Modeling Context for Business Rule Management

Felix Burgstaller, Dieter Steiner, and Michael Schrefl
Department of Business Informatics - Data & Knowledge Engineering
Johannes Kepler University Linz, Austria
Email: {felix.burgstaller, dieter.steiner, michael.schrefl}@jku.at

***Abstract*—Many companies are confronted with an increasingly large number of business rules. Consequently, means for efficient and effective management of these business rules are necessary. However, Business Rule Management Systems (BRMS) frequently provide only simple business rule organization techniques such as collecting business rules into rule sets according to a single criterion. Many fields organize information by contexts that may be hierarchically structured. Similarly, we propose the use of contexts to manage business rules and associated business vocabulary.**

We survey context literature and identify dimensions and possible choices for each dimension to compare and relate different context models. For each dimension we propose the appropriate choice for a context model of business rules and business vocabulary. Based on these choices we present a generic context model for business rules and business vocabulary that can be instantiated multi-level: for a specific domain and for a concrete application. We demonstrate the usefulness of the generic context model for business rules and vocabulary by relating it to a use case in the aeronautical domain, the semantic filtering of digital Notices to Airmen. This real world use case addressed in the SemNOTAM project involves thousands of business rules and extensive business vocabularies. These need to be organized along multiple hierarchical context parameters (e.g. aircraft type, mode of operation) such that they become manageable with regard to definition, maintenance, extension, and determining the business rules and business vocabulary relevant for a specific business case (e.g. flight plan of a pilot, responsibility area of a controller).

I. Introduction

A business rule is a "rule that is under business jurisdiction and derived from business policy" [1, p. 98]. Today's companies are confronted with thousands of such rules and their count is rapidly increasing [2]. Similarly, the business vocabulary, i.e., the business terms and facts, used to define business rules become more and more extensive. Consequently, means for efficient and effective management are necessary. However, BRMS oftentimes provide only simple business rule organization techniques such as partitioning business rules regarding a single criterion. The resulting partitions then represent rule sets. More complex organization techniques considering organizational structures, in particular hierarchical structures, are mostly not supported.

In many fields, e.g., artificial intelligence, ubiquitous computing, or ontologies, contexts described by context models are used to organize knowledge. Likewise, we propose the use of contexts to organize business rules and business vocabularies (BRV) where each context defines the situation in which the contained business rule set and business vocabulary are relevant. Consequently, the readability of BRV increases as the situation does not need to be encoded in their definitions. Furthermore, such a context-based organization of BRV promises several advantages such as hierarchies of business rule sets and business vocabularies (BRSV), automatic determination of relevant BRSV for a business case, increased rule evaluation performance, and easier debugging. These benefits support efficient and effective management of BRV. To develop the presented generic context model for organizing BRV providing these advantages, we employ the information systems (IS) research framework, in particular the design science paradigm, described in [3].

To demonstrate the usefulness of the presented generic context model, it is applied in the research project Semantic Notices to Airmen (SemNOTAM) [4]. A Notice to Airmen (NOTAM) is a safety- and time-critical announcement of alterations to aeronautical conditions which might be of relevance to flight operations personnel. Actually, many of the published NOTAMs are irrelevant to specific flight operations, e.g., a helicopter pilot is not interested in closed gates. Furthermore, the NOTAMs relevant to a specific flight operation are of different importance, e.g., an aerodrome closure is critical whereas a taxiway closure is less important. However, current NOTAM systems perform unsatisfactory in this regard [5] and thus incur information overload.

The objective of SemNOTAM is to overcome the identified lacks by providing fine-grained semantic filtering, importance classification, and grouping of NOTAMs for various business cases, e.g., pre-flight briefing for a pilot or briefing for an air traffic controller. Therefore, a user provides an abstract query, called interest specification. An interest specification comprises basic interests combined by conjunctions and disjunctions. Each basic interest either specifies a time, area, or aircraft the user is interested in. Such an interest specification is then evaluated using business rules. These business rules define, using aeronautical vocabulary, how to determine the relevance of NOTAMs, how to classify the NOTAMs regarding importance, and how to assign NOTAMs to groups. Workshops with pilots revealed that some of these business rules apply across all business cases whereas others apply only for a subset of business cases. Further complicating the tasks are: a large number of rules resulting from the number of business cases, the requirement that relevant NOTAMs must not be filtered, the complexity of the aeronautical domain, as well as the computational complexity due to increasing numbers of NOTAMs, business rules, and business vocabularies [6].

The remainder of this paper is organized as follows: Section II reviews the literature on business rules and contexts and presents a table of dimensions and options for context models. Section III utilizes this table and argues the choices made for a context model organizing BRV. Section IV presents a generic context model implementing the choices and demonstrates its usefulness by instantiating it for SemNOTAM. Section V discusses related work and Section VI concludes the paper.

II. Literature Review

To ascertain a context model for management of business rules, we review the literature regarding business rules and contexts. From this literature review we determine dimensions and options relevant to context models (c.f. TABLE II) by performing qualitative document analysis with inductive category formation as described in [7]. Regarding the IS research framework [3], this review and analysis serves to get a sense of the relevant knowledge base. The identified dimensions and options can be used to decide the relevant context model options for any application scenario.

A. Business Rules

Business rules are relevant to business cases. According to the Business Rules Group (BRG) business rules can assume two perspectives - on the one hand the business and on the other hand the IT perspective [8]. These two perspectives are in fact partly overlapping. The recent standard on Semantics of Business Vocabulary and Business Rules (SBVR) 1.3 by the Object Management Group (OMG) relinquishes this distinction and defines a business rule as a "rule that is under business jurisdiction and derived from business policy" [1].

Orthogonal to the separation into business and IT perspective, business rules are distinguished into different classes as summarized in TABLE I. The BRG [8] names structural assertions (here named business vocabulary) as one business rule class. Contrary, Ross [9] and the Business Rule Community [2] do not consider business vocabulary as a business rule class but nevertheless as substantial to business rules. A business vocabulary entry is either a business term or a fact. A business term is a phrase with specific meaning for a business in a certain context which needs to be explicitly and unambiguously defined, e.g., in a glossary. A fact expresses a relationship between terms, e.g., a customer can buy a car. Such a fact can either be a base fact, i.e., a simple statement, or a derived fact, i.e., derived from other terms and facts. Furthermore, facts can be discriminated into attributes, designating a term as attribute of another term; generalizations, defining a term as super-type of another term; or participation, any other relationship between terms.

These terms and facts are used to define actionable rules. We identify three basic classes of actionable rules: information constraints, derivations, and action enablers. Information constraints subsume constraints [2], [8]–[10] and guidelines [10], i.e., warnings or recommendations. The class derivations is distinguished into computations and inferences [10]. A similar distinction is proposed by Ross [9]. Action enablers define event-condition-action rules, i.e., the actions to execute when events occur [2], [9], [10].

TABLE I
Business rule classifications

Business Vocabulary	**Actionable Rules**
Business terms	Information constraints
Types	Constraints
Literals	Guidelines
Business facts	Derivations
Derivation-based classification	Computations
Base facts	Inferences
Derived facts	Action enablers
Role-based classification	
Attribute	
Generalization	
Participation	

Regardless the classification of a business rule the BRG defines ten "Principles of Rule Independence" in its Business Rules Manifesto [11] such as business rules are to be primary requirements; applied across processes and procedures; deliberate explicated knowledge; declarative, thus distinct from its enforcement; well-formed; and for the sake of business, not technology. Furthermore, a rule-based architecture able to explain the reasoning should be provided, as well as a rule-guided process adopted rather than an exception-based one.

B. Contexts

According to Dey and Abowd context is "any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [12, pp. 3-4], where a user is not necessarily a person. A characterization specific to communication is context as a shared space of knowledge containing the current state of knowledge of the participating agents [13], thus context is inseparable from its use [14]. Brézillon's summary on several workshop discussions in Artificial Intelligence [14] describes context as guiding the focus of attention as well as permitting local reasoning with potentially different languages and forms of reasoning. From an ontology and database viewpoint contexts are means to define knowledge to be considered, limitations and validity of this knowledge, and when to employ the knowledge [13]. Giunchiglia and Bouquet [15] describe context as a box containing linguistic expressions with a collection of parameters describing the box (box metaphor). A concrete context is then identified by values for each of the parameters. Thus, contextualized knowledge, i.e., the knowledge in the box, is separated from contextual knowledge, i.e., the knowledge about contexts. This does not restrict contextualized knowledge from being contextual knowledge for another problem [14].

So, what exactly is a context? Besides the interpretations and definitions of context stated here many others exist in

TABLE II
DIMENSIONS RELEVANT TO CONTEXT MODELS

Dimension	Options
Interpretation of Context	
Nature	Representational / Interactional
Objective of context	Viewpoints / separate concerns
Context Model	
Contexts	
Set of contexts	Dynamic / static
Reasoning	Uniform / individual
Context hierarchies	Yes / no
Context relationships	Independent / related
Effect of context rel.	Contexts / param. / param. values / none
Context Parameters	
Parameters of context	Uniform / individual
Set of parameters	Dynamic / static
Set cardinality	Single / multiple
Parameter hierarchies	Yes / no
Parameter relationships	Independent / related
Effect of param. rel.	Contexts / param. / param. values / none
Parameter values	
Set of param. values	Dynamic / static
Value hierarchies	Yes / no
Value relationships	Independent / related
Effect of value rel.	Contexts / param. / param. values / none
Model Support	
Purpose	Presentation / automatic exec. / tagging
Qualities	
Adaptability[a]	Yes / no
Dynamism	Contexts / param. / param. values / none
Functionalities	
Effects of relationships	Dynamic / materialized / no
Context versioning	Yes / no
Decontextualization[b]	Yes / no
Referencing[b]	Yes / no
Shifting[b]	Yes / no
Sources	
Uniformity	Homogeneous / heterogeneous
Quality	Assured / uncertain
Histories	Yes / no

[a] regarding application domain [b] mechanisms on contexts

various domains. Although Bazire and Brézillon [16] analyzed 150 definitions across several domains and as a result hypothesize that the reason for the diversity in definitions of context is the focus on different topics, no consensus has been found yet.

Consequently, rather than giving a definition, we review context interpretations and context descriptions as well as purposes, qualities, functionalities, and sources supported by context models. For each of these aspects we identify relevant dimensions and their options, as summarized in TABLE II. These dimensions are not independent of each other, e.g., *nature* of context influences *set of parameters*. To develop a domain specific context model the appropriate options need to be chosen for each of the dimensions. Depending on the domain and on the dimension, several or just one option are relevant to a domain specific context model. In the following we delineate the identified dimensions.

For the aspect *interpretation of context* we identified two dimensions. One dimension is the *nature* of context [13], [14], [16], [17] which can be representational (context as information) or interactional (context as a process). These two

TABLE III
INTERACTIONAL AND REPRESENTATIONAL CONTEXT [17]

Representational context	Interactional context
Context as information	Context as process
Context is discrete	Context is continuous; particular to each activity occurrence
Context is static/stable	Context is dynamic; unknown beforehand
Context and activity are separable	Context arises from activity; it is actively produced, maintained and enacted

natures are attributed to the fact that context is interpreted differently depending on whether an engineering (representational context) or a cognitive science (interactional context) viewpoint is taken [13], [14]. TABLE III summarizes the differences between representational and interactional context. Both types are of importance and have to be accounted for as they are different aspects of context [14], [16]. For instance, a conversation takes place in a certain representational context (place, persons, etc.); the interactional context then describes the conversation itself, thus the changing shared knowledge throughout the conversation.

The other dimension regarding *interpretation of context* is the *objective of context*. Contexts can be used as different viewpoints of one and the same object, i.e., each context describes its view of the element. To get an idea of the element the different contexts need to be integrated. An example are orthographic projections which describe different views of an object, e.g., side, front, or back. To conceive the object in three-dimensional space the different views need to be integrated. Contexts describing separate concerns on the other hand consider different objects.

Regarding the aspect *context model* we consider the three dimension groups *contexts*, *context parameters*, and *parameter values* relevant. Regarding contexts, the *set of contexts* can either be static or dynamic. A set is considered dynamic here if elements are often added, removed, or modified; otherwise it is considered static. Regarding *reasoning*, either the same reasoning (uniform) is applied to all contexts or context-dependent reasoning (individual) [14] is applied. Further choices regard whether *context hierarchies* and, more general, *context relationships* are supported [14], [18]–[20]. For these relationships it has to be determined, regarding *effect of context relationships*, which, if any, of the three dimension groups they effect. For instance, hierarchical context relationships could imply inheritance of knowledge and thus effect contexts. Another dimension group of the aspect *context model* are *context parameters*. Context parameters influence the contextualized knowledge; they provide unarticulated information about it. Brézillon [14] describes this as implicit context as this information is not explicated within the context. Explicit context on the other hand describes knowledge within the context. For instance, in a context with country as parameter and France as value it is clear that Paris describes the capital. Not knowing about the parameter value France, Paris could also be one of the many places in the US. Dimensions in *context parameters* with similar names as dimensions in *contexts* have the same meaning and thus only different dimensions are discussed. Depending on the domain, either the same *parameters of context* are used for all contexts (uniform) or each context is described by potentially different parameters (individual) [20]. Many authors, proposing different theories, argue for virtually infinite parameters different from context to context [20]. Additionally, a choice regarding the *set cardinality* of context parameters is necessary. This set can either contain only one context parameter, representing a simple context model, or multiple context parameters. The last dimension group of *context model* is *parameter values*. For this dimension group the same choices as regarding the former two are necessary. An example for the *effect of value relationships* is presented in Homola et al. [19] where the hierarchy of contexts is derived from the hierarchies of parameter values.

The remaining aspect to be discussed is *model support*. The options of dimension *purpose* are derived from the features of context-aware applications by Dey and Abowd [12]: presentation, automatic execution, and tagging. Presentation is the context-dependent presentation of information and services, e.g., depending on the smartphone location, search requests retrieve different results. Automatic execution describes the automatic execution or adaptation of a service depending on the context. Finally, tagging delineates the association of data with context. The dimension group *qualities* contains qualities important to context models. This comprises the *adaptability* to different domains and *dynamism*. *Dynamism* allows to choose whether set of contexts, context parameters, or parameter values are modeled dynamic or not. The dimension group *functionalities* comprises several dimensions. While the effects of relationships to be modeled are chosen in *context model*, concerning dimension group functionalities it is determined whether and how these *effects of relationships* are supported. Furthermore, a choice regarding *context versioning* is necessary. The other dimensions regarding functionalities are mechanisms on contexts. *Decontextualization* [18] describes how sentences of contextualized knowledge can be moved to a target context while representing the same statement as in the original context. *Referencing* [19] allows to refer to contextualized knowledge in another context, i.e., a sentence in context X can refer to a sentence in context Y. *Shifting* [20] describes, using the box metaphor, a change of parameter values and consequently a shift in the interpretation of contextualized knowledge. This is only applicable if the *objective of context* is viewpoints. These and other mechanisms for relating and translating between contexts are deemed critical to the reuse of contexts [13]. The dimension group *sources* concerns the sources which are used to determine the relevant context and thus knowledge. Regarding *Uniformity* it needs to be judged whether the sources are homogeneous or heterogeneous [21]. Moreover, it needs to be determined whether the *quality* is assured or whether quality has to be incorporated into the context model [21], [22]. Another choice concerns whether *histories* of sources [21] should be kept which would allow processing similar to event consumption policies in active databases [23].

TABLE IV
CONTEXT MODEL CHOICES FOR BRV

Decision	**Chosen Options**
Interpretation of Context	
Nature	Representational
Objective of context	Separate concerns
Context Model	
Contexts	
Set of contexts	Dynamic
Reasoning	Uniform & individual
Context hierarchies	Yes
Context relationships	(Related)
Effect of context rel.	Contexts
Context Parameters	
Parameters of context	Uniform
Set of parameters	Dynamic
Set cardinality	Multiple
Parameter hierarchies	No
Parameter relationships	Independent
Effect of param. rel.	None
Parameter values	
Set of param. values	Dynamic
Value hierarchies	Yes
Value relationships	(Related)
Effect of value rel.	Contexts & parameter values
Model Support	
Purpose	Presentation & tagging
Qualities	
Adaptability	Yes
Dynamism	Contexts & param. & param. values
Functionalities	
Effects of relationships	Dynamic / materialized
Context versioning	Yes
Decontextualization	Yes
Referencing	No
Shifting	No
Sources	
Uniformity	Heterogeneous
Quality	Assured
Histories	(Yes)

III. Context Model Requirements for Business Rules and Business Vocabulary

Before we can determine a generic context model for organizing BRV we need to clarify the requirements regarding such a model. In general, all types of business rules should be supported. To determine specific requirements, we choose for every identified dimension in TABLE II the options relevant to a context model organizing BRV. These choices are summarized in TABLE IV and delineated below.

Preferably, the organization into sets is rather stable as it will otherwise be difficult to keep an overview. Furthermore, all contexts and their business rules and vocabulary are known and can be separated from the business rule evaluation. Consequently, the appropriate *nature* of context is representational context. Interactional context could be used to model forces and knowledge influencing a business and thus its business rules. Since the aim of the context model is to organize BRV the *objective of context* is separate concerns.

As businesses need to constantly adapt to changes in markets, new challenges, governance, etc. so need their business rules, business rule sets, and business vocabulary. Thus, the addition and removal of *contexts*, containing business rule sets and business vocabulary, is likely and hence the *set of contexts* is dynamic. As businesses usually employ only one business rule engine the *reasoning* is mostly uniform unless the engine supports multiple forms of reasoning. Additionally, individual reasoning per context complicates the accountability of results. Following from the hierarchical structure of most businesses *context hierarchies* are of interest. Other kinds of *context relationships* might be relevant depending on the application domain. For each kind of relationship its effects need to be considered, e.g., regarding hierarchical context relationships whether a sub-context of another context inherits its knowledge. Regarding *context parameters*, the *parameters of context*, describing contexts, are the same for all contexts to keep the organization of BRV comprehensible. Due to the dynamic nature of businesses it might become necessary to modify the *set of parameters* for all contexts over time, thus this set is dynamic. Single parameters of contexts are sufficiently covered by conventional BRMS thus we focus on multiple parameters. These parameters are rarely related and consequently effect of parameter relationships are not relevant either. Analogously to the parameter set, the *set of parameter values* is dynamic. Most parameter values form *value hierarchies*, e.g., organizational units in a company. Other *value relationships* depend on the application domain. As relationships between parameter values exist their effects need to be modeled. In most cases these effect contexts and parameter values, e.g., as in Homola et al. [19].

The remaining aspect to be discussed is the *model support* to be provided by a context model for BRV organization. Such a model needs to support management and determination of relevant BRV for the current business case at hand. Consequently, the *purposes* presentation and tagging are relevant. Presentation is important as for a given business case the relevant contexts and thus BRV need to be determined. Tagging is vital as BRV need to be associated with the context in which they have to be considered. Automatic execution of business rules may be delivered by the model as well but is not required. Regarding dimension group *qualities*, *adaptability* is to be provided as the generic context model needs to be adapted to specific domains, e.g., SemNOTAM. Concerning *dynamism* all three main dimension groups of *context model* are to be supported. Regarding *functionalities*, it depends on the application domain whether *effects of relationships* need to be dynamically computed or materialization is appropriate. Furthermore, *context versioning* is necessary to ensure accountability for older results. Of the described mechanisms *decontextualization* is relevant for management of BRV. As soon as the contextual knowledge of a context changes and the business rules should still have the same meaning as before, decontextualization is necessary. The same is vital if a business rule is moved to another context without modifying its meaning. *Referencing* a business rule, a term, or a fact in another context might be useful if they are of relevance to the current context. Nevertheless, for such cases other means for modeling, e.g., inheritance, are available. *Shifting* is not relevant as the *objective of context* is separate concerns. The remaining dimension group to discuss is *sources*. In today's business more and more efforts are made to provide homogeneous and integrated data, e.g., data warehouses, but still much data is heterogeneous. Modeling *quality* of data sources is considered valuable but not vital. Consequently, we focus on core requirements in this paper and consider quality as future work. Source *histories* are important if processing methods similar to consumption policies in active databases [23] are needed in the application domain.

TABLE V
THE THREE LEVELS OF THE CONTEXT MODEL FOR BRV ORGANIZATION

Lvl	Model
M2	Generic context model for business rules and vocabulary
M1	Domain specific context model (e.g. SemNOTAM)
M0	Application specific contexts & business cases

IV. Context Model for Business Rules and Business Vocabulary Organization

In this section we present a conceptual, i.e., implementation independent, context model for BRV organization based on the choices made in the previous section. This conceptual model can be implemented in various ways, e.g., using knowledge-based or expert system techniques where the proposed model describes the knowledge base and the reasoning engine is used for context determination and business rule evaluation.

The context *nature* being representational suggests modeling contexts as objects with attributes. A widely used modeling language for object-oriented and conceptual modeling is the Unified Modeling Language (UML), in particular, its class diagram. Consequently, a UML class diagram is also suitable to model contexts for BRV organization. Using standard class diagrams to conceptually model contexts, limitations regarding the adaptability of the resulting context model arise. To overcome these limitations, non-strict multi-level modeling adapting De Lara [24] is employed resulting, after several iterations of the assess-refine cycle described in [3], in the three levels depicted in TABLE V.

Level M2 defines a generic context model for BRV organization independent of any application domain; it is a meta context model for BRV organization. As such it describes the realization of all choices made, in particular the realization of the purposes presentation and tagging. Level M1 instantiates the generic model regarding an application domain and thus constrains and refines the generic context model, e.g., it defines the parameters describing a context, restricts available relationships, and defines the business cases relevant in the domain. Additionally, the effects of relationships as well as the semantics of dynamism, e.g., the consequences of removing a parameter, need to be specified. Level M0 instantiates the domain specific context model and thus contains concrete context instances and their BRV as well as concrete business case instances. The purposes described at M2, presentation and tagging, are realized at this level.

The generic context model for BRV organization and its instantiations for SemNOTAM are depicted in Fig. 1. This model draws on the relation configurator pattern and the modified UML notation described in De Lara et al. [24]. Additionally, potencies for methods are used. For instance, detRelevantCtxs^1(BusinessCase):CtxMetaModel describes a method whose body is defined on the next instantiation level and whose parameter and return type are instances of BusinessCase and CtxMetaModel respectively.

A. M2 - Generic Context Model

The generic context model is designed to support the choices in TABLE IV. The choices regarding the aspect interpretation of context are modeled by considering context an object identified by a tuple of parameter values describing relevant BRV. The parameter values specify when the context is relevant, i.e., the conditions under which the described BRV are relevant. Thus, the parameter values also determine the business cases for which a context is relevant.

These objects, contexts, parameters, parameter values, and business cases, are modeled by the classes CtxMetaModel, Parameter, and BusinessCase in the generic model; parameter values instantiate domain specific parameter classes at M0 and thus are not represented explicitly. The set of contexts, parameters, and parameter values can easily be modified by utilizing the object-oriented multi-level approach to create or deconstruct instances of the corresponding classes. This approach furthermore supports *adaptability* and *dynamism*.

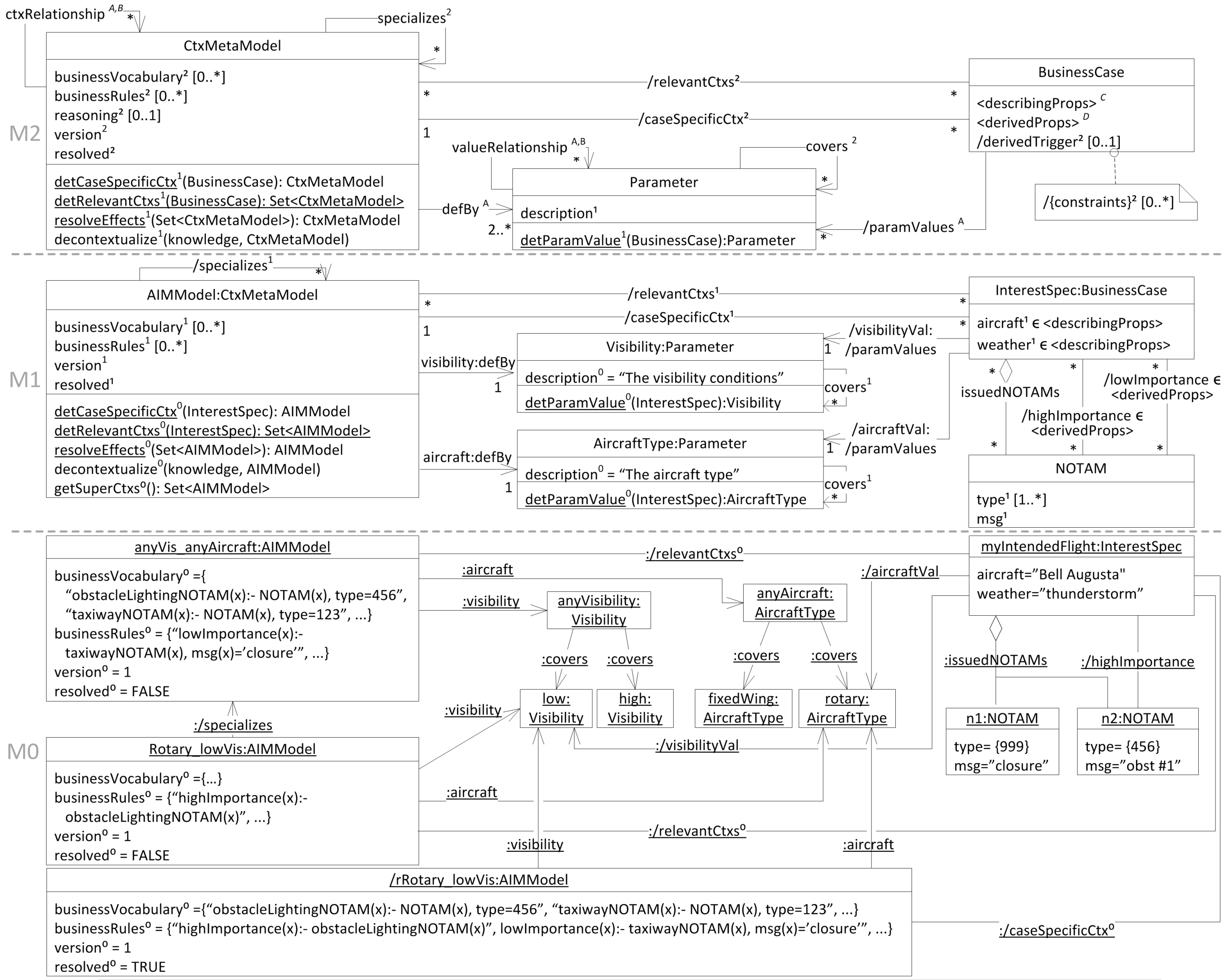
Each context, modeled by CtxMetaModel, specifies parameter values for its parameters modeled by the relationship defBy. In accordance with the *context model* choices, any domain specific context model has to define at least two parameters. The attribute version supports the functionality versioning. The attribute resolved is true if it is a context for which all effects of relationships have been resolved; otherwise it is false. These properties, defBy, version, and resolved, uniquely identify a context. Each context further specifies business rules (businessRules) and business vocabulary (businessVocabulary) relevant to it. The kind of reasoning to be used when evaluating the specified BRV can be defined using the attribute reasoning. Besides reasoning, relationships between contexts, in particular hierarchical context relationships, can exist. These are modeled by the relationships ctxRelationship and specializes respectively. Since specializes is considered an important relationship regarding BRV it is modeled explicitly in the generic context model. Similarly the hierarchical relationship of parameter values, covers, is modeled explicitly. Other parameter value relationships configure valueRelationship at M1. The effects of the relationships are defined in resolveEffects at M1.

Each business case (BusinessCase) specifies properties describing it, as indicated by the placeholder <describingProps>. These properties are used to derive the parameter values of a business case as depicted by the relationship /paramValues between BusinessCase and Parameter. Based on /paramValues the relevant contexts are determined for the business case denoted by the derived relationship /relevantCtxs between BusinessCase and CtxMetaModel. From these relevant contexts the case specific context is derived by merging all relevant contexts and resolving all emerging conflicts. Such conflicts arise for example when a business term is defined differently in two relevant contexts. The resolved context is referred to by /caseSpecificCtx. This derived relationship fulfills the purpose of presentation, i.e., BRV relevant to a business case are determined. Further information regarding the business case is derived from concrete business rules. Depending on the actionable rule class, different kinds of information are derived: the class constraints derive Object Constraint Language (OCL) constraints on the model, guidelines are realized using derived properties (<derivedProps>), derivations derive properties (<derivedProps>), and action enablers derive triggers (/derivedTrigger). The supported business rule classes can be constrained at M1.

The described derived relationships are determined by methods. Assuming heterogeneous sources of assured quality the parameter values of each parameter are determined differently. This is modeled by the class method detParamValue of Parameter determining for a given business case its parameter value regarding the parameter class it is defined for. The class method detRelevantCtxs of CtxMetaModel uses /paramValues to determine the set of contexts relevant to a given business case. detCaseSpecificCtx computes, for a given business case, a single context into which all relevant contexts have been merged and any conflicts resolved. The returned context has the same parameter value set as determined for the business case but specifies itself as resolved (resolved=TRUE). The merging and conflict resolution is defined in the class method resolveEffects. This method resolves, for a given set of contexts, all effects of relationships as well as any arising conflicts. resolveEffects returns a new context containing the merged and conflict resolved BRV. The functionality decontextualization is provided by the method decontextualize. The parameter knowledge identifies the business term, fact, or rule to be decontextualized, whereas the parameter ContextModel states the target context. Tagging is supported by methods like addBusinessTerm(Term) or getters and setters for the attributes businessVocabulary and businessRules (not depicted in Fig. 1).

B. M1 - Domain Specific Context Model

The domain specific context model constrains and refines the generic context model by instantiating it regarding an application domain. With respect to contexts it has to be decided whether different reasoning techniques per context are supported. Moreover, any relationships other than specializes need to be defined using relation configuration on ctxRelationship. The parameters are refined by instantiating domain specific parameters from Parameter and



[A] Need to be configured on the next level using the relation configurator pattern. [B] Need to be reflexive on the next level, e.g., same:ctxRelationship needs to be a reflexive relationship of AIMModel. [C] Placeholder for properties describing the domain-specific business case defined on the next level denoted by ϵ. [D] Placeholder for properties derived from business rules defined on the next level denoted by ϵ.

Fig. 1. The generic context model for BRV organization and its instantiations for SemNOTAM.

configuring `defBy` appropriately, e.g., multiplicities of one ensure that each context is described by the same parameters. Any additional relationships between parameter values besides `covers`, can be defined by configuring relationship `valueRelationship`. Besides context and parameter refinements, the methods' signatures need to be refined and their logic defined.

Similarly to `CtxMetaModel` and `Parameter`, the business case needs to be refined. On the one hand this comprises definition of business case properties describing the domain specific business case. On the other hand, a decision regarding supported actionable rule classes is necessary, e.g., if action enablers are inessential then the attribute `/derivedTrigger` is not relevant to the domain specific context model.

The presented generic model supports dynamism but the semantics of dynamism need to be specified when instantiating a domain specific context model. In particular the semantics of adding, deleting, and modifying contexts, parameters, and parameter values need to be defined. This is a laborious task but once accomplished no additional effort is necessary for dynamism.

In addition to the semantics of dynamism, the effects of relationships need to be specified, i.e., the logic of `resolveEffects` defined. In particular, the semantics of hierarchical relationships of contexts and parameter values are of interest (c.f. TABLE IV). One effect of hierarchies often implemented in programming languages is inheritance. The question is: What does inheritance mean with respect to business rules and vocabulary in the application domain? Two common options are the mutually exclusive additive and most-specific inheritance. Using the former option, all BRV of the context hierarchy relevant to a concrete business case are relevant. With the latter option business rules and vocabulary can be refined, i.e., only the most specific business rules, terms, and facts of the context hierarchy are relevant. A problem incurred by inheritance mechanisms are conflicts from multi-inheritance. When business rules or vocabulary with the same identifier but different implementations are inherited it needs to be decided which one to use. Therefore, conflict resolution strategies have to be defined.

Fig. 1 depicts a detail of the SemNOTAM specific context model. In the following, the previously described process of instantiation is applied for SemNOTAM. The SemNOTAM specific context model is named Aeronautical Information Management Model (`AIMModel`). In SemNOTAM the same reasoning is used in every context, thus the attribute `reasoning` is removed. Furthermore, only hierarchical relationships of contexts and parameter values are of interest, hence no other relationships are configured at M1. The depicted detail shows two parameters, namely `Visibility` and `AircraftType`, of seven identified: airline, meteorological condition (e.g. visibility), flight rule, aviation type, user role, flight phase, and interest (e.g. aircraft or space). These are mandatory for every context instantiating the SemNOTAM specific model and were determined by applying qualitative document analysis and grounded theory (as described in [7]) to transcripts of three problem-centered expert interviews as well as workshop documents and documents from aeronautical authorities. The interviews furthermore revealed that the set of identified parameters is stable. Consequently, no semantics for modifying the set of parameters need to be defined. The method signatures are refined to the domain specific classes. Besides refining method signatures, the methods' logic needs to be defined as is depicted in TABLE VI for several methods.

The business case specific to SemNOTAM is the interest specification. The `InterestSpec` defines the two attributes `aircraft` and `weather` for the placeholder `<describingProps>`. From the values of these attributes the parameter values for the parameters `Visibility` and `AircraftType` are derived using `detParamValue` of the corresponding parameter class. Furthermore, the issued NOTAMs to which the business rules are to be applied are specified. The relationships `/highImportance` and `/lowImportance` are derived properties (from business rules) specifying whether a NOTAM is of high, low, or no importance.

The generic context model supports dynamism whereas SemNOTAM, as a safety-critical application, constrains the supported level of dynamism to addition of contexts and parameter values. Deletion of either would complicate provisioning of accountability for query results. The semantics and effects of hierarchical relationships are similar to Homola et al. [19] where the hierarchy of parameter values determines the hierarchy of contexts. A context specializes another context if it has a more specific, or at least the same, parameter value for all parameters. Thus, `specializes` at M1 is prefixed with a "/" to indicate its derivation from `getSuperCtxs`. The effect of a context specializing another one is the inheritance of BRV. In SemNOTAM additive inheritance semantics are applied.

TABLE VI
PSEUDO-CODE OF THE AIMMODEL METHODS AT M1

detCaseSpecificCtx(InterestSpec i) : AIMModel
```
AIMModel ctx = AIMModel.resolveEffects( i.relevantCtxs )
ctx.setParamValues( i.getParamValues() )
RETURN ctx
```
detRelevantCtxs(InterestSpec i) : Set⟨AIMModel⟩
```
Set⟨AIMModel⟩ specifiedCtxs , Set⟨AIMModel⟩ relevantCtx
specifiedCtxs = lookupCtxs[a]( i.getParamValues() )
FOREACH c IN specifiedCtxs
   relevantCtx = relevantCtx ∪ c ∪ c.specializes
RETURN relevantCtx
```
resolveEffects(Set⟨AIMModel⟩ s) : AIMModel
```
Set⟨Vocabulary⟩ vocabulary, Set⟨Rule⟩ rules
AIMModel specificCtx = detSpecificCtx[b]( s )
FOREACH c IN s
   rules = rules ∪ c.businessRules
vocabulary = resolveConflicts[c]( s )
RETURN new AIMModel( specificCtx.getParamValues(),
                     specificCtx.version, TRUE, rules, vocabulary )
```
getSuperCtxs() : Set⟨AIMModel⟩ *//covers is assumed transitive*
```
Set⟨AIMModel⟩ super, Bool f
FOREACH ctx IN instancesOf(AIMModel)
   f = TRUE
   FOREACH p IN instancesOf(Parameter)
      IF !getParamValue( ctx, p ).covers( getParamValue( this, p ) )
         THEN f = FALSE
   IF f THEN super = super ∪ ctx
RETURN removeObsoleteCtxs[d]( super )
```

[a] Returns the highest version contexts for the given parameter values.
[b] Returns the or one of the most specific contexts in the given set.
[c] Contains the logic to resolve multi-inheritance conflicts.
[d] Keeps only the most recent context versions in the given set.

The only exception are defaults whose value can be refined in sub-contexts, e.g., a default spatial buffer of 70 nm at the uppermost context can be refined to 50 nm at a lower context. Since such defaults are supported `resolveEffects` needs to define conflict resolution strategies for multi-inheritance issues. In case of spatial buffers the resolution strategy is to use the largest buffer. Parts of the described semantics are implemented in the methods depicted in TABLE VI.

C. M0 - Application Specific Contexts & Business Cases

Level M0 instantiates the domain specific context model and consequently contains concrete context and business case instances as well as any information which is derived by business rules. Therefore, the contexts and their relationships, the parameter values and their relationships, and the BRV need to be elicited from business personnel and stakeholders. Consequently, this level realizes tagging, i.e., BRV are assigned to concrete contexts. Furthermore, presentation is realized by determining the contexts relevant to a concrete business case (`/relevantCtxs`). The application of the relevant BRV to the concrete business case corresponds to the purpose automatic execution.

M0 in Fig. 1 depicts two concrete contexts with their BRV. The context `anyVis_anyAircraft` contains business rules and vocabulary relevant for any visibility and any aircraft. In this context the business rule `lowImportance` is specified which defines that NOTAMs communicating a taxiway closure are classified *low importance*. This is the case as aerodromes have other taxiways which can be used instead. `lowImportance` uses the term `taxiwayNOTAM` which is defined in the business vocabulary. Furthermore, the term `obstacleLightingNOTAM` is defined. These terms are defined in the form of predicates testing whether the term applies to a given NOTAM.

The context `Rotary_lowVis` specifies BRV relevant for low visibility and rotary aircrafts (e.g. helicopters) only. The depicted detail shows the business rule `highImportance` classifying any NOTAMs regarding obstacle lighting *high importance* for this context. As both parameter values, `rotary` and `low` are covered by `anyAircraft` and `anyVisibility` respectively, `Rotary_lowVis` specializes `anyVis_anyAircraft`.

The interest specification `myIntendedFlight` describes a flight with a Bell Augusta, a helicopter, during thunderstorms, implying low visibility. The issued NOTAMs to consider are `n1`, a waterlane closure NOTAM, and `n2`, an obstacle lighting NOTAM. The contexts relevant to `myIntendedFlight` are context `Rotary_lowVis` and `anyVis_anyAircraft` as determined by `/detRelevantCtxs`. The derived case specific context `/rRotary_lowVis` computed by `detCaseSpecificCtx (resolved = TRUE)` contains the business vocabulary `obstacleLightingNOTAM` and `taxiwayNOTAM` as well as the business rules `highImportance` and `lowImportance`. Evaluating the rules for the issued NOTAMs `n1` and `n2` of `myIntendedFlight`, NOTAM `n2` is of high importance whereas regarding `n1` no additional information is derived.

V. Related Work

BRMSs like IBM's Operational Decision Management, Oracle Business Rules, or JBoss Drools consist of a business rule repository, a business rule engine, and tools for managing and defining business rules. Most BRMS are capable of organizing business rules into rule sets which can then be loaded and applied. Furthermore, many BRMS support business dictionaries containing business terms and facts. Some allow the definition of business vocabulary for specific rule sets. However, which rule set is to be applied to a specific business case is usually encoded in other systems accessing the business rule repository or decided by the user. Regarding relationships of business rule sets, at most inclusion of rule sets is provided.

Besides actual products, several papers pay attention to business rule management. Schafer and Kreher [25] group business rules into sets regarding a specific situation to simplify maintenance. Similarly, Rai and Anantaram [26] arrange business rules with the purpose of ease of decision making and efficiency of rule processing. Several groupings, e.g., use case, functionality, or business objects they concern, are listed. Another approach links rules with goals, other rules, and other relevant information [27]. All these approaches bear similarities to the presented model. Nevertheless, none of them states improvement of rule readability by their approach nor the support of business vocabularies. Furthermore, relationships between rule sets are not supported. Foremost, these approaches do not mention support of automatic determination of relevant business rules and vocabulary.

Homola et al. [19] present Contextualized Knowledge Repositories (CKR) which are based on the description logic variant $\mathcal{SROIQ}$-RL. Similar to the presented conceptual model they derive the context hierarchy from the hierarchy of parameter values. Additionally, Homola et al. introduce context classes which are automatically imported into all contexts that instantiate the context class' parameter value set. The hierarchical relationship of contexts is formalized in an ontology allowing reasoning on contextual knowledge. They also support the functionality *referencing*. Different from the presented conceptual context model, CKR do not provide an implementation independent generic model which can be adapted to different use cases. Furthermore, CKR assume fixed parameter sets and parameter values, or at least no semantics are defined for dynamism. Moreover, only hierarchical context relationships are supported whereas context versioning and decontextualization are not explicitly supported.

VI. Conclusion

In this paper we have identified the relevant dimensions regarding context models and list their options. Moreover, we have considered the identified dimensions and options regarding business rules and business vocabulary organization and documented the choices.

Based on these choices we have presented a generic conceptual context model for business rules and business vocabulary organization developed by applying the IS research framework described in [3]. This generic model can be instantiated multi-level for a specific domain and for a concrete application within this domain. Notable choices supported include: relationships of contexts and thus of business rule sets and business vocabularies; custom effects of relationships, e.g., custom inheritance mechanisms; custom semantics of modifying sets of contexts, parameters, and parameter values; and automatic determination of the relevant contexts for a given business case. To demonstrate the usefulness of the presented generic context model we instantiate it for SemNOTAM. We discuss the process of instantiation in detail with all its tasks, e.g., defining the inheritance mechanism. Furthermore, we give an example of context determination and rule application.

Advantages of the presented generic model are support of business rule sets and business vocabularies definition through tagging, i.e., the assignment of business rules and business vocabulary to the context in which they are relevant. Furthermore, as each context describes the situation in which their contained business rules and business vocabulary are relevant, the number of conditions needed to define business rules and business vocabulary decreases. This in turn increases the readability of business rule and business vocabulary definitions. Maintenance of business rule sets is simplified as the relevant rule set for a specific business case can be determined automatically. Moreover, the situation in which a rule set and its vocabulary are relevant is explicitly defined and contexts can be versioned. The extension of business rule sets and business vocabularies is supported by tagging. Furthermore, new business rules sets and business vocabularies can be easily added by instantiating a new context.

Additional benefits include the automatic determination of the relevant contexts and thus the relevant business rule sets and business vocabularies for a given business case. Furthermore, the generic context model supports localized reasoning as the reasoning technique can be specified for each context individually. The contextualization of business rules and business vocabularies enables efficient business rule processing as only the relevant ones are evaluated. This increases performance of rule evaluation and also eases debugging. Regarding usability the presented generic model for business rules and business vocabulary organization is rather simple with only a few classes and relationships. Nevertheless, the model demands additional effort for managing contexts, context parameters, and parameter values. However, as long as the number of parameters and parameter values is reasonable this effort is outweighed by the improvement in manageability of business rules and business vocabularies.

Future work includes an in-depth evaluation of the proposed model. This encompasses a comparison of the complexity reduction in managing business rules and business vocabularies using the proposed model with the complexity reduction in conventional approaches as well as the additional complexity introduced by the model. For this comparison the influence of dynamism, i.e., frequent changes of the set of contexts, parameters, and parameter values, need to be taken into account.

Acknowledgement

This research is supported by the Austrian Ministry of Transport, Innovation, and Technology in program TAKE OFF under grant FFG-839006 (SemNOTAM - SemanticNOTAMs: Ontology-based Representation and Semantic Querying of Digital Notices to Airmen).

References

[1] Object Management Group, "Semantics of business vocabulary and business rules (sbvr)," Object Management Group, Tech. Rep., 2015. [Online]. Available: http://www.omg.org/spec/SBVR/1.3/PDF

[2] BRCommunity.com. (2016, Feb.) Business rules community (brcommunity.com): The world's most trusted resource for business rule professionals. Business Rule Solutions, LLC. [Online]. Available: BRCommunity.com

[3] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[4] F. Burgstaller, D. Steiner, M. Schrefl, E. Gringinger, S. Wilson, and S. van der Stricht, "AIRM-based, fine-grained semantic filtering of notices to airmen," in *Integrated Communication, Navigation, and Surveillance Conference (ICNS)*, 2015, pp. D3–1–D3–13.

[5] R. Hoeft, F. Jentsch, and J. Kochan, "Freeing NOTAMs From Teletype Technology," *Flight Safety Digest*, vol. 23, no. 4, pp. 1 – 35, 2004.

[6] F. Burgstaller, D. Steiner, B. Neumayr, M. Schrefl, and E. Gringinger, "Using a model-driven, knowledge-based approach to cope with complexity in filtering of notices to airmen," in *Proceedings of the Australasian Computer Science Week Multiconference*, ser. ACSW '16. New York, NY, USA: ACM, 2016, pp. 46:1–46:10.

[7] P. Mayring, *Einführung in die qualitative Sozialforschung: Eine Anleitung zu qualitativem Denken*. Weinheim: Beltz Verlag, 2002.

[8] Business Rules Group. (2000, Jul.) Defining business rules what are they really? [Online]. Available: http://www.businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf

[9] R. G. Ross, *Principles of the Business Rule Approach*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

[10] B. von Halle, *Business Rules Applied: Building Better Systems Using the Business Rules Approach*, 1st ed. Wiley Publishing, 2001.

[11] Business Rules Group. (2003, Nov.) Business rules manifesto. Business Rule Group. [Online]. Available: http://www.businessrulesgroup.org/brmanifesto.htm

[12] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," in *In HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. Springer-Verlag, 1999, pp. 304–307.

[13] P. Brézillon, "Context in artificial intelligence: I. a survey of literature," 1999.

[14] ——, "Context in artificial intelligence: Ii. key elements of contexts," 1999.

[15] F. Giunchiglia and P. Bouquet, "Introduction to contextual reasoning. an artificial intelligence perspective," in *Perspectives on Cognitive Science*. NBU Press, 1997, pp. 138–159.

[16] M. Bazire and P. Brézillon, "Understanding context before using it," in *Modeling and Using Context*, ser. Lecture Notes in Computer Science, A. Dey, B. Kokinov, D. Leake, and R. Turner, Eds. Springer Berlin Heidelberg, 2005, vol. 3554, pp. 29–40.

[17] P. Dourish, "What we talk about when we talk about context," *Personal and Ubiquitous Computing*, vol. 8, no. 1, pp. 19–30, 2003.

[18] J. McCarthy, "Notes on formalizing context," in *Proceedings of the 13th International Joint Conference on Artifical Intelligence - Volume 1*, ser. IJCAI'93. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, pp. 555–560.

[19] M. Homola, L. Serafini, and A. Tamilin, "Modeling contextualized knowledge," in *Procs. of the 2nd Workshop on Context, Information and Ontologies (CIAO)*, vol. 626, 2010.

[20] M. Benerecetti, P. Bouquet, and C. Ghidini, "Contextual reasoning distilled," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 12, no. 3, pp. 279–305, 2000.

[21] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, 2010.

[22] T. Strang and C. L. Popien, "A context modeling survey," in *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing*, 2004.

[23] S. Chakravarthy, V. Krishnaprasad, E. Anwar, and S.-K. Kim, "Composite events for active databases: Semantics, contexts and detection," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 606–617.

[24] J. De Lara, E. Guerra, and J. S. Cuadrado, "When and How to Use Multilevel Modelling," *ACM Trans. Softw. Eng. Methodol.*, vol. 24, no. 2, pp. 12:1–12:46, December 2014.

[25] A. Schäfer and M. Kreher, "Wie strukturiere ich meine Business Rules für eine effektive Pflege und Steuerung von kritischen, komplexen und dynamischen Geschäftsprozessen," in *GI Jahrestagung (1)*, 2010, pp. 213–218.

[26] V. K. Rai and C. Anantaram, "Structuring business rules interactions," *Electronic Commerce Research and Applications*, vol. 3, no. 1, pp. 54 – 73, 2004.

[27] P. Kardasis and P. Loucopoulos, "Expressing and organising business rules," *Information and Software Technology*, vol. 46, no. 11, pp. 701 – 718, 2004.