Making Expert Knowledge Explicit to Facilitate Tool Support for Integrating Complex Information Systems in the ATM Domain

Thomas Moser[α] Richard Mordinyi[α] Alexander Mikula[β] Stefan Biffl[α]
[α]Complex Systems Design & Engineering Lab, Vienna University of Technology, Austria[1]
{thomas.moser, richard.mordinyi, stefan.biffl}@tuwien.ac.at
[β]Frequentis AG, Vienna, Austria
alexander.mikula@frequentis.com

Abstract

The capability to provide a platform for flexible business services in the Air Traffic Management (ATM) domain is both a major success factor for the ATM industry and a challenge to integrate a large number of complex and heterogeneous information systems. Most of the system knowledge needed for integration is not available explicitly in machine-understandable form, resulting in time-consuming and error-prone human tasks.

In this paper we propose a knowledge-based approach, "Semantically-Enabled Externalization of Knowledge" for the ATM domain (SEEK-ATM), which explicitly models a) expert knowledge on specific heterogeneous systems and integration requirements; and b) allows mapping of the specific knowledge to the general ATM problem domain knowledge for semantic integration. The domain-specific modeling enables a) to verify the integration knowledge base as requirements specification for later design of technical systems integration and b) to provide an API to the problem space knowledge to facilitate tool support for efficient and effective systems integration.

Based on an industry case study, we evaluate effects of the proposed SEEK-ATM approach in comparison to traditional system integration approaches in the ATM domain.

Keywords: complex information systems integration, knowledge-based systems, context-specific knowledge.

1. Introduction

In the *Air Traffic Management* (ATM) domain complex information systems need to cooperate to provide data analysis and planning services, which consist in the core of safety-critical ATM services and also added-value services for related businesses. ATM is a relevant and dynamic business segment with changing business processes that need to be reflected in the integration of the underlying information and technical systems.

A major integration challenge is to explicitly model the knowledge embedded in systems and ATM experts to provide a machine-understandable knowledge model for integration requirements between a set of complex information systems (CIS). CIS consist of a large number of heterogeneous subsystems. Each of these subsystems may have different data types as well as heterogeneous system architectures. In addition, CIS typically have significant quality-of-service demands, e.g., regarding security, reliability, timing, and availability. Many of today's ATM CIS were developed independently for targeted business needs, but when the business needs changed, these systems needed to be integrated into other parts of the organization [7]. Most of the system knowledge is still represented implicitly, either known by experts or described in human-only-readable sources, resulting in very limited tool support for systems integration. The process of adapting the cooperation the business system is traditionally a human-intensive approach of experts from the ATM and technology domains.

Making the implicit expert knowledge explicit and understandable for machines can greatly facilitate tool support for systems integrators and engineers by providing automation for technical integration steps and automatic validation of integration solution candidates. The overall process for systems integration consists of 3 phases (see [15]): first, the elicitation and validation of systems integration requirements (problem space knowledge); second, the description of the architecture and the modeling of the capabilities of technical solu-

[1]The authors would like to thank all other members of the SWIS project (A. Anjomshoaa, C. Fabianek, E. Kühn, S. Miodrag, M. Mor, C. Rihacek, and A. Schatten) performed 2006-2008 at Vienna University of Technology together with Frequentis AG and Austro Control GmbH.

978-0-7695-3575-3/09 $25.00 © 2009 IEEE
DOI 10.1109/CISIS.2009.90

tion candidates (solution space knowledge); and third the bridging of the knowledge models of problem and solution space to identify the most suitable solution candidates.

In this paper we focus on the first phase of system integration to provide the foundation for the later phases. We propose a knowledge-based approach, "Semantically-Enabled Externalization of Knowledge" for the ATM domain (*SEEK-ATM*), which a) explicitly models specific heterogeneous system and expert knowledge on integration requirements using a three-layered ontology architecture for storing knowledge, b) allows mapping of the specific knowledge to the general ATM problem domain knowledge for enabling semantic integration, and c) facilitates tool support for e.g., requirements validation by means of providing homogeneous access to heterogeneous integration knowledge. The knowledge base provides tool access to knowledge models based on a common problem domain model, allowing queries or validation of heterogeneous knowledge sources. The output of this phase is a validated knowledge base of business requirements for integration as input to technical design steps.

We evaluate the effectiveness and efficiency of the *SEEK-ATM* approach in an industrial case study in the ATM domain. Based on two integration scenarios, we determine key performance indicators, like integration effort, integration duration, quality assurance efficiency, model complexity, and level of automation support in order to compare the *SEEK-ATM* approach with traditional system integration approaches in the ATM domain.

The remainder of this paper is structured as the following: section 2 summarizes related work, section 3 motivates research issues, section 4 pictures the use case, sections 5 and 6 introduce and apply the *SEEK-ATM* approach; section 7 discusses the evaluation results. Section 8 concludes and proposes further work.

2. Related Work

This section summarizes related work on semantic data integration using ontologies.

2.1. Semantic Data Integration

Semantic integration of heterogeneous information systems has become a large research field in recent years. Semantic integration aims at solving semantic heterogeneities that can occur between legacy information systems. Goh identified three main categories of semantic conflicts in the context of data integration that can appear: confounding conflicts, scaling conflicts, and naming conflicts [6]. The use of ontologies as a solution approach to semantic integration and interoperability has been studied over the last ten years. Wache et al. reviewed a set of ontology-based approaches and architectures that have been proposed in the context of data integration and interoperability [18]. Good examples for architectures or systems in the context of semantically enhanced data integration are reports on the projects COIN [6], OBSERVER [11], BUSTER [17], COG [10] and CLIO [13]. However, there are few reports on the use of semantically enhanced data integration in safety-critical domains like ATM.

2.2. Ontologies for Semantic Data Integration

Ontologies can support data integration processes by providing a continuous data model [3] that helps bridging semantic gaps between systems and/or processes. Compared to traditional common data models like UML Class Diagrams or Entity Relationship Diagrams (ERDs), ontologies both a) provide methods for integrating data models using automated transformation and b) support the concurrent modeling of different systems [9]. There is a wealth of research reports on the extension of UML to support Ontology Engineering for the Semantic Web [1]. For quality assurance (QA) ontologies can check whether a model has knowledge missing or inconsistent knowledge.

There has been ample research [8] on the use of ontologies for supporting typical software engineering processes like systems integration. Ontology-Driven Architecture (ODA) is introduced, serving as a starting point for the W3C to elaborate a systematic categorization of the different approaches for using ontologies in Software Engineering. The current MDA-based [12] infrastructure provides architecture for creating models and meta-models (e.g. models of the systems to be integrated), define transformations between those models (e.g., transformations between integrated systems), and managing metadata. Though the semantics of a model is structurally defined by its meta-model, the mechanisms to describe the semantics of the domain are rather limited compared to knowledge representation languages. In addition, MDA-based languages do not have a knowledge-based foundation to enable reasoning (e.g., for supporting QA) [2]. System integration can benefit from the integration with ontology languages such as RDF and OWL [4, 5] in various ways, e.g., by reducing language ambiguity, enabling validation and automated consistency checking. Ontology languages provide better support for logical inference, integration and interoperability than MDA-based approaches.

3. Research Issues

Recent projects with industry partners from the safety-critical ATM domain raised concerns about the verification of modern technology-driven integration environments. For certification a major goal was to improve the capability of engineers to verify an integration solution by facilitating team work and tool support.

The data-driven *SEEK* approach [15] has been developed in order to explicitly model the semantics of the problem space, the solution space, and provide a process to bridge problem and solution spaces. The *SEEK* approach, described in [15] more detail, consists of 6 process steps: 1. legacy system description, 2. domain knowledge description, 3. model QA, 4. derivation and selection of integration partners, 5. generation of transformation instructions, and 6. configuration QA. For a typical systems integration scenario, the problem space is described as integration requirements and capabilities, the solution space consists of connectors and data transformation instructions between legacy systems, while the bridging process between both spaces is concerned with finding feasible integration solutions, e.g., with minimal integration costs.

In this paper, we apply the original *SEEK* process to a use case example from the ATM domain and describe the resulting variant of the *SEEK* process, *SEEK-ATM*, with a main focus on the first three process steps, namely the modeling of integration requirements and capabilities for integration knowledge elicitation and QA, resulting in the following research issues.

RI-1. Foundations for tool support for automation of integration steps. Investigate to what extent (e.g., effort saved during process execution) the explicit and machine-understandable semantic modeling of integration knowledge helps to automate time-consuming systems integration steps. Investigate the effect of the automated integration process steps regarding the quality assurance efficiency.

As precondition for RI-1, we needed to ensure that a) the knowledge is complete enough for relevant tool support (Section 6), and b) the knowledge can be accessed (Section 5.4) by tools e.g., by means of an API.

RI-2. More efficient and effective systems integration process steps. Investigate whether the *SEEK-ATM* approach provides an overall more efficient and effective systems integration process regarding key performance indicators like integration effort and duration, QA efficiency, model complexity and level of automation support.

For empirical evaluation we determine the integration effort needed for each process step to compare the steps in the new *SEEK-ATM* approach with traditional methods and measure the effectiveness and efficiency of the available methods and tools.

4. Use Case Description

A requirement of the ATM domain is to provide timely and correct data analyses from a web of heterogeneous legacy applications. The high number of distributed legacy applications with heterogeneous interfaces to their services on the one hand and the need to dramatically improve the flexibility in order to provide new ways of systems integration in a safety-critical environment on the other hand, demanded for a innovative approach like the *SEEK-ATM*.

The ATM use case (Figure 1) represents information that is typically extracted from participants in workshops on requirements elicitation for information systems in the aviation domain.

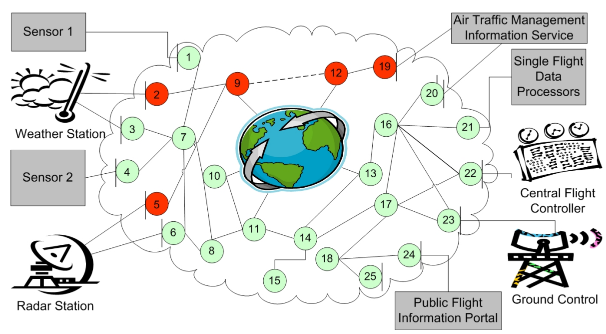

Figure 1: Overview Use Case Example.

The business system ATM Information Service (ATMIS) has to provide information services about flights to business partners via a Public Flight Information Portal (PFIP). ATMIS needs to collect and refine information from at least two other systems: the Central Flight Controller (CFC) and the Single Flight Data Processors (SFDPs).

As input to integration process each data provider, in our case CFC and SFDPs, defines the data content and format he can provide and the quality of service, e.g., the frequency of incoming data such as radar signals; each data consumer, in our case ATMIS, similarly defines his needs for data content, format and quality of service, and may additionally require conditions such as data coming from a defined geographical area and within a defined time window. Finally, the network provider describes the capacity of connectors between the data provider and consumer nodes, and the quality of service of these connectors, e.g., security levels, reliability.

All systems have requirements on reliability, timeliness, safety, service quality, failover, performance, auditability, maintainability, and flexibility. An additional requirement regarding a possible systems integration solution is the capability of agile reaction to any kind of changes due to altered business needs.

There are database-style and/or UML models of the systems interfaces, which work well together in a homogeneously designed set of systems. However, in the ATM domain the systems often exhibit heterogeneous semantics, i.e., similar meaning can be expressed in several ways. Currently, experts in the problem and solution space bridge these semantics as there are so far no machine-readable models available to facilitate comprehensive tool support. However, the limited availability of these experts slows down the pace of strategically desirable integration projects.

5. Making Integration Knowledge Explicit

This section pictures the semantic modeling of heterogeneous knowledge using a set of ontologies as model. The ontology architecture [14] is described in detail as well as the distribution of the modeled information among the layers.

The ontologies used as input models for the derivation of the system configuration are organized using a subdivided architecture, consisting of three different types of ontologies. The ontology types building the semantic model for a specific scenario are the abstract integration scenario ontology (AIS), the domain-specific ontologies, and the integration system ontologies (see Figure 2). The domain ontologies extend the abstract integration scenario ontology by adding concepts describing the common domain knowledge used. In addition, the integration system ontology uses the other two ontologies for aligning its concepts with the more general concepts defined in either the AIS or domain ontology.

5.1. Abstract Integration Scenario Ontology

The abstract integration scenario (AIS) ontology is defined in an application-domain-independent manner, allowing its use across different domains. This domain-independent definition is a powerful mechanism to provide a flexible base for information sharing scenarios, completely independent of a particular domain. The terms in the AIS ontology are defined in an abstract way to simplify the conceivability of the use in different domains.

5.2. Domain Ontology

The domain ontology includes the main shared knowledge between stakeholders of the particular domain (e.g., ATM domain) and hence represents the collaborative view on the information exchanged in an integration scenario. In addition, the domain ontology is the place to model standardized domain-specific information. The customers map their proprietary information, which is defined in the integration system ontologies, to the standardized information in order to allow the interoperability with other participants.

This domain-specific information is used for the detection of semantically identical information provided or consumed by participating applications or organizations, independent of the format or identifiers used for the information, and therefore improves or enables the communication between these organizations. The identification of possible integration partners is simplified and the tool-supported transformation of semantically identical information existing in different formats allows further communication between new partners.

This particular domain-specific knowledge described in the domain ontology can easily be updated or transferred to other *SEEK-ATM* approach-based integration scenarios residing in the same domain. This allows a broad spectrum of new applications in a particular domain to benefit from the described domain knowledge. Instead of modeling the domain knowledge from scratch it is also possible to use as starting point a description of the problem domain, a so-called "world model". The advantage of this approach is the reduced effort for modeling the domain knowledge; however a tradeoff exists in the complexity of typical "world model" ontologies, resulting in a longer waiting time when searching for concrete domain knowledge.

5.3. Integration System Ontology

The integration system ontology (ISO) defines the customer-specific, proprietary view on the information exchanged in an integration scenario. This includes the view on the format of the information (as required by the legacy application), but can also describe the meaning or the use of the specific view on the existing information, since there can exist multiple views for the same information. The ISO defines the structure of the legacy applications, services and messages, i.e., the services provided by a legacy application, the messages provided or consumed by a service and the message segments a message consists of, by adding instances of the concepts defined in either the AIS or domain ontology.

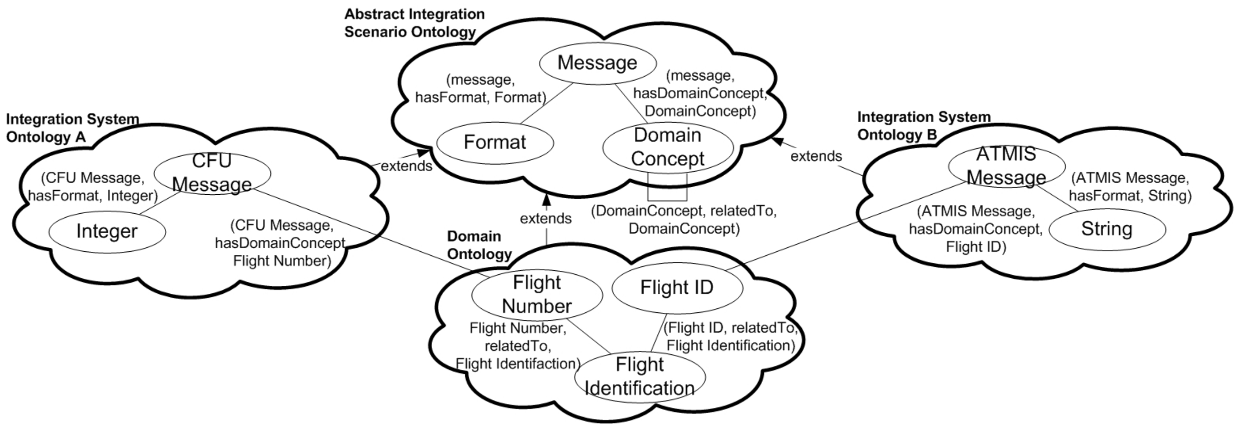

Figure 2: Ontology Architecture Example [14, 16].

The most important part of this description is the definition of the exchanged information, i.e., the definition of the messages either provided or consumed by the legacy applications. The ISO describes the semantic context and the format of each message segment, supported by the domain expert. Each message segment is mapped to exactly one particular domain concept. This defines the semantic context of the information contained in the segment and allows the detection of possible collaborations for an integration scenario. In addition, the format of the information is described, enabling automated transformation between formats.

5.4. SEEK-ATM Process Description

This section summarizes the key factors of the *SEEK-ATM* approach. Figure 3 gives a short overview of the *SEEK-ATM* process steps for requirements elicitation and validation in comparison with a traditional integration approach.

Traditional Integration Approach. In the traditional integration approach, for each legacy information system to be integrated, the Subject Matter Expert (SME) responsible for the particular system describes the requirements and capabilities of the system using human-readable (but typically not machine-readable) language. The outcome of this process step is a set of legacy systems interface description documents. The QA step is performed mostly by humans and mainly consists of a) a comparison of the knowledge represented in the legacy systems interface description documents with the knowledge captured implicitly by the SMEs; and b) a comparison of the accepted set of integration partners and the needed transformation instructions with the knowledge represented in the legacy systems interface description documents and again with the knowledge captured implicitly by the SMEs. As key parts of this knowledge are not available in machine-readable form, tool support for QA is very limited and takes much effort from scarce human experts.

SEEK-ATM Integration Approach. In the *SEEK-ATM* approach, for each legacy information system to be integrated, the SME responsible for the particular system describes the requirements and capabilities (R&Cs) of the system using machine-readable notations. In addition to these R&Cs, the semantic meaning of the exchanged information is externalized by mapping information to more general knowledge represented in the domain ontology. In comparison to the traditional integration process, the outcome of this process step is a set of ontologies describing the R&Cs of the legacy information system to be integrated, as well as the mapping of the information to general domain knowledge. In addition to the description of the R&Cs of the participating systems, the domain expert (DE) describes the common knowledge of the problem domain used in the integration scenario.

This externalized domain knowledge is used by the SMEs while describing the particular legacy systems. The outcome of this process step is an ontology describing the shared domain knowledge of the problem domain used in the integration scenario. This domain ontology can be reused for a set of different integration scenarios in a domain. The QA step in the *SEEK-ATM* integration approach can be very well supported with tools based on ontology-based reasoning. Reasoning allows checks for consistency (e.g., whether information entered in different input masks is consistent) and completeness (e.g., whether all needed information is entered). This allows a much faster and more reliable QA compared to the traditional integration process and allows relieving scarce experts from tedious work.

To summarize the process description, for both the traditional integration process and the SEEK-ATM process the input is the same, but the output differs.

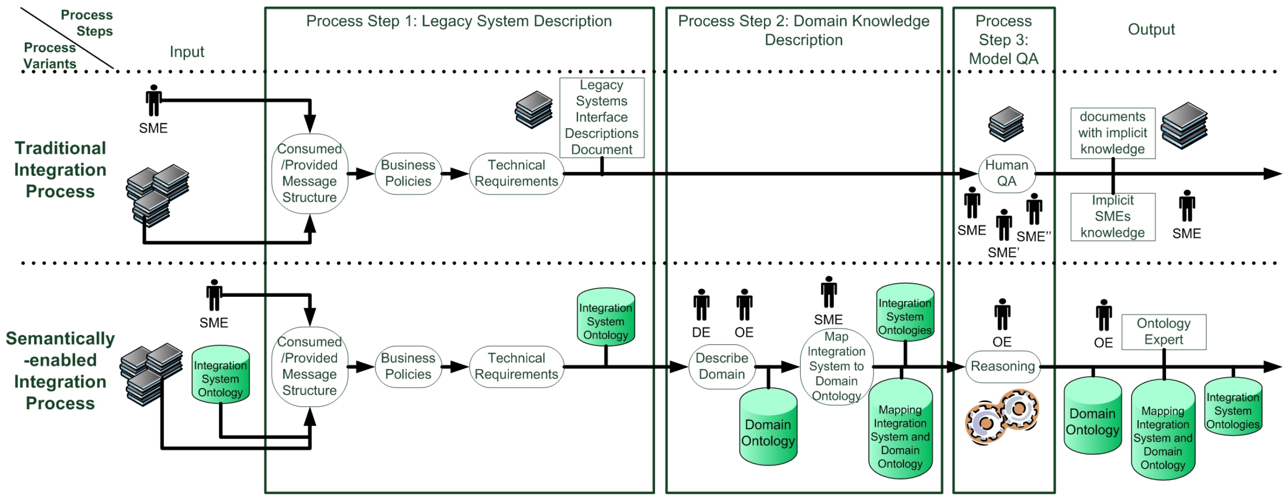

Figure 3: Side-by-side comparison of the traditional and the SEEK-ATM integration process steps.

While the output of the traditional integration process still consists of mainly implicit knowledge, the output of the SEEK-ATM process consists of explicit and machine-understandable knowledge.

6. Added Value from Explicit Knowledge

This section pictures usage scenarios for heterogeneous knowledge integrated using the *SEEK-ATM* approach. In addition, the real usage scenarios from the exemplary ATM use case are described shortly.

The knowledge can be used for a set of queries like checking the consistency of the integrated data (e.g., by measuring type similarity between concepts), or checking the completeness of the mapped concepts (e.g., whether it is possible to fulfill the given requirements with the modeled knowledge.

QA queries. In the use case from the ATM domain, the integrated knowledge can be used for the automated identification of integration partner candidates, the generation of transformation instructions and for the generation of system integration configurations. The following paragraphs summarize these usage examples.

Automated Identification of Integration Partner Candidates. For every consumer service, the set of possible provider services providing the required information is calculated. These sets of pairs of a consumer service and at least one provider service, together with the required transformation instructions are called collaboration candidates. The Domain Expert and the customer SMEs choose one or if applicable more desired collaborations from these collaboration candidates. Then the system integration configuration for these chosen collaborations is calculated by the *SEEK-ATM* Approach.

Generation of Transformation Instructions. After these integration partners are selected, the transformation instructions for these collaborations need to be created. This generation process is semi-automatic and supervised by the Domain Expert. The Domain Expert reviews the generated transformation instructions and has to accept it, in order to be functional.

Generation of System Integration Configuration. The information derived in the previous steps is used to create the final system integration configuration. The configuration is stored in an XML files containing information all the needed instructions to run the system, such as routing tables, transformation instructions, and binding descriptions for connecting to particular legacy systems.

7. Evaluation

In the previous section RI-1 has been addressed. To discuss the RI-2, we started an evaluation by means of the proposed entire *SEEK-ATM* approach. Therefore, we derived four parameters (Table 1) to compare the proposed approach with the traditional one. Table 1 summarizes the effort and duration needed for integration, the quality assurance efficiency, the complexity of the used models, and finally the level of automation support both approaches provide.

The evaluation is based on two scenarios within the ATM use case. The first scenario (Sc. 1) determines the results based on an integration project from the scratch. The second scenario (Sc. 2) assumes that an initial integration project has been accomplished providing a first integration solution, but due to changing business requirements some system adaptations have to be performed, like the need to update the domain model.

Sc. 1 within the ATM use case has the following characteristics: 5 systems (applications) with 30 integration points (services) and 100 data structures (logical entities). In case of Sc. 2, 10 integration points of 3 different systems have been updated resulting in 2 new data structures and 10 updated ones.

Table 1: Comparison of the traditional and the SEEK-ATM approaches.

Evaluation parameters	Traditional approach	SEEK-ATM approach
Integration effort	Sc. 1: 415 PD[2] Sc. 2: 76 PD	Sc. 1: 435 PD Sc. 2: 32 PD
QA efficiency	Low	High
Model complexity	High and distributed	High and centralized
Level of automation support	Low	High

Integration effort. The results of the evaluation show that the overall integration effort is similar for both approaches in case of small number of systems to be integrated and slightly higher for the *SEEK-ATM* approach in case of larger systems. The higher effort comes from the need to manage the domain model, since additional mappings between the integration system ontology and the domain model are needed. The effort to create the integration system ontology or the interface description is similar since in both approaches the conducted SMEs has to cope with the same problem of finding the right information describing the system interfaces with its semantics. The SEEK-ATM has the advantage that in case of adaptation the knowledge already gathered is explicitly given and can be reused in further discussions compared to the traditional approach where this knowledge exists implicitly only.

In case of reconfiguration issues the *SEEK-ATM* process has proven to be more efficient than the traditional approach since once the knowledge has been externalized, it can be reused with little extra effort. Furthermore, in case of the traditional approach each system expert has to be contacted for any kind of changes resulting in discussions. In case of the *SEEK-ATM* approach the domain expert is needed in major changes only where the mapping of the integration system ontology to the domain ontology has to be altered as well. In case of minor changes, affecting the characteristics of the system only, the SMEs are needed. Additionally, performing changes, like structure modifications, based on documents is more difficult and time consuming than compared with ontologies where you deal with classes. Changes can be performed much faster and can be done during the discussion concerning the integration project as well.

The duration of the traditional approach tends to be higher due to error-prone mainly manual process steps resulting in additional efforts to discuss error sources and possible solutions. The proposed *SEEK-ATM* approach reports errors or missing information immediately due to in-time consistency and completeness checks based on ontology reasoning. In case of describing systems, parallel processing is possible in both approaches. However, the following *SEEK-ATM* processing steps are running mainly automated from the third processing step on, while the traditional approach is still human-driven resulting in time consuming and error-prone processing steps. Therefore, the duration depends strongly on of automation support.

QA efficiency. Since the traditional approach focuses on manual validity checks, it is therefore more time consuming and error-prone. This also results in the fact that missing information is often detected in a later integration step. The quality assurance efficiency is measured by the number of failures detected in each system description weighted by the time of detection. The later the failure detected the higher the weighting rate. The *SEEK-ATM* approach uses ontology-based reasoning. This allows performing consistency and completeness checks in-time automatically, resulting in a lower failure rate and in-time notification of the SME about missing/incorrect information. Additionally, since the *SEEK-ATM* approach is mainly automated, it allows returning to any processing state in order to e.g., reproduce errors or revise decisions taken.

Model complexity. The model used in the traditional approach is smaller and therefore less complex compared to the model used in the SEEK-ATM approach, since a considerable part of the integration knowledge is not described explicitly. In the SEEK-ATM approach, the number of relations, i.e., the number of mappings from the integration system ontology to the domain ontology introduces a higher structural complexity. The benefit of a more complex ontology model lies in the way how later integration steps can be supported by a higher level of automation. From the SME's point of view the complexity remains the same in both approaches. For the domain expert the *SEEK-ATM* approach reduces his efforts to the task of managing the structural complexities of the ontologies and to support the SMEs in mapping. In the traditional way the domain experts need to cope with the major part of the complexity, since he is responsible for ensuring the consistency and completeness as well as managing the integration of the SMEs' legacy system descriptions.

[2] PD: Person Day (Full Time Equivalent).

Level of automation support. The *SEEK-ATM* approach supports the user while entering the data with consistency and completeness checks. Additionally, it influences the integration process in later steps by automatically deriving integration partner candidates and automatically generating transformation instructions for message exchange between the integrated systems.

8. Conclusion and Further Work

In this paper, we introduced and evaluated a domain-specific approach for ATM to make expert knowledge on heterogeneous systems and system integration requirements explicit to facilitate tool-support for design and QA. An important contribution of the paper is to enable new research and application areas for semantic techniques that help control complex information system. Major results of our research evaluation of SEEK-ATM in an industrial case study were:

1. Tool support for automation of integration steps. The explicit and machine-understandable knowledge in SEEK-ATM helps to automate time-consuming systems integration steps like consistency and completeness checks. Furthermore, it allows automating later integration processing steps, like deriving integration partner candidates or automatically generating transformation instructions for message exchange between the integrated systems.

2. More efficient and effective systems integration. The evaluation showed that the integration effort needed with the SEEK-ATM approach is slightly higher in case of integration from the scratch, but comparatively a lot smaller when adaptations due to changing business needs have to be performed. In addition, the advantage of centrally storing the domain ontology together with the mappings of individual system knowledge lies in the possibility of an automated QA and automation of further integration steps resulting in less integration efforts and less failures.

Further work will extend the semantic modeling of the problem space to the technical solution space and ultimately ways to bridge problem and solution spaces, as well as include a large-scale evaluation of the S*EEK-ATM* approach using scenarios and integration effort measurements of a real-world integration project.

9. References

[1] K. Baclawski, M. Kokar, P. Kogut, L. Hart, J. Smith, W. Holmes, J. Letkowski, and M. Aronson, "Extending UML to Support Ontology Engineering for the Semantic Web," in Proc. 4th Intl Conf on UML, 2001.

[2] K. Baclawski, M.K. Kokar, P.A. Kogut, L. Hart, J. Smith, J. Letkowski, and P. Emery, "Extending the Unified Modeling Language for Ontology Development," *International Journal of Software and Systems Modeling (SoSyM)*, vol. 1, (no. 2), pp. 142-156, 2002.

[3] C. Calero, F. Ruiz, and M. Piattini, *Ontologies for software engineering and software technology*: Springer, 2006.

[4] D. Gasevic, D. Djuric, and V. Devedzic, "Bridging MDA and OWL Ontologies," *Journal of Web Engineering*, vol. 4, (no. 2), pp. 118-143, 2005.

[5] D. Gasevic, D. Djuric, and V. Devedzic, *Model driven architecture and ontology development*: Springer, 2006.

[6] C.H. Goh, "Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Systems", PhD Thesis, MIT, 1996.

[7] A. Halevy, "Why your data won't mix," *Queue*, vol. 3, (no. 8), pp. 50-58, 2005.

[8] H.J. Happel and S. Seedorf, "Applications of Ontologies in Software Engineering," in Proc. 2nd Intl Wsh on Semantic Web Enabled Software Eng, 2006.

[9] M. Hepp, P. De Leenheer, A. De Moor, and Y. Sure, *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*: Springer, 2007.

[10] R. Lara and J. de Bruijn, "Ontology-based Transformations for the Automotive Industry," in Proc. 1st European Semantic Web Symposium (ESWS2004), 2004.

[11] E. Mena, A. Illarramendi, V. Kashyap, and A.P. Sheth, "OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-Existing Ontologies," *J on Distributed and Parallel Databases*, vol. 8, (no. 2), pp. 223-271, 2000.

[12] J. Miller and J. Mukerji, "Model Driven Architecture (MDA) Object Management Group Draft Specification, http://www.omg.org/docs/ormsc/01-07-01.pdf, 2001.

[13] R.J. Miller, M.A. Hernández, L.M. Haas, L. Yan, C.T.H. Ho, R. Fagin, and L. Popa, "The Clio project: managing heterogeneity," *ACM SIGMOD Record*, vol. 30, (no. 1), pp. 78-83, 2001.

[14] T. Moser, A. Anjomshoaa, and A. Mikula, "FISN Semantic Architecture Document," Frequentis AG, 2007.

[15] T. Moser, R. Mordinyi, A. Mikula, and S. Biffl, "Efficient System Integration Using Semantic Requirements and Capability Models: An approach for integrating heterogeneous Business Services," submitted to 11th Intl Conf on Enterprise Information Systems, 2008 (http://www.ifs.tuwien.ac.at/files/ICEIS_TechRep_v1.pdf).

[16] T. Moser, K. Schimper, R. Mordinyi, and A. Anjomshoaa, "SAMOA - A Semi-automated Ontology Alignment Method for Systems Integration in Safety-critical Environments," in Proc. 2nd Intl Wsh on Ontology Alignment and Visualization, 2008, accepted.

[17] H. Stuckenschmidt, H. Wache, T. Vogele, and U. Visser, "Enabling technologies for interoperability," in Proc. Wsh on the 14th Intl Symp of Computer Science for Environmental Protection, 2000, pp. 35-46.

[18] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner, "Ontology-based integration of information-a survey of existing approaches," in Proc. Wsh on Ontologies and Information Sharing, 2001, pp. 108-117.