The representation of QuantML annotations in UMR - an exploration

Harry Bunt
Tilburg University / Tilburg, The Netherlands
harry.bunt@tilburguniversity.edu

Kiyong Lee
Korea University / Seoul, Korea
ikiyong@gmail.com

Abstract

This paper explores the possibilities and the problems in using Unified Meaning Representations (UMRs) for representing annotations of quantification phenomena, according to the ISO standard scheme QuantML (ISO 24617-12:2025). We show that the semantic information in QuantML annotations can we expressed in UMR, provided that some powerful semantic concepts are introduced and a slightly more general approach is adopted for the representation of multiple scope relations. Conversion functions are defined that transform the XML-based representations of QuantML into UMR structures and vice versa. The consequences are discussed that can be drawn from this regarding the possible role of UMR and the semantics of UMR representations of quantification.

1 Introduction

Quantification is one of the most studied topics in semantics. Its complexity gives rise to a plethora of questions, conceptual, linguistic, logical, and computational. Montague (1971) used a higher-order intensional logic with categorial grammar to treat quantification in natural language. In contrast, Abstract Meaning Representation (AMR) and its extension, Uniform Meaning Representation (UMR) provide a first-order Neo-Davidsonian semantics which does not address many aspects of quantification, but which is attractive for its conceptual simplicity, especially when representing meaning in the form of a rooted graph. Intuitively, the nodes and edges of such a graph are similar to the entity and link structures of QuantML. For these reasons, and further motivated by the increasing popularity of AMR and UMR in natural language processing (see Lee et al., 2025), this paper explores the possibility of representing the rich QuanML annotations of quantification in the form of UMRs.

The organization of this paper is as follows. In Section 2 we consider some of the characteristic features of QuantML and UMR, in particular the 3-level architecture of QuantML with (a) a concrete syntax, which defines an XML-based representation format, (b) an abstract syntax which defines annotation structures in a format-independent way, using set-theoretic constructs, and (c) a semantics, which specifies semantic interpretations of the annotation structures of the abstract syntax. Section 3 discusses some of the fundamental concepts in the annotation of quantification. In Section 4 we examine the potential application of UMR expressions in the semantic annotation of quantification. We do this in two steps. First, we compare the annotation of a variety of quantification phenomena using (a) the XML-based reference format of QuantML, which we will refer to as QuantML/XML, and (b) a UMR-based representation format, which we will refer to as QuantML/UMR. This comparison provides insight into those aspects where the two forms seem little more than notational variants and those aspects for which the relation between the two is more complex. Second, we investigate the possibility of converting representations in QuantML to QuantML/UMR and vice versa. In the concluding Section 5 we discuss how QuanML/UMR could fit into the architecture of ISO SemAF annotation standards and we consider the semantics of Quant/UMR annotations in this context.

2 Background

2.1 QuantML

QuantML (ISO 24617-12:2025) is part 12 of the multi-part ISO standard *Semantic Annotation Framework (SemAF)* for semantic annotation. Following the Principles of semantic annotation (ISO 24617-6:2015), the parts of SemAF have the same 3-level architecture (see also Pustejovsky et al. 2017), consisting of:

1. An abstract syntax, which specifies the class of well-defined *annotation structures* as pairs,

Proceedings of the 21st Joint ACL - ISO Workshop on Interoperable Semantic Annotation (ISA-21), pages 21–30
September 24, 2025, ©2025 the authors

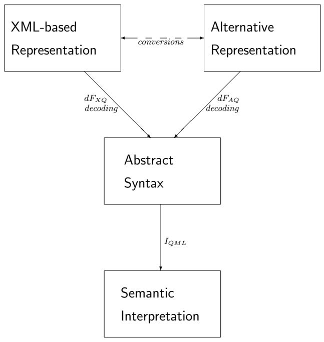

Figure 1: Architecture of SemAF parts.

triples, and other set-theoretical constructs containing quantification-related concepts. Annotation structures consist of *entity structures*, which contain information about a stretch of primary data, and *link structures*, which contain information relating two (or more) entity structures.

2. semantics, which specifies the meaning of the annotation structures defined by the abstract syntax. QuantML has an interpretation-by-translation semantics, which translates annotation structures to discourse representation structures (DRSs, (Kamp and Reyle, 1993)). The use of DRSs is mainly motivated by the fact that this formalism is also used in other SemAF parts.

3. A concrete syntax, which specifies a representation format for annotation structures. The QuantML definition includes an XML-based reference format, primarily motivated by the widespread use of XML in other standards.

The three levels are interrelated by encoding (eF), and interpretation functions; see Figure 1. Since the semantics is defined at the level of the abstract syntax, alternative representation formats may be used that share the same abstract syntax, as indicated in Figure 1, and are thus semantically equivalent. This adds to the interoperability of the annotation schema.

QuantML is semantically rooted in Davidsonian event semantics (Davidson (1967), Parsons (1990)) and in the theory of generalised quantifiers (GQT, Barwise and Cooper (1981), Cooper, 1983). Generalised quantifiers are not the logical counterparts of determiners, such as *"every"* and *"some"*, but of NPs like *"More than fifty students"* and *Three of the five men"*. In Davidsonian semantics, verbs are viewed as denoting events and their NP arguments as denoting participants in the events in certain semantic roles. Combining the two approaches, a sentence with quantified participants like *"More than fifty students protested"* has at least two readings, depending on whether it is taken to describe a single protest event with a set of more than fifty students as agents, or a set of protest events with individual students or smaller numbers of students as agents, the total number of participants involved in the events adding up to more than 50.

QuantML supports the annotation of quantified participation in events by taking into account the following categories of information:

(1)
1. quantification domain
2. determinacy (determine/indeterminate)
3. distributivity (individual/collective/unspecific)
4. individuation (count/mass)
5. involvement (absolute and proportional)
6. semantic role
7. exhaustivity
8. polarity
9. participant scope
10. event scope
11. repetitiveness
12. size of reference domain
13. restrictiveness of modifiers
14. linking of modifiers (inverse or linear)
15. modality (e.g. epistemic)
16. genericity (generic or specific).

The categories 1 - 14 correspond to attributes of XML elements in the concrete syntax of QuantML. Some of these items are optional, in the sense of having a default value: polarity is by default 'positive', exhaustivity is 'negative', event scope is 'narrow', and repetiveness is 'at least once'. The attributes

145and 16 are exceptional in that they exist only in the concrete syntax; they do not correspond to anything in the abstract syntax or the semantics. They have been added purely to support searches in corpora where generic or modal quantification is marked up. For explanations and discussion of all the categories see (Bunt, 2024).

2.2 UMR Formalism

Three equivalent alternative formats are used in AMR-UMR: logical, graph, and PENMAN formats. This paper focuses on the last. form. Example (2) illustrates how meaning is represented in the PENMAN format at the two levels of UMR: the sentence and document levels.[1] Representation 2 is understood as saying that the sentence 's1' contains an instance 's' of the event of two women sharing a pizza 'p' and this occurred yesterday, the date of which depends on the document creation time ('DCT') (see Van Gysel et al. (2022), part 1).

(2) Sentence 1: Two women shared a pizza.

```
Snt1: Two women shared a pizza
      today.
%% Sentence (predicate) level
(s / share
   :agent (w / woman
             :quantity 2)
   :patient (p / pizza
               :quantity 1)
   :temporal (t / yesterday))
%% Document (discourse) level
(s1 / sentence
    :temporal ((DCT :depends-on s1t)
               (s1t :contained s1s)))
```

Some of the variables that are introduced in the PENMAN representation ('s','w', 'p', and 't') at the predicate-structure level. may recur at the document (discourse-structure) level, being prefixed with their root variable 's1' in this level, as seen in (2).

This format makes use of two operators: the slash / for concepts and the colon : for semantic relations. The slash form like (s / share) represents the variable *s* as an *instance* of the semantic concept *share*, not as a word in the text, and it is logically represented as *instance(s, share)*.[2] The concept *share* might be realized as a verb "shared", a noun "share", or a participle "sharing".

The colon ':' indicates a binary relation between two concept variables or between a variable and some logical element like the negation, a numeral, or a name like "John". For example, the relation :agent in example (2) relates the concept variable 'w' for woman to the root concept variable 's' for the 'share' predicate. It is logically represented as *agent(s, w)*, stating that *w* is the agent in the share event *s*.

The current version of the UMR guidelines (Van Gysel et al., 2022) treats coreference, temporal, and modal relations at the document (discourse-structure) level representation, with representations different from those at the sentence level. To be consistent, we propose that they should be represented as follows:

(3) Proposed discourse-level representation

```
(s1 / sentence
    :temporal (d / depends-on
                 :arg1 s1t
                 :arg2 DCT)
    :anchoring (c / contained
                  :arg1 s1s
                  :arg2 s1t))
```

For using UMRin the annotation of quantification, we have to consider the representation of scope. The original AMR format is too limited in this respect, and this is one of the motivations for extending it to UMR. According to the UMR Guidelines (Van Gysel et al., 2022) quantification scope is represented with an 'inverse relation', such as pred-of, as illustrated by example (4):

(4) Someone answered all the questions.

```
(a / answer_01
  :ARG0 (p / person)
  :ARG1 (q / question
           :quant all)
  :pred-of (s / scope
              :arg1 p
              :arg2 q))
```

Here, it is understood that the first argument scopes over the second. This would be clearer if instead of the term *scope* a term like *scopes-over* would be used. On this approach the event predicate remains the root of the structure.

Second, Pustejovsky et al. (2019) have proposed a treatment in which a *scope* concept is used as the root of the representation, rather than the event predicate. On this approach, the sentence in (4) is represented as follows:

(5) Representation in (Pustejovsky et al. (2019)

[1]**?** prefer to call the two ;eve;s of UMR *predicate-structure level* and *discourse-structure* level, respectively. We might follow their practice in this paper. Note also that the discourse-structure level may deal with a single sentence.

[2]By introducing the notion of *instance*, events and properties are treated not as functional types like $(t \rightarrow e)$, but as first-class objects.

```
(s / scope
    :pred (a / answer_01
            :ARG0 (p / person)
            :ARG1 (q / question
                     :quant all)
    :arg1 p
    :arg2 q))
```

Again, the intended interpretation could be made more explicit by using *scopes-over* as the concept name. QuantML distinguishes other scope relations besides *wider*, viz. *equal* and *dual*. In this paper we therefore propose a slightly different representation, in which the scope concept has besides the two scope-linked arguments also an explicit representation of the scope relation. This leads to the following representation:

(6) Representation eith explicit scope relation

```
(s / scope
   :pred (a / answer_01
            :ARG0 (p / person)
            :ARG1 (q / question
                     :quant all)
  :arg1 p
  :arg2 q
  :scopeRel wider)
```

Both the approach proposed by Pustejovsky et al. and the one suggested in (6) support the representation of a single scope relation only. For a sentence with multiple scope relations, like (7), a more general approach is needed.

(7) Some teachers gave every pupil a present.

One possibility is to extend the approach proposed by Pusttejovsky et al. (2019) by allowing a list of scope relations, all of which refer to the same predicate-level substructure.

Alternatively, the scope concept used in (6) can be generalized to a more complex 'scope structure' concept that has a list of scope relations. For a three-argument sentence like (7) this would look schematically as follows.

(8) Snt1 Some teachers gave every pupil a present.

```
(s / scope
   :scoping (sc / scopeLinks
               :op1 (sl1 / scopeLink
                  :arg1 x1
                  :arg2 x2
                  :scopeRel wider)
               :op2 sl2 / scopeLink
                  :arg1 x1
                  :arg2 x3
                  :scopeRel wider)
   :pred (g / give
            :  ...))
```

Even when enriched with a representation of scope, the UMR format is still quite limited in its representation of quantification, being restricted to the use of a `quant` relation in expressions of the form `:quant all`, which corresponds to the representation of involvement, including numerical involvement, as expressed in QuantML/XML by the value of the @involvement attribute. As noted in Section 2.1, QuantML supports the annotation of 16 aspects of quantification, of which involvement is one. In the next section we consider how of these aspects could be represented in UMR.

3 Annotations of Quantification

3.1 Participation in Events

Some of the aspects of quantification listed in (1) can be represented as properties of events or participants, but some aspects cannot, since they express properties of *the way* certain participants are involved in an event. This is the case for the distributivity of a quantification, as illustrated by the example *"The three men had a beer after moving the pianos"*. Here we see the same three men involved individually in drinking beer and collectively in moving pianos.

Another case is the exhaustivity of a quantification. Consider the difference between *"Two women smiled"* and *"TWO women smiled"*. In the latter case, no more (and no less) than two women smiled, and there is an implication that certain other women did not smile. This distinction can hardly be captured by a property of the women who did smile.

A third case is the *event scope*. This sometimes overlooked aspect concerns the relative scoping of events and participants. For example, the sentence *"Ninety-six passengers survived a crash"* describes in one interpretation a crash in which 96 participants survived, and in another interpretation the total number of passengers who survived in a number of crashes. In the former interpretation, an event scopes over set of passengers, in the latter interpretation it's the other way round.

In QuantML, *participation* and *predication* structures are used for indicating how events and participants are related.[3] Such structures contain the specification of semantic role, distributivity, and event scope (and optionally exhaustivity and polarity). These aspects of quantification can be represented in UMR by introducing the participation and prediction concepts. The use of these concepts in

[3]Predication structures are used for copular verbs and verbs with adjectival complements

QuantML/XML and QuantML/UMR is illustrated in (13) and (14) and discussed in the next section.

3.2 Domains of Quantification

Quantifications in natural language have a certain domain. The domain defined by the head noun of a quantifying NP, such as the domain of women in *"two women"* and the domain of students in *"all the students"* is called the *source domain*. Occurrences of NPs are nearly always intended to refer to a subset of the source domain - in *"two women smiled"* the quantification most likely does not refer to the set of all women, and in *"all the students protested"* the quantification does not range over the set of all students in the world, but they refer to certain contextually determined subsets of women and students, respectively. This more restricted domain is called the *reference domain*[4]hl, 19??). For NPs with bare head nouns the reference domain and the source domain may coincide, but in general the specification of a reference domain combines nouns, modifiers and conjunctions, as illustrated in (9)

(9) Twenty valuable ((Chinese vases) and (Japanese drawings))

Modifiers are interpreted in QuantML using a concept of *modification* that has some similarities with *participation*. A modification structure captures properties of the way a modifier relates to its arguments. These properties include distributivity, restrictiveness, and form of linking (linear or inverse). The following examples illustrate the distributivity and linking of modifiers, respectively.

(10) a. I'm carrying these heavy books to the new library building.

b. Two students from every Dutch university participated in the talks.

For describing such properties QuantML makes use of <entity> elements with a @domain attribute whose value refers to a reference domain, represented by a <refDomain> element. Such an element may contain references to multiple subdomains and modification structures. The recursion in such representations ends when a subdomain consists of a single component without modifiers. Such a domain is represented by a <sourceDomain> element. For example, the quantification domain of "older (men and women)" is represented in (11).

[4] Also known as 'context set' (Westerst{a

(11) Older men and women.

Markables:
m1 = "older men and women",
m2 = "older", m3 = "men and women",
m4 = "men", m5 = "women".

```
<entity xml:id="x1" target="#m1" domain="#x2"
  individuation="count" involvement="some"/>
<refDomain xml:id="x2" target="#m3"
    :subdomains="#y1 #y2" determinacy="indet"
    restrictions=#r1"/>
<adjMod xml:id="r1" target="#m2"
    pred="older">
<sourceDomain xml:id="y1" target="#m4"
    pred="woman"/>
<sourceDomain xml:id="y2" target="#m5"
    pred="woman"/>
```

Complex quantification domains can be represented in UMR by introducing the concepts of reference domain, source domain, and modification structure, leading to the following representation.

(12) Older men and women.

```
(x1 / entity
   :individuation count
   :involvement some
   :domain (x2 / refDomain
    :subdomains (c / conjunction
       :op1 (y1 / sourceDomain
          :pred man)
       :op2 (y2 / sourceDomain
          :pred woman))
    :determinacy / indet
    :restr (m / modification
       :distributivity individual
       :restrictiveness restrictive
       :linking linear))
```

When we compare the XML and the UMR representations of this sentence we can see some interesting similarities and differences. These are analysed in the next section.

4 Comparing and Converting Representations

The unit of annotation in QuantML is a clause, i.e. a grammatical unit describing an event (or set of events) and the participants involved. The QuantML/XML annotation of the clause in example (2) is represented as follows.

(13) Two women shared a pizza.

Markables: m1 = "Two women", m2 = "women",
m3 = "shared", m4 = "a pizza", m5 = "pizza".

```
<event xml:id="e1" target="#m2" pred="share"/>
<entity xml:id="x1" target="#m1" domain="#x2"
    individuation="count" involvement="2"/>
<refDomain xml:id="x2" target="#m2"
   subdomains="#x3"     determinacy="indet"/>
<sourceDomain xml:id="x3" target="#m2"
    pred="woman"/>
```

```
<entity xml:id="x4" target="#m4" domain="#x5"
    individuation="count" involvement="some"/>
<refDomain xml:id="x5" target="#m5"
    subdomains="#x3" determinacy="indet/>
<sourceDomain xml:id="x3" target="#m5"
    pred="pizza"/>
<participation event="#e1" participant="#x1"
    semRole="agent" distributivity="individual"/>
<participation event="#e1" participant="#x2"
    semRole="patient" distributivity="individual"/>
<scope arg1="#x1" arg2="#x4" scopeRel="wider"/>
```

Using the representation of scope shown in (6), the corresponding QuantML/UMR representation is as follows.

(14) Two women shared a pizza.

```
%% Sentence (predicate) level
    Snt1: Two women shared a pizza.
    (sc / scope
       :scoping (sLs / scopeLinks
         :op1 (sL1 / scopeLink
             :arg1 x1
             :arg2 x4
             :scopeRel wider)
       :pred (sh / share
         :arguments (a / argStructure
          :op1 (p1 / participation
             :semRole agent
             :participant (x1 / entity
                :individuation count
                :involvement 2
                :domain (x2 / refDomain
                  :determinacy indet
                  :subdomains
                     (x3 / sourceDomain
                          :pred woman))))
              :distributivity collective
              :eventScope narrow)
           :op2 (p2 / participation
             :semRole patient
             :participant (x4 / entity
                :individuation count
                :involvement 1
                :domain (x5 / refDomain
                  :determinacy indet
                  :subdomains
                     (x6 / sourceDomain
                          :pred pizza))))


                :pred pizza
                :individuation count
                :involvement 1)
             :distributivity individual
             :eventScope narrow)))
```

Inspecting the clause representations in (13) and (14) and the NP representations in (11) and (12) we can observe the following correspondences ,

(15) a. For all entity structures[5], i.e. for the XML elements <event>, <entity>, <cardinality>, <amount> and the elements representing modifiers (<adjMod>, <relClause>, <ppMod> and <nnMod>) therre is a simple correspondence:[6] :

```
<E xml:id="x" a1="v1" ... ak="vk" />
⇐⇒
(x / E
    :a1 v1
    ...
   :ak vk)
<E xml:id="x" a1="v1" ... ak="vk" />
⇐⇒
(x / E
    :a1 v1
    ...
   :ak vk)
```

b. For participation and predication link structures, the corresponding UMR structure is the same, except that in this case the variable that is introduced does not have a counterpart in the XML representation.

```
< a1="v1" ...  ak="vk" >
⇐⇒
(z / L
    :a1 v1
     ...
    :ak vk)
```

c. For XML attributes with value type 'IDREF" (i.e. a pointer to another substructure of the representation):

aj="#xj"
⇐⇒
the UMR structure that corresponds to the XML element with xml:id="xj".

For example, the following correspondence holds because of clause (15a) and the present clause:

```
<refDomain xml:id="x2" subDomains=
   "#x3" determinacy="indet"/>
<sourceDomain xml:id="x3"
   pred="woman">
⇐⇒
(x2 / refDomain
     :determinacy indet
     :subDomain
             (x3/sourceDomain
                :pred woman))
```

[5]In the sense of the distinction between entity structures and link structures.

[6]ISO annotation schemes include a fine-grained representation of the anchoring of annotation components in the primary data using maarkables, following the TEI standard. Since UMR does not represent the anchoring of annotation components, we suppress for comparison the markables in QuantML/XML representations in the rest of this paper.

d. For QuantML/XML attributes with value type ’IDREFS” (i.e. a list of pointers to other substructures):

```
aj="#xj #x2...  #xk"
⇐⇒
(op1 U1
:op2 U2
...
:opk Uk)
```

where Uj is the UMR structure that corresponds to the XML element with identifier ’xj’, i.e. the entity structure with xml:id=”xj”.

Together, the items in (15) define correspondences between NP representations, as exemplified in (16) for the phrase *“Seven black kittens”*.

(16)
```
<entity xml:id="x1" domain="#x2"
    individuation="count" involvement="#c1"/>
<cardinality xml:id="c1" numRel="greater-
    or-equal" number="7">
<refDomain xml:id="x2" subDomains="#x3"
    restrs"#r1" determinacy="indet" />
<source xml:id="x3" pred="kitten"/>)
<mod xml:id="#m1" pred="black">
    distributivity="individual">
⇐⇒
(x1 / entity
  :domain (x2 / refDomain
     :subdomains (x3 / sourceDom
                        :pred kitten)
     :restrs (m1 / mod
          :distributiviy individual
          :pred black
     :determinacy indet
  :individuation count
  :involvement (c1 / cardinality
     :numRel greater-or-equal
     :number 7))
```

The correspondences between parts of XML- and UMR-representations suggest that it should not be too difficult to define conversion functions $F_{X\to U}$ and $F_{U\to X}$ that transform XML-representations to UMR-representations and vice versa. A complication, however, is formed by a structural difference between the two.

Where a QuantML/XML representation is a list of XML elements, describing events and participants, connected by pointers, a UMR representation is a rooted nested structure with an event node (also called ‘pred’ node in UMR) as the root and with the participation in the event as properties of the event. Moreover, for a sentence with scope relations, which is almost every sentence, the proposals for scope representation discussed in Section 2.2 add a scope node as the root directly ‘above’ the event node, By contrast, scope relations in QuantML/XML are simply items in the list of XML elements, arbitrarily positioned.

A conversion form XML to UMR has to make the converted scope representation and position and place it immediately ‘above’ the converted event representation.

To define the XML - UMR conversion function we note hat a QuantML/XML structure represents a clause. In the list of elements of such a representations three parts can be distinguished: (1) an <event> element, (2) a list of <scope> elements, and (3) elements describing participants and participation. We use the notation in (17) to designate such a list (17):

(17) $L = [L_e, L_s, L_p]$

For the elements in L_e and L_p the XML-UMR conversion function is defined by the correspondences in (15).

For converting the scope relations $l_{s1}, l_{s2}, ...l_{sn}$ forming L_s it should be noted that their representation in QuantML/XML depends on the fact that they occur in a clause representation with a single <event> element, whereas in UMR they are the value of the ‘pred’ relation in a scope representation. Their conversion to UMR is therefore part of the conversion of the clause representation as a whole, specified in (18).

(18) $F_{X\to U}(L) =$
```
(s / scope
   :scoping (sLs / scopeLinks
      :op1 (sL1 / scopeLink
                 :arg1 x1
                 :arg2 x2
                 :scopeRel sr1)
      ...
      :opk (sLk / scopeLink
                 :arg1 xj
                 :arg2 xk
                 :scopeRel srk)
   :pred FX→U(Le,Lp))

where
FX→U(Le,Lp) = ⋃³(FX→U(Le),FX→U(Lp),
                 "arguments")
and
FX→U(Lp) = ( a / argumentStructure
   :op1 FX→U(Lp1)
   ...
   :op2 FX→U(Lp2)
   :opk FX→U(Lpk))
```

The operation $\bigcup^3$ is defined as follows. Given two UMR representations, the operation adds the second argument to the first argument through the relation expressed by the third argument. Schematically:

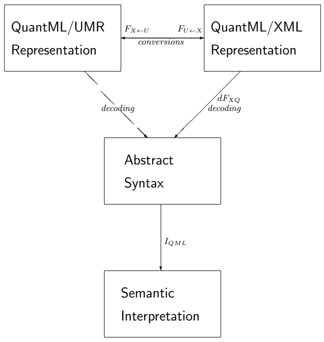

Figure 2: SemAF Architecture with UMR .

(19) $\bigcup^3$

```
   ( (a / A,        (b / B,       R )
       :a1 v1          :b1 w1
       :  ...           ...
       :ak vk)          :bn wn)
 = (a / A
       :a1 v1
       :  ...
       :ak vk
       :R (b / B
             :b1 w1
             :  ...
             :bn wn))
```

Applied to the QuantML/XML representation in (13) this conversion function produces the QuantML/UMR representation in (14).

Looking in the other direction, the correspondence relations in (15) are equally useful for converting subexpressions of UMR representations to XML, due to their symmetry. For example, based on the correspondence (15a), the conversion of entity structure representations is defined in (20).

(20) $F_{U \to X}$

```
((x / E
       :a1 v1
       :...
       :ak vk)) =
 = <E xml:id="x" a1="v1"...ak="vk" />
```

The structural differences and the different treatments of scope are easier to convert from UMR to XML than in the other direction. The list of scope relations in the `(sLs / scopeLinks)` list is converted to the list of their converted representations, with all the elements linked to the <event> element corresponding to the value of `:pred` relation in the `(s / scope)` representation. This is illustrated in (21).

Since QuantML/UMR representations can be converted to QuantML/XML, the existence of a decoding function from annotations in the latter format have to expressions of the QuantML abstract syntax shows that the QuantML/UMR representations have the same semantics as those of QuantML/XML.

To complete the reasoning about the two formats, we consider the decoding of QuantML/UMR representations directly by a decoding function to expressions of the QuantML abstract syntax.

5 Conclusions: UMR in the SemAF architecture

The inter-convertibility of the XML-based and the UMR-based representations of QuantML annotations means that QuantML/UMR fits perfectly into the three-level architecture of ISO SemAF annotation schemes, as visualized in Figure 2. It does involve the addition to UMR of some of the fundamental concepts of QuantML, notable those of participation (in events), of predication for dealing with copular verbs, and of adnominal modification, and it requires a more general treatment of scope.

As noted in Section 2.1, annotations made by a SemAF annotation schema have a semantics, defined by an interpretation function which is applicable to the structures of the abstract syntax. In the case of QuantML, the semantics combines event semantics and generalized quantifier theory with Discourse Representation Theory. This semantics applies to the representation structures defined by a concrete syntax via ‘decoding’ functions that express the semantic information in these representations in the set-theoretic structures of the abstract syntax.

From the inter-convertibility of QuantML/XML and QuantML/UMR plus the existence of a decoding function for QuantML/XML, defined in ISO 24617-12:2025 and Bunt (2023), it follows that QuantML/UMR representations can also been decoded in the QuantML abstract syntax, and thus share the semantics of QuantML/XML.

UMR and its predecessor AMR are commonly viewed as a representation format for first-order logic. Indeed, Bos (2016) presented a first-order semantics for AMR. The enrichment of AMR to UMR with scope links, inspired by QuantML (see Pustejovsky...????) and the extension to generalized quantifiers, as well as with other than individual participation, brings the expressiveness of UMR to a higher level. Since the semantics of the QuantML abstract syntax is second-order (viz. second-order DRT), needed for dealing with generalized quantifiers, it follows that QuantML/UMR representations also have a second-order semantic interpretation.

(21) $F_{U\to X}$(s / scope
:scoping (sLs / scopeLinks
:op1 (sL1 / scopeLink
:arg1 x1
:arg2 x2
;scopeRel R1)
...
:opk (sLk / scopeLink
:arg1 x1
:arg2 x2
;scopeRel Rk))
::pred (e / event
:name ‘eventName’
:arguments (a / argumentStructure
:participant (x1 / entity
:domain d1
:individuation i1
:involvement q1
:semRole sR1
:distributivity di1
:opk (pk / participation
:participant (xk / entity
:domain dk
:individuation ik
:involvement qk
:semRole sRk
:distributivity dik

:individuation ik
: ...)))

= $F_{U\to X}$((sL1 / scopeLink
:arg1 x1
:arg2 x2
;scopeRel R1))
...
$F_{U\to X}$((sLk / scopeLink
:arg1 xi
:arg2 xj
;scopeRel Rk))
$F_{U\to X}$((e / event
:name ‘eventName’
:arguments (a / argumentStructure
:op1 (p1 / participation
:participant (x1 / entity
:individuation i1
: ...))
:...
:opk (pk / participation
:participant (xk / entity
:individuation ik
: ...))

= <scope arg1=”#x1 arg2=”x2” socpeRel=”R1”>
...
<scope argi=”#xi arg2=”xj” socpeRel=”Rk”>
<event xml:id=”e” pred=”eventName”>
<participation event=”#e” participant=”#x1” distributivity=”d1” semRole=”sR1” >
< entity xxml:id=x1 individuation=”i1”...>
...
<participation event=”#e” participant=”#xk” distributivity=”dk” semRole=”sRk” >
< entity xxml:id=x1 individuation=”ik”...>
...
etc.

References

J. Barwise and R. Cooper. 1981. Generalized Quantifiers and Natural Language. *Linguistics and Philosophy*, 4:159–219.

J. Bos. 2016. Expressive Power of Abstract Meaning Representations. *Computational Linguistics*, 42:3:527–535.

J. Bos. 2020. Separating Argument Structure from Logical Structure in AMR. In *Proceedings Second International Workshop on Designing Meaning Representations (DMR 2020)*, pages 13–20. Association for Computational Linguistics ACL.

H. Bunt. 2019. An annotation scheme for quantification. In *Proceedings 14th International Conference on Computational Semantics (IWCS 2019)*, pages 31–42, Gothenburg, Sweden.

H. Bunt. 2023. The Compositional Semantics of QuantML. In *Proceedings of the 19th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-19)*, pages 56 – 65, Nancy, France.

H. Bunt. 2024. Combining annotation schemes through interlinking. In *Proceedings ISA-20, Twentieth Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 84–94, Turin, Italy.

R. Cooper. 1983. *Quantification and syntactic theory*. Reidel, Dordrecht.

D. Davidson. 1967. The Logical Form of Action Sentences. In N. Resher, editor, *The Logic of Decision and Action*, pages 81–95. The University of Pittsburgh Press, Pittsburgh.

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

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*. International Organisation for Standardisation ISO, Geneva.

ISO. 2025. *ISO24617-12:2025, Language Resource Management: Semantic Annotation Framework (SemAF) - Part 12: Quantification*. International Standard. International Organisation for Standardisation ISO, Geneva.

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

Kiyong Lee, Harry Bunt, James Pustejovsky, Alex Chengyu Fang, and Chongwon Park. 2025. Representing ISO-annotated dynamic information in UMR. In *Proceedings of the Sixth International Workshop on Designing Meaning Representations (DMR 2025)*, pages 49–58.

R. Montague. 1971. The proper treatment of quantification in ordinary language. In R.Thomason, editor, *Formal Philosophy*. Yale University Press, New Haven.

T. Parsons. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press, Cambridge, MA.

V. Petukhova, H. Bunt, et al. 2008. LIRICS Semantic Role Annotation: Design and Evaluation of a Set of Data Categories. In *LREC*.

J. Pustejovsky, 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.

J. Pustejovsky, H. Bunt, and A. Zaenen. 2017. Designing annotation schemes: From theory to model. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 21–72. Springer, Berlin.

J. Pustejovsky, K. Lai, and N. Xue. 2019. Quantification and Scope in Abstract Meaning Representations. In *Proceedings of the First International Workshop on Designing Meaning Representations (DMR 2019), Florence, Italy*, pages 28–33. Association for Computational Linguistics.

J. Van Gysel, M. Vigus, J. Zhao, and N. Xue. 2022. *Uniform Meaning Representation 0.9 Specification*. Unpublished document.