Orientation Configuration and Dimensionality Markup Language

Version 1.0

Cynthia Goodman Jasper Phillips Cory Massaro Zachary Yocum

February 26, 2013

Contents

1 Introduction

This document is a specification of Orientation Configuration and Dimensionality Markup Language (OCDML), a natural language annotation schema aimed at capturing the intrinsic orientation and dimensionality of spatial entities in addition to configuration relations between entities. This specification builds, in part, on ISO-Space[2], an annotation standard being developed at Brandeis University for markup of spatial and spatio-temporal information.

OCDML anticipates an annotation task with the goal of identifying spatially relevant entities and categorizing those entities according to certain intrinsic spatial properties. Our model presumes that there exist constraints on the intrinsic properties of spatial entities that humans rely on.[1] In *The Design of Everyday Things*[1], cognitive scientist Donald A. Norman discusses physical objects that people interact with regularly in their daily lives in terms of "affordances" which are, in essence, the converse of such constraints. Adopting such terminology, OCDML is intended to provide a formal means of capturing orientational spatial affordances of spatial entities that are exploited by natural language.

Examples of the kinds of spatial language that OCDML is intended to capture are listed in example (1).

(1) a. I found a baby bird in the road at my work today.

b. ... an empty flask stood on the table beside him.

c. If I held the iPhone flat on its back, the way it would lay on a table ...

For instance, in example (1a), the spatial entity *road* can be categorized as a 2-dimensional surface area whose intrinsic 'top' affords supporting, but also as a 3-dimensional 'tunnel' or 1-dimensional 'path', which both afford traversal. In addition, the entity denoted by *bird* can be categorized as a biped with an intrinsic 'bottom' that affords being supported. Though the spatial properties of these entities are not encoded explicitly in the lexical items themselves, they are exploited when interpreting the relation introduced by the spatial preposition *in*. In the case of example (1a) it is the intrinsic 'bottom' of the bipedal *bird* and the intrinsic 'top' surface of the *road* which are selected for in the configuration.

Similarly, for example (1b), it is the intrinsic 'bottom' surface of the *flask*, taken as a 3-dimensional rectangular prism or cylinder, that is selected for by *standing*, and the *table*, much like the *road* from the previous example, can be categorized as a 2-dimensional supporting surface whose 'top' affords being stood on. Also, assuming *him* can be categorized as a biped with an intrinsic 'left' and 'right', *beside* accesses the 'left' or 'right' for the location of *table* in a figure-ground relation between the *table* and *him*.

In example (1c), the *iPhone* can be categorized as a 3-dimensional rectangular-prism with intrinsic 'left', 'right', 'front', 'back', 'top' and 'bottom' surfaces. These surfaces, excluding the 'top' and 'bottom', afford laying. In this case, the *back* surface of the *iPhone* happens to be referenced explicitly, which can be classified as a 2-dimensional surface which has its own intrinsic 'left', 'right', 'front', 'back', 'top', and 'bottom' edges.

[1]We do not suggest that these constraints are the same as those formalized in modern universal laws of physics or mathematics. Rather, we presume that these constraints belong to the set of common knowledge of human beings in line with the notion of naïve physics from Western philosophical and cognitive psychological traditions.

2 Goal

The goal of our schema is to adequately capture the intrinsic spatial orientation and dimensionalities of spatial entities from natural language text for the purpose of making inferences about their participation in spatial relations. Our ultimate aim is, given a list of spatial entities, to be able to classify them according to the ways in which they participate within orientationally dependent spatial relations and be able to supply their 'default' configuration. E.g., we would want to be able to classify hominids as entities, which possess intrinsic lefts, rights, fronts, backs, tops and bottoms, whose default configuration, when acting as a figure in a figure-ground configuration, is a vertical 'bottom-to-top' relation with some supporting surface. The primary applications in spatial reasoning that OCDMLwould be suited to are text-to-scene generation applications.

3 Corpus Selection

Some corpora consisting of spatial language already exist in the form of scene descriptions for projects such as ImageCLEF and WordsEye. These projects contain descriptive texts that necessarily comprise spatial information, though they are not necessarily completely natural. We may test against such data, for our annotation, we have opted to select more linguistically natural prose from highly descriptive portions of literary texts. Preliminarily, we have programatically identified and isolated stage directions and scene descriptions from dramatic works in the public domain. Such documents can easily (and legally) be accessed via Project Gutenberg; the Library of Congress's website (loc.gov) and the Internet Archive (archive.org) also provide large literary corpora of public domain and Creative Commons-licensed works. We are also evaluating other domains to broaden the variety our corpus.

4 Tag Types

This section enumerates the inventory of OCDML tag types.

4.1 Spatial Entity

The SPATIAL_ENTITY tag is used to capture participants in spatial relations. OCDML inherits the ISO-Space SPATIAL_ENTITY tag which possesses `dimensionality`, `form`, `latLong`, `mod`, `countable`, `amount`, `quant`, and `scopes` attributes. OCDML extends the SPATIAL_ENTITY tag with the following attributes: `top_bottom`, `front_back`, and `left_right`. Table 1 enumerates the relevant ISO-Space attributes and OCDML-specific attributes for the SPATIAL_ENTITY tag.

`dimensionality` This attribute designates the number of spatial dimensions occupied by the entity. Under our model, we are taking spatial entities to be point-sets, so a value of POINT indicates a 0-dimensional entity consisting of a single point with no edges or surfaces. A value of LINE indicates a 1-dimensional entity with up to two bounding points with a single edge and no surfaces. A value of AREA indicates a 2-dimensional entity with some number of bounding edges, points and two surfaces. A value of VOLUME indicates a 3-dimensional entity with at least one surface and possibly a number of bounding edges and points.

Attribute	Value
id	se1, se2, se3, ...
dimensionality	POINT, LINE, AREA, VOLUME
mod	A spatially relevant modifier
line_type	SEGMENT, RAY, LINE, LOOP, OTHER
area_type	3-GON, 4-GON, DISC, ANNULUS, OTHER
volume_type	TRI_PRISM, RECT_PRISM, PYRAMID, SPHERE, TORUS, CYLINDER, CONE, BIPED, QUADRUPED, OTHER
left_right	INTRINSIC or RELATIVE
front_back	INTRINSIC or RELATIVE
top_bottom	INTRINSIC or RELATIVE

Table 1: Attributes for SPATIAL_ENTITY

`line_type` This attribute is used to classify the entity based on a set of 1-dimensional primitive types.

`area_type` This attribute is used to classify the entity based on a set of 2-dimensional primitive types.

`volume_type` This attribute is used to classify the entity based on a set of 3-dimensional primitive types.

`left_right` This attribute is a bit that indicates whether the entity possesses an intrinsic axis of orientation whose polar extremes are 'left' and 'right'. One heuristic which may help to determine whether a spatial entity possesses an intrinsic `left_right` axis is the *linear array test*. The *linear array test* can be used by asking "Could several of these entities be arranged in a 1-dimensional array such that the 'left' boundary of each entity in the arry abuts the 'right' boundary of another?" If the answer is "no", then the `left_right` attribute would be RELATIVE; if "yes", the value would be INTRINSIC.

`front_back` This attribute is similar to the `left_right` attribute. A value of INTRINSIC indicates the entity possesses an intrinsic axis of orientation whose polar extremes are 'front' and 'back'. A value of RELATIVE indicates the opposite. The *linear array test* heuristic can be modified for this case such that arrangement is 'front-to-back'.

`top_bottom` This attribute is similar, again, to both `left_right` and `front_back`. A value of INTRINSIC indicates the entity possesses an intrinsic axis of orientation whose polar extremes are 'top' and 'bottom'. The linear-array-test can be applied for this attribute as well to test if the entities can be stacked 'top-to-bottom'.

Example (2) lists examples of extents which would be appropriate candidates for the SPATIAL_ENTITY tag:

(2) a. ...a baby [**bird**$_{se1}$] in the [**road**$_{se2}$] at my work today.

b. ...an empty [**flask**$_{se3}$] stood on the [**table**$_{se4}$] beside [**him**$_{se5}$].

c. If [**I**$_{se6}$] held the [**iPhone**$_{se7}$] flat on its [**back**$_{se8}$], the way it would lay on a [**table**$_{se9}$] ...

4.2 Orientation Signal

The ORIENTATION_SIGNAL tag is used to capture additional information about the orientation of a SPATIAL_ENTITY that is not contributed by the lexical item associated with the SPATIAL_ENTITY extent itself. ORIENTATION_SIGNAL tags trigger CONFIGURATION_LINK tags, which are discussed in section 5.1. Table 2 enumerates the attributes for this tag type and example (3) lists examples of appropriate extents that it is intended to capture. We are currently investigating whether it is preferable to extend the ISO-Space SPATIAL_SIGNAL tag rather than establish this tag type.

Attribute	Value
`id`	`os1`, `os2`, `os3`, ...
`orientation_type`	LATITUDINAL, LONGITUDINAL, VERTICAL, OTHER

Table 2: Attributes for SPATIAL_ENTITY

`orientation_type` The `orientation_type` attribute is used to identify the way in which the signal informs the relation between spatial entities. This attribute is used to specify which axis of orientation of the figure entity is exploited for orienting the figure relative to the ground. LATIDUNIAL refers to the left-right axis, LONGITUDINAL refers to the front-back axis, and VERTICAL refers to the top-bottom axis. The additional value of OTHER is supplied for signals which establish an orientation not based on the three previously defined axes of orientation.

Examples of extents captured by the the ORIENTATION_SIGNAL tag are given in example (3):

(3) a. ... an empty flask [**stood**$_{os1}$] on the table beside him.

b. If I held the iPhone [**flat**$_{os2}$] on its back, the way it would [**lay**$_{os3}$] on a table ...

5 Link Types

5.1 Configuration Link

The CONFIGURATION_LINK relates two SPATIAL_ENTITY tags. The attributes are enumerated in table 3, though our model for configuration relations is still developing, and we would like to ensure the highest degree of interoperability with the existing treatment of orientation relations within ISO-Space. The combination of the `figure_config` and `ground_config` attributes are used to specify the configuration between the participating figure and ground entitities. The `figure_config` attribute is compulsory whereas the `ground_config` attribute is optional; example (4a) shows an optional `ground_config` whereas in example (4b) the `ground_config` is requisite. The `coercion` attribute can be used to indicate that the dimensionality of the figure entity is coerced to that of the ground or vice versa. The `direction` attribute is used to indicate the asymmetrical directionality of the relation in terms of figure-to-ground or ground-to-figure.

Attribute	Value
id	cl1, cl2, cl3, ...
figure_config	LEFT, RIGHT, FRONT, BACK, TOP, BOTTOM, ANY
ground_config	LEFT, ANY, FRONT, BACK, TOP, BOTTOM, OTHER
dim_coercion	FIGURE, GROUND, or NONE
trigger	An ID of a orientation_signal tag that triggers the link
direction	FIGURE_GROUND, GROUND_FIGURE

Table 3: Attributes for CONFIGURATION_LINK

References

[1] Donald A. Norman. *The Design of Everyday Things*, chapter 4. Basic books, 2002.

[2] James Pustejovsky, Jessica L. Moszkowicz, and Marc Verhagen. Iso-space: The annotation of spatial information in language. In *Proceedings of ISA-6: ACL-ISO International Workshop on Semantic Annotation*, Oxford, England, January 2011.