On the principles of interoperable semantic annotation

Harry Bunt
TiCC, Tilburg Center for Cognition and Communication
Tilburg University, The Netherlands
harry.bunt@uvt.nl

Abstract

This paper summarizes the research that is leading to ISO standard 24617-6, which describes the approach to semantic annotation that characterizes the ISO semantic annotation framework (SemAF). It investigates the consequences and the risks of the SemAF strategy of developing separate annotation schemes for certain classes of semantic phenomena, with the long-term aim to combine these schemes into a single, wide- coverage scheme for semantic annotation. The principles are discussed for linguistic annotation in general and semantic annotation in particular that underly the SemAF effort. The notions of abstract syntax and concrete syntax are described with their relation to the specification of a metamodel and the semantics of annotations. Overlaps between the annotation schemes defined in SemAF parts are discussed, as well as semantic phenomena that cut across these schemes.

1 Introduction

ISO standard 24617-6, "Principles of semantic annotation", sets out the approach to semantic annotation that characterizes the ISO semantic annotation framework (SemAF). In addition, it provides guidelines for dealing with two issues regarding the annotation schemes defined in the different parts of SemAF: inconsistencies that may arise due to overlaps between annotation schemes, and semantic phenomena that cut across SemAF-parts, such as negation, modality, and quantification.

The purpose of ISO 24617-6 is to provide support for the establishment of a consistent and coherent set of international standards for semantic annotation. It does so in three ways. First, by making explicit which basic principles underly the approach followed in the SemAF parts that have already produced ISO standards (Part 1, Time and events; Part 2, Dialogue acts); and in the parts that are under way (Part 4, Semantic roles; Part 7, Spatial information; Part 8, Discourse relations). This approach lends methodological coherence to SemAF and helps to ensure consistency between existing, developing, and future SemAF parts. Second, by identifying overlaps between SemAF parts, and indicating how these may be dealt with. Third, by identifying common issues that cut across SemAF parts and which are not or only partially covered, where possible indicating directions for how these issues may be tackled.

Semantic annotation enhances primary data with information about their meaning. Given the current state of the art in semantics, it is unlikely that any existing formalism for representing semantic information would have general support from the research community. In practice, moreover, semantic annotation tasks often have the limited aim of annotating certain specific semantic phenomena, such as semantic roles, discourse relations, or coreference relations, rather than annotating the full meaning of primary data. Therefore a strategy was adopted to devise separate standards in different SemAF parts, with annotation schemes for specific semantic phenomena; over time, these schemes could develop into a wide-coverage framework for semantic annotation.

This 'crystal growth' strategy has proved fruitful in making progress in the establishment of standardized annotation concepts and schemes in support of the development of interoperable resources, but it also entails certain risks: (1) annotation schemes defined in different SemAF parts are not necessarily mutually consistent; (2) it may not be possible to combine the schemes, defined in different parts, into a coherent single scheme if they incorporate different views or employ different methodologies; and (3) some semantic phenomena are outside the scope of all SemAF parts but cannot be disregarded entirely in some parts, which may lead to unsatisfactory treatments of these phenomena. The methodological principles and guidelines provided in this standard are designed to minimize these risks.

Mutual consistency of SemAF parts is essential for making the integration possible of annotation schemes defined in different parts. Three aspects of consistency among annotation schemes can be distinguished:

- methodological consistency, i.e. the same approach is followed with respect to the distinction between abstract and concrete syntax and their interrelation, and with respect to their semantics;
- conceptual consistency, i.e. different schemes are based on compatible underlying views regarding their basic concepts, e.g. verbs are viewed as denoting states or events, rather than relations;
- terminological consistency, i.e. terms which occur in different annotation schemes have the same meaning in every scheme, and the same term is used across annotation schemes for indicating the same concept.

The rest of this paper is organized as follows. Section 2 summarizes certain principles for standard annotation schemes in general, and some that are specific for the annotation of semantic information. Section 3 outlines the methodological basis of SemAF, taking these principles into account. Section 4 discusses cases of overlaps between annotation schemes and the consistency issues that these give rise to. Section 5 discusses a number of semantic phenomena whose annotation cuts across SemAF parts. The paper ends with conclusions in Section 6.

2 Annotation principles and requirements

The ISO efforts aiming to develop standards for semantic annotation rest on a number of basic principles for semantic annotation, some of which have been laid out by Bunt & Romary (2002; 2004) and developed further in Bunt (2010; 2013); others have been formulated as general principles for linguistic annotation and are part of the ISO Linguistic Annotation Framework (LAF, ISO 24623:2012). The latter are often of a very general nature, such as the principle that segments of primary data are referred to in a uniform and TEI-compliant way, and the principle that the use of multiple layers over the primary data is supported, with stand-off annotation and a uniform way of cross-referencing between layers.

The use of layers of annotation is of particular relevance for SemAF because it allows different layers to be used for different types of semantic information, such as one layer for the annotation of events, time and space, and another one for semantic roles, each with their own annotation scheme. While this allows in principle the use of layers which are not mutually consistent, the ‘crystal growth’ strategy of SemAF is designed to allow the annotation schemes for the various types of semantic information to grow into a single coherent annotation scheme.

Of particular relevance for SemAF is also the distinction between ‘annotations’ and ‘representations’ (Ide & Romary, 2004). An annotation is any item of linguistic information that is added to primary data, independent of a particular representation format. A representation is a rendering of an annotation in a particular format, e.g. as an XML expression. This distinction has incited the development of a methodology for developing semantic annotation schemes with an ‘abstract syntax’ of annotations and a ‘concrete syntax’ of representations. This methodology is described in Section 3.

Other general principles for designing annotation schemes include empirical validity; theoretical justification; learnability for humans and machines; generalizability; completeness; and compatibility with existing good practices. Of special importance are moreover the requirements of extensibility and variable granularity:

Extensibility ISO standard annotation schemes are designed to be language-, domain- and application-independent, but some applications or some languages may require specific concepts which are not relevant in other applications or languages. Therefore, annotation schemes should allow extension with language-, domain-, or application-specific concepts.

Variable granularity One way to achieve good coverage is to include annotation concepts of a high level of generality, which cover many specific instances. Since an annotation scheme which uses only very general concepts would not be optimally useful, this leads to the principle that annotation schemes should include concepts with different levels of granularity. This is also beneficial for its interoperability, as it gives more possibilities for conversion to and from existing annotation schemes and the standard scheme.

The idea behind annotating a text, which dates from long before the digital era, is to add information to a primary text information in order to support its understanding. The semantic annotation of digital source text has a similar purpose, namely to support the understanding of the text by humans as well as by machines. Therefore, semantic annotations must satisfy the principle of 'semantic adequacy':

Semantic adequacy: semantic annotations add information to source data in a form that has a well-defined semantics, ensuring the annotations to be machine-interpretable.

3 The methodological basis of SemAF

3.1 Steps in the design of an annotation scheme

An annotation scheme determines which information may be added to primary data, and how that information is expressed. When an annotation scheme is designed from scratch, the first step to take is a conceptual analysis of the information that annotations should capture. This analysis identifies the concepts that form the building blocks of annotations, and specifies how these blocks may be used to build annotation structures. This step corresponds to what is known in ISO projects as the establishment of a 'metamodel', i.e. the expression of a conceptual view of the information in annotatations. The second step, indicated by '2' in Figure 1, articulates this conceptual view as a formal specification of categories of entities and relations, and of how annotation structures can be built up from elements in these categories. This formal specification defines the *'abstract syntax'* of an annotation language.

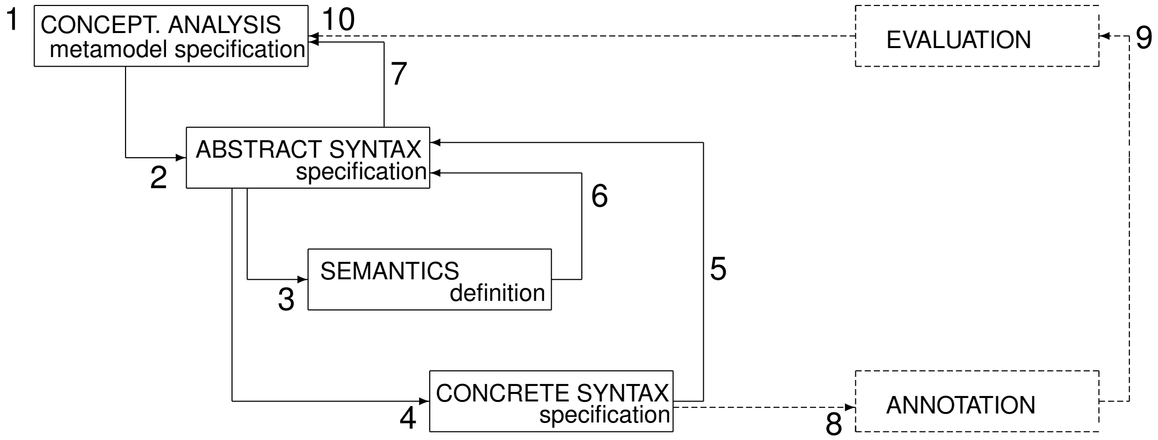

Figure 1: Steps and feedback loops in the CASCADES method

While these two steps make explicit what information can be captured in annotations, they do not specify how annotations should be represented, for example as XML strings, as logical formulas, as graphs, as feature structures, or otherwise; the abstract syntax defines the specification of information in terms of set-theoretic structures. The definition of a representation format for annotation structures occurs in step 4 in Figure 1, the specification of a concrete syntax.

Step 3 is the specification of the meaning of the structures defined by the abstract syntax, i.e. the specification of a semantics for annotation structures. By definition, a representation defined by the concrete syntax has the meaning of the abstract annotation structure that it represents.

This method for designing an annotation scheme is called CASCADES: **C**onceptual analysis, **A**bstract syntax, **S**emantics, and **C**oncrete syntax for **A**nnotation language **DES**ign. Figure 1 visualizes the CASCADES method, of which the central concept of an abstract syntax for annotations with the specification of a semantics, was introduced in Bunt (2010). The dotted parts of Figure 1 are discussed in Section 3.3.

The CASCADES method is useful for 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. Apart from supporting the design of an annotation scheme from scratch, this method also provides support for improving an existing annotation scheme. This support consists not only in the distinction of four well-defined design steps but also of procedures and guidelines for taking these steps and using feedback loops, as discussed in Section 3.3.

3.2 Metamodels, abstract syntax, concrete syntax, and semantics

A metamodel of an annotation scheme is a schematic representation of the relations between the concepts that are used in annotations. Over the years, two slightly different notions of a metamodel have been used in ISO projects, namely: (a) as a representation of the relations between the most important concepts that are mentioned in the document in which the standard is proposed; (b) as a representation of the relations between the concepts denoted by terms that occur in annotations. Metamodels of type (a) may be helpful for nontechnical readers to better understand an annotation scheme; those of type (b) are a visualization of the abstract syntax of the scheme, and are helpful to see at a glance what information the annotations may contain. Note that a type (a) metamodel may have a type (b) metamodel as a proper part.

The abstract syntax of an annotation scheme specifies the information in annotations in terms of set-theoretical structures such as the triple $\langle e_1, e_2, R_i \rangle$ which relates the two arguments e_1 and e_2 through the relation R_i. More generally, these structures are n-tuples of elements which are either basic concepts, taken from a store of basic concepts called the 'conceptual inventory' of the abstract syntax specification, or n-tuples of such structures. An *annotation structure* is a set of *entity structures*, which contain semantic information about a region of primary data, and *link structures*, which describe a semantic relation between two such regions.

A concrete syntax specifies a representation format for annotation structures, such as the representation of a triple like $\langle e_1, e_2, R_i \rangle$ by a list of three XML elements, of which the element <srLink event="#e1" participant="#x1" semRole="agent"/> represents the relation and the other two elements represent two entity structures.

A representation format for annotation structures should ideally give an exact expression of the information contained in annotation structures. A concrete syntax, defining a representation format for a given abstract syntax, is said to be *ideal* if it has the following properties:

- Completeness: every annotation structure defined by the abstract syntax can be represented by an expression defined by the concrete syntax;
- Unambiguity: every representation defined by the concrete syntax is the rendering of exactly one annotation structure defined by the abstract syntax.

The representation format defined by an ideal concrete syntax is called an *ideal representation format*. Due to its completeness, an ideal concrete syntax defines a function from annotation structures to representations, and due to its unambiguity there is also an inverse function from representations to annotation structures. It follows that for any two ideal representation formats are interoperable: there is a complete meaning-preserving mapping from one format to the other. Figure 2 visualizes the relations between abstract syntax, semantics, and multiple ideal concrete syntactic specifications.

An ideal concrete syntax can be derived systematically from an abstract syntax. For example, a concrete syntax defining XML representations can be constructed as follows:

1. For each element of the conceptual inventory specify an XML name;

2. For each type of entity structure $\langle m, s \rangle$ define an XML element with the following attributes and values:

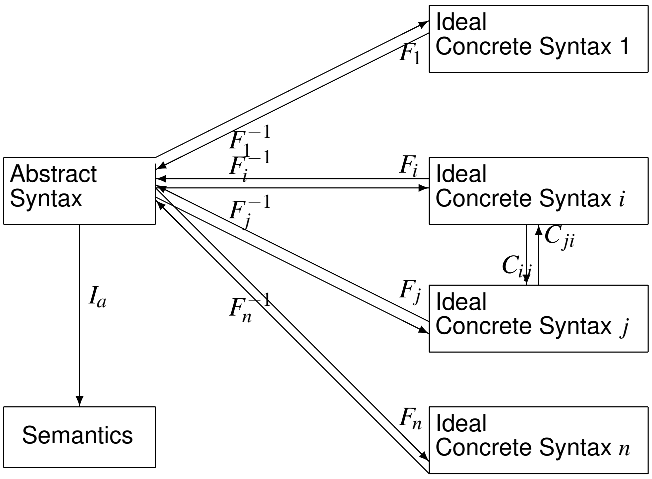

Figure 2: Relations Relations between abstract syntax, semantics, and concrete syntax of annotations.

- the special attribute 'xml:id', whose value is an identifier of the element;
- the special attribute 'target', whose value represents the markable *m*;
- attributes whose values represent the components of *s*.

3. For each type of link structure define an XML element with three attributes, whose values refer to the representations of the linked entity structures and to the relation that links them.

Bunt (2011) proposes to provide a semantics based on Discourse Representation Theory (DRT, Kamp & Reyle, 1993). The use of Discourse Representation Structure (DRSs) has an advantage over the use of first-order logic, with which DRSs are formally equivalent, since DRSs were designed to facilitate incremental construction. This can be exploited when constructing DRSs systematically from the components of an annotation representation.

The CASCADES approach defines a semantics for abstract annotation structures. Such a semantics can exploit the fact that entity structures and link structures are n-tuples of semantic concepts, the significance of an element in being encoded by its position. Bunt (2014) shows how annotation structures can be translated into DRSs in a compositional way, combining the translations of the component entity structures and link structures.

3.3 Steps forward and feedback in the design process

While the procedures for making the CASCADES steps are helpful for defining well-founded annotation schemes, it would be unrealistic to think that annotation schemes can be designed simply through a linear sequence of steps, from conceptual analysis to the specification of a representation format. Realistic design processes require feedback loops.

Pustejovsky and colleagues have introduced the 'MAMA' cycle for developing an annotation scheme (see Moszkowicz, 2012; Pustejovsky and Stubbs, 2012) , which distinguishes four steps: (1) Model; (2) Annotate; (3) Evaluate; and (4) Revise. In step (1) an annotation scheme is constructed, which can subsequently be revised and improved by repeating the cycle <2,3,4,1> until the scheme is stable.

In the CASCADES method, feedback cycles can occur between each of the four design stages, as shown in Figure 1. The feedback cycle <5;4> is especially useful when combined with the 'inner cycle <3;6>, to form the iterative feedback loop <5;<3;6>*;4>. This feedback loop is central to the application of the CASCADES method for improving an existing representation format, for detecting and resolving semantic deficiencies, or for turning an existing format into an annotation scheme that meets the requirements of the ISO Linguistic Annotation Framework and the requirement of semantic adequacy. In practice, the design of semantic annotations mostly starts from an existing representation format. An abstract syntax (with a semantics) can then be constructed that fits the representations and meets the LAF requirements and the requirement of semantic adequacy by following the iterative feedback loop <5;<3;6>*;4 >, commencing with the reconstruction of an abstract syntax.

The CASCADES method has been used in this 'reverse engineering' mode in the development of ISO-TimeML (ISO 24617-1), starting from TimeML, and in preliminary studies for the definition of an ISO standard for discourse relation annotation starting from the annotations in the Penn Discourse Treebank (PDTB) (see Bunt, Prasad & Joshi, 2012). Ide et al. (2011) have 'reverse-engineered' an abstract syntax for the PDTB representation format with the aim of designing a GrAF representation (Ide & Suderman, 2001) for these annotations, and have shown that, even without specifying a semantics for this abstract syntax, this leads to significant improvements.

The CASCADES design steps and feedback loops integrate perfectly with the MAMA development cycles, as shown in Figure 1, viewing the CASCADES steps and feedback loops together as an implementation of the Model stage of the MAMA cycle, and the CASCADES feedback loops as an implementation of the Revise stage, to which the MAMA cycle adds the stages of 'Annotation' and 'Evaluation' in between the CASCADES stages of Concrete Syntax specification and Conceptual Analysis. This integration clarifies the relation between the Model and Revise stages in the MAMA cycle. Intuitively, revising an existing annotation scheme should involve some of the same activities as the Model stage; the CASCADES steps make this explicit, since the feedback loops for revising an annotation scheme are also part of the modelling stage.

3.4 Optional elements in an annotation scheme

The abstract - concrete syntax distinction opens up interesting possibilities for optional elements in annotations and their representations

In a given annotation task it may be relevant to take information into account which does not form part of the focus of the annotation scheme but which may be useful for performing the task. For example, in coreference annotation it is useful to identify the noun phrases that are potential antecedents of referential pronouns according to their grammatical number and their grammatical or natural gender, depending on which of these properties is relevant in the grammar of the language under consideration. It is therefore useful to annotate the number and gender of noun phrases and pronouns. This may now be supported by an annotation scheme which includes the representation of gender and number in the concrete syntax but does not include this information in the abstract syntax, and therefore does not deal with the semantics of number of gender annotations.

Another form of optionality is that the concrete syntax defines default values for certain attributes. For example, an attribute 'polarity', with possible values "positive" and "negative", can be assumed to have the value "positive" by default. Optional components of this kind do correspond to elements in the abstract syntax, and do have a semantics.

A third kind of optionality is when semantic information may take more or less elaborate forms. An example is the annotation of attribution and argument type for discourse relations. In an explorative study which applies the CASCADES method to re-engineer the annotation scheme of the Penn Discourse Treebank (see Bunt, Prasad & Joshi, 2012) the entity structures that annotate arguments of discourse relations are defined as follows: *"An Argument Entity Structure is a pair $\langle m,s \rangle$ consisting of a markable m and the semantic information s, which is either vacuous (i.e. the entity structure only identifies the markable corresponding to an argument of a discourse relation), or contains information about the attribution of the argument and/or specifies the type of the argument."* Allowing the semantic information in these entity structures to be vacuous is a way of saying that the semantic information does not have to include certain components. This form of optionality is useful for dealing with information which is not always applicable or is irrelevant in certain cases.

3.5 Theory, practice and evaluation of annotation schemes

The CASCADES method of designing an annotation scheme can be viewed as a *theory* of annotation scheme design. Two ideas are central to this theory:

- annotations mean something; they are not just labels or XML strings that can mean whatever someone would like them to mean;[1]
- the choice of particular tag names and tag structures is of secondary importance; of primary importance is the determination of concepts and conceptual structures which the annotation scheme allows to be represented.

The CASCADES theory thus offers a way of evaluating the 'soundness' of an annotation scheme, namely the extent to which its representations are complete and unambiguous. Extended with the MAMA steps of Annotation and Evaluation, it moreover offers the steps for evaluating the empirical validity of an annotation scheme and for combining the feedback from an empirical evaluation with that of revising the annotation scheme in a theoretically sound way.

For practical purposes, if a certain annotation task calls for terminological and conceptual deviations from an existing annotation scheme, it may be sufficient to check that there is a mapping between the two sets of terms and between the respective representation structures. If a conceptual deviation is in fact a conceptual *extension*, the of course such mapping will fully work in one direction only.

4 Overlaps between annotation schemes

4.1 Spatial and temporal relations as semantic roles

The annotation schemes of ISO-TimeML (ISO 24617-1) and ISOspace (ISO 24617-7) include relations between events and their place and time of occurrence, as well as relations between temporal and spatial entities. The annotation scheme of SemAF-SR (ISO 24617-4) views semantic roles as relations between events and their participants, including spatial and temporal participants.

SemAF-SR defines the following eight semantic roles of a spatial or temporal character: (1) Location; (2) Initial-location; (3) Final-location; (4) Path; (5) Distance; (6) Duration; (7) Initial-time; and (8) Final-time. These concepts also occur in ISOspace or in ISO-TimeML, sometimes using exactly the same terms. For example, ISOspace defines a 'path' as a 'series of locations', like a road or a river, which can be used to get from one location to another. ISOspace is inconsistent in this respect with SemAF-SR, which defines Path as *Intermediate location or trajectory between two locations, or in a designated space, where an event occurs*, and thus views a path as inherently related to an event. So whereas 'path' is a spatial object in ISOspace, it is a relational notion in SemAF-SR.

ISOspace also defines 'event-path' as the dynamic notion of a trajectory followed in a motion, which is in essence the same concept as the semantic role Path in SemAF-SR. There is, on the other hand a difference between the way ISOspace views an event-path and the way SemAF-SR views a Path role, since the latter is a *relation* whereas the ISOspace notion is a *spatial object*.

A general question is whether all the distinctions among spatial and temporal relations that are made in ISOspace and ISO-TimeML should be reflected in distinctions between semantic roles in SemAF-SR. For example, ISOspace uses the attribute 'goalReached', with possible values "true", "false" and 'uncertain", in order to distinguish between cases like *John arrived in Boston*, where John reached his destination, from *John left for Boston*, where we don't know if he did. SemAF-SR so far has no provisions for making this distinction.

[1] An exception is the case of an instance of the second kind of optionality, discussed in Section 3.4, which does not have a semantics.

4.2 Events

Events take central stage both in ISO-TimeML and in ISOspace. For the sake of consistency, ISOspace inherits the typology of events defined in ISO-TimeML. On the other hand, ISOspace makes a basic distinction between motion events and non-motion events that cuts through the ISO-TimeML typology; whether this can lead to consistency problems needs to be investigated. Events are also of central importance in SemAF-SR, which views semantic roles as relations between events and their participants, but does not assume any particular typology of events.

The ISOspace distinction between motion events and non-motion events does seem relevant for semantic role assigment, since only motion verbs have spatial entities in roles like Initial Location, Path, and Final Location. Motion verbs used in a negative sentence, such as *John did not leave home* seem to require a different spatial role for characterizing the relation between *leave* and *home*, which is not available in ISO-SR. The same is true for *John stayed at home*.

4.3 Discourse relations in dialogue

The study of semantic relations in discourse is very much focused on the intersentential relations that lend coherence to a text; however, these relations may occur also in dialogue, not only within but also between speaker turns (see e.g. Tonelli et al., 2010; Petukhova et al., 2011; Lascarides & Asher, 2007). The ISO 24617-2 annotation scheme for dialogue act annotation therefore includes the concept of a 'rhetorical relation', however, it leaves open which specific relations may be used in dialogue annotation, recommending annotators to use the relations defined in the forthcoming standard ISO 24617-8 This is a good example of how the annotation schemes of different SemAF parts can be combined.

Utterances in dialogue may also be related by other semantic relations than those that are found in written text. The ISO dialogue act annotation scheme defines two other relations: (1) 'feedback dependence', which occurs when a dialogue act provides or elicits feedback about the success of processing (recognizing, understanding, or accepting) one or more previous dialogue acts – the 'scope' of the feedback act; and (2) 'functional dependence', for dialogue acts that due to their communicative function depend for their semantic content on a preceding dialogue act, such as an answer being dependent on a question. These relations are not present in any existing annotation scheme for discourse relations, presumably because of their focus on written discourse. The ISO annotation scheme for discourse relations inherits these relations from the ISO-24617-2 scheme.

5 Ubiquitous semantic phenomena

5.1 Quantification

Quantification phenomena arise whenever a predicate is applied to one or more sets of individuals, as in *Three men moved both pianos*. Quantification has been studied extensively, but not so much in relation to events, times and places. Still, in principle any relation between two sets of entities is quantified, as are the relations between events and temporal entities, for instance by means of temporal quantifiers such as *always, sometimes, every Monday*. For this reason, ISO-TimeML has some provisions for time-related quantification. The attribute 'quant' has been introduced for this purpose as one of the attributes of temporal entities.

Quantification cannot be analysed satisfactorily by means of attributes of temporal entities, however, since quantification phenomena are not properties of the entities participating in a predication, but are aspects of relations, as the following example illustrates, where three men are involved collectively in moving a piano and individually in drinking a beer.

(1) The three men had a beer before moving the piano.

An analysis of quantification in terms of feature structures has been proposed by Bunt (2005; 2013b) which can be the basis for annotating quantification in such a way that components of annotation structures correspond to the linguistic expression of quantification. This supports a semantic interpretation that can be combined with a compositional semantics of noun phrases, which is useful since many of the features of quantifications are expressed syntactically in noun phrases. The semantic adequacy of the proposal is demonstrated by a systematic translation of annotation structures into discourse representation structures.

5.2 Quantities and measures

Duration, length, volume, weight, price, and many other ways of measuring quantities of something are linguistically expressed by means of a unit of measurement plus a numerical indication, such as *one and a half hour, 90 minutes, just over two kilos*. Semantically, a measure is an equivalence class formed by pairs $\langle n,u\rangle$ where n is a numerical predicate and u is a unit (Bunt, 1985). Given the relations between the units in a particular system of units, like 1 hour = 60 minutes, any of the equivalent pairs can serve as a representative of the class. Units can be complex, like kilowatt-hour or meter per second. Formally, a unit is either a basic unit or a triple $\langle u_1,u_2,Q\rangle$ where $Q = \times$ (multiplication) or $Q = /$ (division) and u_1 and u_2 are (possibly complex) units.

The abstract syntax of annotations for quantities can be defined by introducing pairs $\langle n,u\rangle$, where u is either an elementary unit or a triple, as indicated above. A corresponding XML-based concrete syntax uses an element ‘amount’ with attribute-value pairs for the numerical part and the unit part, as in the following representation of *three miles*:

```
(2) <amount xml:id=”a1” target=”#m1” num=”3” unit=”mile”/>
```

ISOspace includes amounts of space for measuring distances; ISO-TimeML includes amounts of time for measuring durations. In both cases, only elementary units are considered; the above approach can be used to generalize this for units of velocity, for example, as illustrated in the following representation of *sixty miles per hour*:

```
(3) <amount xml:id=”am1” target=”#m1” num=”60” unit=”#u1”/>
    <unit xml:id=”u1” target=”#m2” unit1=”mile” unit2=”hour” operation=“division”/>
```

Amount expressions involving comparisons, as in *We walked more than five miles*, may be treated as involving an existential quantification over locations, as: *There is an amount of space greater than 5 miles that we walked*:

```
(4) <event xml:id=“e1” target=“#m2” pred=“walk”/>
    <entity xml:id=“x1” target=“#m1”/>
    <srLxink event=“#e1” participant=“#x1” roleType=“agent”/>
    <amount xml:id=“d1” target=“#m3”/>
    <amount xml_id=“d2” target=“#m4” num=“5” unit=“mile”/>
    <relation arg1=“#d1” arg2=“#d2” relType=“greaterThen”/>
    <srLink event=“#e1” participant=“#d1” roleType=“distance”/>
```

5.3 Negation, modality, factuality, and attribution

Negation, modality, factuality and attribution are different but related aspects of the factual content of an utterance or a text. Consider the following example from the Penn Discourse Treebank:

(5) “The public is buying the market when in reality there is plenty of grain to be shipped”, said Bill Biederman, Allendale Inc. director.

Even though Biedermann says *“in reality”*, it would be incorrect to conclude from this text that there is plenty of grain to be shipped. The source to which a statement is attributed is crucial to take into account: if the Wall Street Journal would report directly(rather than quote somebody) that there is plenty of grain to be shipped, then it would probably be more justified to draw this conclusion.

Negations evidently also have a strong influence on which information can be extracted from a text. ISO-TimeML makes use of an attribute ‘polarity’, with possible values “positive” and ‘negative”, as one of the attributes of an event. Positive and negative are just two extremes or a scale of possibilities, however. Modalities as expressed by *probably, maybe* and *surely*, as well as the attribution of the claim to a certain source, all have an influence on the possibilities of extracting factual information from a text. Expressions of modality have been studied by Karttunen (1971; 2012). The factuality of statements about events has been studied by Sauri (2008) and annotated in the FactBank (Sauri & Pustejovsky, 2009). See also Morante & Daelemans (2011) and Pareti (2012; 2015) for work on the annotation of negation, modality and attribution.

5.4 Modification and qualification

5.4.1 Nominal modification

The modification of nominal expressions, e.g. by adjectives, prepositional phrases, or relative clauses, gives rise to many of the same issues as the expression of quantification; in particular, issues of scope and distribution arise in much the same way. Consider the following example of a text next to a box of bell peppers:

(6) Bell peppers for fifty pesos

This is ambiguous as to whether *for fifty pesos* applies to the individual bell peppers in the box or to the whole lot (collective reading). Adjectives and prepositional phrases, used as modifiers, can be viewed as one-place predicates, whose application to a set of arguments gives rise to quantificational issues, as noted in Section 4.1. The ambiguity of (6) is due to an ambiguity in the way the predicate is applied to its arguments. This suggests an approach to the annotation of modification in terms of annotation structures that consist of a predicate, a set of arguments, and the type of relation between them (such as the ‘restrictive modifier’ relation type). Such a structure allows the distribution of the modification to be a property of the relation type. In an XML representation, such an annotation could look as follows:

(7) a. heavy boxes
b. <entity id=“x1” target=“#m2” signature=“set”/>
<property id=“p1” target=“#m1 />
<modLink id=“m1” head=“#x1” modifier=“#p1” relType=“restrModifier” distribution=“individual”/>

Modification by means of relative clauses gives rise to all the issues that are known to arise in quantifications, as can be seen by transforming a quantified sentence into a modified noun phrase – see the sentence pairs (8) and (9):

(8) a. That crane moved thirty big pipes.
b. Thirty big pipes moved by that crane.

(9) a. Two students read the six papers.
b. The six papers read by two students.

Sentence (8b) has the same ambiguity as (8a) in the distributive aspect of the quantification, i.e. whether the crane moved the pipes one by one or all in one go. Similarly, (9b) has the same ambiguity as (9a) with respect to the scopes of the quantifications.

In view of the analogy between modification and quantification, it seems commendable to develop an approach to the annotation of modification integrated with that of quantification.

5.4.2 Qualification

The notion of a 'qualifier' has been introduced in ISO 24617-2 in order to make more subtle distinctions between dialogue act types then would be possible by just using the set of communicative functions defined in the annotation scheme. Although this set is fairly comprehensive, it is not sufficient for dealing with subtle differences like those in (10).

(10) A: Would you like to have some coffee?
a. B: Only if you have it ready.
b. B: Maybe; how much time do we have?
c. B: Maybe later
d. B: Coffee, wonderful!
e. B: Coffee? At midnight??

These examples show the conditional acceptance of an offer (a); an uncertain acceptance (b); an uncertain rejection (c); an acceptance with pleasure (d); and a rejection with surprise (e). In order to take such modalities into account, which can occur with every dialogue act that has a responsive character, Petukhova and Bunt (2010) proposed the use of qualifiers for certainty, conditionality, and sentiment. These are optional elements in the abstract syntax of dialogue act annotations, which means that they do not have to be used, but if they are, then they have a semantic interpretation.

Qualifiers may be an interesting addition in other SemAF-parts as well, such as in the annotation of semantic roles. For example, the Agent role is defined in SemAF-SR as the involvement of *a participant who acts intentionally or consciously*. So when annotating a sentence like *Peter dropped his plate on the kitchen floor* the question arises whether this was done intentionally or not. If it was, then this could be made explicit by means of a intentionality qualifier. Similarly for discourse relation annotation, in examples like *but unexpectedly, but perhaps*, or *but fortunately* in order to annotate not just a contrastive relation but also the speaker's certainty or sentiment .about what happened, contrary to expectation.

6 Conclusions and future work

Efforts that aim to improve the interoperability of semantically annotated resources, taking place under the umbrella of the ISO Semantic Annotation Framework (SemAF), have as their most important characteristic the use of an abstract syntax underlying concrete annotation representations and the specification of a semantics of annotation structures. The importance of this approach is that it ensures that any two representation formats which have 'complete' expressive power and are 'unambiguous', are semantically interoperable: representations in one format can be converted to those in the other. We have also shown that this approach opens interesting alternative possibilities for the use of optional elements in semantic annotations.

In this paper we have identified various semantic phenomena that cut across SemAF annotation schemes for semantic roles, for time and space, for events, for discourse relations and for dialogue acts; for some of these phenomena (such as quantification and nominal modification) we have indicated promising directions for how they may be dealt with. Together with the analysis given in this paper of the overlaps between SemAF annotation schemes, this contributes to an agenda for future work that aims at the establishment of powerful annotation schemes for interoperable semantic annotation.

References

Asher, N. (1993). *Reference to abstract objects in discourse.* Dordrecht: Kluwer.

Bunt, H. (1985). *Mass Terms and Model-Theoretic Semantics.* Cambridge University Press.

Bunt, H. (2005). Quantification and Modification Represented as Feature Structures. In *Proceedings 6th International Workshop on Computational Semantics (IWCS-6)*, Tilburg, Netherlands, pp. 54–65.

Bunt, H. (2010). A methodology for designing semantic annotation languages exploring semantic-syntactic ISO-morphisms. In *Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL 2010)*, Hong Kong: City University, pp. 29–46.

Bunt, H. (2013). A methodology for designing semantic annotations. TiCC Technical Report TR 2013-001, Tilburg University.

Bunt, H. (2013b). *The annotation of quantification and its interpretation*. University of Potsdam.

Bunt, H. (2014). Annotations that effectively contribute to semantic interpretation. In H. Bunt, J. Bos, and S. Pulman (Eds.), *Computing Meaning, Vol. 4*, pp. 49–70. Dordrecht: Springer.

Bunt, H., R. Prasad, and A. Joshi (2012). First steps toward an ISO standard for the annotation of discourse relations. In *Proceedings 7th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-7), Istanbul*, Paris: ELRA, pp. 60–69.

Bunt, H. and J. Pustejovsky (2010). Annotating temporal and event quantification. In *Proceedings 5th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-5)*, Hong Kong: City University, pp. 15–22.

Bunt, H. and L. Romary (2002). Towards Multimodal Content Representation. In K. S. Choi (Ed.), *Proceedings of LREC 2002, Workshop on International Standards of Terminology and Language Resources Management*, Las Palmas, pp. 54–60. Paris: ELRA.

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.

Hovy, E. and E. Maier (1992). *Parsimonious or profligate: how many and which discourse structure relations? ISI research report*. Marina del Rey: Information Sciences Institute, University of Southern California.

Ide, N. and H. Bunt (2010). Anatomy of annotation schemes: Mapping to GrAF. In *Proceedings 4th Linguistic Annotation Workshop (LAW IV)*, Uppsala.

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

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

ISO (2012b). *ISO 24617-2:2012, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 2: Dialogue acts*. Geneva: ISO.

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

ISO (2014b). *ISO 24617-7: 2014, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 7: Spatial information*. Geneva: ISO.

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

ISO (2015b). *ISO CD 24617-8:2015, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 8: Semantic relations in discourse*. Geneva: ISO.

Kamp, H. and U. Reyle (1993). *From Discourse to Logic*. Dordrecht: Kluwer Academic Publishers.

Karttunen, L. (1971). Implicative verbs. *Language 47*, 340–358.

Karttunen, L. (2012). Simple and phrasal implicatives. In *Proceedings of SEM 2012*, Montreal, pp. 124–131. Association for Computational Linguistics.

Lascarides, A. and N. Asher (2007). Segmented discourse representation theory: Dynamic semantics with discoure structure. In H. Bunt and R. Muskens (Eds.), *Computing Meaning, Vol. 3*, pp. 87–124. Dordrecht: Springer.

Morante, R. and W. Daelemans (2011). Annotating modality and negation for a machine learning evaluation. In *CLEF 2011 Labs and Workshop, Notebook Papers.*

Pareti, S. (2012). The independent encoding of attribution relations. In *Proceedings 8th Joint ACL ? ISO Workshop on Interoperable Semantic Annotation (ISA-8, "ISA in Pisa")*, Pisa, pp. 48–55.

Pareti, S. (2015). Annotating attribution relations across languages and genres. In *Proceedings 11th Joint ACL ? ISO Workshop on Interoperable Semantic Annotation (ISA-11)*, London.

Petukhova, V. and H. Bunt (2010). Introducing communicative function qualifiers. In *Proceedings Second International Conference on Global Interoperability for Language resources (ICGL-2), Hong Kong*, pp. 132 – 132.

Petukhova, V., L. Prévot, and H. Bunt (2011). Discourse relations in dialogue. In *Proceedings 6th Joint ACL - ISO Workshop on Interoperable Semantic Annotation (ISA-6)*, Oxford, pp. 18–27.

Prasad, R. and H. Bunt (2015). Semantic relations in discourse: The current state of ISO 24617-8. In *Proceedings 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11)*, pp. 80–92.

Prasad, R., N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber (2008). The Penn Discourse TreeBank 2.0. In *Proceedings 6th International Conference on Language Resources and Systems (LREC 2008)*, Marrakech.

Pustejovsky, J. and J. Moszkowics (2010). The role of model testing in standards development: The case of iso-space. In *Proceedings 8th International Conference on Language Resources and Evaluation(LREC 2012, Istanbul*. ELDA, Paris.

Pustejovsky, J. and A. Stubbs (2012). *Natural Language Annotation for Machine Learning*. O'Reilly.

Sauri, R. (2008). *A Factuality Profiler for Eventualities in Text.* Ph.D. Thesis, Brandeis University.

Sauri, R. and J.Pustejovsky (2009). Factbank: a corpus annotated with event factuality. *Journal of Language Resources and Evaluation 43 (3)*, 227–268.

Tonelli, S., G. Riccardi, R. Prasad, and A. Joshi (2010). Annotation of discourse relations for conversational spoken dialogs. In *Proceedings 7th International Conference on Language Resources and Systems (LREC 2010)*, Genoa.