The ISA-17 Quantification Challenge
Background and Introduction

Harry Bunt
Tilburg University

ISA-17@IWCS 2021
Groningen (online)

Tilburg University

ISO Semantic Annotation Framework

- Part 1: Time and events (ISO-TimeML, Pustejovsky, 2012)
- Part 2: Dialogue acts (Bunt, 2012); Second Edition 2020
- Part 4: Semantic roles (Palmer a.o., 2014)
- Part 6: Principles of semantic annotation (Bunt, 2016)
- Part 7: Spatial Information (ISO-Space, Pustejovsky & Lee, 2015)
- Part 8: Semantic relations in discourse (DR-Core, Prasad & Bunt, 2016)
- Part 9: Reference annotation (Romary, 2019)
- Part 11: Measurable Quantitative Information (Hao, 2020)
- Part 12: Quantification (proposal, 2019)

Annotating quantification: requirements

- Take linguistic, semantic, computational knowledge about quantification into account – ***GQT (****NPs as quantifiers),* ***event semantics, DRT***

- Compatible with ISO ***Principles of semantic annotation*** (ISO 24617-6)
- Semantically adequate: annotations have a well-defined semantics – *compositional translation into* ***DRSs***
- Compatible with ISO Linguistic Annotation Framework: Distinguish between ***annotations*** (content) and ***representations*** (format);
- Compatible with ISO-TimeML, ISO-SpaceML, and ISO-Semantic Roles.

QuantML meets these requirements.

Annotation Theory

ISO Principles for Semantic Annotation (ISO 24617-6):

A markup language for semantic annotations has a 3-part definition:

- **Abstract syntax** defines well-formed ***annotation structures*** as set-theoretical constructs, like pairs, triples, n-tuples.
- **Concrete syntax** specifies a format for the ***representation*** of annotation structures, e.g. in XML.
- **Semantics** assigns meanings to annotation structures.

Annotation scheme, formal specification

Formally, a semantic annotation scheme is a triple:

$$\langle AS_a, \quad CS_a, \quad Sm_a \rangle$$

$$= \langle\langle CI_a, \quad AC_a, \quad F_a^{-1}\rangle, \quad \langle V_a, \quad CC_a, \quad F_a\rangle, \quad \langle M_a, I_a\rangle\rangle$$

conceptual inventory	annotation structures	decoding	vocabulary representations	encoding	model interpretation

Annotation Theory

A **concrete syntax should be *'ideal'*** for a given **abstract syntax**:

- ***Complete***: For every annotation structure defined by the abstract syntax it defines a representation (F_a).
- ***Unambiguous***: Every representation is the rendering of exactly one annotation structure (F_a^{-1}).

All 'ideal' representation formats for a given abstract syntax are semantically equivalent.

The formal specification of QuantML meets these requirements.

Quantification phenomena covered in QuantML

- Definiteness/determinacy: Source and reference domain
- Absolute and proportional domain involvement
- Domain size
- Distributivity (collective, individual, unspecific)
- Individuation (count/mass/count parts)
- Participant scopes
- Event scope
- Repetitive events
- Negation scope
- Scope and distributivity of adjectives, PPs, NN modifiers, rel. clauses
- Proper names, definite (singular) descriptions
- Possessives
- Cumulativity
- Exhaustivity

Loose ends – not covered in QuantML

- Generics and habituals
- Quantitatively detailed involvement and domain size
- Reciprocals
- Reflexives
- Anaphoric possessives

Some of these may be dealt with by means of ‘annotation scheme plug-ins’ (Bunt, IWCS 2019), based e.g. on ISO 24617-11 Measurable Quantitative Information and on ISO 24617-9 Reference Annotation Framework.

Annotation Scheme Plug-ins

An **annotation scheme plug-in** is a (miniature) annotation scheme

$L_p = \langle AS_p, CS_p, Sm_p \rangle$

combined with a host annotation scheme through a *'plug interface'.*

$AS_p = \langle CI_p, AC_p, F_p^{-1} \rangle$:

conceptual inventory CI_p; annotation structures AC_p, decoding F_p^{-1}

$CS_p = \langle VC_p, CC_p \rangle, F_p \rangle$:

vocabulary VC_p; representation structures CC_p, encoding F_p

$Sm_p = \langle M_p, I_{ap}$:

model M_p; interpretation function I_p

Plug-interface for DA semantic content

A **plug-interface** for a host annotation scheme:

a miniature triple ${}_hY_p = \langle AS_y, CS_y, Sm_y \rangle$
defining link structures for combining elements of host and plug-in schemes.

$AS_y = \langle \{\}, \{\text{link structures}\}, \text{decoding} \rangle$

$CS_y = \langle \{\}, \{\text{link representations}\}, \text{encoding} \rangle$

Sm_y: interpretation of new link structures.

Combined: structures and representations and their semantics:

Host + plug-in + interface: $AC = AC_y \cup AC_y \cup AC_y$
$CC = CC_y \cup CC_y \cup CC_y$, etc.