Towards Flexible and Efficient Process and Workflow Support in Enterprise Modeling

Andreas Demuth, Markus Riedl-Ehrenleitner, Roland Kretschmer, Peter Hehenberger, Klaus Zeman, and Alexander Egyed

Johannes Kepler University (JKU) Linz
{firstname.lastname}@jku.at

Abstract. In enterprise modeling, organizational structures as well as an enterprise's processes and important artifacts (e.g., business knowledge stored in documents) are captured formally using different kinds of models. These models are not only used for documentation purposes, but they are also used to provide guidance for employees. For example, the models may impose rules on artifact access (i.e., who is allowed to view or manipulate certain artifacts) or they may define workflows for individual processes (e.g., which employees should perform which adaptation steps, and in which order). However, existing enterprise modeling approaches typically support only coarse-grained artifacts. For instance, only individual files can be associated with workflow tasks. Unfortunately, enterprise artifacts are typically of high complexity (e.g., spreadsheets contain millions of data cells). Therefore, it is not sufficient to provide employees only with information about the artifacts involved in a task, but it is necessary to provide more detailed information (e.g., which cells in a spreadsheet are relevant).
In this paper, we introduce a novel approach to enterprise modeling that addresses the issue of too coarse-grained support for enterprise artifacts. Our approach relies on a generic knowledge-sharing platform, called DesignSpace, in which all aspects of an enterprise are integrated and stored at a fine level of granularity. The DesignSpace supports fine-grained representation of enterprise artifacts and their linking to tasks in defined workflows. Moreover, it supports automatic, efficient, and generic workflow support. First case studies suggest that the approach is feasible and provides significant improvements in terms of efficiency compared to state-of-the-practice enterprise modeling solutions.

Key words: Enterprise Modeling, Workflow Support, Consistency

1 Introduction

Generally, enterprise modeling captures the organizational structure as well as important processes of an enterprise [1, 2, 3, 4, 5]. The organizational structure depicts an enterprise's organizational hierarchies and includes detailed information about how individual areas of the enterprise are interrelated (e.g., chain of command, reporting hierarchy) [1, 6]. The modeled processes define at a quite abstract level which information is available in the enterprise and how information should be shared and propagated [6]. Typically, for each process there is a detailed workflow that defines guidelines for employees about how to execute the process step-by-step; it consists of a series of ordered tasks that have to be perfomed, where each tasks is assigned to either an individual or a group of employees [4, 6, 7]. For the sake of simplicity, we will refer to those responsible for performing tasks (i.e., individual members of the organization or groups thereof) as *agents* in this paper.

Indeed, capturing these aspects in enterprise models with graphically appealing visual notations is beneficial for building awareness and making communication about the organization more efficient [1, 8]. However, capturing these aspects is not enough. Especially for processes, it is crucial that they are not only well defined, but that they are also executed properly. Thus traceability and process enforcement are essential for applying enterprise models successfully and with maximum effectiveness [7, 9]. Unfortunately, most existing approaches to enterprise modeling do not support these aspects in enough detail. Specifically, process enforcement is typically limited to the definition of workflows and the assignment of defined tasks to agents (e.g., [7, 10, 5]). Tasks often including coarse-grained information about the involved enterprise artifacts. After performing an assigned task, it is typically marked as completed by the corresponding agent (e.g., through an enterprise management system) and the next task is started. For example by notifiying another agent about the new, pending task. While this is indeed a first step in the right direction, it is by far not sufficient as typically traceability is limited to who performs which task, and which enterprise artifacts are involved (e.g., a spreadsheet for calculating the total personnel costs) [8]. With artifacts of increasing complexity, such as spreadsheets with millions of cells and highly complex calculations, the coarse-grained level of traceability that is available in existing approaches does not suffice to provide meaningful guidance for agents. However, traditional approaches cannot tackle this issue as they generally do not consider enterprise artifacts at a level finer than individual files (at least in a generic fashion) [8].

Furthermore, adaptations to files performed by agents during workflows may introduce contradictions (also called inconsistencies) between knowledge captured in the adapted file and knowledge existing in other files—or even within the same file. For instance, after adding information about a new employee in a spreadsheet, it may be necessary to perform another adaptation that updates the total number of employees for the specific organizational unit in the organization model. Unfortunately, such inconsistencies that may be introduced to enterprise knowledge typically cannot be detected with existing approaches.

In this paper, we present a novel approach to enterprise modeling that addresses the issue of coarse-grained traceability and enterprise artifact integration, and missing guidance and detection of errors made during workflows. We introduce a platform, called DesignSpace, that allows for the integration of enterprise knowledge at arbitrary levels of granularity. The DesignSpace supports enterprise models and artifacts of arbitrary formats and notations, as well as traceability between these models and enterprise artifacts. For efficient process and workflow support, it provides various mechanisms such as incremental consistency checking, automatic change impact analysis, knowledge propagation, or change notifications. First case studies with a prototype implementation of the DesignSpace indicate that the proposed approach is technically feasible, scalable, and usable in practice. Please note that the DesignSpace as a platform for knowledge sharing has been published previously [11]. The novel contribution of this paper is the application of the DesignSpace approach to the domain of enterprise modeling to address the issues discussed above.

2 Illustrative Example

As a simple, yet illustrate example, consider the following scenario that involves the organization model of an enterprise and a spreadsheet that is used for calculating the enterprise's personnel costs, grouped by individual departments. Assume that a large enterprise's European accounting department has just hired an intern for several months. For this hiring process, a workflow is defined that contains two tasks: i) add the intern's information to the organization model, and ii) add the intern's information to the enterprise's cost-calculation spreadsheet. For the first task, an agent uses an organization modeling tool and adapts the organization model. To execute the second task, another agent has to add the intern to the list of employees and he must insert the intern's personal information and salary in the corresponding spreadsheet, which is linked with the task. As the agent performs this task, he encounters two challenges. First, he must find the exact location within the document where information about the accounting department is kept. Second, to obtain the correct salary for the cost calculation, the agent must look up the salary for interns in the accounting department, which is already defined in the spreadsheet. Note that for a complex spreadsheet with millions of cells, dozens of pages, and numerous complex calculations, this task becomes quite error prone and the agent may, by accident, add the intern to the wrong department (e.g., the logistics department), or he may use the wrong salary (e.g., the salary for junior accountants, or interns in the logistics department). Moreover, the intern's personal information (e.g., name, social security number) is entered twice by different agents. If either of them enters information incorrectly, an inconsistency is introduced. Unfortunately, this inconsistency likely remains undetected as there is no connection between the organization model and the cost-calculation spreadsheet.

Overall, this simple example highlights how the lack of fine-grained integration of enterprise artifacts and models and the resulting lack of traceability and consistency checking may lead to contradictions in enterprise knowledge.

3 DesignSpace

Next, we discuss in detail the capabilities of the *DesignSpace* with respect to enterprise modeling, workflow support, and knowledge management. The De-

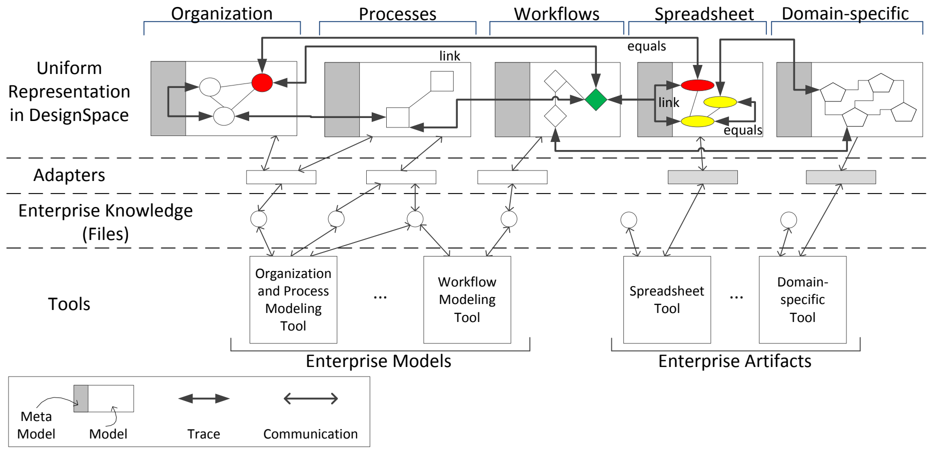

Fig. 1: Approach Overview.

signSpace is a cloud-based knowledge integration and service platform that enables effective and efficient knowledge sharing and management by providing an integrated view on knowledge of various sources and in various formats. The DesignSpace also allows arbitrary inter-dependencies to be established within this representation of enterprise knowledge, therefore augmenting the knowledge already available in the individual models and artifacts. Moreover, it provides services that enable agents to work with diverse enterprise artifacts, to handle inconsistencies between individual pieces of enterprise knowledge, and to perform transformations of knowledge during the execution of tasks. An overview of the DesignSpace approach is depicted in Fig. 1.

3.1 Integration Services

The DesignSpace is a mirror of existing enterprise knowledge, reflecting the contents of existing files using a uniform representation. The existence of the DesignSpace is generally transparent to agents as they can continue to work with common modeling or artifact editing tools to edit enterprise knowledge. However, the DesignSpace's integration of knowledge enables a range of knowledge sharing and information services that, for example, allow agents to define interdependencies among enterprise models and artifacts—even if these are produced and edited in different tools and stored in different files with different formats. These services will be discussed in detail in Section 3.2.

There are two ways of integrating existing enterprise knowledge in the DesignSpace, which we discuss next: file-based integration and tool-based integration.

File-based Integration. File-based knowledge integration takes as input the existing files that contain enterprise knowledge (e.g., enterprise model files or spreadsheets). The contents of these files and are parsed and mapped to the unified representation used in the DesignSpace by *file-adapters*. For each type of file to be integrated with the DesignSpace, a corresponding, file-type-specific adapter is used. Note that the tool adapter also defines at which level of granularity files are mapped to the DesignSpace's unified representation. For instance, an adapter may work at a very fine-grained level of granularity and map the individual cells of a spreadsheet to the DesignSpace (i.e., for a single spreadsheet, information about each of its cells is available in the DesignSpace). However, for certain primitive enterprise artifacts this may not be required (e.g., for image files). When knowledge—and thus the file in which the knowledge is stored—evolves, this is reflected in the DesignSpace by incrementally updating the uniform representation. This synchronization is fully automated by the file-adapters.

Tool-based Integration. Even though the default way of integrating enterprise knowledge is via file-adapters, the DesignSpace also supports and encourages the use of *tool-adapters* that map tools' internal data structures (and thus the knowledge that is edited with the tool) to the DesignSpace's uniform representation on-the-fly. Tool-adapters have a grey background in Fig. 1. In the figure, tool-adapters are used for enterprise artifacts and file-adapters are used for enterprise models. This is by coincidence and not a prescription of our approach. The major benefit of tool-adapters is that they allow for the augmentation of tools with additional information. For example, inconsistencies may be highlighted directly within a tool.

Overall, the DesignSpace is agnostic to the enterprise knowledge it manages and thus supports arbitrary knowledge. Only adapters are aware of which knowledge is integrated with the DesignSpace and how the mapping to the uniform representation of the DesignSpace is done.

3.2 Knowledge Sharing and Management Services

The DesignSpace provides extensive support for sharing and managing enterprise knowledge efficiently.

Traceability Support. Once an enterprise's knowledge is integrated with the DesignSpace through adapters, as discussed above, traceability information may be added. The DesignSpace allows for the definition of traceability between arbitrary pieces of knowledge. Traces may be used within or between individual enterprise models or artifacts, as shown in Fig. 1. For example, it is possible to link a specific cell in a spreadsheet to another cell within the same file, to link individual cells of a spreadsheet with certain workflow tasks, or to link specific files to certain agents (e.g., to define access rights or responsibilities; not depicted in the figure). Different kinds of traces are available in the DesignSpace: untyped and typed traces.

Untyped traces. Arbitrary pieces of knowledge (i.e., parts of enterprise models or artifacts) can be connected via *untyped traces*. Such traces do simply consist of two connection ends where each end points to at least one piece of knowledge.

Typed traces. Typed traces are defined using meta-information. For typed traces, it is defined explicitly which kinds of knowledge they may link. Moreover, typed traces may have explicit semantics assigned. For instance, to define responsibilities within an enterprise, a typed trace named `responsibility trace` may be defined that allows only the linking of (parts of) enterprise artifacts to agents. Each such trace then defines a specific responsibility. As another example, a simple *equality trace* may be defined for a certain type of enterprise artifact knowledge (e.g., spreadsheet cells) and it may require the linked pieces of knowledge to be actually equal (depicted as traces named *equals* in Fig. 1). The semantics of such a trace are straightforward: it requires the linked spreadsheet cells to have equal values. Similarly, a complex equality traces could be defined between knowledge pieces of different types, along with explicit information about how equality is defined. Such a trace could be established, for example, between spreadsheet cells and enterprise model elements. Equality could be defined as both the spreadsheet cell and the enterprise model element must contain the same value. For every typed trace it can be checked if the trace is correct or if there is a mismatch between the linked knowledge and the desired relation (i.e., the semantics) are violated.

Consistency Checking. A key feature of the DesignSpace is consistency checking. Generally, consistency is given if enterprise knowledge is free of inconsistencies (i.e., contradictions). Such contradiction may occur not only between knowledge stemming from different sources (e.g., a spreadsheet cell and an enterprise model element represent the same knowledge, but both have different values), but also between pieces of knowledge that stem from a single source (e.g., two cells in a spreadsheet that should have the same value actually have different values). Indeed, a contradiction indicates that either an agent made a mistake during knowledge adaptation, or that different agents have a different understanding. In the DesignSpace, consistency checking is done automatically based on defined traces (see above) and also on explicitly stated, domain-specific, and adaptable consistency rules. These rules are defined for certain types of knowledge and specify desired conditions that must hold between individual pieces of knowledge. For instance, a consistency rule may state that each pending task during a workflow must be assigned to an agent, and that the assigned agent must belong to the organizational unit that is responsible for performing that workflow.

Change Impact Analysis and Change Notification. Whenever agents perform tasks during a workflow, they may adapt and change enterprise knowledge. In particular, they may adapt pieces of knowledge that were previously adapted by other agents. Or they may change knowledge which other agents depend on. In the DesignSpace, every single adaptation of enterprise knowledge is analyzed automatically by the change impact service for its potential effects on other agents (based on defined traces). If a piece of knowledge is changed, the agents that are linked to the changed part either directly through a trace or transitively through a chain of traces can be notified about the change. This allows these potentially affected agents to review the change and to react to it accordingly.

Knowledge Transformation and Propagation. As discussed above, adaptations of enterprise knowledge may lead to inconsistencies. In addition to the detection of inconsistencies, the DesignSpace also supports mechanism for repairing inconsistencies and propagating adaptations in a way that re-establishes consistency among pieces of knowledge.

4 Fine-grained Enterprise Artifact Integration

Let us now discuss specifically how enterprise knowledge can be integrated with the DesignSpace. In particular, we present how traditional enterprise models and arbitrary enterprise artifacts can be integrated.

4.1 Organization, Process, and Workflow Models

In enterprise modeling, models that describe the organizational structure, the processes, and their associated workflows are commonly used. Typically, these models provide fine-grained information and they are stored by modeling tools in common file formats that are well-structured and well-documented (e.g., XML files with defined schemas); the structure of the models typically resembles a graph-structure. Therefore, integration of these models typically requires a straight-forward mapping of the well-structured model file contents to the DesignSpace's uniform representation. For XML files, for example, it is sufficient to use a single, generic adapter that takes as input an XML schema and a model file to perform the integration of the corresponding model with the DesignSpace.

However, even if custom enterprise modeling tools are used that do not allow for the export of models to common file formats, at most one adapter is required per model type; and, indeed, the number of different models is usually quite limited.

4.2 Enterprise Artifact Models

Besides the integration of the typical enterprise models discussed above, other enterprise artifacts must also be integrated with the DesignSpace. Compared to the integration of the typical enterprise models, as discussed above, the integration of arbitrary enterprise artifacts is more challenging. Specifically, this is because of diversity of enterprise artifacts, which range from spreadsheets, over semi-structured documents containing natural language, to domain-specific files. Especially the latter imposes challenges as for different business domains, different domain-specific knowledge is required. For example, technology companies require hardware models and source code of software to be integrated, whereas for commercial banks the domain-specific enterprise artifacts may contain risk-analysis and stock exchange information. Indeed, for each kind of these enterprise artifacts, a corresponding adapter is required that maps the artifact to the DesignSpace. In contrast to Section 4.1, this mapping is less straight-forward because the files that contain the domain-specific knowledge are often created by custom-built tools. Moreover, before building an adapter, it must be determined at which level of granularity the knowledge should be reflected in the DesignSpace. The level of granularity necessary does not only depend on the knowledge contained in an enterprise artifact and its structure, but also on how and where the artifact is used in workflows. If an artifact is only used as-is during workflows (e.g., an image for the enterprise's official letter head, which is only embedded in documents for customer correspondence but never changed), it is sufficient to represent the artifact at file-level-granularity. If, however, different parts of an artifact are adapted during workflows (e.g., a spreadsheet containing entry-level salaries for different positions), it may be required to represent the artifact in the DesignSpace at a quite fine-grained level so that for different workflows and different tasks it can be defined exactly which parts of the artifact are to be adapted.

5 Efficient Process and Workflow Support

Next, based on our illustrative example, we discuss how the integration of enterprise knowledge with the DesignSpace makes task execution more efficient for agents and how it prevents inconsistencies from being introduced unnoticed.

First, both required tasks in the workflow, which is highlighted with a green background in Fig.1, can be linked to specific pieces of knowledge (e.g., to the department that hired the intern in the organization model, or the salary for interns in that department in the spreadsheet). This not only increases efficiency, but it also reduces the chance of errors being made.

Moreover, since the organization model and the enterprise's cost-calculation spreadsheet are integrated with the DesignSpace, consistency can be checked between the knowledge stored in the corresponding files. A consistency rule can be used that expresses, for instance, that for each employee that appears in the spreadsheet there must be a corresponding entry in the organization model that has exactly the same information (e.g., the employee must not belong to different departments). The traces between employee entries in the organization model and the spreadsheet can be generated automatically in the DesignSpace's knowledge transformation service based on unique indentifiers (e.g., the employees name, date of birth, and social security number). Indeed, if such a trace cannot be established, for example because one of the two responsible agents entered an incorrect date of birth, this is identified as an inconsistency by the DesignSpace. If the trace can be created, the consistency checking service can navigate the trace and check whether both pieces of information meet the desired condition (i.e., equality), and detects an inconsistency if there is a contradiction (e.g., the department differs). If an inconsistency is detected either during trace generation or during consistency checking, both agents are informed through the notification service that there is an inconsistency that needs to be fixed. The knowledge involved in this inconsistency is highlighted with red background in Fig. 1. Since the consistency rule requires the specified information to be equal, either of the responsible agents may review in the file linked to his workflow task whether he entered the employee information correctly. If, for example, the agent that adapted the organization model finds that he entered the information correctly, it follows automatically that the information is incorrect in the involved spreadsheet. In this case, the DesignSpace's knowledge transformation and propagation services may be used to automatically propagate the correct information and update the spreadsheet accordingly.

To check whether the correct salary was entered in the spreadsheet during the second task of the workflow, another trace can be generated that, based on the employee's specific job, department, and geographic location automatically traces to the correct salary. This equality trace can then be checked by the consistency checker. If an inconsistency is detected (as illustrated by the elements highlighted with yellow background in Fig. 1), in this case only the agent who performed the second task is informed because the contradiction is within the spreadsheet, which was not edited by the agent that handled the first task. As with the first inconsistency, the change propagation service of the DesignSpace may be used by the agent to establish equality and thus eliminate the inconsistency.

Note that in this scenario, the knowledge transformation service of the DesignSpace could have been used to insert the correct salary automatically. However, by detecting an inconsistency and informing the agent, it remains possible to purposely ignore an inconsistency, for example because the intern has negotiated a higher salary.

In this section, we have shown by example how the fine-grained integration of enterprise knowledge and the DesignSpace's various services enable efficient detection and handling of inconsistencies during workflows.

6 Validation

To demonstrate the general feasibility of our approach, we used a prototype implementation of the DesignSpace as well as adapters for commonly used modeling and enterprise artifact editing tools. In three case studies the practical applicability and scalability was assessed.

6.1 Prototype Implementation and Tool Integration

To date, the core knowledge integration and knowledge sharing and management services of the DesignSpace, as discussed in Section 3.1 and Section 3.2, have been implemented.[1] These services include: i) data storage mechanisms that allows for cloud-based mirroring of arbitrary enterprise models and artifacts, ii) traceability, iii) consistency checking, iv) trace-based change notification for knowledge changes, and v) an editor with basic visualization capabilities.

Currently, tool-adapters are available for various tools to synchronize existing enterprise knowledge automatically with the DesignSpace.

Modeling Tool. For modeling tools, we have developed a tool adapter for the

[1] Prototype available at `isse.jku.at/tools/dsspc/xadr.zip` (pw: dsisse).

IBM Rational Software Architect (IBM RSA). The RSA does not only support software architecture modeling, but it provides general support for building models for diverse domains, including business process models and workflows.

Spreadsheet Tools. For spreadsheet software, we implemented a tool adapter for Microsoft Excel. The adapter performs synchronization with the DesignSpace at cell level—each cell that contains information is mapped to the DesignSpace.

Domain-specific Tools. We have implemented adapters for three tools commonly used by technology companies: i) Eclipse IDE, a source code development tool commonly used in software engineering, ii) ProEngineer, an integrated 3D CAD/CAM/CAE solution commonly used in various engineering domains (e.g., mechatronical systems), and iii) IBM RSA.

6.2 Case Studies

The DesignSpace has been applied in three different case studies.

EPlan. In this case study, engineers used the DesignSpace's services to establish traceability between enterprise artifacts, specifically between EPlan electrical models and Java source code. Consistency between the electrical model and source code was checked based on a set of user-defined, domain-specific rules. Even with large models and source code bases, engineers did not encounter any issues regarding the responsiveness of the DesignSpace. Knowledge adaptations were handled live during workflows by tool-adapters and consistency information as well as change notifications were provided in tools without noticable delays.

ACCM Robot Arm. The DesignSpace has also been used in the mechatronics domain as a platform for designing a robot arm. The project involved various kinds of enterprise artifacts. For example, mechanical calculations were provided in the form of multiple Excel spreadsheets. IBM RSA was used to build models of the robot's controller software. 3D CAD models and Matlab simulation models were also built and integrated with the DesignSpace. All artifacts were represented in the DesignSpace and traceability between the artifacts was established. The DesignSpace's data services were used to check consistency among artifacts and to notify engineers about relevant artifact changes.

ACCM Visualization Experiment. In this application, different enterprise artifacts for mechatronical development projects (e.g., requirements, mechatronic design models) were integrated in the DesignSpace and traces between those artifacts were established by domain experts. Students were then asked to perform defined refactorings (i.e., adaptations of existing knowledge) using the services provided by the DesignSpace, which they did with overall great success.

Summary. Overall, the DesignSpace has been used successfully in different domains to integrate enterprise artifacts, to establish traceability among them, to check consistency, and to notify agents about knowledge changes.

6.3 Threats to Validity

Next, we discuss some possible threats to the validity of our approach.

Integration Effort While for typical formats of enterprise models it is possible to use standardized adapter, this is often not possible for domain-specific enterprise artifacts that may require specialized adapters. In practice, this means that enterprises adopting our approach will need to consult with information engineering experts to decide on these questions. However, the DesignSpace provides a sophisticated programming interface as well as an extensive documentation. Therefore, building file- or tool-adapters is quite straight-forward.

Trace Creation and Management Traces are one of the cornerstones of our approach, thus it is crucial that they are established and also managed, which seems to be error-prone when considering the vast amount of knowledge and the diversity of artifacts that exist in today's enterprises. However, the DesignSpace supports different ways of establishing and managing traces. First, heuristic algorithms may be used to generate and maintain traces automatically. Second, the DesignSpace provides tools that support agents in creating and managing traces. In first industrial applications (see above), agents have not encountered any issues with creating or managing traces. Thus, using the DesignSpace to integrate and trace diverse enterprise artifacts has been shown to be feasible.

Kinds of Integrated Models and Artifacts In the case studies presented above, only enterprise artifacts but no enterprise models have been used. However, in principle there is no difference between tracing enterprise artifacts and enterprise models, as both are represented uniformly within the DesignSpace. Moreover, the software models that have been used in the case studies are structurally quite similar to typical enterprise model (i.e., graph-like data structures). Therefore, we believe that our approach is applicable also to enterprise models.

7 Related Work

Antunes et al. [12] proposed the use of ontologies to enable sophisticed EA analysis through the use of description logics. While the DesignSpace is generally agnostic to ontologies, the ontologies of enterprise models and artifacts can also be managed by the DesignSpace. The additional information can be used by services to perform, for example, more sophisticated consistency checking. A benefit of using the DesignSpace is that it does not require the explicit definition of ontologies. Moreover, the DesignSpace has been built with incrementality as key feature—any change to enterprise knowledge is analyzed immediately and automatically, and feedback to potentially affected agents is available instantly.

Florez et al. [1] proposed an approach that allows for the explicit modeling of imperfections in enterprise models that occur because of, for instance, incorrect information or missing information sources. Note that when using the DesignSpace services for modeling an enterprise, such imperfections are detected automatically. Information about these imperfections is available to modelers instantly. However, are free to ignore inconsistencies for the time being, thus accepting the detected imperfections temporarily. Moreover, the DesignSpace is in principle capable of detecting imperfections in enterprise models represented in any language.

In [13], Florez et al. presented requirements for enterprise model analysis tools. The DesignSpace meets critical requirements: it supports metamodel extensibility and partial model conformity, it provides a framework for implementing analysis methods, it displays and exports analysis results, and it allows for storing analysis results within the models.

8 Conclusion and Future Work

In this paper, we presented how the cloud-based knowledge-sharing platform *DesignSpace* can be applied in enterprise modeling to address common issues of too coarse-grained support for enterprise artifacts in workflows. First case studies with a prototype implementation suggest that the approach is feasible and scalable. For future work, we plan to provide more adapters for different kinds of existing enterprise knowledge and to apply our prototoype in industrial companies.

References

1. H. Florez, M. E. Sánchez, and J. Villalobos, "iarchimate: A tool for managing imperfection in enterprise models," in *EDOC Workshops*, pp. 201–210, 2014.
2. V. A. Pinto, C. L. de Rezende Rohlfs, and F. S. Parreiras, "Applications of ontologies in enterprise modelling: A systematic mapping study," in *ER Workshops*, pp. 23–32, 2014.
3. G. Berio and F. B. Vernadat, "New developments in enterprise modelling using cimosa," *Computers in Industry*, vol. 40, no. 2, pp. 99–114, 1999.
4. S. Gudas, A. Lopata, and T. Skersys, "Approach to enterprise modelling for information systems engineering," *Informatica*, vol. 16, no. 2, pp. 175–192, 2005.
5. F. Vernadat, "Ueml: towards a unified enterprise modelling language," *International Journal of Production Research*, vol. 40, no. 17, pp. 4309–4321, 2002.
6. J. Petrikina, P. Drews, I. Schirmer, and K. Zimmermann, "Integrating business models and enterprise architecture," in *EDOC Workshops*, pp. 47–56, 2014.
7. S. Sunkle and H. Rathod, "Visual modeling editor and ontology api-based analysis for decision making in enterprises - experience and way ahead," in *EDOC Workshops*, pp. 182–190, 2014.
8. U. Frank, "Enterprise modelling: The next steps," *Enterprise Modelling and Information Systems Architectures*, vol. 9, no. 1, pp. 22–37, 2014.
9. F. B. Vernadat, "Enterprise modeling and integration (emi): Current status and research perspectives," *Annual Reviews in Control*, vol. 26, no. 1, pp. 15–25, 2002.
10. D. Barone, E. Yu, J. Won, L. Jiang, and J. Mylopoulos, "Enterprise modeling for business intelligence," in *The practice of enterprise modeling*, pp. 31–45, Springer, 2010.
11. A. Demuth, M. Riedl-Ehrenleitner, A. Nöhrer, P. Hehenberger, K. Zeman, and A. Egyed, "DesignSpace – An Infrastructure for Multi-User/Multi-Tool Engineering," in *SAC*, 2015. Accepted for publication.
12. C. Antunes, A. Caetano, and J. L. Borbinha, "Enterprise architecture model analysis using description logics," in *EDOC Workshops*, pp. 237–244, 2014.
13. H. Florez, M. E. Sánchez, and J. Villalobos, "Extensible model-based approach for supporting automatic enterprise analysis," in *EDOC*, pp. 32–41, 2014.