# Standard Upper Ontology

# A Documentation Trial

*Draft Version 0.1 - October 2000*

*Matthew West*

*matthew.r.west@is.shell.com*

## Introduction

### Background

The Institution of Electrical and Electronic Engineers (IEEE) established a discussion list for a Standard Upper **Ontology** (SUO) in the 2nd quarter 2000.

During wide ranging discussions a "Starter Ontology" was developed by David Whitten (1). This gave rise to discussions about a number of issues around the development of an SUO. These included:

- The use of **meta level** constructs as part of a foundation for an ontology;
- Principles of analysis of the world to ensure a consistent mapping into a representation in logic;
- The valence of relations at a foundation level in an ontology;
- The use of the Knowledge Interchange Format (KIF) and different ways of representing concepts using KIF;
- The use of natural language in the development of the SUO, and the importance of understandability of the result, including the target audience for the standard.

The use of spatio-temporal analysis as an underlying basis for analysing individuals.

### Purpose

This document presents an alternative start point to the Whitten starter ontology, so that the differences in approach can become evident.

This document is intended as an experiment at several levels:

- How to document an ontology as a standard;
- How to use KIF to create an ontology;
- What content an upper ontology should have.

The intention is to present some ideas about these things by example, and to promote discussion and understanding on the matter within the SUO community.

Comments on any of these are welcomed. Comments should be sent either to the editor, or to the SUO list.

### Target Audience

The target audience for this document is the members of the IEEE SUO discussion group, a group with a wide range of backgrounds and interests in ontology. Also included in the target audience are people with an interest in ontology or those who may be new to the list and who have not been party to the discussions (or who have not been able to keep up with them).

### Documentation Conventions

- Any word that is not in every day use (e.g. ontology) or a word that is used with a special technical meaning (e.g. type, sort) is included in the "Definitions and Abbreviations" section.
- A word with a definition in the "Definitions and Abbreviations" section is given in bold face on its first occurrence.

- Abbreviations are given in full on their first occurrence, and are included in the "Definitions and Abbreviations" section.

# Definitions and Abbreviations

## Definitions

| | |
|---|---|
| Meta-level | assertions about the **model**, rather than about what is being modelled |
| Metaphysics | study of the nature of things |
| Model | limited representation of the universe or some part of it for some purpose |
| Ontology | conceptual model consisting principally of classes, relations, and constraints |

## Abbreviations

| | |
|---|---|
| ACE | |
| E-R | Entity Relationship. |
| FOL | First order logic. |
| IEEE | Institution of Electrical and Electronic Engineering. |
| KIF | Knowledge Interchange Format. |
| NL | Natural language. |
| SUO | Standard Upper Ontology. |
| UCS | Universal Character Set |
| UML | The Unified Modelling Language. |

# Technical Discussion

## Formalisms and Forms

When creating a model you need to use a formalism of some sort to create it in. Examples include:

- First order logic (FOL);
- Entity-Relationship (E-R);
- Natural language (NL).

Different formalisms have different capabilities and properties.

### First Order Logic

FOL has some special properties:

- ...
- ...
- ...

This means that for things expressed in first order logic you can:

- ..
- ...
- ...

FOL does have restrictions in its capability to express some things:

- ...
- ...
- ...

These are the price that must be paid for the power that it supplies.

First order logic can take a number of forms, such as:

- Symbolic logic,  consisting of a special set of symbols, designed to be processable by logicians,
- KIF, an alpha-numeric form designed to be easily computer processable,
- Restricted natural language, such as ACE, which are easier to understand by humans, but harder for computers to process.

### Natural Languages

Natural languages have the ultimate expressive power. However, they can be ambiguous, and do not have the properties of FOL. In particular this makes it difficult to process automatically.

### Entity-Relationship Models

Entity-relationship models (or perhaps more correctly entity type, relationship type, attribute type models) are derived from the expression of information in columns and rows (tables). Entity types represent tables, relationship types represent columns that contain cross references between tables, and attribute types represent columns of terminal elements with (generally) text and numbers.

There is a basic capability to express constraints through the cardinality of the relationship types (for an object at one end of a relationship the minimum and maximum numbers of objects at the other end of the relationship that it can have a relationship with of this type). However, this ability is limited in what it can express.

Some versions of entity relationship languages (sometimes labelled "object-oriented") have extensions beyond the basic capabilities. For example the Unified Modelling Language (UML) (6) and EXPRESS (ISO10303-11) (4) both support subtype-supertype relationships, and provide a more capable constraint specification capability. However, even these are unable to express that one entity type is a member of another entity type, which means that as models become more complex, they become more abstract in nature, and eventually become a meta-model.

The main advantage of E-R models is that they map very well into the relational database formalism, which provides a powerful implementation environment. This makes E-R a good language choice for the meta-models of other languages.

## Modelling Style

In any language it is possible to say essentially the same thing in different ways. For readers it is helpful to have a consistent style, since this will help to pick up inconsistencies, errors, and particularly missing information.

Style covers things like:

- layout;
- choice of alternative structures (e.g. relations or axioms).

## World Views (Theories)

### Introduction

One of the key things about an ontology is the underlying view of the world that it has, and the views it can support. This is essentially a matter of **metaphysics**.

For most ontologies this is not too much of a problem, since there is a particular domain that is being covered, and there is usually a well established viewpoint for that purpose. The problem that faces an upper ontology is that it's purpose is to support many viewpoints, enabling them to be integrated, rather than being suitable for some particular purpose.

This means that the most intuitive viewpoint which comes from language, is not necessarily the most appropriate. Amongst other things it is essentially based on the metaphysics of 2000 years ago, and some progress has been made since then. "Some notes on the nature of things" (2) gives such a viewpoint.

On the other hand our intuitive view is one that should be supported (within the range in which it works) by anything underlying.

The next sections illistrates the point by developing features of an intuitive view of the world, identifying some issues that arise. The following section presents a view based on states, and shows how these issues are resolved.

### An Intuitive View

### *Outline*

The essence of an intuitive view is that the only individuals allowed are "whole" individuals. So a person, or a car, but not a parent or a child, because these are not essential classes to being a person. Similarly, whole individuals play a role in activities or relations. This gives rise to a number of issues.

### *Essential and Incidental Classes*

Under this view of the world, classes can be divided into two broad sorts, essential and incidental. Essential classes are those that a whole individual must belong to to be a whole individual. Incidental classes are just those that an individual happens to belong to from time to time. For example a door might be a member of opened or it might be a member of closed from time to time.

Essential classifications are timeless, because it is the whole individual that is a member of the essential class. However, this is not true for incidental classifications. Here we need to identify the period during which the door was open. This is usually done with a temporal relation, where either start and end date/time elements are added to the relation, or the relation itself participates in a relation that indicates it is true during this period. Either way one is essentially identifying for which period the individual is a member of the class.

### *Issue - Classes/sets with changing membership*

One of the problems that arise from this is that whilst sets are supposed to be defined by their members, with these incidental classes, membership appears to be changing over time. This is at least inconvenient.

### *The roles things play*

Things can play a role in either activities or relationships (where a relationship is a state of affairs that exists between two (or more) individuals. So a person can be chairman in a meeting activity, or a person can be the parent of a child.

A relationship is often turned into a relation of the form say parent-of, which indicates the role played by the first element of the relation with respect to the second. This is helpful to humans reading or writing in KIF, but does not mean that the relation represents a role. The roles are actually the places in the relation. So in this ase the first element would play the role of parent, and the second element would play the role of child.

There is no inherent order to a relationship. Both roles have equal status. However, when this is represented as a relation, there is an order to the roles, and whilst the particular order is not important, consistency obviously is. So if I write child-of instead of parent-of, these are semantically the same thing, but syntactically the order of the elements is probably different.

In EPISTLE where we have been modelling using entity relationship models, the same ordering issue is not present. However, it can still be tempting to name the relation equivalent in terms of the role one element plays towards another. However, we have a policy of naming in a non-directional way. So rather than say, part-of, we would use composition, and recognise whole and part as the places in the relation. Similarly, subtype-of would be specialisation, with subtype and supertype as the roles.

Anyway, regardless of naming conventions, the important thing here is to recognise that the roles are the places in a relation, and not the relation itself.

Relationships between individuals are generally temporal in nature, i.e. they hold for some period of time. So just as with incidental classifications, we need to deal with this, with the same, or similar solutions being available.

### *Issue - how do you deal with the temporal nature of relationships between individuals?*

With activities, things are different. A relation is sometimes used to represent an activity, with each of the places inndicating the role played in the activity. However, it is usually fairly easy to see that this is a short cut for each of the elements (or group of elements) being separately related to the element representing the activity itself. Thus you really have an activity, and a number of relations representing the involvement of the things in the activity. The types of involvement are the roles things play, again for the time that they are involved in the activity. So a chairman relation might indicate how a person was involved in an activity.

### *Issue - How can roles sometimes be represented by places in relations, and sometimes by relations?*

### *Artifacts - individuals designed to play a role*

The story does not stop here. When we make an artifact with an intended purpose, one of their essential classes is the role it is intended to play. For example, a pump is designed to play the role of a pump in a pumping activity.

Further, from time to time we have incomplete information. So for example, as an employer, one might know that an employee was a parent, without any knowledge of the children they have. You might then want to incidentally classify the person as a parent to capture what you do no.

So now roles can be essential or incidental classes.

*Issue - How can roles sometimes be places in relations, sometimes be relations, and sometimes be incidental or essential classes?*

### A Spatio-temporal View

*An outline of the basic principles*

Practically, a spatio-temporal view means putting our ordinary concept of individual (things like my car, me, the computer I am working on, the writing of this) into a wider context. For example it means that you admit that an individual can have states and that these are objects you might want to know something about. States are just temporal parts of individuals, and are no more remarkable than spatial parts of individuals, such as my head.

A bit scarier is the generalisation of this. This says that any possible spatio-temporal extent, i.e. any piece of space-time (and whatever material is in it) can be connsidered as a thing.

However, this is not something to get too excited about. Remember set theory, and how there are all those sets out there that can be constructed from the individuals in your universe? But we choose to be interested in only a few, which we call classes, or properties, or sorts or types or whatever. The same holds with spatio-temporal extents. We are interested in some of them, the ones that exhibit interesting patterns and are thus members of classes we are familiar with. The rest may exist but are of no interest.

Some examples of things that are spatio-temporal extents are:

- physical objects;
- activities (something happening);
- states (an individual for a period of time during which something applies);
- events (an instantaneous change in state);
- points in time;
- periods of time.

There is another nice parallel too, just as a set is uniquely defined by its membership, so with spatio-temporal theory, an individual is uniquely defined by its spatio-temporal extent. So now you know under what circumstaces a piece of clay and a vase are the same object, and you can give an objective reason for your rationale (though this doesn't prevent you from mapping to other viewpoints).

*So how does this help with the issues arising from an intuitive viewpoint?*

*Classes with changing membership*

Instead of temporal classification and changing class membership, the state of an individual that is a member of the class is identified. Now the classification is timeless,and so membership of the class is unchanging.

Of course, we still do not know the future, but because we have a static view of time, classifications that we discover with time are still true now, just unknown.

*Temporal relations*

With states there is no need for temporal relations. The time dimension is handled by relating appropriate states of the individuals in a relationship. Relations are now always timeless.

*The variable nature of roles*

Roles classify individuals or their states.

- In a parent-of relation, the state (of a person) in the first place is a member of the parent class, the state (of a person) in the second place is a member of the child class.

- In a meeting activity, there is a state (of a person) that is classified as chairman, and this state has a timeless involvment relation to the meeting activity.

*Roles as incidental or essential classes*

When we have incomplete information, as in the case of the employee where we know they are a parent, but not about their children, we have no problem. The state of the person that is a parent is the same state that would be involved in the parent-of relation, if we later found out about their children, we would be able to add to it.

This is generally the case with incidental classifications, as identified above also.

For essential classifications by roles, when the individual is an artifact, and it has an intended role, it is simply that an individual is being classified not a state.

# Some Principles for Ontology Development and Documentation

Principles for ontology development combine elements of style and metaphysics to provide some heuristics to guide ontology development. Different choices in style and metaphysics will give different principles. As an example, some principles developed for creating entity relationship models is given in "Developing high quality data models" (3). More recently a more formal approach has been taken by Guarino & Welty (7).

## Documentation Principles

Definitions should be given in terms of:

- A noun phrase without the definite or indefinite article;
- The first element should indicate the supertype for this type;
- Further elements should indicate distinguishing features.

Further, definitions for relations should be in the form:

<supertype> that indicates <what one role has to do with the other>

Relations should be named in terms of the noun part of the verb (often ending in -tion, rather than a role. naming in terms of a role only allows the role in one direction to be named.

For example, a relation called composition has as roles whole and part. A relation called connection has two roles, both of which are connected.

Note, the roles indicate a (class of) state of the thing that plays the role.

# A Trial Ontology

## Introduction

### Motivation

This ontology presents a small number of key concepts. A number of things are attempted:

- A way (or ways) of documenting an ontology to maximise the utility and comprehensibility of the ontology;
- Present part of a particular theory of individuals;

### Approach

KIF is a machine based language, and in the end any system that uses the SUO will have to interact with humans. Therefore a choice is made here to embed natural language elements within this trial SUO, both to aid review its review, and to support machine/human interaction. The key elements that are added are:

*names*

There seems to be a preference amongst this community that KIF names should be suitable for use by programmers rather than humans, with forms like instance-of, instance_of, or InstanceOf, preferred to instance of. So this is intended to support text strings as objects and allow them to be related to the objects they name. Note it will be possible to give the same text name to different concepts or to give different names (perhaps in different languages) to the same concept, although the SUO name will be different.

*Definitions*

Whilst the SUO will provide axiomatic definitions of the concepts it defines, these will not be intelligible to most people. Therefore this concept supports the use of a text string to provide a natural language definition of the concept. The definition is designed to be a replacement for the term defined in a larger piece of text.

Note that more than one definition is possible allowing for different levels of formality and language.

*Notes*

Notes allow additional information to be added by way of explanation beyond the formal definition that may be useful to the reader.

*Examples*

Examples of what is (or is not) a member of a concept can be very powerful in conveying what is meant by the concept. Provision is made to give these examples as text.

*Assignment*

With relatively large strings about (and for general convenience) an assignment relation is added. This allows a label to act as a reference to some other expression. ":=" is used as the assignment function.

### Alternatives

The choices above are not the only ones that could be made. You could choose to have all the NL documentation separate from the axioms. However, in the end they need to be put together.

### Implementation

The concepts mentioned above are about how you describe the concepts rather than being a part of the ontology itself. So they are considered here as extensions to KIF, rather than being part of the ontology itself.

## Ontology

### Preliminaries

This section introduces the key concepts that are needed to describe everything else. These are:

- thing;
- instance-of;
- subtype-of;
- collection;
- class;
- naming;
- written-name;
- text-string;
- definition;
- relation;
- phrase;
- note;
- sentences;
- exemplification:
- text-example:
- English-language;
- natural-language;
- language.

Define thing in terms of everything being a thing.

```
(forall ?x (instance-of ?x SUO/thing)
```

Give thing a name for use by humans.

```
(SUO/naming SUO/thing "thing")
```

Say the name was in English.

```
(instance-of "thing" SUO/English-language
```

Give thing a definition for humans (well non-logicians anyway).

```
(SUO/definition SUO/thing
  "thing that exists")
```

Say the definition is in English.

```
(instance-of "thing that exists" SUO/English-language)
```

Add a note about thing, by way of giving a corollary

```
(SUO/note SUO/thing
  "There is not anything that is not a thing.")
```

The text of the note is in English.

```
(instance-of "There is not anything that is not a thing." SUO/English-
language)
```

Give the basic definition of both instance of and collection in terms of each other. If something has instances, it is a collection. (or the null set). Instance-of is the relation that collects things into collections.

```
(forall ?x (instance-of ?x SUO/collection)
(exists ?y)
(or (instance-of ?y ?x) (= ?y SUO/null-set))
```

Instance-of is a subtype of relation.

```
(subtype-of instance-of SUO/relation)
```

Collection is a subtype of thing.

```
(subtype-of SUO/collection SUO/thing)
```

Give subtype-of a English name.

```
(SUO/naming subtype-of "specialisation")
(instance-of "specialisation" SUO/English-language)
```

Give subtype-of and English definition.

```
(SUO/definition subtype-of
"relation that indicates that all instances of the subtype are instances of
the supertype")
```

Give a name for human usage to instance-of.

```
(SUO/naming instance-of "instance of")
```

The name is in English.

```
(instance-of "instance of" SUO/English-language)
```

Provide an English definition for instance-of.

```
(SUO/definition instance-of
  "relation that indicates a thing is an instance of a collection")
(instance-of "relation that indicates a thing is an instance of a
collection" SUO/English-language)
```

Give collection an English name.

```
(SUO/naming SUO/collection "collection")
(instance-of "collection" SUO/English-language)
```

Give collection an English definition.
```
(SUO/definition SUO/collection "thing that has instances")
(instance-of "thing that has instances" SUO/English-language)
```

Class is a subtype of collection.
```
(subtype-of SUO/class SUO/collection)
```

Give class an English name.
```
(SUO/naming SUO/class "class")
(instance-of "class" SUO/English-language)
```

Define class.
```
(SUO/definition SUO/class
"collection whose membership is defined by a non-trivial axiom")
```
I do not know how to say this axiomatically.

Relation is a subtype of class.
```
(subtype-of SUO/relation SUO/class)
```

Give relation an English name.
```
(SUO/naming SUO/relation "relation")
(instance-of "relation" SUO/English-language)
```

Give relation an English definition.
```
(SUO/definition SUO/relation
"class whose instances are tuples")
(instance-of "class whose instances are tuples" SUO/English-language)
```

Define the roles of the places for a naming relation.
```
(=> (SUO/naming ?x ?y)
   (& (instance-of ?x SUO/thing)
      (instance-of ?y SUO/written-name)))
```

Naming is a subtype of relation.
```
(subtype-of SUO/naming SUO/relation}
```

Give naming a name.
```
(SUO/naming SUO/naming "naming")
(instance-of "naming" SUO/English-language)
```

Give naming an English definition.
```
(SUO/definition SUO/naming
"relation that indicates a name is a reference to a thing")
(instance-of "relation that indicates a name is a reference to a thing"
SUO/English-language)
```

Written name is a subtype of text string.
```
(subtype-of SUO/written-name SUO/text-string)
```

Give written name a name in English.
```
(SUO/naming SUO/written-name "written name")
(instance-of "written name" SUO/English-language)
```

Define written name in English.
```
(SUO/definition SUO/written-name
```

```
"encoding in some script that is a reference to some thing")
(instance-of "encoding in some script that is a referent to some thing"
SUO/English-language)
```

Text strings are classes.

```
(subtype-of SUO/text-string class)
```

Give text string a name in English.

```
(SUO/naming SUO/text-string "text string")
(instance-of "text string" SUO/English-language)
```

Give a definition in English for text string.

```
(SUO/definition SUO/text-string
"encoding of a sequence of script characters as UCS binary characters")
```

Give a name to the naming relation in English.

```
(SUO/naming SUO/naming "naming")
(instance-of "naming" SUO/English-language)
```

Give a definition to the naming relation in English.

```
(SUO/definition SUO/naming "relation that indicates a text string is a
referent to a thing")
(instance-of "relation that indicates a text string is a referent to a
thing" SUO/English-language)
```

Give an English  name to the definition relation.

```
(SUO/naming SUO/definition "definition")
(instance-of "definition" SUO/English-language)
```

Define the places in a definition relation.

```
(=> (SUO/definition ?x ?y)
  (& (instance-of ?x SUO/thing)
    (instance-of ?y SUO/phrase)))
```

A phrase is a subtype of text string.

```
(subtype-of SUO/phrase SUO/text-string)
```

Give phrase an English name.

```
(SUO/name SUO/phrase "phrase")
(instance-of "phrase" SUO/English-language)
```

Give phrase an English definition.

```
(SUO/definition SUO/phrase
"text string that is made up of words and can be part of a sentence")
```

Give an English name to definition.

```
(SUO/naming SUO/definition "definition")
(instance-of "definition" SUO/English-language)
```

Give an English definition of definition.

```
(SUO/definition SUO/definition
"relation that indicates a phrase provides the meaning of a thing")

(instance-of "relation that indicates a phrase provides the meaning of a
thing" SUO/English-language)
```

Define the roles in a note relation.

```
(=> (SUO/note ?x ?y)
  (& (instance-of ?x SUO/thing)
    (instance-of ?y SUO/sentence)))
```

Sentence is a subtype of text string.

```
(subtype-of SUO/text-sentence SUO/text-string)
```

Give an English name to sentence.

```
(SUO/naming SUO/text-sentence "sentence")
(instance-of "sentence" SUO/English-language)
```

Give an English definition of a sentence.

```
(SUO/definition SUO/text-sentence
"complete unit of text with meaning that makes a statement, gives a
command, or asks a question")
(instance-of "complete unit of text with meaning that makes a statement,
gives a command, or asks a question" SUO/English-language)
```

Give and English definition for note.

```
(SUO/definition SUO/note
"relation that indicates some text is informative about some thing")
(instance-of "relation that indicates some text is informative about some
thing" SUO/English-language)
```

Define the roles for an exemplification relation.

```
(=> (SUO/exemplification ?x ?y)
  (& (instance-of ?x SUO/collection)
    (instance-of ?y SUO/text-example)))
```

Text example is a subtype of text string.

```
(subtype-of SUO/text-example SUO/text-string)
```

Give an English name to exemplification.

```
(SUO/naming SUO/exemplification "exemplification")
(instance-of "exemplification" SUO/English-language)
```

Give an English definition for exemplification.

```
(SUO/definition SUO/exemplification
"relation that indicates an example is an instance of the collection")
(instance-of "relation that indicates an example is an instance of the
thing" SUO/English-language)
```

Give an English name to the English language.

```
(SUO/naming SUO/English-language "English")
(instance-of "English" SUO/English-language)
```

Give an English definition of the English language.

```
(SUO/definition SUO/English-language "written and spoken natural language
that originates from the country of England")
(instance-of "written and spoken language that originates from the country
of England" SUO/English-language)
```

English is a natural language.

```
(instane-of SUO/English-Language SUO/natural-language)
```

Natural languges are languages.

```
(subtype-of SUO/natural-language SUO/language)
```

Give natural language an English name.

```
(SUO/naming SUO/natural-language "natural language")
(instance-of "natural language" SUO/English-language)
```

Give natural language an English definition.

```
(SUO/definition SUO/natural-language
"language that is used by humans")
```

Languages are classes.

```
(subtype-of SUO/language SUO/class)
```

Give language a name in English.

```
(SUO/naming SUO/language "language")
(instance-of "language" SUO/English-language)
```

Give language a definition.

```
(SUO/definition SUO/language
"class that is a grammar and vocabulary for encoding information")
(instance-of "class that is a grammar and vocabulary for encoding
information" SUO/English-language)
```

### Spatio-temporal model of individuals

Spatio-temporal extent is a subtype of thing.

```
(subtype-of SUO/spatio-temporal-extent SUO/thing)
```

Give individual an English name.

```
(SUO/naming SUO/individual "individual")
(instance-of "individual" SUO/English-language)
```

Give individual a definition in English.

```
(SUO/definition SUO/spatio-temporal-extent "thing that possibly exists in
space-time")
(instance-of "thing that possibly exists in space-time" SUO/English-
language)
```

Add a note in English.

```
(SUO/note SUO/individual "An individual can be any spatio-temporal extent
and includes whatever is in it.")
(instance-of "An individual can be any spatio-temporal extent and includes
whatever is in it." SUO/English-language)
```

Object, point in time, period of time and event are subtypes of spatio-temporal extent.

```
(subtype-of SUO/object SUO/spatio-temporal-extent)
(subtype-of SUO/point-in-time SUO/spatio-temporal-extent)
(subtype-of SUO/period-of-time SUO/spatio-temporal-extent)
(subtype-of SUO/event SUO/spatio-temporal-extent)
```

Individual and state are subtypes of object..

```
(subtype-of SUO/individual SUO/object)
```

Give individual an English name.

```
(SUO/naming SUO/individual "individual")
(instance-of "individual" SUO/English-language)
```

All states are states of some individual,

```
(forall ?x (instance-of ?x SUO/state))
```

```
(exists ?y (instance-of ?y SUO/individual))
(SUO/state-of ?x ?y)

(subtype-of SUO/state SUO/object)
```

Activity and physical object are subtypes of object.

```
(subtype-of SUO/acivity SUO/object)
(subtype-of SUO/physical-object SUO/object)
```

## Some Reflections

## References


1 Whitten, D. "Starter Ontology", 2000, http://www. ...

2 West, M.R. "Some notes on the nature of things", 2000, http://www.

3 West, M.R. "Developing high quality data models Version 2", 1995, http://www.stepcom.ncl.ac.uk/EPISTLE/ ...

4 ISO10303-11 "EXPRESS Language Reference Manual"

5 "SUO KIF", http://www. ...

6 ISO ......, "The Unified Modelling Language", 2000, http://www.

7 Guarino, N., Welty, C. "