (©2007 Sven Abel, Wilhelm Hasselbring, Niels Streekmann, and Mathias Ulsar. Used with permission)


```
  <?xml version="1.0" ?>
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:cim="http://iec.ch/TC57/2000/CIM-schema-cimu09b#">
  - <cim:Substation rdf:ID="_28159176D77E2467">
      <cim:Naming.name>LEGAZZI</cim:Naming.name>
      <cim:Substation.MemberOf_SubControlArea
        rdf:resource="#_55D3DE366B2AD032" />
    </cim:Substation>
  - <cim:EnergyConsumer rdf:ID="_2963867E4A4B1669">
      <cim:EnergyConsumer.pfixed>19.78</cim:EnergyConsumer.pfixed>
      <cim:EnergyConsumer.qfixed>06.10</cim:EnergyConsumer.qfixed>
      <cim:EnergyConsumer.LoadArea
        rdf:resource="#_9C1602456B178B75" />
      <cim:Equipment.MemberOf_EquipmentContainer
        rdf:resource="#_A9D1427B3784CD78" />
      <cim:Naming.name>4711</cim:Naming.name>
    </cim:EnergyConsumer>
  </rdf:RDF>
```

between companies. After the initial data is exchanged, only updates need to be exchanged deVos & Rowbotham (2001) afterwards.

This mechanism is mostly used within the scope of topology exchange, the mechanisms dealing with EAI messages differ a bit in terms of serialization and tooling.

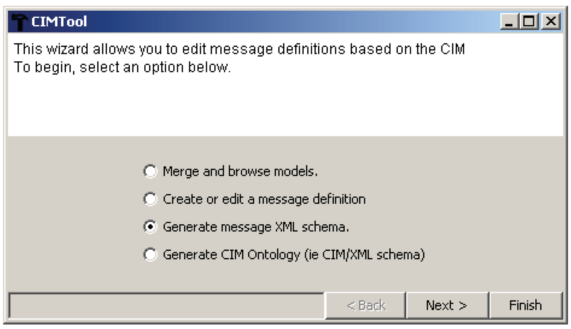
When looking at enterprise level data exchange, we have to deal with more simple structures, in our case XML documents and schemes. The overall process is illustrated in figure 2.

Starting with an XMI (XML metadata interchange) model, a different tool from XPetal is used, the so called open source CIMTool (http://www.cimtool.org). The CIMTool loads the IEC-CIM model as XMI file, this provides the proper base model for the code transformations. Afterwards, a wizard based interface is used to create an OWL representation of the needed payload. After completing those steps, we have a fully thorough semantic definition of our needed EAI payload. Having different base models, we can still use our generic editor to create the payload's semantic description. Afterwards, we have to do different steps to complete the xml scheme needed. We once more start the CIMTool, but instead of creating an OWL description, we create a flat or a nested xml schema based on code transformations of the OWL model. This leads to a proper fully semantically and syntactically compliant IEC-CIM based XML scheme. It is possible to include one's own namespaces and routing headers for the used EAI platform and the schema is ready to deploy.

This overall process really simplifies the creation of meaningful payloads for EAI and increases the overall semantic quality of the needed messages. The approach has several advantages over the existing ones:

- Meaningful semantics are supplied by the IEC-CIM that is used as a domain ontology. A common language can be established on the whole enterprise message bus.
- Model-driven development facilitates the ease of creating the payloads.
- Tools provide both a generic and deterministic way of creating the XML schemes which makes both for ease validation and introduction into the development department.
- Most of the tools available are provided as open source tools. This lowers the costs of getting acquainted with the new technology.

Figure 4. Choices of the CIMTool wizard (©2007 Sven Abel, Wilhelm Hasselbring, Niels Streekmann, and Mathias Ulsar. Used with permission)



- Within the creation process, domain experts are mainly needed when defining the messages. Afterwards, the deployment engineers can transform the OWL descriptions into proper and more technology oriented payloads.

Anyway, there are still some disadvantages.

- A proper versioning of the underlying models is needed in order to structure the code generation process. The semantics of the different artefacts incorporated in the process must be kept consistent.
- The maturity of the used tools differs between the different code levels. While the XML tools already have reached a good maturity, the OWL tools have far less overall capabilities and functions. This sometimes restricts the engineer in modelling the proper payloads (e.g. when constraining objects).
- The overall amount of data exchanged is increased due to the use of XML in comparison to pure CSV (Comma Separated Value) or binary data.

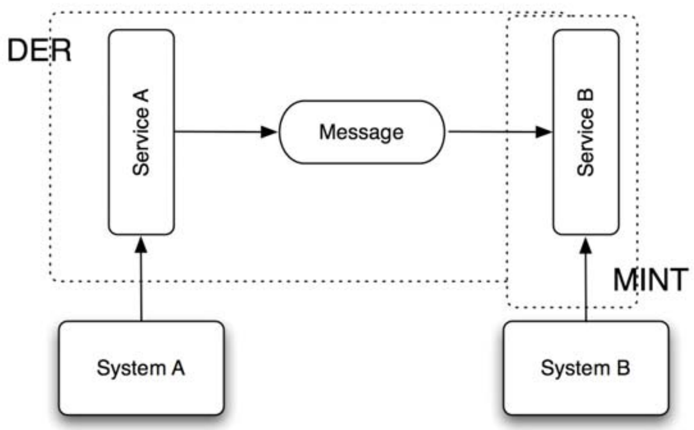
The overall approach is successfully used in the project and has proven to be a good decision. Rapid prototyping of the needed payloads with agreed semantics has been extremely easy compared to the previous approaches. The use of IEC-CIM, MDD, UML and UMM (UN/CEFACT Modeling Methodology) has increased the overall quality and decreased time-to-deployment for the needed payloads for coupling heterogeneous systems.

4. SCOPES OF THE TWO USE CASES

As described in the last sections, MINT and DER are two completely independent projects. We described how to create a flexible architecture and semantically standardized payloads. For both projects, it results in a comprehensive set of services which are provided as SOAP-based WebServices. This allows for performing interactions between both projects by connecting their concepts on a service level.

Figure 5 shows which roles DER and MINT play in the integration of two systems and indicates where their intersection is. The systems

Figure 5. The projects: roles in system integration (©2007 Sven Abel, Wilhelm Hasselbring, Niels Streekmann, and Mathias Ulsar. Used with permission)



are integrated using services which interact by exchanging messages. DER concentrates on the generation of standardized messages. One of the focuses of MINT on the other side is the generation of service adapters for legacy systems and the orchestration of these services. Hence the intersection point of the projects would be the orchestration and generation of adapters that use generated standard messages.

Lessons Learned

In the course of the two projects, the authors learned some lessons that can help readers when taking the decision of using MDI and MDSD based approaches. Compared to a traditional development approach, our approach clearly needs more preparation time. There are two reasons for that. On the one hand, the approach is new and therefore somewhat unknown for all participants, which results in a learning curve at the beginning of the project.

On the other hand the success of our projects highly depends on the modelling and the 'ground work'. This means that it is even more important to ensure a high-quality yet flexible conceptualisation. In order to assure this, we involved domain experts at an early stage in MINT and used standards in DER. This has helped us to ensure a high quality and a standard-compliant solution.

We also realized that our approach is easier for new participants to join an ongoing project. The reason for that is the clear structure that is the result of our model based approach. Having this in mind, we result in a more transparent solution that should also be easier to maintain in the future (this statement of course has to be validated by the time).

Another fact is the necessity for interoperability techniques such as a clear interface specification and the usage of a flexible and easy to use intermediate exchange format. This is required because of the high number of different (and more or less independent) components that need to communicate within the solution. Approaches such as Services-Oriented Architectures can help achieving this.

5. CONCLUSION AND FURTHER ACTIONS

Within this chapter, the model-driven integration approach in complex information systems was introduced by giving two practical examples. The chapter used the experiences the authors have made in two different research projects in the public utilities domain. The result of this chapter is a brief introduction of this topic and it demonstrated the different possibilities to solve the problem. While MINT relies on WebServices to integrate systems on a message level, the DER project focuses on creating a common information model on a semantic basis (RDF) and uses WebServices only for transportation and technical integration between the heterogeneous IT landscape.

In order to ensure quality, MINT uses domain experts that should ensure the validity and the applicability and DER addresses quality issues by using a well defined standard (IEC-CIM) as a basis for creating the RDF messages. The two approaches can be combined to improve the quality of software development using MDI and MDSD techniques.

Acknowledgment

The research project MINT is supported by the German *Federal Ministry of Education and Research* in the scope of the *Forschungsoffensive Software Engineering 2006*. The DER project is supported and partly funded by the *EWE AG*, Oldenburg, Germany.

Additional Reading

Integration and migration of legacy systems in the focus of model-driven software development is also an issue that is faced by standardization organizations like the OMG, who introduced the Architecture-Driven Modernization Initiative (ADM).

In the practice of software engineering there are different approaches to increase the quality of software by using domain models to reduce the linguistic gap between domain experts and software engineers. These approaches differ from the ones described above since they do not focus on the integration of existing systems. Examples are the Eclipse Modeling Framework (Eclipse, 2007), Microsoft's Software Factories (Greenfield, 2004) and language workbenches (Fowler, 2005).

In addition to the model driven integration approach, an interesting and up to a certain extent even completing approach is the usage of ontologies for different formats that are connected using Ontology Mapping or Alignment approaches. An overview about different concepts of this is given in (Doan & Madhavan & Domingos, & Halevy, 2002), (Ehrig & Sure, 2004), (Abels & Haak & Hahn, 2005) and Rahm & Bernstein (2001).

More on the CIM and its scopes for message-based integration can be found in Uslar et al., (2005). Other scopes of use are a bit outside the model-driven development process, more general info on the CIM can be found in Shahidehpour & Yang, (2003). This source provides a useful overview on how to use the IEC TC 57 standard and the CIM in context with both SCADA technology and message-based coupling of systems.

Future (needed) Research

The future of software development tends toward model-driven development. Some of the main questions are addressed in the chapter: the role of standards, integration of existing systems and communication with domain experts. These will also be the topics in future research. The emerging standards in the MDA/ADM surroundings and domain specific standards like the IEC-CIM will be the basis for future high quality software. One of the main drivers will be on how fast the maturity of the overall process models and the tooling grows.

Domain models and domain-specific languages will play a more central role in software development. But still a lot of research has to be done on usability and the granularity of the languages and the decision on when to use standard models and transformations and when to use specific models and specialized languages and transformations.

REFERENCES

Abels & Haak & Hahn (2005). Identification of Common Methods Used for Ontology Integration Tasks. In: Proceedings of the first international ACM workshop on Interoperability of Heterogeneous Information Systems (IHIS05), CIKM conference proceedings. ACM, Sheridan publishing.

ADM Task Force. Why do we need standards for the modernization of legacy systems. White Paper. Retrieved June, 1, 2007, from http://adm.omg.org/legacy/ADM_whitepaper.pdf

Becker, Steffen & Brogi, Antonio & Gorton, Ian & Overhage, Sven & Romanowsky, Alexander & Tivoli, Massimo (2006). Towards an Engineering Approach to Component Adaptation. In `Architecting Systems with Trustworthy Components`, *Proceedings of Dagstuhl Seminar 04511*.

Brand, Klaus-Peter & Buchholz, Bernd (2003). *Systemanforderungen an interoperable Geräte und Systeme der Stationsautomatisierung*. In (Schwarz, Karlheinz (Edt.): Offene Kommunikation nach IEC 61850 für die Schutz- und Stationsleittechnik.

Bundesministerium für Wirtschaft (2004). *Gesetz für den Vorrang Erneuerbarer Energien* (EEG) in der Fassung vom 21.7.2004, BGBL, I:1918

Czarnecki, Krzysztof & Eisenecker, Ulrich (2000). *Generative Programming - Methods, Tools, and Applications.* Addison-Wesley.

Dag, Hasan & Urkan, Ulmut (2004). *An XML Based Data Exchange Model for Power System Studies.* ARI - The Bulletin of the Istanbul Technical University, 2.

deVos, Arnold (2000). *Simplified RDF Syntax for Power System Model Exchange.* Longdale Consultants, 2000, available at http://www.langdale.com.au/CIMXML/.

deVos & Rowbotham (2001). *Knowledge Representation for Power System Modeling,* In (IEEE Publishing Edt.): Proceedings of the PICA 2001 (The 22nd International Conference on Power Industry Computer Applications), IEEE Power Engineering Society.

deVos & Widergren & Zhu (2001). *XML for CIM Model Exchange.* In (IEEE Publishing Edt.): Proceedings of the PICA 2001 (The 22nd International Conference on Power Industry Computer Applications), IEEE Power Engineering Society.

Doan & Madhavan & Domingos, & Halevy (2002). Learning to map between ontologies on the semantic web. In: Proceedings of the Eleventh International WWW Conference, Hawaii, US.

Ehrig & Sure (2004). Ontology Mapping - An Integrated Approach. Proceedings of the First European Semantic Web Symposium. Lecture Notes in Computer Science, Vol. 3053, Springer Verlag, Heraklion, Greece, May 2004, S. 76-91.

Elvesæter, Brian & Hahn, Axel & Berre, Arne-Jørgen & Neple, Tor (2005). Towards an Interoperability Framework for Model-Driven Development of Software Systems, Proceedings of the 2005 International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2005).

European Parlament (2003). Richtlinie 2003/54/EG des Europäischen Parlaments und des Rates vom 26.Juli 2003 über die gemeinsamen Vorschriften für den Elektrizitätsbinnenmarkt und zur Aufhebung der Richtlinie 96/92/EG.

Evans, Eric (2004). *Domain-Driven Design – Tackling Complexity in the Heart Of Software.* Addison-Wesley.

Fowler, Martin (2005). Language Workbenches: The Killer-App For Domain-Specific Languages? Retrieved 2007/05/30, from http://www.martinfowler.com/articles/languageWorkbench.html

Greenfield, Jack & Short, Keith & Cook, Steve & Kent, Stuart (2004). *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools.* Wiley.

Hasselbring, Wilhelm & Reussner, Ralf & Jaekel, Holger & Schlegelmilch, Jürgen & Teschke, Thorsten & Krieghoff, Stefan (2004). The Dublo Architecture Pattern for Smooth Migration of Business Information Systems: An Experience Report. *Proceedings of the 26th International Conference on Software Engineering* (pp. 117-126). IEEE Computer Society Press.

Hess, Andreas & Humm, Bernhard & Voss, Markus (2006). Regeln für serviceorientierte Architekturen hoher Qualität. *Informatik-Spektrum, 29*(6), 395-411.

Van den Heuvel, Willem-Jan & Hasselbring, Wilhelm & Papazoglou, Mike (2000). Top-Down Enterprise Application Integration with Reference Models. *Australian Journal of Information Systems, 8*(1), 126-136.

IEC (2003). IEC - International Electrotechnical Commission: IEC 61970-301: *Energy management system application program interface*

(EMS-API) – Part 301: Common Information Model (CIM) Base. International Electrotechnical Commission.

IEC (2004). IEC - International Electrotechnical Commission: IEC 61970-501: *Energy management system application program interface (EMS-API) – Part 501: CIM RDF Schema – Revision 4*. International Electrotechnical Commission.

IEC (2004)b. IEC - International Electrotechnical Commission: Draft IEC 61970: *Energy Management System Application Program Interface (EMS-API) – Part 503: CIM XML Model Exchange Format - Draft 3b*. International Electrotechnical Commission.

IEEE (1990). Institute of Electrical and Electronics Engineers: IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries.

Jeusfeld, Manfred A. & Backlund, Per & Ralyté, Jolita (2007). Classifying Interoperability Problems for a Method Chunk Repository, Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007).

Object Management Group, OMG (2003). MDA Guide Version 1.0.1.

Marx Gómez & Brehm (2007). *KMU-Software-Umfrage 2006 zur Nutzung betrieblicher Standardsoftware kleiner und mittelständischer Unternehmen in Deutschland,* Dep. of Business Information Systems, University of Oldenburg, Study.

MultiSpeak (2003). *MultiSpeak Version 2.2* Specification (10/07/03), NRECA, Virginia, available online at http://www.multispeak.org.

Neumann (2003). *Comparison of IEC CIM and NRECA* MultiSpeak, UISOL.

Podmore, Robin et al. (1999). *Common Information Model - a Developer's Perspective*. In (Sprague, R. Edt.): Proceedings of the 32nd Hawaii International Conference on System Sciences, IEEE Publishing.

Rahm & Bernstein (2001). A survey of approaches to automatic schema matching, The VLDB Journal, 10, 2001.

Reussner, Ralf (2005). *MINT – Modellgetriebene Integration von Informationssystemen*, Description of Work, Forschungsoffensive „Software Engineering 2006".

Robinson, Greg (2002)a. *Key Standards for Utility Enterprise Application Integration* (EAI), Proceedings of the Distributech 2002, Miami, Pennwell, 2002.

Robinson, Greg (2002)b. *Model Driven Integration (MDI) for Electric Utilities*, Proceedings of the Distributech 2002 Miami, Pennwell, 2002.

Shahidehpour, Mohammad & Wang, Yaoyu (2003). C*ommunication and Control in Electric Power Systems: Applications of Parallel and Distributed Processing*, IEEE Press Series on Power Engineering, 2003.

Uslar, Mathias et al (2005): *Interaction of EMS related Systems by Using the CIM Standard*, In: Walter Leal Filho, Jorge Marx Gómez, Claus Rautenstrauch (Hrsg.): ITEE 2005: Second International ICSC Symposium on Information Technologies in Environmental Engineering: Proceedings, Otto-von-Guericke Universität Magdeburg, Shaker Verlag, 2005.

Winter, Andreas & Ziemann, Jörg (2006). *Model-based Migration to Service-Oriented Architectures*. In U. Kaiser, P. Kroha, A. Winter (Eds.), *3. Workshop Reengineering Prozesse (RePro 2006) Software Migration* (pp. 16-17). Mainz: Johannes Gutenberg University Mainz.

Zhou, E.Z. (2000). *XML and data exchange for power system analysis*. In (IEEE Edt.): IEEE Power Engineering Review, 20, 66-68.

Ziemann, Jörg & Leyking, Katrina & Kahl, Timo & Werth, Dirk (2006). Enterprise Model driven Migration from Legacy to SOA. In R. Gimnich, A. Winter (Eds.), *Workshop Software-Reengineering und Services* (pp. 18-27). Koblenz: University of Koblenz-Landau.

Zimmermann, Olaf & Doubrovski, Vadim & Grundler, Jonas & Hogg, Kerard (2005). *Service-oriented architecture and business process choreography in an order management scenario: rationale, concepts, lessons learned.* In OOPSLA Companion (301-312).

Zimmermann, Olaf & Köhler, Jana & Leymann, Frank (2006). *The Role of Architectural Decisions in Model-Driven Service-Oriented Architecture Construction.* In L.A. Skar, A.A. Bjerkestrand, *Best Practices and Methodologies in Service-Oriented Architectures, OOPSLA 2006 Workshop.*

ENDNOTES

[1] http://www.interop-noe.org

[2] http://www.athena-ip.org