DMLex

This is a recommended object model (i.e. not a serialization, not a data exchange format) for newly created, born-digital dictionaries (reimagined here as "lexicographic resources") which are simultaneously human-oriented and machine-understandable.

1. DMLex Core

The DMLex Core is for monolingual lexical resources, where headwords, definitions, examples etc. are all in one and the same language.

1.1. `LexicographicResource` object type

A data set which can be viewed and used by humans as a dictionary and - simultaneously - ingested, processed and understood by software agents as a machine-readable database. Terminological note: *lexicograpic* resource, not *lexical*.

Properties:

- `language` (optional, IETF language code)
  - The language of headwords, definitions, examples.
- `transcriptionScheme` (optional, reference to some external authority - which?)
  - The scheme (e.g. IPA) in which the `transcription` property of `Pronunciation` objects is given.

Children:

- `Entry` (one or more)

1.2. `Entry` object type

A part of a lexicographic resource which contains information related to exactly one headword.

Child of:

- `LexicographicResource`

Properties:

- `headword` (non-empty string)
  - The headword can be a single word, a multi-word expression, or any expression in the source language which is being described by the entry in the lexicographic resource.
- `homographNumber` (number, optional)

Children:

- `PartOfSpeech` (zero or more)

- `Pronunciation` (zero or more)
- `InflectedForm` (zero or more)
- `Sense` (zero or more)

1.3. `Sense` object type

A part of an entry which groups together information relating to one of the (possibly multiple) meanings (or meaning potentials) of the entry's headword.

Child of:

- `Entry`

Properties:

- `listingOrder`
  - Can be implicit from the serialization.
- `indicator` (optional, non-empty string)
  - A short statement that indicates the meaning of a sense and permits its differentiation from other senses in the entry.
- `definition` (optional, non-empty string)
  - A long statement that describes and or explains the meaning of a sense.

Children:

- `Usage` (zero or more)
- `Example` (zero or more)

1.4. The difference between entries and senses

An **entry** is a container for "formal" properties of the headword such as orthography, morphology, syntax and pronunciation. A **sense** is a container for those of the headword's properties which are statements about semantics and pragmatics.

1.5. `PartOfSpeech` object type

Any of the word classes to which a lexical item may be assigned, e.g. noun, verb, adjective, etc.

Child of:

- `Entry`

Properties:

- `value` (non-empty string)
  - Can be constrained by the DMLex Controlled Vocabularies Module.

1.6. `Usage` object type

An indication of some restriction on the use of the lexical item. The restriction can be pragmatic (time, region, register), semantic (domain, semantic type) or formal ('no plural').

Child of:

- `Sense`
- `Pronunciation`
- `InflectedForm`

Properties:

- `value` (non-empty string)
  - Can be constrained by the DMLex Controlled Vocabularies Module.
  - Its type (eg. whether register, temporal, geographic etc) can be specified by the by the DMLex Controlled Vocabularies Module.

1.7. `Pronunciation` object type

Information about the pronunciation of its parent.

Child of:

- `Entry`
- `InflectedForm`

Properties (at least one):

- `transcription` (non-empty string)
- `recording` (string: name or URL of a sound file)

Children:

- `Usage` (zero or more)

1.8. `InflectedForm` object type

An inflected headword is a form of the inflectional paradigm of its parent.

Child of:

- `Entry`

Properties:

- `label` (non-empty string) e.g. 'plural'
  - Can be constrained by the DMLex Controlled Vocabularies Module.
- `value` (non-empty string)

Children:

- `Usage` (zero or more)
- `Pronunciation` (zero or more)

1.9. `Example` object type

An instance of a lexical item's usage in a specific sense.

Child of:

- `Sense`

Properties:

- `text` (non-empty string)

2. DMLex Bilingual Module

Extends DMLex Core to support the encoding of bilingual lexicographic resources.

2.1. Extensions to `LexicographicResource` object type

Additional properties:

- `translationLanguage` (optional, IETF language code)
- `transcriptionScheme` (optional, reference to some external authority - which?)
  - The scheme (e.g. IPA) in which the `transcription` property of `TranslationPronunciation` objects is given.

2.2. Extensions to `Sense` object type

Additional children:

- `HeadwordTranslation` (zero or more)

2.3. `HeadwordTranslation` object type

The translation equivalent of the headword in one of its senses.

Child of:

- `Sense`

Properties:

- `text` (non-empty string)
  - Can be a single word, a multi-word expression, or indeed any expression in the target language.

Children:

- `TranslationPartOfSpeech` (zero or more)
- `TranslationUsage` (zero or more)
- `TranslationPronunciation` (zero or more)

- `TranslationInflectedForm` (zero or more)

2.4. `TranslationPartOfSpeech` object type

Any of the word classes to which the translation may be assigned, e.g. noun, verb, adjective, etc.

Child of:

- `HeadwordTranslation`

Properties:

- `value` (non-empty string)
  - Can be constrained by the DMLex Controlled Vocabularies Module.

2.5. `TranslationUsage` object type

An indication of some restriction on the use of its parent. The restriction can be pragmatic (time, region, register), semantic (domain, semantic type) or formal ('no plural').

Child of:

- `HeadwordTranslation`
- `TranslationPronunciation`
- `TranslationInflectedForm`

Properties:

- `value` (non-empty string)
  - Can be constrained by the DMLex Controlled Vocabularies Module.
  - Its type (eg. whether register, temporal, geographic etc) can be specified by the by the DMLex Controlled Vocabularies Module.

2.6. `TranslationPronunciation` object type

Information about the pronunciation of its parent.

Child of:

- `HeadwordTranslation`
- `TranslationInflectedForm`

Properties (at least one):

- `transcription` (non-empty string)
- `recording` (string: name or URL of a sound file)

Children:

- `TranslationUsage` (zero or more)

2.7. `TranslationInflectedForm` object type

A form of the inflectional paradigm of its parent.

Child of:

- `HeadwordTranslation`

Properties:

- `label` (non-empty string) e.g. 'plural'
  - Can be constrained by the DMLex Controlled Vocabularies Module.
- `value` (non-empty string)

Children:

- `TranslationUsage` (zero or more)
- `TranslationPronunciation` (zero or more)

2.8. Extensions to `Example` object type

Additional children:

- `ExampleTranslation` (zero or more)

2.9. `ExampleTranslation` object type

The translation of an example.

Child of:

- `Example`

Properties:

- `text` (non-empty string)

3. DMLex Entry Structuring Module

3.1. `SenseGroup` relation type

Represents the fact that a group of senses (all belonging to the same entry) should be grouped when presented to a human user. Typically, when an entry has a large number of senses, it is a convenience to the human user to group them into a smaler number of groups by some broad criterion, such as by part of speech or by semantic similarity.

Participants:

- `Sense` (two or more)

Properties:

- `indicator` (optional, non-empty string)
  - A short statement that indicates the broad meaning that unites the senses in this group and permits their differentiation from other senses in the entry.

3.2. `Subsense` relation type

Represents the fact that one sense (the subordinate sense) should be treated as a subsense of another sense (the subordinate). Both senses belong to the same entry.

Participants:

- the superordinate `Sense` (exactly one)
- the subordinate `Sense` (exactly one)

3.3. `Subentry` relation type

Represents the fact that one entry (= the subordinate entry) should be treated as a subentry inside the sense (= the superordinate sense) of another entry.

Participants:

- the superordinate `Sense` (exactly one)
- the subordinate `Entry` (exactly one)

Properties:

- `listingOrder`
  - Can be implicit from the serialization.

4. DMLex Crossreferencing Module

4.1. `Variant` relation type

Represents the fact that two entries are understood by the lexicographer as variants (for example masculine and feminine counterparts, spelling variants).

Participants:

- `Entry` (two or more)

4.2. `Opposition` relation type

Represents the fact that two senses (typically - but not necessarily - belonging to two different entries) have opposite meanings. This includes antonyms, converses and so on.

Participants:

– `Sense` (exactly two)

4.3. `Similarity` relation type

Represents the fact that two or more senses (typically - but not necessarily - belonging to two different entries) have the same or similar meanings. This includes synonyms, near synonyms, immediate hypernyms/hyponyms and so on.

Participants:

– `Sense` (two or more)

4.4. `Pertainment` relation type

Represents the fact that two or more senses (typically - but not necessarily - belonging to two different entries) are related to each other, in ways other than opposition and similarity.

Participants:

– `Sense` (two or more)

5. DMLex Inline Markup Module

5.1. `Placeholder` markup type

Marks up a substring inside a headword (or inside a headword translation) which is not part of the expression itself but stands for things that can take its place, or constitutes some kind of meta-notation. Examples:

– `beat [sb.] up`
– `continue [your] studies`

Markup of:

– `headword` property of `Entry`
– `text` property of `HeadwordTranslation`

5.2. `Headword` markup type

Marks up a substring inside an example (or inside an example translation) which corresponds to the headword (or to a translation of the headword).

Markup of:

– `text` property of `Example`
– `text` property of `ExampleTranslation`

6. DMLex Controlled Vocabularies Module

This module makes it possible to describe constraints on the values of certain plain-text properties of objects defined in DMLex Core and in DMLex Bilingual Module.

6.1. Extensions to `LexicographicResource` object type

Additional properties:

- **`labelLanguage`** (IETF language code)
  - The language of the display values of labels.

Additional children:

- **`PartOfSpeechLabel`** (zero or more)
- **`TranslationPartOfSpeechLabel`** (zero or more)
- **`UsageLabel`** (zero or more)
- **`TranslationUsageLabel`** (zero or more)
- **`InflectedFormLabel`** (zero or more)
- **`TranslationInflectedFormLabel`** (zero or more)

6.2. `PartOfSpeechLabel` and `TranslationPartOfSpeechLabel` object types

- A **`PartOfSpeechLabel`** represents one of several allowed values for the **`value`** property of **`PartOfSpeech`** objects.
- A **`TranslationPartOfSpeechLabel`** represents one of several allowed values for the **`value`** property of **`TranslationPartOfSpeech`** objects.

Properties:

- **`value`** (non-empty string)
- **`displayValue`** (optional)

Children:

- **`LabelMapping`** (zero or more)

6.3. `UsageLabel` and `TranslationUsageLabel` object types

- A **`UsageLabel`** represents one of several allowed values for the **`value`** property of **`Label`** objects.
- A **`TranslationUsageLabel`** represents one of several allowed values for the **`value`** property of **`TranslationLabel`** objects.

Properties:

- **`type`** (one of: **`normative`**, **`register`**, **`temporal`**, **`geographic`**, **`sociocultural`**, **`domain`**, **`frequency`**, **`attitude`**)
- **`value`** (non-empty string)
- **`displayValue`** (optional)

Children:

- `LabelMapping` (zero or more)

6.4. `InflectedFormLabel` and `TranslationInflectedFormLabel` object types

- An `InflectedFormLabel` represents one of several allowed values for the `label` property of `InflectedForm` objects. Properties and children same as above.
- A `TranslationInflectedFormLabel` represents one of several allowed values for the `label` property of `TranslationInflectedForm` objects.

Properties:

- `value` (non-empty string)
- `displayValue` (optional)

Children:

- `LabelMapping` (zero or more)

6.5. `LabelMapping` object type

Represents the fact that an item in the controlled vocabulary is equivalent to item provided by en external authority.

Parents:

- `PartOfSpeechLabel`
- `TranslationPartOfSpeechLabel`
- `UsageLabel`
- `TranslationUsageLabel`
- `InflectedFormLabel`
- `TranslationInflectedFormLabel`

Properties:

- `sameAs` (URI)