QuantML specification

1. Overview

Following the methodology described in ISO standard 24617-6 (Principles of semantic annotation),[1] the specification of the QuantML markup language consists of four parts:

1. The metamodel, providing an overview of the concepts that may occur in annotations, and the relations between them.
2. The abstract syntax, including an inventory of the concepts from which annotations are built up and of the possible ways of combining them in set-theoretical constructs like pairs and triples, called 'annotation structures'.
3. A concrete syntax, defining a representation format for annotation structures.
4. The semantics, defining an interpretation of annotation structures.

Abstract and concrete syntax are related through encoding and decoding functions, and abstract syntax and semantics through an interpretation function. (See further the Annotation Guidelines.) Subsequent subsections describe these four components.

2. Metamodel

The aspects of quantification discussed in the preceding subclauses provide the building blocks for the QuantML annotation scheme. Figure 1 displays the QuantML metamodel, showing the concepts that go into annotations and how they are related. The internal structure of some concepts is not shown, such as the possible ways of modifying an NP head, or the structure of domain size specifications.

The backbone of the metamodel is formed by sets of events, sets of participants, and the relations between them. Event sets are specified by the characteristic predicate of a domain (typically as expressed by a verb), and in the case of repetitive events an additional numerical predicate. (The annotation of other properties of events is considered in ISO 24617-1, Time and events). Sets of participants in events are characterized by the following properties:

1. the source domain from which the participants are drawn;
2. the determinacy, indicating whether the reference domain of the quantification is a proper part of the source domain or coincides with it;
3. the individuation of the reference domain (individual objects, possibly also their parts, or quantities of masses);
4. the quantitative involvement of the reference domain, determining the size of the participant set;
5. the size of the reference domain, or of groups, subsets, or parts of the reference domain;
6. the relative scopes of participants in different semantic roles.

Participation relations are characterized by five properties: semantic role, polarity, distributivity, exhaustiveness, and relative scope of events and participants. Events and their participants are linguistically expressed; markables identify these expressions in the primary data. Participation relations and scope relations are not verbalized, and hence not associated with a markable.

Different from what Fig. 1 may suggest, events can also play a role as participants in other events. This is the case not only for NPs with an event-denoting head noun, such as *"meeting", "concert",* or *"sunrise",* but also for the use of a gerund or an infinitival form as a verbal complement, as in *"I think (that) Bill is ill".*

[1] This ISO standard is based on Bunt (2015). All references can be found in the Bibliography part of this site.

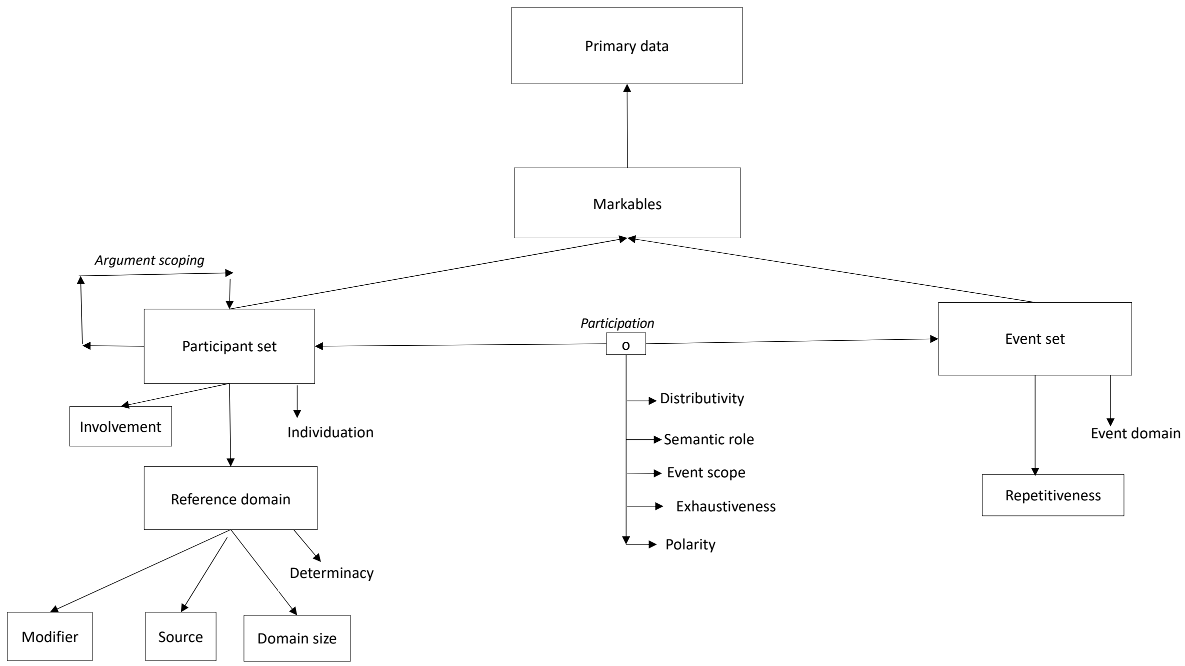

Figure 1: QuantML metamodel for the annotation of quantification

3. Abstract syntax

3.1 General

The annotation structures defined by the abstract syntax are *n*-tuples of elements which are either basic concepts, taken from a store of basic concepts called the 'conceptual inventory', or 'annotation structures'. The elements of the conceptual inventory are instances of the concepts that make up the metamodel and form the basic elements in annotation structures. Two types of structure are distinguished: *entity structures* and *link structures.* An entity structure contains semantic information about a segment of primary data and is formally a pair $\langle m, s\rangle$ consisting of a markable and certain semantic information. A link structure contains information about the way two or more segments of primary data are semantically related. Participant sets and event sets are annotated by means of entity structures; participation relations are characterised by link structures. QuantML additionally includes certain structures for expressing proportional and numerical aspects of quantifications and for quantification in relation to negation and restrictive as well as non-restrictive modification.

3.2 Conceptual inventory

The conceptual inventory of QuantML provides the following concepts:

1. Predicates:
    a. Predicates that correspond to (occurrences of) lexical items of the language of the primary data, notably nouns, verbs, adjectives, and prepositions. Such predicates are designated by canonical forms of lexical items, such as verb stems. These predicates form an open class, the content of which depends on the language of the primary data, the subject matter of the annotated material, and the use of lexical resources like WordNet and VerbNet.
    b. The numerical relations 'less-than', 'equal', and 'greater-than-or-equal'.
    c. Non-numerical quantitative predicates corresponding to determiners such as *"a few", "a little", "several", "many"* in English, or *"beaucoup", "quelques", "plusieurs",* in French. This

is an open class that is language-dependent, but always contains the language-independent predicates . *“some”, “all”, “no”, and “most”.*

   d. Predicates corresponding to proportional determiners such as ‘most’, ‘all’, ‘half’, ‘two-thirds’.
   e. Semantic role predicates: ‘Agent’, ‘Theme’, ‘Pivot’, and others defined in ISO 24617-4.[2]
   f. Predicates that express a modality. This is a language-dependent open class.
   g. The possessive relation ‘Poss’.
2. The non-negative real numbers.
3. Dimensions, such as ‘weight’, ‘volume’, ‘length’, and ‘surface’, and units of measurement, which may be *basic*, like ‘kilo’, ‘liter’, ‘mile’, which are ‘basic’ units, or *derived,* like ‘square meter’ or ‘meter per second’.[3]
4. Concepts that function as parameters in the specification of the semantic interpretation function:
   a. ‘determinate’, ‘indeterminate’ for specifying a determinacy;
   b. ‘collective’, ‘individual’, ‘sampled’, and ‘unspecific’ for specifying distributivity;
   c. ‘count’, ‘mass’ and ‘count+parts’ for specifying individuation;
   d. ‘exhaustive’ and ‘non-exhaustive’ for specifying exhaustivity;
   e. ‘wide’ and ‘narrow’ for specifying event scope;
   f. ‘wider’, ‘dual’, and ‘equal’, for specifying relative scopes of participants;
   g. ‘positive’, ‘wide negative’, and ‘narrow negative’ for specifying a polarity and its scope;
   h. ‘maximal’, ‘medium’, and ‘minimal’ for specifying the scope of a modal operator;
   i. ‘inverse’ and ‘linear’ for specifying whether scope inversion occurs;

3.3 Entity structures

Participant sets are characterised by entity structures of the form (1), which specify a reference domain (‘ε_D‘), its individuation ‘v’, its involvement (‘q’), and optionally a domain size (‘N’) and some non-restrictive modifiers (‘w’).

(1) $\varepsilon_p = \langle m, \langle \varepsilon_D, v, q, [N], [w] \rangle\rangle$

The reference domain is described by an embedded entity structure of the form (2), specifying a domain ‘D’, a list of restrictions ‘r’ (optional), and a determinacy (‘dt’).

(2) $\varepsilon_D = \langle m_D, \langle D, [r], dt \rangle\rangle$

A domain specification (‘D’) is either (a) a single predicate that characterises a source domain, typically as described by an NP head identified by the markable m_D, or (b) a pair formed by a domain specification and a list of restrictions, or (c) a sequence of domain specifications (for conjunctive NP heads). Restrictions are expressed by modifiers, assumed to be intersective. Five types of modifier structure are distinguished, depending on whether they are expressed by adjectives, nouns, possessive phrases, preposition phrases, or relative clauses. The abstract syntax of these structures is as follows.

1. Adjectival modification structures are pairs: ε_{AD} = ⟨markable, ⟨predicate, distributivity⟩⟩.
2. Noun-noun modification is always distributive. A modifying noun can itself be modified by an adjective, as in *“chemical waste dump”*, or by another noun, as in *“corona virus infection”*. The possible complexities of this kind are of little interest for the annotation of quantification, and are not considered in QuantML.
3. PP modification: the semantic information in the PP consists of the relation expressed by the preposition and the information of the embedded NP. For PP-modification both the distributivity is relevant and the scoping (inverse or ‘*linear’* linking). An entity structure ε_{PP} for PP modification therefore contains a quadruple ⟨relation, participant structure, distributivity, linking⟩.

[2] QuantML does not prescribe the use of any particular set of semantic roles. Role sets other than ISO 24617-4 (see e.g. Bonial et al., 2011) can be used as plug-ins (see Bunt, 2019).
[3] See ISO 80000-1:2009, 3 Terms and Definitions (3.10 Base unit, 3.11 Derived unit), and ISO 2417-11:2021, Measurable quantitative information.

4. Possessive modification: possessive modifiers can take a variety of forms, including possessive pronouns, genitive constructions (such as *"every student's essays"*), and PPs with a possessive preposition. Semantically, possessives can be analysed in terms of a possessor, a possessee, and a binary relation 'Poss' between them (Peters and Westerståhl, 2013). In QuantML they are treated like PPs, except that the relation between the discourse referents of the modified NP and the possessor(s) is invariably the Poss relation. An entity structure for possessive modification thus contains a quadruple ⟨Poss, participant structure, distributivity, linking⟩.
5. Relative clause modification: the semantic information to be captured in annotations consists of the specification of the events and participants as described in the relative clause, plus the semantic role that the participants indicated by the head play in the RC's events, the distributivity of the modification, and whether inverse linking occurs. The entity structure for a relative clause thus contains a quadruple ⟨semantic role, clause structure, distributivity, linking⟩.

The domain involvement component q in (1) is either a proportional predicate, such as 'most', 'at least half' or 'two-thirds', or a size specification, i.e. a numerical or measure predicate, like 'twelve', 'or 'no more than 300 grams'. Non-numerical quantitative predicates such as 'some', 'many', 'all', 'several', 'no' or 'much' can be viewed either way.

Entity structures for sets of events contain a predicate that characterizes an event domain, analogous to the source domain of a participant set, optionally a numerical predicate indicating the repetitiveness of an event (default: at least once, and a specification of being generic (default: specific)..

(3) ε_{EV} = ⟨m, ⟨event domain, repetitiveness, genericity⟩⟩

The QuantML abstract syntax defines the following types of entity structure ⟨m, s⟩:

1. Participant structures: s = ⟨ε_D, v, q, [N], [w]⟩, as in (12).
2. Domain specification structures: s = ⟨D, [r], dt⟩, as in (13).
3. Involvement specifications:
   a. Numerical: s is a pair consisting of a numerical relation and a non-negative real number or a measure structure;
   b. Relative: s is a proportional or non-numerical, approximative quantitative predicate.
4. Event structures: s = ⟨event domain, repetitiveness⟩;
5. Clause structures: s = ⟨event structure, list of entity structures, list of participation or predication links, list of scope relations⟩.
6. Modifier structures:
   a. Adjectival structure: s = ⟨property, distributivity⟩;
   b. NN structure: s = ⟨property⟩;
   c. PP structure: s = ⟨relation, participant set, distributivity, linking⟩;
   d. RC structure: s = ⟨ semantic role, clause structure, distributivity, linking⟩;
   e. Possessive structure: s = ⟨Poss, participant set, distributivity, linking⟩.
8. Modality structures: s = ⟨modal predicate, scope⟩
9. Measure structures: s is a pair consisting of a real number and a (basic or derived) unit.

3.4 Link structures

The abstract syntax defines link structures for (1) participation in an event, (2) participation in an adjectival predication, and (3) the relative scoping of participants. Participation structures (4a) connect participants to events specifying their semantic role, the distributivity, whether the quantification over events has wide or narrow scope, the polarity, the exhaustiveness of the participation and possibly a modality. Predication link structures (4b) do the same for copular constructions. Scope link structures (4c) indicate a scope relation between two participant entity structures.

(4) a. L_P = ⟨ε_{EV}, ε_p, semantic role, distributivity, event scope, exhaustiveness, polarity, [modality] ⟩

b. L_C = ⟨ε_p, ε_{EV}, predicate, distributivity, exhaustiveness, polarity, [modality] ⟩

c. L_{SC} = ⟨ε_{p1}, ε_{p2}, scope relation⟩

Formally**,** the following link structures are defined:

1. Participation links: A septet or octet ⟨event structure, participant structure, semantic role, distributivity, exhaustiveness, event scope, polarity, modality ⟩, with optional modality.
2. Predication links: A sextet or septet ⟨participant structure, event structure, adjective, distributivity, exhaustiveness, polarity, modality ⟩, with optional modality.
3. Scope relation links: ⟨participant entity structure, participant entity structure, scope relation⟩.

4. Concrete XML-based syntax

A concrete syntax is specified in the form of an XML representation of annotation structures. These structures are built up from atomic attribute values, which are XML constants that name elements of the conceptual inventory of the abstract syntax, such as 'det' and 'indet'. For each type of entity structure of the abstract syntax, an XML element is defined which has an attribute @xml:id, whose value is a unique name for the information in the element, and an attribute @target, whose value anchors the annotation in the source data through markables. For convenience, the same predicate names are used as in the abstract syntax. NP heads may be complex due to the combination of modifiers and conjunctions, as in *"precious(ancient(Chinese figurines and drawings) and Thai sculptures)*"; the element <complexDomain> is used for representing such structures.

The following XML elements are defined for representing the entity structures of the abstract syntax:

1. <entity>, for representing participant structures:
   @refDomain – value type IDREF, @individuation – (count | mass | count+parts),
   @involvement – IDREF, optionally: @size – IDREF and @qualifiers - IDREF;
2. <event>: @pred – CDATA, optionally @rep – IDREFS and @genericity – (generic | specific);
3. <refDomain>: @components - IDREFS, @restrictions - IDREFS, @determinacy – (det | indet);
4. <sourceDomain>: @pred - CDATA;
5. <domain>: @components - IDREFS, @restrictions - IDREFS;
6. <adjMod>: @pred – CDATA, @distr – (individual | collective | sampled | unspecific);
7. <nnMod>: @pred – CDATA, optionally @restrictions; - IDREFS
8. <ppMod>: @pRel - CDATA, @pEntity - IDREF, @distr – (individual | collective | sampled | unspecific), @linking – (linear | inverted);
9. <possMod>: @possessor - IDREF, @distr – (individual | collective | sampled | unspecific), @linking – (linear | inverted);
10. <relClause>: @semRole - CDATA, @distr – (individual | collective | sampled | unspecific), @linking – (linear | inverted);
11. <cardinality>: @numRel - (greater_than-or_equal | equal | less-than), @number – CDATA;
12. <relativeSize>: @pred – CDATA;
13. <measure>: @numRel - (greater_than-or_equal | equal | less-than), @number - CDATA, @unit - CDATA.

XML elements for the three types of link structure defined by the abstract syntax:

1. <participation>: @event - IDREF, @participant - IDREF, @semRole - CDATA, @distr – (individual | collective | sampled | unspecific), @evScope: (wide | narrow), and optionally @exhaustiveness – (exhaustive | non-exhaustive), @polarity – (positive | wide-negative | narrow-negative), and @modality - CDATA. The following values are default and may be suppressed: exhaustiveness = "non-exhaustive", abbreviated "nex", event scope = "narrow", polarity = "positive".
2. <predication>: @participant - IDREF, @event – IDREF, @predicate – CDATA, @distr – (individual | collective | sampled), and optionally @exhaustiveness – (exhaustive | non-exhaustive), @polarity – (positive | wide-negative | narrow-negative), and @modality - CDATA. Default values, which may be suppressed: exhaustiveness = "non-exhaustive", polarity = "positive".
3. <scoping>: @arg1, @arg2, both with values of type IDREF, @scopeRel – (wider | dual | equal).

Some attributes, such as @pred and @pRel, have values derived from nouns, verbs, adjectives or prepositions in the annotated data. These values are represented by canonical forms of the lexical items of the language of the data such as verb stems and singular forms of nouns. (For convenience the same forms are used in the abstract and in the concrete syntax.) Since every use is associated with a markable, it is in principle possible to associate different word senses with different occurrences.

The use of this concrete syntax is exemplified in (5). Default values of attributes have been suppressed.

(5) The three men moved both pianos

Markables: m1 = The, m2 = The three men, m3 = The men, m4 = three, m5 = men, 6 = moved, m7 = both, m8 = both pianos, m9 = pianos

QuantML/XML:

```
<entity xml:id=”x1” target=”#m2” refDomain=”#x2” individuation=”count” involvement=“#q1” size=“#c1”/>
<refDomain xml:id=”x2” target=”#m3” components=”#x3” determinacy=”det”/>
<sourceDomain xml:id=”x3” target=”#m4” pred=”man”/>
<relativeSize xml:id=“q1” target=“#m1” pred=”all”/>
<cardinality xml:id=“c1” target=“#m4” numRel=“equal” number=“3”/>
<event xml:id=”e1” target=”#m6” pred=”move”/>
<entity xml:id=”x4” target=”#m8” refDomain=”#x5” individuation=”count” involvement=“#q2” size=“#c2”/>
<relativeSize xml:id=“q2” target=“#m7” pred=”all”/>
<cardinality xml:id=“c2” target=”#m7” numRel=“equal” number=“2”/>
<refDomain xml:id=”x5” target=”#m” components=”#x6” determinacy=”det”/>
<sourceDomain xml:id=”x6” target=”#m9” pred=”piano”/>
<participation event=”#e1” participant=”#x1” semRole=”agent” distr=”collective”/>
<participation event=”#e1” participant=”#x4” semRole=”theme” distr=”individual”/>
<scoping arg1=”#x1” arg2=”#x4” scopeRel=”wider”/>
```

5 Semantics

QuantML annotations have a compositional semantics, in the sense that the interpretation of an annotation structure is obtained by combining the interpretations of its entity structures and participation link structures, in a manner that is determined by its scope link structures. The specification of the QuantML semantics has been cast in the form of a compositional translation of annotation structures to discourse representation structures (DRSs). The DRSs used in this semantics reflect the combination of GQT with neo-Davidsonian event semantics, viewing natural language quantifiers as properties of sets of participants involved in sets of events. This form of semantics is convenient for combining annotations of quantification with other types of semantic information, using he ISO Semantic Annotation Framework (SemAF, ISO 24617), which also uses DRSs in some of its parts; otherwise, second-order logic would be equally suitable. The present sub-clause gives a brief outline of the semantics; a systematic specification is provided in part ‘QuantML semantics’ of this site.

According to the metamodel of Fig. 1, the main components of an annotation structure are the structures that describe participant sets, event sets, and the participation relations between them. Example (6) shows the XML-representations of these components for the quantifier expressed by the NP *‘Thirty-two Chinese students”* and the verb “*enrolled*”. The DRS for the NP introduces a discourse referent for the participant set[4], and includes conditions expressing that the participants are students and are Chinese, and that there are 32 of them; the DRS for the verb introduces a discourse referent E that refers to a set of ‘enroll’ events.

(6) a. $[\ X \mid x \in X \ \rightarrow\ [\ \text{student}(x), \text{Chinese}(x)\ ], |X| = 32\ ]$

b. $[\ E \mid e \in E \ \rightarrow\ \text{enroll}(e)\ ]$

[4] All discourse referents for sets of participants or events are required to be non-empty. Discourse referents for events have default repetitiveness of 1. The conditions $|X| \geq 1$ and \E| = 1 are therefore suppressed in all DRSs.

The DRS for the participation link introduces two discourse referents, one for a set of events and one for a set of participants, and relates these sets through the semantic role Agent. Since the event scope is narrow, the event set referent is within the scope of a quantification over the participant set:

(7) $[\ X \mid x \in X \ \rightarrow\ [\ E \mid e \in E \ \rightarrow\ \text{agent}(e, x)\ ]\ ]$

The DRSs of (6) and (6) are combined using the 'glue merge' operation, defined in the next section, with the result (8).

(62) $[\ X \mid |X| = 32,\ x \in X \ \rightarrow\ [\ \text{student}(x),\ \text{Chinese}(x),\ [\ E \mid e \in E \ \rightarrow\ [\ \text{enroll}(e),\ \text{agent}(e, x)\ ]]]]$

For a verb with multiple arguments, the interpretations of the link structures are combined in a way that reflects the relative scoping of the arguments, similar to the way (9) reflects an event scope. This is illustrated by the analysis of sentence (8) for the wide-scope reading of *"All the students".* Note that this NP is interpreted as a universal quantifier over the reference domain defined by the predicate 'student_0'.

(8) All the students read three papers.

(9) $[\ X \mid \text{student}_0(x) \rightarrow x \in X\ ,\ x \in X \rightarrow [\ \text{student}(x),\ [\ Y \mid y \in Y \rightarrow [\ \text{paper}(y),\ [\ E \mid e \in E\ \rightarrow$
$[\ \mid \text{read}(e),\ \text{agent}(e, x),\ \text{theme}(e,y)\ ]]]]\ ]$