Proceedings of ISA-18 Workshop at LREC2022, pages 102–109
Marseille, 20 June 2022

Intuitive and Formal Transparency in Semantic Annotation Schemes

Harry Bunt
Tilburg University
harry.bunt@tilburguniversity.edu

Abstract

This paper explores the application of the notion of *transparency* to annotation schemes, understood as the properties that make it easy for potential users to see the scope of the scheme, the main concepts used in annotations, and the ways these concepts are interrelated. Based on an analysis of annotation schemes in the ISO Semantic Annotation Framework, it is argued that the way these schemes make use of 'metamodels' is not optimal, since these models are often not entirely clear and not directly related to the formal specification of the scheme. It is shown that by formalizing the relation between metamodels and annotations, both can benefit and can be made simpler, and the annotation scheme becomes intuitively more transparent.

Keywords: semantic annotation, annotation methodology, metamodels, annotation schema design, ISO standard

1. Introduction

Interoperable semantic annotation has been a concern of the International Organization for Standardization ISO for the last 15-20 years. After a number of exploratory and feasibility studies such as Ide & Romary (2001), Bunt & Romary (2002), Bunt et al. (2005) and Ide & Pustejovsky (2010) the development of a suite of annotation standards was launched, in particular the Semantic Annotation Framework (SemAF), ISO 24617. In view of the complexity of semantic annotation, and taking into account the differences in maturity of approaches to various aspects of semantic analysis, as well as the lack of consensual approaches to some areas of semantics, it was decided to design SemAF as a suite of separate standards for the annotation of different aspects of semantic content.

The first standard in this suite, ISO 24617 Part 1, published in 2012, was a revamped version of the TimeML annotation scheme (Pustejovsky et al., 2003), which was a *de facto* standard for the annotation of temporal information. This standard is therefore informally known as 'ISO-TimeML'. Similarly, Part 2, also published in 2012, was a streamlined version of the existing DIT++ annotation scheme for dialogue act annotation (Bunt, 2009).

During the revamping of TimeML and the streamlining of DIT++, certain methodological aspects of the design of linguistic annotation schemes in general and semantic annotation schemes in particular, crystallized out. Discussions on the details of representing annotation of temporal information as XML expressions made it clear that annotation standards should not be established at the level of representation formats, but at a more abstract level, focusing on the use of standardized *concepts*. This insight came hand in hand with embracing the notion of *data categories*, as specifications of concept definitions according to well-defined terminological standards, to be documented not only in individual standards but also in a global *data category registry* (DCR, Broeder et al., 2010).

The Linguistic Annotation Framework (Ide & Romary, 2004; ISO 24612) captures some of the fundamental insights about linguistic annotation that emerged in the process, such as the importance of stand-off annotation and the distinction between annotations and representations. *Annotations* capture linguistic information about certain stretches of primary data, irrespective of a particular representation format; *representations* describe annotations in a particular format, such as XML. ISO standards should thus be specified at the level of annotations, rather than representations.

The distinction between annotations and representations is one of the cornerstones of the principles of semantic annotation formulated by Bunt (2010; 2014), and laid down in the methodological standard ISO 24617-6 (2016), which aims at securing the quality and methodological consistency of further SemAF parts. The annotation/representation distinction is implemented in this standard in requiring SemAF standards to have a 3-level architecture consisting of an abstract syntax, a concrete syntax, and a semantics, as displayed in Fig. 1. In this architecture, a concrete syntax for a given abstract syntax (plus semantics) is required to be *complete* and *unambiguous*. Completeness means that every well-formed annotation structure defined by the abstract syntax has a representation that encodes it; unambiguity means that every representation encodes exactly one structure of the abstract syntax. A representation format with these properties is called *ideal*. Due to the properties of completeness and unambiguity, all ideal representation formats for a given abstract syntax are semantically equivalent.

In addition to the three levels of this architecture, a tradition in the formulation of ISO standards for language resources is the presentation of a 'metamodel' as a visual view of the types of concepts involved in annotations. Figure 1 shows this architecture in a

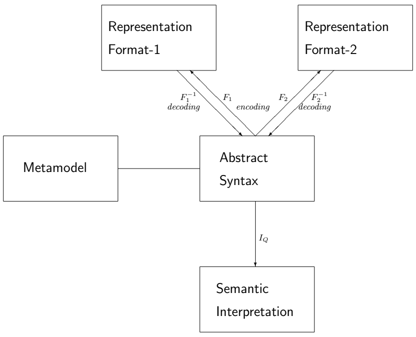

Figure 1: Architecture of SemAF standards.

schematic fashion, with two alternative representation formats specified by two different concrete syntax specifications.

Metamodels have often played an important role in developing SemAF parts, in making the scope of an annotation scheme explicit, and indicating its main concepts to be used in annotations, and their interrelations. From a methodological point of view, a question that remains, however, is what exactly is the status of the metamodel. Figure 1 shows the metamodel as somewhat hanging loose, not really connected with the ingredients of the formal specification of an annotation scheme. Is it just an easily interpretable pretty picture?

The paper proposes an answer to this question. It does so by showing that there can be a tight connection between a metamodel and actual annotations, and that such a coupling makes the annotation scheme intuitively transparant for its users. This notion of transparency is given a formal basis by defining a relation of 'instantiation' between metamodels and annotations. The simplicity of this relation largely determines the transparency of the annotation scheme. Formal transparency as the basis of intuitive transparency. This will be illustrated by several SemAF metamodels and in particular by showing how the aim of transparency helps to simplify both the metamodel and the annotations in the standard under development for quantification phenomena.

This paper is organised as follows. Section 2 discusses the development of annotation schemes in the ISO Semantic Annotation Framework and the role of metamodels in the process. Section 3 discusses the notion of transparency applied to metamodels. Section 4 develops the idea that annotations can be viewed as instances of a metamodel, formalizing the notion of a metamodel as a graphical structure and showing how XML annotations can be mapped to that format. The metamodel and annotations of QuantML (ISO WD 24617-12) are used to illustrate this. Section 5 discusses the advantages

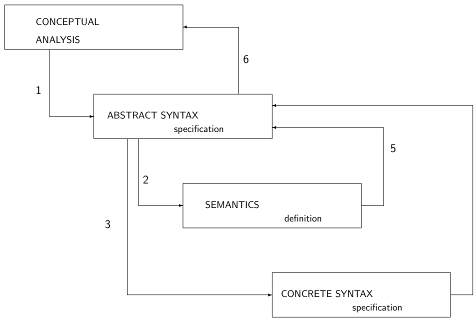

Figure 2: CASCADES development process.

of treating annotations as instances of a metamodel, and concludes by considering the methodological consequences of this idea.

2. Metamodels in SemAF

2.1. The CASCADES development process

For the development of the individual SemAF parts, a process methodology has been developed called CASCADES (**C**onceptual analysis, **A**bstract syntax, **S**emantics, and **C**oncrete syntax for **A**nnotation **DES**ign) (Bunt, 2015) which has been included in the methodological standard ISO 24617 Part 6, Principles. Figure 2 shows the steps in this process, starting with a conceptual analysis phase and ending in the specification of a concrete syntax.

The CASCADES model derives its usefulness in the first place from enabling a systematic design process, in which due attention is given to the conceptual and semantic choices on which more superficial decisions such as the choice of particular XML attributes and values should be based. Second, the model provides methodological support by means of procedures for how to make the step from one level of decision-making to the next, in particular for (1) how to construct an abstract syntax given a metamodel (step 1 in Fig. 2); (2) how to define a formal semantics for a given abstract syntax (step 2); and (3) how to map an abstract syntax to an XML-based concrete syntax.

Realistic design processes require feedback loops. Figure 2 shows three such loops. First, the specification of an abstract syntax is a way to formalise the conceptual analysis in the initial stage of the process. This formalisation may very well clarify or alter some aspects of the initial analysis; CASCADES step 6 is for feeding the results of the formalisation back into the conceptual analysis. Second, the specification of a concrete syntax, defining a specific representation format, may by virtue of its concreteness motivate adaptations in the underlying abstract syntax; step 4 is for this feedback in the design process. Third, since the definition of a semantics for an abstract syntax is the best way to find inadequacies in the latter, this may be fed back into the abstract syntax specification (step 5). And finally, the latter two feedback loops may well be combined: if the feedback in step 4 has resulted in a revised specification of the abstract syntax, them this will require adaptations to be made in the semantics (step 2), which may be fed back again into the abstract syntax specification (step 5). This cycle $\langle 2; 5 \rangle$ may be repeated until the abstract syntax and its semantics are satisfactory and stable, at which point the annotation language is considered to be semantically adequate. The concrete syntax should now be adapted to this abstract syntax (step 3) - which in turn may have consequences that should be fed back (step 4). In fact, the 'outer cycle' $\langle 3; 4 \rangle$ does not make much sense to perform if not combined with the 'inner cycle' $\langle 2; 5 \rangle$, resulting together in the feedback loop (1):

(1) $\langle 4; \langle 2; 5 \rangle^*; 3 \rangle^*$

This feedback loop is particularly important not only for systematically developing a consistent design, starting from scratch with of conceptual analysis, but also for being applied to a pre-existing representation format, in order to detect semantic deficiencies, or to develop an annotation language that better meets the requirements of the ISO Linguistic Annotation Framework and the requirements of semantic adequacy.

3. Transparency in Metamodels

An annotation scheme is intuitively more transparent if presented with a metamodel that is conceptually clear and informative. Conceptual clarity can be achieved by using a relatively small number of well-defined concepts. In several SemAF documents, such as ISO 24617-2 and ISO WD 24617-12, a metamodel is presented together with a discussion of basic concepts in order to support this aspect of the model's clarity. Informativeness means that the metamodel gives a good indication of the concepts that make up annotations according to this scheme.

Figure 3, for example, shows the metamodel for reference annotation in ISO 24617-9 (2019). This metamodel indicates that (1) referring expressions are anchored to segments in the primary data; (2) such expressions refer to entities that play a role in a discourse ('discourse entities'); and (3) that two kinds of relations are distinguished: relations between referring expressions ('lexical relations', like synonymy) and relations between discourse entities ('objectal relations', like identity).

Its simplicity makes this metamodel exemplary in its clarity, but it is not very informative: it hardly provides any information about the concepts that go into annotations according to this annotation scheme. Moreover, a critical look at the metamodel raises various questions: What is the significance of the frame around the top

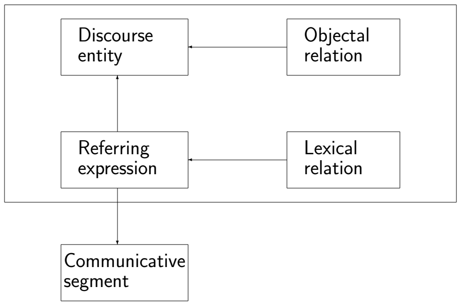

Figure 3: Metamodel for coreference annotation (ISO 24617-9).

four boxes? What do the arrows from objectal relations and lexical relations to discourse entities and referring expressions signify? Do the arrows from referring eXpressions to discourse entities and communicative segments have the same significance? Altogether, this metamodel does not contribute much to making the Reference Annotation Schema (ISO 24617-9) transparent to its users.

Figure 4 shows the metamodel underlying QuantML annotations, as proposed in ISO WD 24617-12, i.e. Part 12 of the SemAF suite, which is currently under development. Arrows with multiple heads indicate the possibility of multiple linking (like for the participation in events) or an attribute having multiple values (like for the reference domain of a quantification being defined by a source domain and multiple modifiers). This metamodel contains all and only those concepts of which instances may occur in QuantML annotations. In the next section, we formalize the relation between this metamodel and the annotations that it supports.

4. Annotations as Metamodel Instances

This section explores the idea that annotations can be regarded as instances of the metamodel. This idea is based on the observation that a metamodel provides information about relevant combinations of concepts. For example, the QuantML metamodel in Fig. 4 says that a set of participants may be involved in a set of events in a variety of ways, characterized by five concepts: distributivity, semantic role, event scope, polarity, and exhaustiveness. The annotation of a given item of primary data, such as the sentence *"Only three of the fifty-two students protested"* will for example say that there is a participant set of three students, taken from the reference domain consisting of 52 students, individually involved as Agents, with positive polarity, and exhaustively (none of the other students protested). The annotation thus combines instances of the concepts in the metamodel. The annotation of this sentence represented in QuantML/XML format is shown in (7) below. This representation has a straightforward mapping to a graphical representation in terms of com-

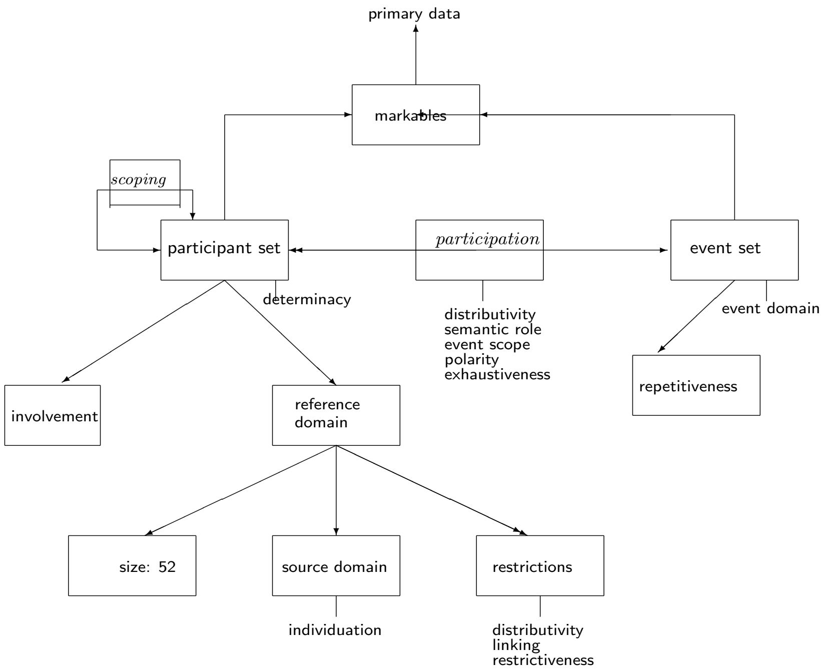

Figure 4: Metamodel for quantification annotation (ISO WD 24617-12:2021)

ponents of the metamodel. Boxed entities in the latter representation correspond to XML elements, and strings associated with boxes correspond to (string) values of attributes within such elements; arrows from boxes to boxes indicate attributes with structured values. Double-headed arrows indicate the possible multiplicity of relata (such as multiple sets of participants involved in certain events) To further explore the formal relation between metamodels and annotations, we first formalize the graphical representation of metamodels used in Fig. 4 and subsequently introduce the notion of 'instance' of such a graph.

4.1. Metamodels as M-Graphs

First, inspecting the metamodel shown in Fig. 4, we note that there are four types of ingredients:

(2) Metamodel ingredients:

1. boxes containing structured concepts (such as participant sets and reference domains), source domains. Some of these are linked to markables, others are not;
2. boxes containing unstructured objects (such as size, involvement, and repetitiveness; these are not linked to markables);
3. structured labeled relations (such as participation, scoping);
4. unlabeled arrows emanating from boxes of type 1 and connecting these to boxes of type 2.

These types of ingredients can be formally defined as follows:

(3)
1. An M-Box is a quadruple ⟨markable, element type, simple concept list, complex concept list ⟩.
2. A simple concept list is a list of concepts of which the instances are unstructured entities.
3. A complex concept list is a list of concepts of which the instances are structured entities (represented by M-Boxes that are pointed to).
4. An M-link is a triple ⟨M-Box, M-label, M-Box⟩.
5. An M-label is a triple ⟨label-name, simple concept list⟩.

A Metamodel Graph (M-Graph) is a collection of M-boxes and M-links. Metamodels of the form of Fig. 4 can be formalized as M-graphs using the following mapping relation.

(5) Mapping metamodel diagrams to M-Graphs:

1. Boxes containing structured concepts are mapped to M-Boxes.
2. Boxes containing unstructured objects are mapped to elements in the simple concept list of the M-Box at the tail of the arrow to such boxes.
3. Structured labeled relations are mapped to M-links with the same label name.
4. Unlabeled arrows connecting two boxes with structured concepts are mapped to elements in the complex concept list of the M-Box at the tail.

Using these formal definitions and mappings to M-Graphs, the QuantML metamodel can be formally

(4) QuantML metamodel as M-graph:
$M_{QuantML}$ =
{ ⟨markable, event set, [repetitiveness, event domain]⟩,
⟨markable, participant set, [determinacy, involvement, reference domain]⟩,
⟨markable, reference domain, [size, source domain, restrictions]⟩,
⟨markable, source domain, [individuation]⟩,
⟨⟨markable, event set, [repetitiveness, event domain]⟩,
⟨participation, [distributivity, semantic role, event scope, polarity, exhaustiveness]⟩,
⟨markable, participant set, [determinacy, involvement, reference domain]⟩⟩,
⟨⟨markable, participant set, [determinacy, involvement, reference domain]⟩,
⟨scoping, [argument scope]⟩,
⟨markable, participant set, [determinacy, involvement, reference domain]⟩⟩ }

specified as the M-Graph in (9).

It may be noted that the status of the ingredients in the QuantML metamodel is in some cases not entirely clear. Concepts like determinacy, polarity, and exhaustiveness are clearly unstructured, but for concepts like involvement, markables, and size it isn't obvious whether they are structured or unstructured. The 'restrictions' concept is clearly one with internal structure, so why does the metamodel not say anything about that? Why are some boxes with structured concepts linked to markables, others not? The formalization of metamodels as M-Graphs helps to make these issues explicit and resolve them. It may be noted here that by identifying M-Boxes with diagram boxes containing structured concepts, as in (5), every box in this diagram should either be linked to a markable or should contain an unstructured concept. This is not the case: the boxes 'source domain' and 'restrictions' contain structured objects but are not linked to markables. Moreover, the concepts of 'involvement', 'size', and 'repetitiveness' are in fact structured, which is not indicated in the metamodel, and which requires them to also be linked to markables. We will return to these issues below.

4.2. MI-Graphs

Just as metamodels, represented graphically in terms of boxes connected by labelled and unlabelled arrows, can be formalized as M-Graphs, similarly annotations can be represented graphically in much the same way, which can be formalized as 'instances' of M-Graphs. Such instances are called 'MI-Graphs', and are formally defined as follows.

A Model Instantiation Graph (MI-Graph) is a collection of nodes connected by labeled structured edges, called MI-links, and labeled unstructured edges. Nodes have the form of boxes, called MI-boxes, which consist of a name (like 'participant set'), a markable, a list of attribute-value pairs, and zero or more directed edges labeled by attribute names (like 'size' and 'domain') which point to other boxes. Formally, an MI-Graph is a collection of MI-boxes and MI-links such that all MI-boxes are linked to one or more other MI-boxes, and all MI-links connect two MI-boxes. The following definitions formalize the notions of MI-box and MI-link.

(6) **Definition.** An MI-Graph is an instance of an M-Graph, i.e.:

- An MI-Box is a quadruple ⟨m, e, AV, AMI⟩, where m is a markable, e is an element type of the M-Graph (such as 'event set'), AV is a list of instances of unstructured concepts, and AM is a list of instances of structured concepts, labeled with names of attributes that have structured values.
- An MI-label is a pair ⟨label-name, AV⟩, with AV as above.
- An MI-link is a triple ⟨MI-Box, MI-label, MI-Box⟩.

4.3. Annotations as MI-Graphs

Example (7) shows the QuantML annotation of "Only three of the fifty-two students protested" in XML.

(7) **Primary data:**
"Only three of the fifty-two students protested."
Segmentation:
m1 = three of the fifty-two students, m2 = the fifty-two students, m3 = students, m4 = protested.
Annotation in QuantML/XML:
<event xml:id="e1" target="#m4" pred="protest"
<entity xml:id="x1" target="#m1" domain="#x2" involvement="#n1" determinacy="indet" size= "52"/>
<refDomain xml:id="x2" target="#m2" source="#x3" restrs=""/>
<sourceDomain xml:id="x3" target="m3" individuation="count" pred="student"/>
<numPred xml:id="n1" numRel="equal" num="3"/>
<event xml:id="e1" target="m4" pred="protest"/>
<participation event="e1" participant="x1" semRole="agent" distr="individual" eventScope= "narrow" exhaustiveness= "exhaustive" polarity= "positive"/>

The corresponding MI-Graph is not easily obtained from this representation, since the latter includes the specification of the reference domain size as a property of the participant set, whereas in the metamodel it is a property of the reference domain. This suggests a lack of transparency in the annotation scheme. To remedy this, the simplest solution is to move the @size attribute from <event> elements (corresponding to participant sets) to <refDomain> elements.

The XML representation can be converted into an MI-Graph by applying the function F_{XG}, defined in (8). This function takes a QuantML/XML annotation structure A_X as a parameter and converts its constituent XML elements into MI-Boxes and MI-Links.

(8)
1. $F_{XG}(A_X$, <entity xml:id="x" target="#m" involvement="#xi" domain="#y" determinacy="d"/>) =
   ⟨m, participant set, [⟨'d'⟩], [⟨involvement, $F_{XG}(A_X$,#xi)⟩, ⟨domain, $F_{XG}(A_X$,#y)⟩]⟩
2. For any identifier of the form $\#z$, $F_{XG}(A_X$, $\#z$) = the result of applying F_{XG} to the A_X-element with xml:id="z".
3. For any constant c, $F_{XG}(A_X$, c) = c.
4. $F_{XG}(A_X$, <event xml:id="e" target="#m" pred="P" rep="#r"/>) =
   ⟨m, event set, [⟨$F_{XG}(A_X$,#r)⟩, ⟨domain, $F_{XG}(A_X$,P)⟩]⟩
5. $F_{XG}(A_X$, <participation event="#e" participant="#x" semRole=A distr="individual" exhaustiveness="exhaustive" polarity= "positive"/>) =
   ⟨$F_{XG}(A_X$,#e), ⟨participation, [$F_{XG}(A_X$, A), individual, exhaustive, positive]⟩, $F_{XG}(A_X$,#x)⟩
6. $F_{XG}(A_X$, <numPred xml:id="n" target="#m" numRel="R" num="#k"/>) =
   ⟨m, numPred, [⟨$F_{XG}(A_X$,R)⟩, ⟨nu, $F_{XG}(A_X$,#k)⟩]⟩
7. And so on.

Similarly, the inverse function F_{GX} converts an MI-Graph into a QuantML/XML annotation structure.

The MI-Graph corresponding to the QuantML/XML representation in (7) is shown in (9), in which the annotations of involvement (participant set size "three") and reference domain size ("fifty-two") have for the sake of readability been simplified to numbers.

(9) Annotation (8) as MI-graph:
{ ⟨m4, event set, [protest]⟩,
⟨m1, participant set, [indeterminate, 3, 52],
[⟨domain, F_{XG}(#x2)]⟩,
⟨m2, reference domain, [], [source domain,
F_{XG}(#x3)]⟩,
⟨m3, source domain, [count, student]⟩,
⟨⟨m4, event set, [protest]⟩,
⟨participation, [individual, agent, narrow,
positive, exhaustive]⟩,
⟨m1, participant set, [indeterminate, 3, 52],
[⟨domain, F_{XG}(#x2)]⟩,
⟨scoping, [argument scope]⟩ } }

After applying the recursively embedded calls to F_{XG}, and using the visualization meethod of M-Graphs that is behind the diagram in Fig. 4 the MI-Graph in (7) can be rendered graphically as shown in Fig. 5. Comparing this representation with the metamodel in Fig. 4, we can see clearly that the QuantML metamodel is optimally 'transparent' in the sense of giving users of the annotation scheme an immediate impression of the annotations that the scheme supports. The relative simplicity of the graphical representation as a metamodel instantiation graph is rather surprising, given the complexity of quantification phenomena in natural language. The graphical representations of annotations can also be viewed as better human-readable than the XML-representations. As the conversion function $F_X G$ makes explicit, XML expressions can be automatically converted to this graphical format, which opens a possibility for easy inspection of QuantML annotations.

5. Discussion and Conclusions

The formalization of metamodels as M-Graphs, and the notion of instantiating an M-Graph to represent annotations, is useful for defining metamodels with greater precision and to see that a metamodel is an abstraction of individual annotations. In that sense, the metamodel is maximally informative, and maximally transparent. While formalizing the informal box-and-arrow representation of the QuantML metamodel as an M-Graph, we encountered several issues that the metamodel did not address properly.

First, as noted in section 4.1, every non-relational box is intended to correspond to a structured concept, characterized by a number of features.[1] The fact that some of these concepts are linked to a markable while others are not, suggests that some of them are expressed in the primary data while others are not. This is not really the case: involvement, size, source domain, domain restrictions, and repetitiveness are all expressed in the data. Specifications of size and repetitiveness are possibly complex quantitative predicates, like *slightly more than 12 ounces*, *between 40 and 45*; involvement specification can also use such predicates, as well as vague predicates like *not much, just a few, quite a lot* and proportional indications like *nearly all, by*

[1] In terms of the QuantML abstract syntax, not considered in this paper, every non-relational box corresponds to a so-called 'entity structure' and every relational box to a 'link structure'. Entity structures by definition contain semantic information about a stretch of primary data, and are thus always linked to a markable,

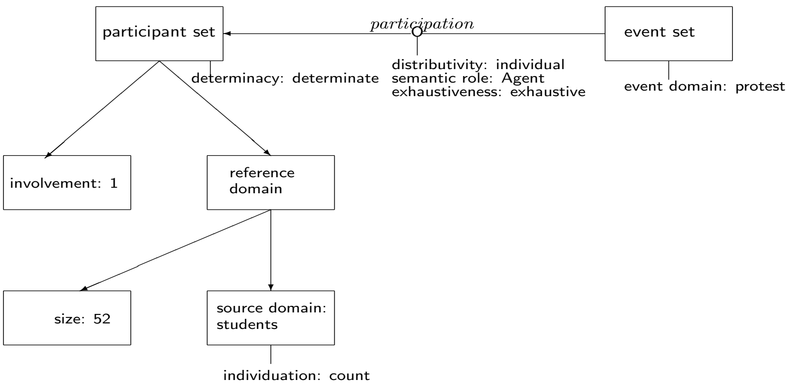

Figure 5: QuantML annotation as instantiation of the metamodel .

far most. The QuantML metamodel is deliberately unspecific about how much detail such structures should be covered in its annotations, imagining that annotation scheme plug-ins, possibly based on the ISO standard for annotating measurable quantitative information (ISO 2417-9XX), could be added on for this purpose. Where this is a viable strategy remains to be seen.

Second, another deliberate choice in the metamodel concerns the lack of detail about of reference domain restrictions. Such restrictions can take a variety of forms natural language, such as adjectives, nouns, relative clauses, prepositional phrases, and possessive phrases. Each of these forms comes with slightly different semantic structures, and it would clutter up the metamodel to make these all explicit. This could perhaps be resolved by specifying one or more separate sub-metamodels for the various forms of restriction.

From a methodological point of view, the explorations in this paper shed new light on the relation between metamodels and annnotation representations, as depicted in Fig. 5, and on the role of this relation in the CASCADES development process depicted in Fig. 2. Conceptually, the metamodel of an annotation scheme is closely related to the abstract syntax specification, as the CASCADES model in Fig. 2 also suggests, but in this paper we have shown that the metamodel can also be tightly coupled with a particular annotation representation format through the notion of instantiation. This is shown in Fig. 6. For the CASCADES design model, it suggests that it may be useful to add a step where a forward jump is made from metamodel specification to the establishment of an annotation representation format, and a backward jump in the opposite direction - this is shown in Figure 7.

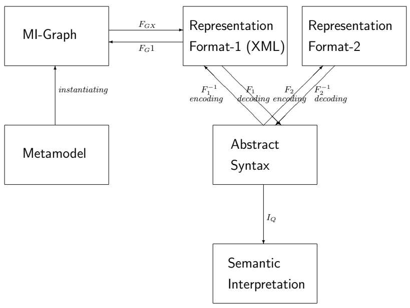

Figure 6: Architecture of SemAF standards with metamodel instantiation.

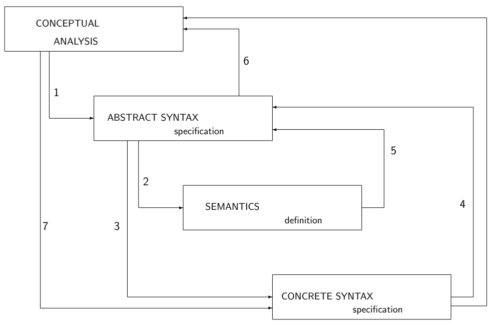

Figure 7: CASCADES development process with metamodel instantiation.

6. References

Broeder, D., M. Kemps-Snijders, D. Uytvanck, M. Windhouwer, P. Wittenburg, and C. Zinn (2010). A data cataegory registry and component-based metadata framework. In *Proceedings of LREC 2010*. ELRA, Paris.

Bunt, H. (2009a). A framework for dialogue act specification. In R. C. D. Heylen, C. Pelachaud and D. Traum (Eds.), *Proceedings of EDAMLAAMAS Workshop "Towards a Standard Markup Language for Embodied Dialogue Acts, Budapest*, pp. 36–36.

Bunt, H. (2015). On the principles of semantic annotation. In *Proceedings 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-*

11), London, pp. 1–13.

Bunt, H., M. Kipp, M. Maybury, and W. Wahlster (2005). Fusion and coordination for multimodal interactive information presentation. In O. Stock and M. Zancanaro (Eds.), *Multimodal Intelligent Information Presentation*, pp. 325–339. Dordrecht: Springer.

Bunt, H. and L. Romary (2004). Standardization in Multimodal Content Representation: Some methodological issues. In *Proceedings of LREC 2004*, Lisbon, pp. 2219–2222. Paris: ELRA.

Bunt, H. and A. Shiffrin (2006). Methodologial aspects of semantic annotation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genova, Italy*, Paris. ELRA.

Ide, N. and J. Pustejovsky (2010). What does interoperability mean, anyway? toward an operational definition of interoperability. In *Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL 2010)*, Hong Kong.

Ide, N. and L. Romary (2001). Standards for language resources. In *Proceedings of the IRCS Workshop on Linguistic Databases*, Philapdelphia.

Ide, N. and L. Romary (2004). International Standard for a Linguistic Annotation Framework. *Natural Language Engineering 10*, 211–225.

ISO (2009). *24612:2009, Language Resource Management - Linguistic Annotation Framework (LAF)*. International Organisation for Standardisation ISO, Geneva.

ISO (2012). *ISO 24617-1: 2012, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 1: Time and events*. Geneva: International Organisation for Standardisation ISO.

ISO (2014). *ISO 24617-4: 2014, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 4: Semantic roles*. Geneva: International Organisation for Standardisation ISO.

ISO (2015). *ISO 24617-6:2015, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 6: Principles of semantic annotation*. Geneva: International Organisation for Standardisation ISO.

ISO (2019a). *ISO 24617-9:2019, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 9: Reference annotation framework (RAF)*. Geneva: International Organisation for Standardisation ISO.

ISO (2019b). *ISO/WD 24617-12:2019, Language Resource Management: Semantic Annotation Framework (SemAF) - Part 12: Quantification*. Geneva.: International Standard. International Organisation for Standardisation ISO.

ISO (2021, November). *ISO/WD 24617-12: 2021, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 12: Quantification (Second Draft)*. Geneva: International Organisation for Standardisation ISO.

Petukhova, V. and H. Bunt (2008). LIRICS semantic role annotation: design and evaluation of a set of data categories. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco. ELRA, Paris.

Pustejovsky, J., H. Bunt, and K. Lee (2010). ISO-TimeML. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Malta. ELDA, Paris.

Pustejovsky, J., J. Castano, R. Ingria, R. Gaizauskas, G. Katz, R. Saurí, and A. Setzer (2003, January). TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, Tilburg, Netherlands, pp. 337–353.