TOWARD GENERIC CONSISTENCY PATTERNS IN MULTI-VIEW ENTERPRISE MODELLING

Research paper

Awadid, Afef, University Paris 1 Pantheon-Sorbonne, Paris, France, afef.awadid@malix.univ-paris1.fr

Bork, Dominik, University of Vienna, Vienna, Austria, dominik.bork@univie.ac.at

Karagiannis, Dimitris, University of Vienna, Vienna, Austria, dk@dke.univie.ac.at

Nurcan, Selmin, University Paris 1 Pantheon-Sorbonne, Paris, France, selmin.nurcan@univ-paris1.fr

Abstract

Modelling methods are applied for decades focussing on the analysis and design of holistic conceptual representations of entire enterprises. Creation and interpretation of such holistic representations in one single model is unrealistic. It is therefore widely adopted to refer to multi-view enterprise modelling methods. Such methods decompose the holistic model into views that focus only on some aspects while omitting others. The views, however, rather usually than rarely overlap. Usability and utility of multi-view enterprise models therefore significantly depends on consistency between all views. This paper explores enterprise modelling methods with a focus on multi-view consistency, thereby contributing a thorough investigation of the syntactic and semantic overlaps between views. It then abstracts from concrete examples to derive generic consistency patterns which can be specified in a novel formalism. Utility of the formalism is evaluated in two case study applications. One strength of this approach is its conceptual nature, enabling its adoption by method engineers who not necessarily have a computer science background. The consistency patterns facilitate specification of modelling methods while the formalism eases the validation of models, and the implementation of multi-view modelling tools.

Keywords: Multi-view modelling, consistency, patterns, enterprise modelling, metamodelling.

1 Introduction

Enterprise modelling (EM) has been common practice for decades (Sandkuhl et al., 2016). EM aims to establish a holistic representation of the entire enterprise, thereby creating models of "goals and processes, organization and products structures, IT-systems" (Zdravkovic et al., 2015). With the technological changes (e.g., Internet of Things), the economic changes (e.g., platform ecosystems or product-service systems), and the fierce competition due to the possibilities of innovative start-ups, enterprises are today more than ever liable to continuous change. This change is often referred to as 'transformation' or more recently as 'digital transformation', considering the paramount role of information technology.

Enterprise models are powerful for managing this digital transformation, especially when utilized properly (cf. Sandkuhl et al., 2016). EM can act as a formalized knowledge base enabling intersubjective understanding and machine-processing (Bork and Fill, 2014; Giannoulis et al., 2013) or for creating genuinely flexible, controllable, and usable models of an enterprise (Bork and Alter, 2018). It is not feasible to cover all aspects of an enterprise in one overarching model, therefore multi-view modelling methods have been employed. These methods decompose the overarching enterprise model into multiple views to foster understanding by human beings. The separation of the enterprise models into, e.g., structural, behavioral, technological, procedural, capability, contextual views (Loucopoulos and Kavakli, 2016; Zdravkovic et al., 2015) is mostly redundant, i.e., syntactic and/or semantic overlaps are predominant. The description of these overlaps and the preservation of consistency between all views are vital, e.g., for utility of the method and proper tooling (Bork, 2015; Lopez-Herrejon and Egyed, 2011).

When looking at the specification of EM methods, a high level of detail is given for the modelling language and its decomposition into views. However, only scarce information can be found regarding the view overlaps and consistency requirements. On the one hand, authors might deem that consistency is inherently specified by having a closer look at the decomposition specifications. On the other, specifying consistency requirements comprehensively for large EM languages is particularly time consuming and challenging. One aspect that contributes to this situation is the lack of: *a procedure* one can follow, and *a formalism* one can use. This hinders validation of models, machine processing of models, and the conceptualization of EM tools. The research applied to realize the paper at hand was guided by these research questions: *RQ1: Which overlaps between multi-view conceptual models can be distinguished? RQ2: How can generic consistency patterns be specified in a technology-agnostic manner?*

This paper contributes a formalism that enables the precise specification of consistency requirements that targets the aforementioned weaknesses in a procedural way. Following the design science research (DSR) methodology, we develop generic multi-view consistency patterns as a new artefact of type *construct*. A DSR construct artefact can be a “concept, assertion, or syntax that has been constructed from a set of statements, assertions, or other concepts” (Peffers et al., 2012, p. 401). As an evaluation, the artefact is applied within two real EM case studies showing its suitability (Peffers et al., 2012, p. 402).

This paper first provides a sound foundation on multi-view enterprise modelling, consistency between views, and related works (Section 2). Afterwards, consistency issues of several enterprise modelling methods are analyzed in Section 3. The identified, concrete consistency requirements are then abstracted toward generic consistency patterns and a formalism for their specification in Section 4. In Section 5, the patterns and the formalism are used to specify the consistency requirements of two enterprise modelling methods. A SWOT analysis is presented in Section 6. Finally, some conclusions are drawn.

2 Foundations and Related Works

2.1 Multi-view Enterprise Modelling

The notions "view" and "multi-view modelling" have various meanings depending on the application domain (Kheir et al., 2013). A prominent perception is employed in enterprise architecture management by the ISO/IEC/IEEE 42010 (International Organization for Standardization, 2011) standard. There, a view is defined as a “work product expressing the architecture of a system from the perspective of specific system concerns”. The view is derived by the requirements of specific stakeholders and their purposes and stays on a conceptual level.

Within this paper, a multi-view modelling method is regarded as a specialized instantiation of the generic modelling method framework as introduced by Karagiannis and Kühn (2002) (see Figure 1). A modelling method is composed of: (i) a *modelling language* that defines the *syntax* (concepts and their relationships, e.g., in form of a meta model), the *semantics* (explicit properties in concept schemata), and the notation (the graphical representation of the constructs), (ii) a *modelling procedure* that establishes the steps to be followed in order to attain various modelling goals, and (iii) *mechanisms & algorithms* that process the knowledge in diagrammatic models (Karagiannis, Buchmann, et al., 2016).

Our generic viewpoint-based multi-view modelling theory is then applied in relation to the generic modelling method framework. Each view is an instance of a viewpoint. The viewpoint specifies the modelling language that one can use when creating a view of that specific type. Thus an analogy to the relationship between meta model and model in conceptual modelling (Bork et al., 2015) is also employed between viewpoint and view. Based on whether or not one overarching meta model exists, *projective* and *synthetic* multi-view modelling methods can be distinguished (Cicchetti et al., 2011a). In the former case, views can be derived by projecting certain concepts out of the overarching meta model. By contrast, in the latter case every view has its distinct meta model. Considering the design process of multi-view modeling methods, projective follows a top-down approach, i.e., first the overarching meta model is developed, then the projection operators that derive the viewpoints from it. In the synthetic case, a bottom-up approach is followed by integrating previously unrelated and distinct meta models.

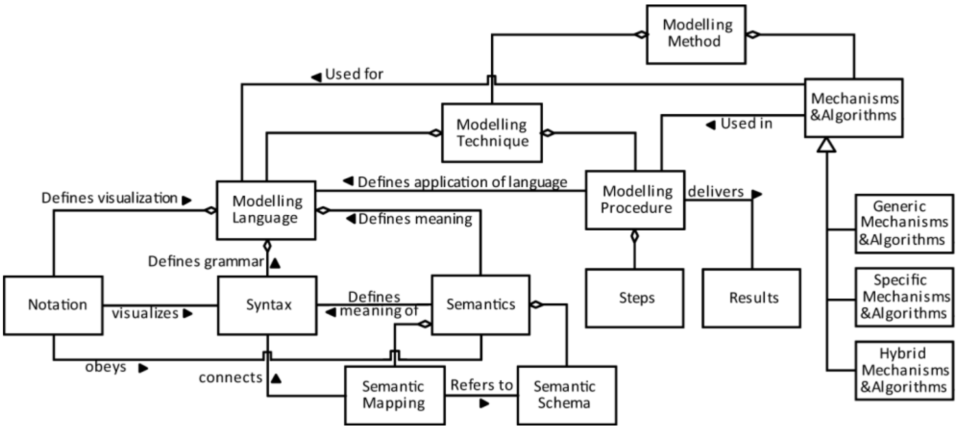

Figure 1. Components of modelling methods (Karagiannis and Kühn, 2002)

2.2 Multi-View Consistency

When multiple views represent the same system under study, overlaps are inevitable. Those overlaps can be of *syntactic* nature in cases where at least one syntactic element is part of two viewpoints, and/or *semantic*, in cases where different syntactic elements in two viewpoints represent the same aspect of the system under study. The notion of *inter-view consistency* then refers to the extent to which the information contained in multiple views does not contradict each other. Figure 2 visualizes different kinds of inter-view relationships proposed by Persson et al. (2013). In the following, we will concentrate on such methods that have *syntactic* and *semantic overlaps*. If syntax and semantics of two views are orthogonal/independent to/of each other, we don't consider them a multi-view modeling method but a loose coupling of multiple modeling languages. Consistency requirements do not exist in such cases.

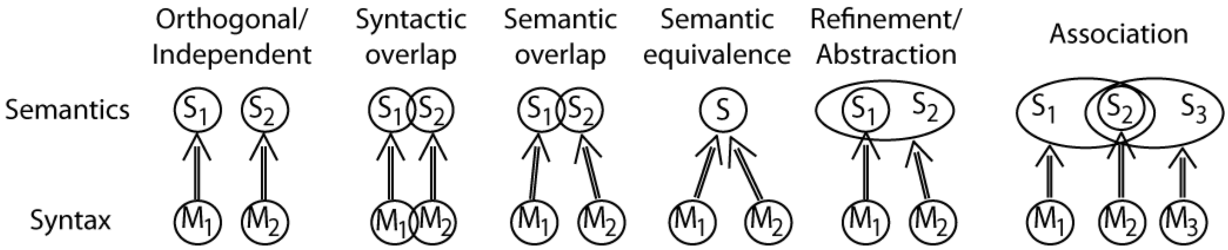

Figure 2. Inter-view Relationships (Persson et al. 2013)

A multi-view model is characterized as being consistent, if all views are syntactically and semantically consistent to each other. Keeping the multiple views consistent and providing suitable visualization means is crucial for the utility and the applicability of a multi-view modelling method (Bork, 2015), since the views are interrelated and their semantics and/or syntax is overlapping. From here arises the challenging and foremost question we intend to address in this paper: *How to identify and formally specify such overlaps by means of consistency requirements?* In order to derive an answer, we first review related works, before we contribute our new formalism and procedure for specifying consistency requirements of multi-view modeling methods.

2.3 Related Works

Identification and management of consistency between multiple views is not new in itself. Lots of research can be found in the field of systems analysis and design, not only limited to enterprise modelling. Regardless of the concerned field, there are primarily two preoccupations when dealing with consistency: (1) Consistency preservation, i.e., how to ensure consistent views; and (2) consistency checking, i.e., how to verify whether the views are consistent (Awadid and Nurcan, 2017). Considering these two preoccupations, two main research streams can be derived which will be delineated in the following. Afterwards, some approaches to formally specify requirements of modelling methods are discussed.

In the first research stream, different consistency preservation techniques have been proposed. One of the most common techniques is model transformations in model-driven development. Model transformations are often used to ensure consistency between a source model and a target model. Rules are specified that define which patterns in a source model can be transformed into patterns in the target model. When considering the involved models and the transformation specification as a graphs, Triple Graph Grammars (TGG) can be employed (Guerra et al., 2013). In this vein, (Hermann et al., 2015) propose a formal synchronization framework based on TGGs in order to preserve consistency between source and target models. Moreover, model transformations have been used to automatically restore consistency between multiple concurrently modified models (e.g., (Cicchetti et al., 2011b; Ehrig et al., 2015; Poskitt et al., 2014). One other recent approach for consistency preservation was published by Karagiannis et al. (2016). The approach is based on semantic graphs derived from diagrammatic models, and queries acting upon these graphs. As application scenarios, view transformations, view synchronization, and passive view consistency checks are presented.

The focus of the second research stream is on supporting consistency checking of distinct models. Examples of this stream appear in the area of software and systems modelling with UML, where several consistency checking formalisms have been proposed. See e.g., Object Constraint Language (Millan et al., 2009; Object Management Group (OMG), 2014), Query View Transformation Language (Object Management Group (OMG), 2016), or Alloy (Farias et al., 2017; Gammaitoni et al., 2017). Many other approaches build on the constraint programming paradigm as a basis for checking consistency between UML models (e.g., (Cabot et al., 2014; Mazo et al., 2011; Vierhauser et al., 2012)).

All of the mentioned works apply technical means either to preserve (first research stream) or to check (second research stream) consistency. Accordingly, they are powerful in the context of automating transformations between views or the transformative generation of views. However, all of these approaches require a deep understanding of the formalisms, targeting at technologically affine users. Furthermore, they all focus on syntactic aspects, omitting semantics and notation to a great extent.

When searching for related works in the field of requirements engineering and specification techniques for modelling methods, only limited literature can be found that also comprehends inter-viewpoint consistency. FDMM (Fill et al., 2012), is a formalism for describing ADOxx meta models and models. FDMM has been successfully applied for divers EM methods, e.g., HORUS (Fill et al., 2013) or RUPERT (Johannsen and Fill, 2016). Its strength is in its expressive power for meta models conceptualized on the ADOxx meta modelling platform (Fill and Karagiannis, 2013). A recent approach for the specification of modelling methods by utilizing a domain-specific language called MM-DSL has been proposed by (Visic et al., 2015). Whereas the approach seems appealing, it is currently in a draft state. Due to their formal foundation, the application of FDMM and MM-DSL currently requires preparation and studying of the relevant publications.

Compared to these approaches, the focus of this paper is on investigating the specifics of inter-view consistency requirements between conceptual modelling views and proposing a technology-agnostic specification formalism by re-purposing the generic modelling method framework (see Figure 1). This formalism provides conceptual support for both, consistency preserving and checking of multi-view conceptual models. Thus, the contribution of this paper is not a technical solution but a procedure and formalism to elicit consistency requirements guiding method engineers during the specification of multi-view modelling methods. The approach aims to balance the requirement of being precise enough for tool developers while remaining to being applicable by non-technical users using pen and paper.

3 Toward the Identification of Consistency Patterns

This section reports on the results of an explorative research approach we applied to identify prevalent consistency patterns in multi-view EM methods. At first, we analysed several different methods, e.g. MEMO (Frank, 2014), Semantic Object Model (SOM) (Ferstl et al., 2016; Ferstl and Sinz, 2013), Enterprise Knowledge Development (EKD) (Barrios and Nurcan, 2004; Loucopoulos et al., 1997; Loucopoulos and Kavakli, 1999), and ARIS (Scheer and Schneider, 2006). Given that each modelling method describes an enterprise in its own way, many different descriptions were investigated. However, even informal specifications of view relationships were scarcely found. In the following, we will therefore focus on the EKD and SOM methods. This is mainly due to two reasons: (1) The chosen methods are among the few ones for which consistency requirements have been explicitly defined to some extent, and (2) the space limitation of the paper. Furthermore, we concentrate the analysis on the business process related aspects of the methods, as business process modelling itself requires the integration of multiple perspectives (Awadid, 2017; Awadid and Nurcan, 2016; Letsholo et al., 2012; Sousa et al., 2007).

3.1 Consistency Patterns in EKD

This section briefly introduces the business process modelling part of the EKD method with an emphasis on their consistency requirements. An EKD business process is comprehensively modelled by three interrelated views: the *Actor-Role* view, the *Role-Activity* view, and the *Business Objects* view. Table 1 shows a brief description of the EKD business process views (Awadid and Nurcan, 2016).

EKD Viewpoint	**Viewpoint Description**
Actor-Role view	A high-level view of the association between actors and the different responsibilities (called roles) that they hold in different processes.
Role-Activity view	A detailed view of the activities in which a role of a given actor is involved.
Business Objects view	Depicts resources (physical or informational) that are required by one or more activities being performed by a role.

Table 1. EKD business process modelling viewpoints.

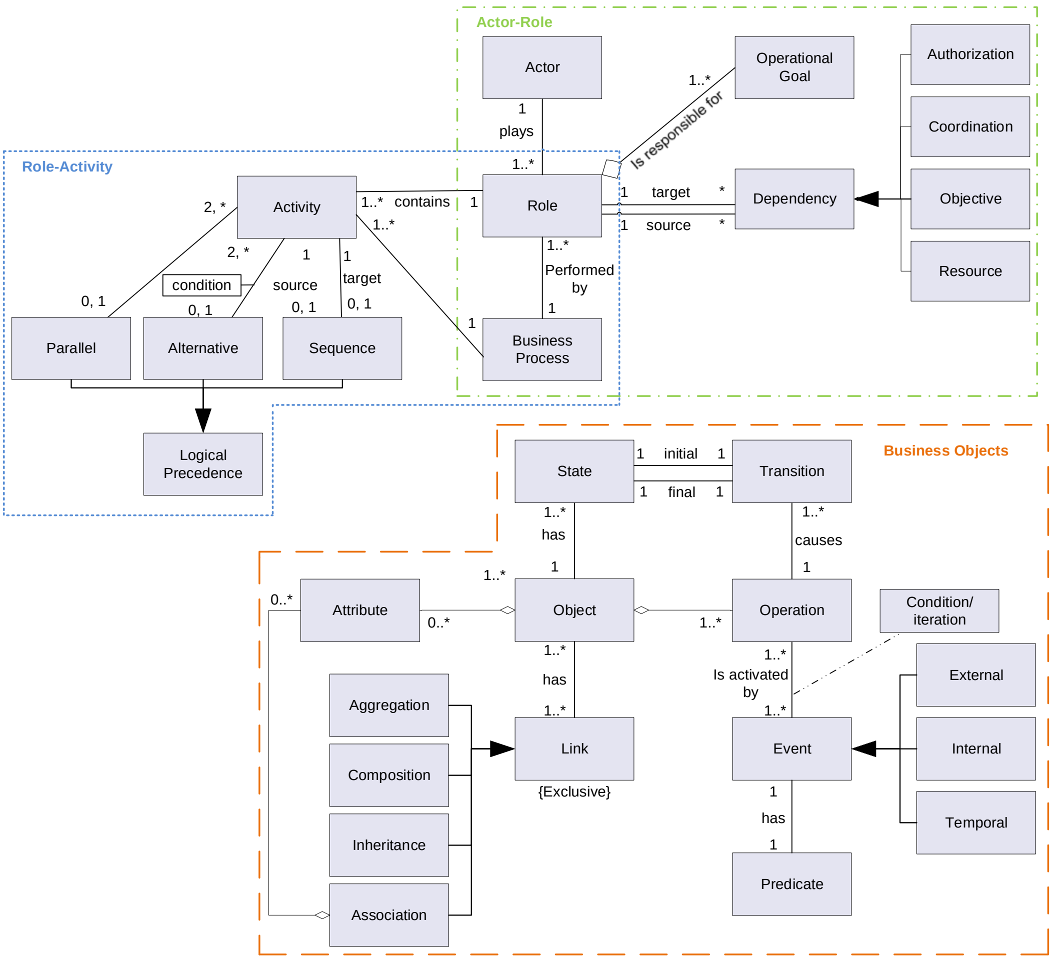

Figure 3. EKD business process meta model (adapted from (Awadid and Nurcan, 2016))

Figure 3 visualizes the three meta models of the viewpoints together with their relationships. It can be seen, that the meta models comprise syntactic overlaps (cf. Figure 2), e.g., the concepts *Role* and *Business Process* are part of the *Actor-Role* and the *Role-Activity* viewpoint. Besides the syntactic overlaps which can be identified rather easy by looking at the meta models, EKD also has semantic overlaps (cf. Figure 2), e.g., between the concept *Object* in the *Business Objects* viewpoint and the concept *Activity* in the *Role-Activity* viewpoint by means of a *use / produce* relationship. A second semantic overlap relates a *Business Process* in the *Actor-Role* view with at least one *Activity* in the *Role-Activity* view. Such semantic overlaps are scarcely visible in meta models and therefore require expert domain knowledge to be identified.

Figure 4 visualizes a simplified EKD multi-view business process model with some highlighted overlaps. It shows a car booking process centering the actors *Customer*, *Travel Agent* and *Cab Operator* and their roles in the Actor-Role view on the upper left corner. On the lower area, the Role-Activity view details the set of activities performed by a certain role (e.g., as highlighted for the *Cab driver*). On the upper right corner, the corresponding Business Objects view is visualized, depicting the objects involved in the business process, e.g., the *Cab booking request* object in the Business Objects view which is derived from the *Dependency Resource* relationship *A phone call* in the Actor-Role view.

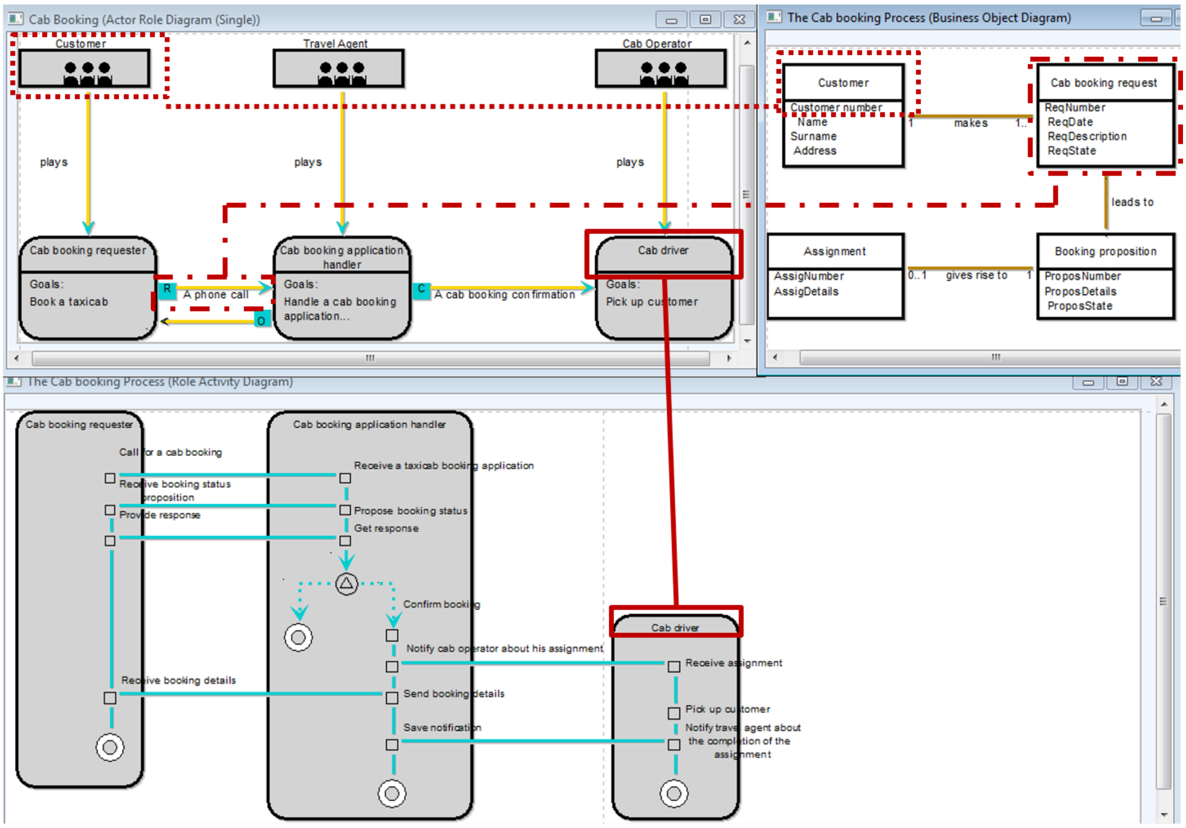

Figure 4. An example of a multi-view business process model in EKD.

In the following, the semantic overlaps are investigated in more detail. The concept of an *overlapping concept* is introduced. An overlapping concept (OC) is a concept that is represented in two different viewpoints by either the same syntactic element (i.e., *syntactic view overlaps*), or by different syntactic elements (i.e., *semantic view overlaps*). Referring to the discussed syntactic view overlap of the concept *Role* in the EKD meta model depicted in Figure 3, one would expect that the exact same concept (regarding its syntax, semantics, and notation) is part of two views. However, the realization as visualized in the example in Figure 4 shows, that the concept *Role* has two different notations: a flat rounded rectangle with *responsible goals* in the Actor-Role viewpoint, and an elevated rounded rectangle with all *contained activities* in the Role-Activity viewpoint.

Table 2 shows an excerpt of the identified overlaps in EKD. For each overlapping concept (OC), its counterparts in the corresponding EKD viewpoints (VP) are listed. The symbol 'x' means that no counterpart exists in the current viewpoint. Parenthesis are used to indicate whether an instance in a viewpoint is an object or a relation. Brackets are furthermore used to define the attributes that establish the semantic linkages between multiple viewpoints.

EKD VP / OC	Actor-Role	Role-Activity	Business Objects
Actor	Actor (Object)	X	Business Object (Object)
Role	Role (Object)	Role (Object)	X
Operational Goal	Role (Object) {Operational Goal}	A sequence of at least 2 activities (Object) related to that operational goal {relates}.	X
Dependency Resource	Dependency Resource (Relation)	X	Business Object (Object)
Business Object	X	Resource {Used or produced}	Business Object (Object)

Table 2. Excerpt of semantic view overlaps of the EKD method.

3.2 Consistency Patterns in SOM

The backbone of the SOM methodology is an enterprise architecture which uses different perspectives on a business system via a set of models. These models are grouped into three model layers referring to a business plan, business process models, and resource models. In order to reduce complexity, each model layer is subdivided into several viewpoints, each focusing on specific aspects of a model layer. Table 3 lists and briefly describes the viewpoints of the SOM business process layer.

SOM Viewpoint	Viewpoint Description
Interaction Schema	A structural perspective on the business process. It shows the relationships between business objects and business transactions, thereby revealing the employed coordination principles.
Task-Event Schema	A behavioural perspective on the business process by visualizing the sequence of tasks necessary to execute a business process.
Transaction Decomposition	A tree-based structure of the decomposition rules applied to the set of initial business transactions.
Object Decomposition	A tree-based structure of the decomposition rules applied to the set of initial business objects.

Table 3. SOM business process modelling viewpoints.

Figure 5 shows the integrated SOM business process meta model. It can be seen, that the SOM business process model comes with syntactic view overlaps (cf. Figure 2), e.g., the concept *Business Transaction* is part of the *Interaction Schema*, the *Task-Event Schema*, and the *Transaction Decomposition* viewpoints. Moreover, semantic view overlaps can be derived, e.g., one *Business Object* is related to *at least one Task* in the Task-Event Schema by means of a *comprises* relationship.

In Figure 6, a simplified SOM business process model with all four views is visualized: Transaction Decomposition on the top left, Object Decomposition on the lower left, Interaction Schema on the top right, and the Task-Event Schema on the lower right. The business process describes on a high level of abstraction how a *seller* is coordinating the *sell products* process with the *buyer*. A transaction-based coordination is employed, decomposing the initial business transaction into a sequence of an *initiating transaction 'advertise products'*, a *contracting transaction 'negotiate conditions'*, and an *enforcing transaction 'sell product'*. Due to limited space, readers interested in a detailed introduction to SOM need to be referred to the literature (Ferstl et al., 2016; Ferstl and Sinz, 2006).

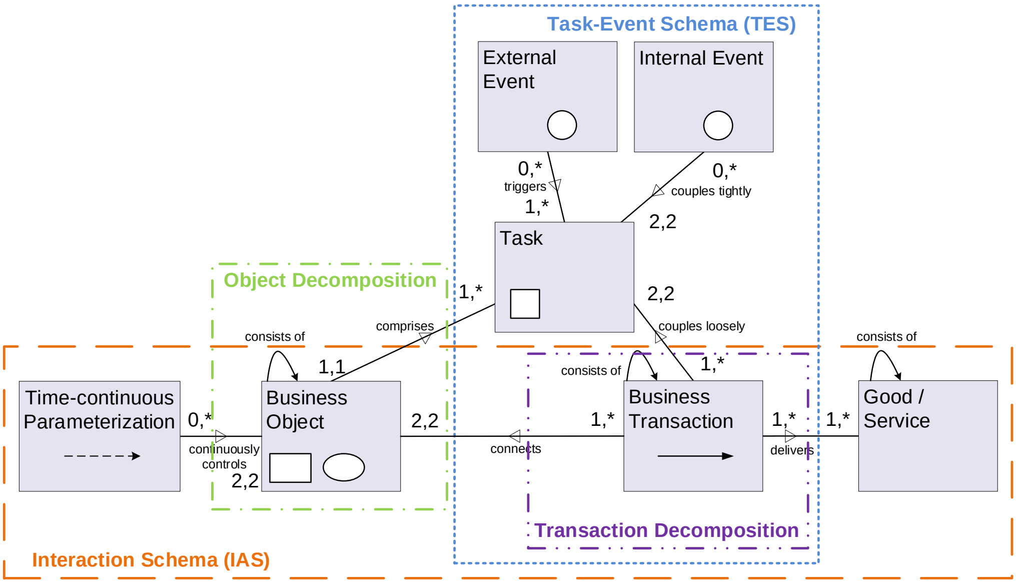

Figure 5. SOM business process meta model (adapted from (Ferstl and Sinz, 2006)).

It can be derived from Figure 6, that the four views share a number of syntactic view overlaps (e.g., the *seller* concept is part of multiple views), as well as semantic view overlaps (e.g., *buyer* is not only a concept in the Object Decomposition view but also an attribute of three *Tasks* in the Task-Event Schema). For readability purposes only some of the overlaps are visualized in the figure.

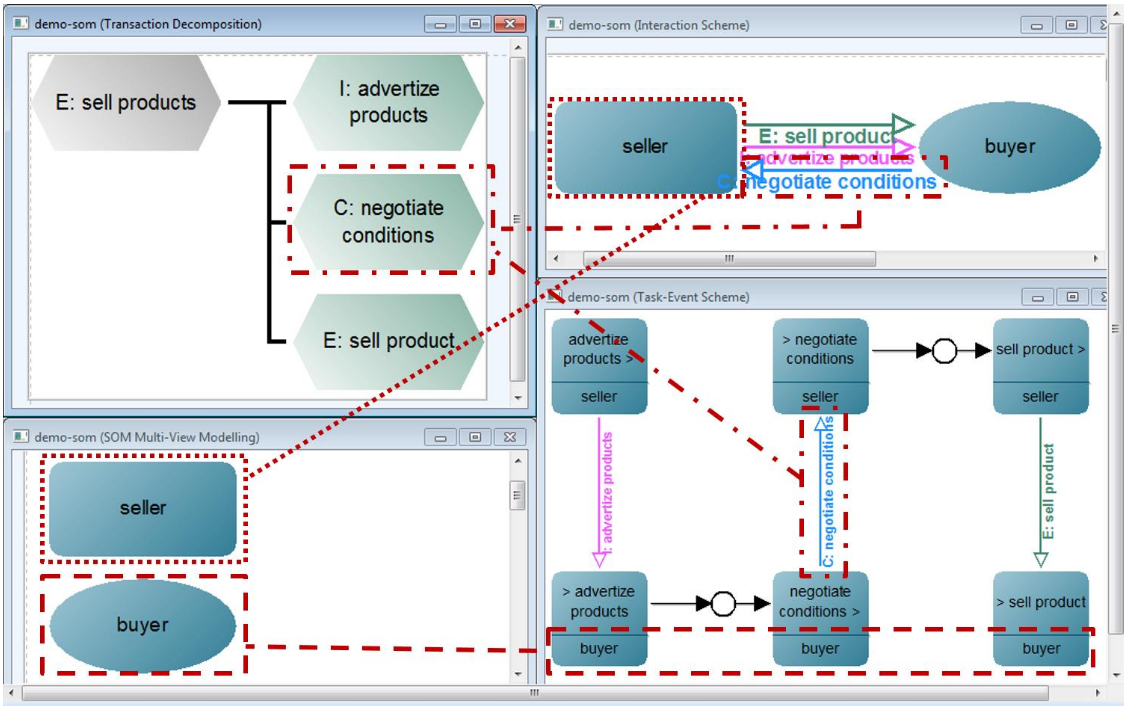

Figure 6. An example of a multi-view business process model in SOM.

In Table 4, the semantic view overlaps between the SOM business process model views are described in more detail by applying the overlapping concept as introduced previously. In SOM, some concepts, e.g., the Business Transaction are part of several viewpoints. The meta model visualized in Figure 5 indicates, that in all three viewpoints the same syntactic concept is used. However, the Business Transaction concepts is represented by a *modelling class* in the Transaction Decomposition, but by a *relation class* in the Interaction Schema and Task-Even Schema (indicated by the parenthesis in Table 4).

SOM VP / OC	Interaction Schema	Task-Event Schema	Object Decomposition	Transaction Decomposition
Business Transaction	Business Transaction (Relation)	Business Transaction (Relation); For each Business Transaction exactly two Tasks (Objects) need to be coupled {send/ receive task}.	X	Business Transaction (Object)
Business Object	X	For each Business Object at least one Task (Object) needs to be comprised.	Business Object (Object)	X
Task	Business Transaction (Relation) {couples}	Task (Object)	Business Object (Object) {comprises}	Business Transaction (Object) {couples}

Table 4. Excerpt of semantic view overlaps of the SOM method.

A different observation is related to the semantic view overlap between the Task concept in the Task-Event Schema (TES) and the Business Transaction. This semantic relationship between the two concepts is realized by an attribute of the Task concept in the Task-Event Schema. Hence, every Task in the Task-Event Schema has an attribute that refers to the corresponding Business Transaction by means of a *couples* relationship (indicated by the brackets {} in Table 4).

4 Generic Consistency Patterns

In the following, the concrete patterns identified in the preceding sections are abstracted to derive generic patterns that are independent of any modelling method. Furthermore, a generic formalism for multi-view consistency pattern specification is proposed.

4.1 A Generic Pattern Specification Framework

The utilization of meta models not only for the specification of a modelling language's syntax but as the primary source for consistency specifications (Buchmann and Karagiannis, 2016), or the integration between conceptual models and ontologies (Fill and Burzynski, 2009), is already common practice. However, when focusing only on the syntactic aspects, semantics and notation of the modelling languages are not considered. As the analysis in Section 3 showed, especially the semantic view overlaps are challenging to be identified and managed. Consequently, and by referring to the generic modelling method framework (see Figure 1), the multi-view patterns comprise all modelling language components. Thus, three criteria for multi-view patterns can be derived: *syntax*, *semantics*, and *notation*. As a fourth criterion, *cardinality* is introduced. Cardinality is an established concept for specifying constraints on modelling languages. This criterion supports cases, where no 1:1 mapping between syntactic concepts is given. Table 5 provides a description of each consistency pattern criteria.

Pattern Criteria	**Description**
Syntax	The syntactic elements of the viewpoints affected by the overlap. A syntactic element can be either a modelling class, a relation class, or an attribute that represents a semantic link to a modelling class or relation class.
Semantics	The semantic meaning of the concepts affected by the overlap.
Notation	The graphical representation (i.e. visualization) of the affected concepts.
Cardinality	The quantitative aspects for the syntactic elements of the affected viewpoints.

Table 5. Multi-view consistency pattern specification criteria.

4.2 A Formalism for the Specification of Multi-View Consistency Patterns

Using the criteria introduced in Table 5, one can create a generic specification of a multi-view consistency pattern. The generic patterns always analyse a pair of two concepts of two viewpoints (see Figure 7). We believe this is the only feasible approach in order to manage the complexity of identifying consistency issues and developing consistency-preserving mechanisms. In cases where more than two views ($n > 2$) are given, the patterns need to be applied $n * (n - 1)$ times in order to realize a comprehensive specification of the multi-view consistency requirements. Every application comprises an analysis of all pairs of concepts of the involved viewpoints.

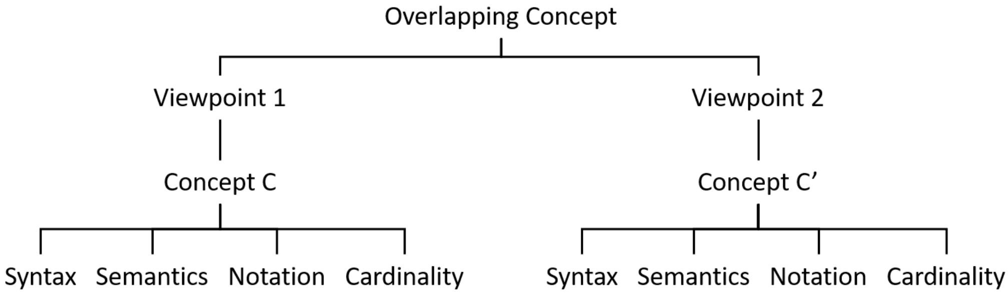

Figure 7. Overlapping concept pattern in multi-view models.

A generic specification of a multi-view consistency pattern for a source concept (C, equation 1) and an overlapping concept in a second viewpoint (C', equation 2) is given by the following quadruple:

(1) C = (Syn, Sem, Not, Card)

(2) C' = (Syn', Sem', Not', Card')

For each criterion in the Consistency Pattern *Pattern (C, C')* can be defined, whether a similarity is given or a translation needs to be specified, indicated by the → symbol in equation 3.

(3) Pattern (C, C') = (Syn → Syn', Sem → Sem', Not → Not', Card → Card')

An overlapping concept can be *semantically equivalent* but *syntactically and notationally different*, or *syntactically and semantically equivalent*, but *notationally different* etc. All consistency pattern criteria are independent from each other, therefore similarity needs to be analysed independently for all of them.

5 Evaluation

Evaluation of the proposed generic multi-view consistency patterns is performed by two use cases (Peffers et al., 2012, p. 402). This evaluation shows the suitability of the artefact, the generic consistency patterns and the formalism, in a real-world scenario with real enterprise modelling methods. Thus, the generic consistency pattern formalism is first used to specify the consistency patterns of the EKD method and then also for the SOM method. Table 6 and Table 7 list an excerpt of the consistency patterns together with a brief rationale. For each criterion, similarity is indicated by '=' and dissimilarity by '!='.

EKD VP / OC	Actor-Role view	Role-Activity view	Business Objects view
Actor	C= (**SYN** =Actor, **SEM** = a participant of the business process, **NOT** = rectangle, **CARD** = 1)	X	C'= (**SYN** =Business Object, **SEM** = data produced/used in the business process, **NOT** = class, **CARD** = 1)
	Rationale: For any Actor in the Actor-Role VP, one Business Object in the Business Objects VP needs to be given. **Pattern(C, C') = (Syn != Syn', Sem = Sem', Not != Not', Card = Card')**		

Role	C= (**SYN** = Role, **SEM** = the responsibilities of a business process participant, **NOT** = Rounded Rectangle with Goals, **CARD** = 1)	C'= (**SYN** = Role, **SEM** = the responsibilities of a business process participant, **NOT** = Rounded Rectangle with Activities, **CARD** = 1)	X
	Rationale: For any Role in the Actor-Role VP, one Role in the Role-Activity VP needs to be given. **Pattern(C, C') = (Syn = Syn', Sem = Sem', Not != Not', Card = Card')**		
Operational Goal	C= (**SYN** = Operationsl Goals attribute, **SEM** = the goals associated to a specific role within the business process, **NOT** = list of goals within the Role notation, **CARD** = 1)	C'= (**SYN** = Activity, **SEM** = a set of activities necessary to achieve a goal, **NOT** = small square placed within the corresponding Role notation, **CARD** = 2..*)	X
	Rationale: For any Operational Goal in the Actor-Role VP, at least two activities in the Role-Activity VP need to be given. **Pattern(C, C') = (Syn != Syn', Sem != Sem', Not != Not', Card != Card')**		
Dependency Resource	C= (**SYN** = Resource Dependency, **SEM** = a necessary resource for a goal, **NOT** = arrow with 'R', **CARD** = 1)	X	C'= (**SYN** = Business Object, **SEM** = data required, **NOT** = class, **CARD** = 1)
	Rationale: For any Resource Dependency in the Actor-Role VP, one Business Object in the Business Objects VP needs to be given. **Pattern(C, C') = (Syn != Syn', Sem != Sem', Not != Not', Card = Card')**		

Table 6. Excerpt of EKD consistency patterns specifications.

SOM VP / OC	Interaction Schema	Task-Event Schema	Object Decomposition	Transaction Decomposition
Business Transaction	C= (**SYN** = Business Transaction, **SEM** = corrdination of business objects, **NOT** = arrow with transaction type as prefix, **CARD** = 1)	C'= (**SYN** = Business Transaction, **SEM** = corrdination of tasks of a business process, **NOT** = arrow with transaction type as prefix, **CARD** = 1)	X	C''= (**SYN** = Business Transaction, **SEM** = Business Transaction, **NOT** = Rectangle, **CARD** = 1)
	Rationale: For any Business Transaction in the Interaction Schema, one Business Transaction in the Task-Event Schema and the Transaction Decomposition need to be given. **Pattern(C, C') = (Syn = Syn', Sem != Sem', Not = Not', Card = Card')** **Pattern(C, C'') = (Syn = Syn'', Sem != Sem'', Not != Not'', Card = Card'')**			
Business Object	X	C'= (**SYN** = Responsible Business Object attribute, **SEM** = responsibility for task execution, **NOT** = rounded rectangle, **CARD** = 1..*)	C= (**SYN** = Business Object, **SEM** = either an environmental object or an object of discourse, **NOT** = Ellipse (env. Obj.) or Rectangle (Obj. of Disc.), **CARD** = 1)	X
	Rationale: For any Business Object in the Object Decomposition view, at least one task in the Task-Event Schema needs to be assigned. **Pattern(C, C') = (Syn != Syn', Sem != Sem', Not != Not', Card != Card')**			

Table 7. Excerpt of SOM consistency patterns specifications.

6 Discussion

The paper at hand introduced a procedure and a generic formalism for specifying multi-view consistency requirements. In the following, a discussion of the presented approach is performed by means of a SWOT analysis, thereby critically reflecting on strengths, weaknesses, opportunities, and threats.

Strengths: The presented approach contributes towards a precise specification of requirements for multi-view EM. Existing approaches are technology-driven and require specific knowledge in computer science - or even programming languages - to be utilized. By contrast, the presented approach builds upon a generic modelling method framework that enables to narrow down multi-view consistency requirements along the generic criteria syntax, semantics, notation, and cardinality. It therefore not only fills a research gap, but it also facilitates applicability by technology-agnostic users.

Weaknesses: The approach requires an understanding of conceptual modelling in general and the application domain of the method at hand. Moreover, the approach might be perceived as an unnecessary burden when thinking of modelling methods that only serve the purpose of grasping an overview of some complex phenomenon without the goal of processing the models. Hence, this approach is only meaningful, when consistency is perceived necessary.

Opportunities: The presented approach contributes towards comprehensive and precise specifications of multi-view modelling methods. This is beneficial for: a) modellers interested to learn/understand a modelling method and apply it correctly; b) researchers aiming to validate/extend a given modelling method, and c) tool developers responsible for realizing a modelling tool. In our future research, we aim to integrate the formalism in a model-driven approach therefore enabling e.g., the Open Models Laboratory (OMiLAB) community (Bork and Miron, 2017; Karagiannis, Mayr, et al., 2016) in analysis and design of multi-view modelling methods and their implementation on a meta modelling platform like ADOxx (Fill and Karagiannis, 2013). A second stream of future research will focus on the operationalization of the consistency patterns. In this regard, the patterns ease the model-driven development of consistency preserving mechanisms, e.g., by applying model queries and model transformations.

Threats: One thread to the validity of the presented approach is that it might be biased by the investigated EM methods. Moreover, one threat to validity is the potential bias by the authors who were in charge of both, conducting the analysis and performing the evaluation. We aim to work on both threats in our future work by publishing this approach and by enabling the modelling method community to apply and evaluate it with more multi-view modelling methods.

7 Conclusions

Grounded on a modelling method framework, this paper introduced a generic procedure and a formalism for the specification of consistency requirements of multi-view modeling methods. Applicability was evaluated by two case studies. Thereby, multi-view consistency requirements for the Enterprise Knowledge Development (EKD) and the Semantic Object Model (SOM) enterprise modelling methods were specified using the formalism. The case studies showed, that conventional meta model-based specifications, focusing on syntactic aspects only, are neither complete nor adequate with regards to the specific characteristics of semantically overlapping viewpoints. The proposed formalism tackles these shortcomings. Eventually, utilization of the formalism supports the design and development of consistency-preserving multi-view modeling methods.

Future research will focus on three aspects: (1) Exploitation of the formalism and its application to analyze a broader set of multi-view modelling methods. (2) Extension of the formalism, e.g., with regards to conditional consistency aspects. For some multi-view modelling methods (e.g., ComVantage (Buchmann, 2016)), constraints need to further incorporate the attribute values of concepts in multiple views. Hence, it is not sufficient to consider the meta model level of a method, but also the model level when specifying overlaps. (3) Operationalization of the formalism in a model-driven approach in order to further increases the value of the approach. Consistency patterns defined by the user may then be automatically transformed into consistency-preserving mechanisms.

References

Awadid, A. (2017), "Supporting the consistency in multi-perspective Business Process Modeling: A mapping approach", in Brighton, U. (Ed.), *2017 11th International Conference on Research Challenges in Information Science (RCIS)*, IEEE Computer Society, pp. 414–419.

Awadid, A. and Nurcan, S. (2016), "Towards Enhancing Business Process Modeling Formalisms of EKD with Consistency Consideration", *Tenth IEEE International Conference on Research Challenges in Information Science (RCIS)*, IEEE, Grenoble, France, pp. 1–12.

Awadid, A. and Nurcan, S. (2017), "Consistency requirements in business process modeling: a thorough overview", *Software & Systems Modeling*, pp. 1–19.

Barrios, J. and Nurcan, S. (2004), "Model Driven Architectures for Enterprise Information Systems", in Persson, A. and Stirna, J. (Eds.), *Advanced Information Systems Engineering. CAiSE 2004. Lecture Notes in Computer Science, Vol 3084*, Springer, Berlin, Heidelberg, pp. 3–19.

Bork, D. (2015), *A Development Method for the Conceptual Design of Multi-View Modeling Tools with an Emphasis on Consistency Requirements*, University of Bamberg.

Bork, D. and Alter, S. (2018), "Relaxing Modeling Criteria to Produce Genuinely Flexible, Controllable, and Usable Enterprise Modeling Methods", in Fellmann, M. and Sandkuhl, K. (Eds.), *9th International Workshop on Enterprise Modeling and Information Systems Architectures (EMISA)*.

Bork, D., Buchmann, R. and Karagiannis, D. (2015), "Preserving Multi-view Consistency in Diagrammatic Knowledge Representation", in Zhang, S., Wirsing, M. and Zhang, Z. (Eds.), *8th International Conference on Knowledge Science, Engineering and Management. Lecture Notes in Computer Science, Vol 9403*, Vol. 1, Springer Cham, pp. 177–182.

Bork, D. and Fill, H.G. (2014), "Formal Aspects of Enterprise Modeling Methods: A Comparison Framework", *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp. 3400–3409.

Bork, D. and Miron, E.-T. (2017), "OMiLAB - An Open Innovation Community for Modeling Method Engineering", *International Conference of Management and Industrial Engineering (ICMIE) - Management in the Innovation Society*, Niculescu Publishing, Bucharest, Romania, pp. 64–77.

Buchmann, R.A. (2016), "Modeling Product-Service Systems for the Internet of Things: The ComVantage Method", in Karagiannis, D., Mayr, H.C. and Mylopoulos, J. (Eds.), *Domain-Specific Conceptual Modeling*, Springer International Publishing, pp. 417–437.

Buchmann, R.A. and Karagiannis, D. (2016), "Enriching Linked Data with Semantics from Domain-Specific Diagrammatic Models", *Business & Information Systems Engineering*, Springer Fachmedien Wiesbaden, Vol. 58 No. 5, pp. 341–353.

Cabot, J., Clarisó, R. and Riera, D. (2014), "On the verification of UML/OCL class diagrams using constraint programming", *Journal of Systems and Software*, Elsevier, Vol. 93, pp. 1–23.

Cicchetti, A., Ciccozzi, F. and Leveque, T. (2011a), "A hybrid approach for multi-view modeling", *Electronic Communications of the EASST - Recent Advances in Multi-Paradigm Modeling (MPM 2011)*, Vol. 50, available at: http://www.easst.org/eceasst/.

Cicchetti, A., Ciccozzi, F. and Leveque, T. (2011b), "Supporting incremental synchronization in hybrid multi-view modelling", *International Conference on Model Driven Engineering Languages and Systems*, pp. 89–103.

Ehrig, H., Ermel, C., Golas, U. and Hermann, F. (2015), "Model Synchronisation", *Graph and Model Transformation*, Springer, pp. 249–294.

Farias, K., Oliveira, T., Gonçales, L.J. and Bischoff, V. (2017), "Modeling Composition of UML Profiles with Alloy", *International Journal of Computer Science and Software Engineering*, Vol.

6 No. 10, pp. 2409–4285.

Ferstl, O.K. and Sinz, E.J. (2006), “Modeling of Business Systems Using SOM.”, *Handbook on Architectures of Information Systems.*, Vol. I No. 43, pp. 347–367.

Ferstl, O.K. and Sinz, E.J. (2013), *Grundlagen Der Wirtschaftsinformatik*, 7th ed., Oldenbourg, München.

Ferstl, O.K., Sinz, E.J. and Bork, D. (2016), “Tool support for the semantic object model”, in Karagiannis, D., Mayr, H.C. and Mylopoulos, J. (Eds.), *Domain-Specific Conceptual Modeling: Concepts, Methods and Tools*, Springer Berlin Heidelberg, pp. 291–310.

Fill, H.-G. and Burzynski, P. (2009), “Integrating Ontology Models and Conceptual Models using a Meta Modeling Approach”, *11th International Protégé Conference*.

Fill, H.-G., Hickl, S., Karagiannis, D., Oberweis, A. and Schoknecht, A. (2013), “A Formal Specification of the Horus Modeling Language Using FDMM”, *Wrtschaftsinformatik*, pp. 1165–1179.

Fill, H.-G. and Karagiannis, D. (2013), “On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform”, *Enterprise Modelling and Information Systems Architectures*, Vol. 8 No. 1, pp. 4–25.

Frank, U. (2014), “Multi-perspective enterprise modeling: Foundational concepts, prospects and future research challenges”, *Software and Systems Modeling*, Vol. 13 No. 3, pp. 941–962.

Gammaitoni, L., Kelsen, P., by Alfonso Pierantonio, C., Blanchette, J., Bordeleau, F., Kosmatov, N., Taentzer, G., et al. (2017), “F-Alloy: a relational model transformation language based on Alloy”, *Software & Systems Modeling*, p. in print.

Giannoulis, C., Zikra, I., Bergholtz, M., Zdravkovic, J., Stirna, J. and Johannesson, P. (2013), “A comparative analysis of enterprise modeling approaches for modeling business strategy”, *The Practice of Enterprise Modeling: Proceedings of the 6th IFIP WG 8.1 Conference on the Practice of Enterprise Modeling*, pp. 193–204.

Guerra, E., de Lara, J., Orejas, F., Paige, R., Gray Guerra, J.E., de Lara, J. and Orejas, F. (2013), “Inter-modelling with patterns”, *Softw Syst Model*, Vol. 12, pp. 145–174.

Hermann, F., Ehrig, H., Orejas, F., Czarnecki, K., Diskin, Z., Xiong, Y., Gottmann, S., et al. (2015), “Model synchronization based on triple graph grammars: correctness, completeness and invertibility”, *Software & Systems Modeling*, Springer, Vol. 14 No. 1, pp. 241–269.

International Organization for Standardization. (2011), “ISO/IEC/IEEE 42010:2011 - Systems and software engineering -- Architecture description”.

Johannsen, F. and Fill, H.-G. (2016), “Supporting business process improvement through a modeling tool”, *Domain-Specific Conceptual Modeling*, Springer, pp. 217–237.

Karagiannis, D., Buchmann, R. and Bork, D. (2016), “Managing Consistency in Multi-View Enterprise Models: An Approach based on Semantic Queries”, *Twenty-Fourth European Conference on Information Systems (ECIS 2016)*, p. Research paper 53.

Karagiannis, D. and Kühn, H. (2002), “Metamodelling Platforms”, *Third International Conference EC-Web 2002*, Vol. LNCS 2455, p. 182.

Karagiannis, D., Mayr, H.C. and Mylopoulos, J. (2016), *Domain- Specific Conceptual Modeling*, edited by Karagiannis, D., Mayr, H.C. and Mylopoulos, J., Springer Berlin Heidelberg, available at:https://doi.org/10.1007/978-3-319-39417-6.

Kheir, A., Naja, H., Oussalah, M. and Tout, K. (2013), “Overview of an Approach Describing Multi-views / Multi-abstraction Levels Software Architecture”, *Evaluation of Novel Approaches to Software Engineering*, pp. 132–140.

Letsholo, K., Chioasca, E. and Zhao, L. (2012), "An Integration Framework for Multi-Perspective Business Process Modeling", in Moser, L., Parashar, M. and Hung, P. (Eds.), *Proceedings of the 2012 IEEE Ninth International Conference on Services Computing*, IEEE, pp. 33–40.

Lopez-Herrejon, R.E. and Egyed, A. (2011), "C2MV2 : Consistency and Composition For Managing Variability in Multi-View Systems", *2011 15th European Conference on Software Maintenance and Reengineering*, pp. 347–350.

Loucopoulos, P. and Kavakli, E. (2016), "Capability Oriented Enterprise Knowledge Modeling: The CODEK Approach", in Karagiannis, D., Mayr, H.C. and Mylopoulos, J. (Eds.), *Domain-Specific Conceptual Modeling*, Springer International Publishing, Cham, pp. 197–215.

Loucopoulos, P. and Kavakli, V. (1999), "Enterprise Knowledge Management and Conceptual Modelling", in Chen, P.P. (Ed.), *Conceptual Modeling, LNCS 1565*, Springer-Verlage Berlin Heidelberg, pp. 123–143.

Loucopoulos, P., Kavakli, V., Prekas, N., Rolland, C., Grosz, G. and Nurcan, S. (1997), *Using the EKD Approach: The Modelling Component*, Paris, available at: https://hal.archives-ouvertes.fr/hal-00707997.

Mazo, R., Lopez-Herrejon, R.E., Salinesi, C., Diaz, D. and Egyed, A. (2011), "Conformance checking with constraint logic programming: The case of feature models", *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual*, pp. 456–465.

Millan, T., Sabatier, L., Thi, T.L.E., Bazex, P., Percebois, C., Recherche, I. De, Université, D.T., et al. (2009), "An OCL extension for checking and transforming UML models", *Proceedings of the 8th Int. Conference on Software Engineering, Parallel and Distributed Systems Engineering*, Cambridge, United Kingdom, pp. 144–149.

Object Management Group (OMG). (2014), "Object Constraint Language", *Object Constraint Language - Version 2.4*, available at:https://doi.org/10.1167/7.9.852.

Object Management Group (OMG). (2016), "MOF Query/View/Transformation Specification Version 1.3", available at: http://www.omg.org/spec/QVT/ (accessed 15 November 2017).

Peffers, K., Rothenberger, M., Tuunanen, T. and Vaezi, R. (2012), "Design Science Research Evaluation", in Peffers, K., Rothenberger, M. and Kuechler, B. (Eds.), *DESRIST*, Springer-Verlag Berlin Heidelberg, pp. 398–410.

Persson, M., Törngren, M., Qamar, A., Westman, J., Biehl, M. and Tripakis, S. (2013), "A Characterization of Integrated Multi-View Modeling in the Context of Embedded and Cyber-Physical Systems", in Ernst, R. (Ed.), *Proceedings of the Eleventh ACM International Conference on Embedded Software*, IEEE Press, Montreal, Quebec, Canada, p. 10:1--10:10.

Poskitt, C., Dodds, M., Paige, R.F. and Rensink, A. (2014), "Towards rigorously faking bidirectional model transformations", *AMT 2014-Analysis of Model Transformations Workshop Proceedings*, pp. 70–75.

Sandkuhl, K., Fill, H.G., Hoppenbrouwers, S., Krogstie, J., Leue, A., Matthes, F., Opdahl, A.L., et al. (2016), "Enterprise modelling for the masses – From elitist discipline to common practice", *Lecture Notes in Business Information Processing*, Vol. 267, pp. 225–240.

Scheer, A. and Schneider, K. (2006), "ARIS – Architecture of Integrated Information Systems", in Bernus, P., Mertins, K. and Schmidt, G. (Eds.), *Handbook on Architectures of Information Systems*, Springer Berlin Heidelberg, pp. 605–623.

Sousa, P., Pereira, C., Vendeirinho, R., Caetano, A. and 12, J. (2007), "APPLYING THE ZACHMAN FRAMEWORK DIMENSIONS TO SUPPORT BUSINESS PROCESS MODELING", *Digital Enterprise Technology*, pp. 359–366.

Vierhauser, M., Grünbacher, P., Heider, W., Holl, G. and Lettner, D. (2012), "Applying a consistency checking framework for heterogeneous models and artifacts in industrial product lines", *International Conference on Model Driven Engineering Languages and Systems*, pp. 531–545.

Visic, N., Fill, H.-G., Buchmann, R.A. and Karagiannis, D. (2015), "A domain-specific language for modeling method definition: From requirements to grammar", *Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference On,* pp. 286–297.

Zdravkovic, J., Stirna, J., Kirikova, M., Karagiannis, D. and Winter, R. (2015), "Advanced Enterprise Modeling", *Business & Information Systems Engineering*, Vol. 57 No. 1, pp. 1–2.