

The Universal Networking Language (UNL)

Specifications

Version 3 Edition 2

UNL Center
UNDL Foundation
1 July 2003

Introduction

The UNL is an acronym for “Universal Networking Language”. It is a computer language that enables computers to process information and knowledge across the language barriers. It is an artificial language that replicates, in the cyber world, the functions of natural languages in human communication. As a result, it enables people to express all knowledge conveyed by natural languages. It also enables computer to intercommunicate, thus providing people with a linguistic infrastructure for distributing, receiving and understanding multilingual information.

The UNL expresses information or knowledge in the form of semantic network with hyper-node. Different from natural languages, UNL expressions are unambiguous. In the UNL semantic network, nodes represent concepts, and arcs represent relations between concepts. Concepts can be annotated.

Although the UNL is a language for computers, it has all the components of a natural language. It is composed of words expressing concepts called “Universal words”, also referred to as UWs which are inter-linked with other UWs to form sentences. These links, known as “relations”, specify role of each word in a sentence. The subjective meaning intended by the speaker can be expressed through “attributes”. In addition, the “Knowledge Base (UNLKB)” is provided to define semantics of UWs. The UNLKB defines every possible relations between concepts including hierarchical relations and inference mechanism based on inclusion relations between concepts. Thus, the UNLKB provides semantic background of the UNL to make sure the meaning of the UNL expressions is unambiguous.

Chapter 1: UNL Expression

Binary relations are the building blocks of UNL expressions. They are made up of a relation and two UWs. This section deals with the definition and interpretation of the binary relations of the UNL expression.

There are two forms for expressing the UNL expressions, one is the table form and the other is the list form. The table form of a UNL expression is more readable than the list form, but the list form of a UNL expression is more compact than the table form. Here, only the table form is explained and the list form is shown in Chapter 5.

Any component, such as a word, phrase or title and, of course, a sentence of a natural language can be represented with UNL expressions. A UNL expression therefore consists of a UW or a (set of) binary relation(s). In UNL documents, a UNL expression for a sentence is enclosed by the tags {unl} and {/unl} inside [S] and [/S]. If a UNL expression consists of a UW, this UW should be enclosed further by the tags [W] and [/W]. If necessary, the whole sentence can also be expressed as a scope. In this case, the Compound UW-ID of the scope should be enclosed by [W] and [/W].

Thus, the UNL expression of a sentence is the following:

```
{unl}  
<Binary Relation>  
...  
{/unl}
```

or,

```
{unl}  
[W]  
<UW><Attribute List>  
[/W]  
{/unl}
```

or,

```
{unl}  
[W]  
":"<Compound UW-ID><Attribute List>  
[/W]  
<Binary Relation>  
...  
{/unl}
```

Syntax of a binary relation

A binary relation is made up as follows:

<Binary Relation>	::= <Relation Label> [":"<Compound UW-ID>] "(" {{ <UW ₁ > [":" <UW-ID ₁ >]} { ":" <Compound UW-ID ₁ > }}[<Attribute List>] "," {{ <UW ₂ > [":" <UW-ID ₂ >]} { ":" <Compound UW-ID ₂ > }}[<Attribute List>] ")"
<Relation Label>	::= a relation, see "Chapter 2: Relations"
<UW>	::= an UW, see "Chapter 3: Universal Words"
<Attribute List>	::= { ":" <Attribute> } ...
<Attribute>	::= an attribute, see "Chapter 4: Attributes"
<UW-ID>	::= two characters of '0' – '9' and 'A' – 'Z'
<Compound UW-ID>	::= two-digit decimal number (00 – 99) 00 is used for representing the main sentence, which can be omitted.

Compound UW-IDs are strings of two digits used to identify each instance specified by Compound UWs. Compound UWs are groups of binary relations (so-called "Scope-Nodes") that can be referred to as a UW.

For instance, the following shows an example of a UNL expression of the sentence "I can hear a dog barking outside".

```
{unl}  
aoj(hear(icl>perceive(agt>thing,obj>thing)).@entry.@ability, I)  
obj(hear(icl>perceive(agt>thing,obj>thing)).@entry.@ability, :01)  
agt:01(bark(agt>dog).@entry, dog(icl>mammal))  
plc:01(bark(agt>dog).@entry, outside(icl>place))  
{/unl}
```

In the above UNL expressions, "aoj", "agt" and "obj" are the **relation labels**, "hear(icl>perceive(agt>thing,obj>thing))", "I", "bark(agt>dog)", "dog(icl>mammal)" and "outside(icl>place)" are the **UWs**, and ":", which appears three times in the example, shows the Compound UW-ID. The Compound UW-ID appears in the position of a UW, the so-called "**scope-node**", and is used to cite or refer to a Compound UW previously defined. Binary relations indicated by the Compound UW-ID define the contents of the scope. A scope-node always begins with ":" followed by the two digits of a Compound UW-ID.

UW-IDs are omitted from the above UNL expression. When a UW is unique in a UNL expression, the UW-ID can be omitted.

The UW-ID is used to indicate some referential information, for example that there are two or more different occurrences of the same concept (they are not co-referent). Normally, if the same UW occurs more than once, it is in all cases understood to refer to the same entity or occurrence. For example, if one man greeted another man, the same UW would be used twice -- “man(icl>male person)” and it is possible to distinguish one from the other with the UW-IDs:

man(icl>male person):01 for the first and
man(icl>male person):02 for the other, to make it clear that the first man did not greet himself.

Chapter 2: Relations

This section deals with the definition and interpretation of the relation labels of the UNL.
The relations between UWs in binary relations have different labels according to the different roles they play. These Relation-Labels are listed and defined below.

Relation Labels

A relation label is represented as strings of 3 characters or less.
There are many factors to be considered in choosing an inventory of relations. The principles for choosing relations are as follows.

Principle-1 Necessary Condition

When an UW has relations between more than two other UWs, each relation label should be set so as to be able to identify each relation on the premise that there is enough knowledge about the concept of each UW expressed.

Principle-2 Sufficient Condition

When there are relations between UWs, each relation label should be set so as to be able to understand the role of each UW only by referring to the relation label.

The following are the relations defined according to the above principles.

agt (agent)

Agt defines a thing that initiates an action.

agt (do, thing)

Syntax

agt [“.”<Compound UW-ID>] “(“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “)”

Detailed Definition

An agent is defined as the relation between:

UW1 - do, and

UW2 - a thing

where:

- UW2 initiates UW1, or
- UW2 is thought of as having a direct role in making UW1 happen.

Examples and readings

agt(break(agt>thing,obj>thing), John(icl>person))

agt(translate(agt>thing,gol>language,obj>information,src>language),
computer(icl>machine))

agt(run(icl>act(agt>volitional thing)), car(icl>vehicle))

agt(break(agt>thing,obj>thing), explosion(icl>event))

John breaks ...

computer translates ...

car runs ...

explosion breaks ...

Related Relations

An agent is different from **cag** in that an agent initiates the action, whereas a co-agent initiates a different, accompanied action.

An agent is different from **ptn** in that an agent is the focused initiator of the action, whereas a partner is a non-focused initiator.

An agent is different from **con** in that an agent is the focused initiator of the action, whereas a condition is an indirect, usually unfocused, influence on the action.

and (conjunction)

And defines a conjunctive relation between concepts.

and (uw, uw)

Syntax

and ["<Compound UW-ID>"] (" {<UW1>|<Compound UW-ID>} ", " {<UW2>|<Compound UW-ID>} ")

Detailed definition

A conjunction is defined as the relation between:

UW1 – a concept, and

UW2 – another concept,

where:

- The UWs are different, and
- UW1 and UW2 are seen as grouped together, and
- what is said of UW1 is also said of UW2.

Examples and readings

and(quickly, easily)

and(sing(agt>person), dance(agt>person))

and(Mary(icl>person), John(icl>person))

... easily and quickly

... singing and dancing

... John and Mary

Related Relations

A conjunction is different from **or** in that with **and** things are grouped together to say the same thing about both of them, whereas with **or** we separate them to indicate that what is true about one is not true about the other.

A conjunction is different from **cag** in that when the agents are conjoined, both initiate an explicit event, whereas with **cag**, the co-agent initiates an implicit event.

A conjunction is different from **ptn** in that when the agents and partners are conjoined, both are in focus, whereas with **ptn**, the partner is not in focus (as compared to the agent).

A conjunction is different from **coo** and **seq** in meaning, although in many cases the same expressions can be used for both. A conjunction only means that terms are grouped together; no information about time is implied. **Coo**, on the other hand, means that the terms are in the same time, whether they are considered to be grouped together or not. In turn, **seq** means that the terms are ordered in time, one after the other.

aoj (thing with attribute)

Aoj defines a thing that is in a state or has an attribute.

aoj (uw(aoj>thing), thing)

aoj (thing, thing)

aoj (be, thing)

Syntax

aoj ["<Compound UW-ID>"] (" {<UW1>|<Compound UW-ID>} ", " {<UW2>|<Compound UW-ID>} ")

Detailed definition

A thing with an attribute is defined as the relation between:

UW1 – a state or a thing which represents a state, and

UW2 – a thing,

where:

- UW1 is an attribute or state of UW2, or
- UW1 is a state associated with UW2.

Examples and readings

aoj(red(aoj>thing), leaf(pof>plant))	... <u>leaf is red.</u>
aoj(available, information)	This <u>information is available</u> for ...
aoj(nice, ski(agt>person))	<u>Skiing is nice.</u>
aoj(teacher(icl>occupation), John(icl>person))	<u>John is a teacher.</u>
aoj(have(aoj>thing,obj>thing), I)	<u>I have</u> a pen.
obj(have(aoj>thing,obj>thing), pen(icl>writing instrument))	
aoj(know(aoj>thing,obj>thing), John(icl>person))	<u>John knows</u> ...
aoj:01(difficult(aoj>thing,obj>thing), it)	It is difficult for John.
aoj(:01, John(icl>person))	

Related Relations

A thing with an attribute is different from **mod** in that **mod** gives some restriction, whereas **aoj** gives a state or characteristic.

A thing with an attribute is different from **ben** in that a beneficiary is quite independent from a focused event or state. This event or state can be considered as exerting a good or bad influence, whereas **aoj** has a closer relation and can be considered as describing a state or characteristic.

A thing with an attribute is different from **obj** in that **obj** defines a thing which is directly affected by an action or phenomenon, whereas, **aoj** defines a thing in a state.

bas (basis for expressing a degree)

Bas defines a thing used as the basis (standard) for expressing a degree.

bas (degree, thing)

Syntax

bas [“:”<Compound UW-ID>] (“ {<UW1>|“:”<Compound UW-ID>} “,” {<UW2>|“:”<Compound UW-ID>} “)”

Detailed definition

A basis is defined as the relation between:

UW1 – a degree, and

UW2 – a thing,

where:

- UW1 is a degree expressing similarity or difference, such as “more”, “most”, “less”, “same”, “similar”, “as much as”, “at least” etc., and
- UW2 is something used as the basis for evaluating the characteristics or quantity of some other (focused) thing.

Examples and readings

bas(more(aoj>thing), 7)	Ten is three <u>more than seven.</u>
bas(more(icl>how), Jack(icl>person))	Betty weighs <u>more than Jack (does).</u>
bas(same(icl>how), girl(icl>female person).@pl)	We treat boys exactly <u>the same as girls</u>
bas(at least, :01)	It'll <u>cost at least 500 dollars.</u>
qua:01(dollar(icl>money).@pl, 500)	
man(beautiful, more(icl>how))	A tulip is <u>more beautiful than a rose</u>
bas(more(icl>how), rose(icl>flower))	
aoj(:01, John(icl>person))	John is <u>more quiet than shy.</u>
man:01(quiet(aoj>thing), more(icl>how))	
bas:01(more(icl>how), shy(aoj>thing,mod<thing))	

ben (beneficiary)

Ben defines an indirectly related beneficiary or victim of an event or state.

ben (occur, thing)

ben (do, thing)

ben (uw(aoj>thing), thing)

Syntax

ben [":<Compound UW-ID>] "(" {<UW1>|":<Compound UW-ID>} "," {<UW2>|":<Compound UW-ID>} ")

Detailed Definition

A beneficiary is defined as the relation between:

UW1 – an event or state, and

UW2 – a thing,

where:

- UW2 is thought of as being indirectly affected by UW1, as the beneficiary or victim.

Examples and readings

ben(give(agt>thing,gol>thing,obj>thing)),
Mary(icl>person))

To give ... for Mary.

ben(good(aoj>thing,mod<thing)),
John(icl>person))

It is good for John to ...

Related Relations

A beneficiary is different from **aoj** in that **aoj** has a close relation and can be considered as describing a state characteristic, whereas a beneficiary is quite independent from a focused event or state, but this event or state can be considered as exerting a good or bad influence.

cag (co-agent)

Cag defines a thing not in focus that initiates an implicit event that is done in parallel.

cag (do, thing)

Syntax

cag [":<Compound UW-ID>] "(" {<UW1>|":<Compound UW-ID>} "," {<UW2>|":<Compound UW-ID>} ")

Detailed definition

A co-agent is defined as the relation between:

UW1 - an action, and

UW2 - a thing

where:

- There is an implicit action that is independent of, but accompanies, UW1, and
- UW2 is thought of as initiating the implicit action, and
- UW2 and the implicit action are seen as not being in focus (as compared to the agent's action).

Examples and readings

cag(walk(icl>do), John(icl>person))
cag(live(icl>do), aunt(icl>person))

To walk with John

To live with ... aunt

Related relations

A co-agent is different from **agt** in that differing independent actions occur for an agent and a co-agent. Moreover, an agent and its action are in focus, while a co-agent and its action are not in focus.

A co-agent is different from the **ptn** in that the co-agent initiates an action that is independent of an agent's action, whereas a partner initiates the same action together with an agent.

A co-agent is different from **con** in that a co-agent initiates a non-focused action, whereas a condition is an indirect influence on the focused action.

cao (co-thing with attribute)

Cao defines a thing not in focus that is in a parallel state.

cao (uw(aoj>thing), thing)

cao (thing, thing)

Syntax

cao [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “”

Detailed definition

A co-thing with an attribute is defined as the relation between:

UW1 – a state or a thing which represents a state, and

UW2 – a thing,

where:

- There is an implicit state that is independent of, but accompanies, UW1, and
- UW2 is in an implicit state, or
- UW2 is associated with an implicit state.

Examples and readings

cao(exist(icl>be), you)

... be with you

Related relations

A co-thing with an attribute is different from **aoj** in that there is a different, independent state for the thing with an attribute and a co-thing with an attribute, respectively.

cnt (content)

Cnt defines an equivalent concept.

cnt (thing, thing)

Syntax

cnt [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “”

Detailed Definition

A content is defined as the relation between:

UW1 – a thing, and

UW2 – a thing,

where:

- UW2 is a content or explanation of UW1.

Examples and readings

cnt(UNL(icl>Universal Networking Language),
Universal Networking Language)

UNL, Universal Networking Language

cnt(Internet(icl>communication network),
amalgamation(icl>harmony))

The Internet: an amalgamation

cnt(language generator,
deconverter.@double_quote)

a language generator “deconverter”...

cob (affected co-thing)

Cob defines a thing that is directly affected by an implicit event done in parallel or an implicit state in parallel.

cob (occur, thing)

cob (do, thing)

cob (uw(aoj>thing,obj>thing), thing)

Syntax

cob [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “”

Detailed Definition

A “co-object” is defined as the relation between:

UW1 – an event or state, and

UW2 – a thing,

where:

- UW2 is thought of as directly affected by an implicit event done in parallel or an implicit state in parallel.

Examples and readings

cob(die(obj>living thing), Mary(icl>person))	... <u>dead with Mary</u>
obj(injure(icl>hurt(agt>thing,obj>living thing), John(icl>person))	John <u>was injured</u> in the accident <u>with his friends</u>
cob(injure(icl>hurt(agt>thing,obj>living thing), friend(icl>comrade).@pl)	
pos(friend(icl>comrade).@pl, he)	

Related Relations

A co-object is different from **obj** in that the **obj** is in focus, whereas **cob** is related to a second, non-focused implicit event or state.

con (condition)

Con defines a non-focused event or state that conditions a focused event or state.

```
con (occur, occur)
con (occur, do)
con (occur, uw(aoj>thing))
con (do, occur)
con (do, do)
con (do, uw(aoj>thing))
con (uw(aoj>thing), occur)
con (uw(aoj>thing), do)
con (uw(aoj>thing), uw(aoj>thing))
```

Syntax

con ["<Compound UW-ID>"] (" {<UW1>|<Compound UW-ID>} ", " {<UW2>|<Compound UW-ID>} ")

Detailed definition

A condition is defined as the relation between:

UW1 – a focused event or state, and

UW2 – a conditioning event or state,

where:

- UW1 and UW2 are different and
- UW2 is thought of as having an indirect or external role in making UW1 happen, that is, as some conditioning or inhibiting factor (real or hypothesized) that influences whether or when UW1 can happen.

Examples and readings

aoj:01(tired(aoj>thing,mod<thing), you)	<u>If you are tired</u> , we will <u>go</u> straight home.
con(go(icl>move(agt>thing,gol>place,src>place)), :01)	

coo (co-occurrence)

Coo defines a co-occurrent event or state for a focused event or state.

```
coo (occur, occur)
coo (occur, do)
coo (occur, uw(aoj>thing))
coo (do, occur)
coo (do, do)
coo (do, uw(aoj>thing))
coo (uw(aoj>thing), occur)
coo (uw(aoj>thing), do)
coo (uw(aoj>thing), uw(aoj>thing))
```


Syntax

coo ["<Compound UW-ID>] "(" {<UW1>|<Compound UW-ID>} "," {<UW2>|<Compound UW-ID>} ")

Detailed definition

A co-occurrence is defined as the relation between:

UW1 – a focused event or state, and

UW2 – a co-occurrent event or state,

where:

- UW1 and UW2 are different, and
- UW1 occurs or is true at the same time as UW2.

Examples and readings

coo(run(icl>act(agt>volitional thing)),	... was <u>crying while running</u>
cry(icl>weep(agt>volitional thing))	
coo(red(aoj>thing,mod<thing),	... <u>is red</u> while ... <u>is hot</u>
hot(aoj>thing,mod<thing))	

Related Relations

A co-occurrence is different from **seq** in that **seq** describes events or states that do not occur at the same time, but one after the other, whereas **coo** describes events that occur simultaneously.

A co-occurrence is different from **tim** in that **coo** relates the times of events or states with other events or states, whereas **tim** relates events or states directly with points or intervals of time.

dur (duration)

Dur defines a period of time during which an event occurs or a state exists.

dur (occur, period)
dur (occur, event)
dur (occur, state)
dur (occur, occur)
dur (occur, do)
dur (occur, uw(aoj>thing))
dur (do, period)
dur (do, event)
dur (do, state)
dur (do, occur)
dur (do, do)
dur (do, uw(aoj>thing))
dur (uw(aoj>thing), period)
dur (uw(aoj>thing), event)
dur (uw(aoj>thing), state)
dur (uw(aoj>thing), occur)
dur (uw(aoj>thing), do)
dur (uw(aoj>thing), uw(aoj>thing))

Syntax

dur ["<Compound UW-ID>] "(" {<UW1>|<Compound UW-ID>} "," {<UW2>|<Compound UW-ID>} ")

Detailed definition

A duration is defined as the relation between:

UW1 – an event or a state, and

UW2 – a period during which the event or state continues.

Examples and readings

dur(work(agt>person), hour(icl>period))	... <u>work nine hours</u> (a day)
qua(hour(icl>period), 9)	
dur(talk(icl>express(agt>thing,gol>person,obj>thing),	... <u>talk</u> ... <u>during meeting</u>
meeting(icl>event)	... <u>come during</u> (my) <u>absence</u>
dur(come(icl>move(agt>thing,gol>place,src>place),	

absence(icl>state))

fmt (range: from-to)

Fmt defines a range between two things.

fmt (thing, thing)

Syntax

fmt [“.”<Compound UW-ID>] “(“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “)”

Detailed definition

A range (from-to) is defined as the relation between:

UW1 – a range-initial thing, and

UW2 – a range-final thing,

where:

- The UWs are different, and
- UW2 describes the beginning of a range and UW1 describes the end.

Examples and readings

fmt(a(icl>letter), z(icl>letter))

the alphabets from a to z

fmt(Osaka(icl>city), New York(icl>city))

the distance from Osaka to New York

fmt(Monday(icl>day), Friday(icl>day))

the weekdays from Monday to Friday

Related Relations

A range is different from **src** and **gol** in that for **src** and **gol** the initial and final states of certain **obj** are characterized with respect to some event, whereas **fmt** makes a similar characterization but without linking the endpoints of a range to some event.

A range is different from **plf** and **plt** or **tmf** and **tmt** in that **fmt** defines endpoints of a range without reference to any sort of event, whereas **plf**, **plt**, **tmf** and **tmt** delimit events.

frm (origin)

Frm defines an origin of a thing.

frm (thing, thing)

Syntax

frm [“.”<Compound UW-ID>] “(“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “)”

Detailed definition

An origin is defined as the relation between:

UW1 – a thing, and

UW2 – an origin of the thing,

where:

- UW2 describes the origin such as the original position of UW1.

Examples and readings

frm(visitor(icl>person), Japan(icl>country))

a visitor from Japan

gol (goal: final state)

Gol defines a final state of object or a thing finally associated with the object of an event.

gol (occur(gol>thing), thing)

gol (do(gol>thing), thing)

Syntax

gol [“.”<Compound UW-ID>] “(“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “)”

Detailed definition

A final state is defined as the relation between:

UW1 – an event, and

UW2 – a state or thing,

where:

- UW2 is the specific state describing the **obj** (of UW1) at the end of UW1, or
- UW2 is a thing that is associated with the **obj** (of UW1) and the end of UW1.

Examples and readings

gol(change(gol>thing,obj>thing,src>thing),
red(aoj>thing,mod<thing))
gol(deposit(icl>save(agt>thing,obj>thing)),
account(icl>record))

the lights changed from green to red

millions were deposited in a Swiss bank
account

Related Relations

A final state is different from **tmf** and **plf** in that **gol** describes qualitative characteristics and not time or place.

A final state is different from **src** in that **gol** describes the characteristics of the **obj** at the final state of the event.

ins (instrument)

Ins defines an instrument to carry out an event.

ins (do, concrete thing)

Syntax

ins [“:<Compound UW-ID>] “(“ {<UW1>|“:<Compound UW-ID>” “,” {<UW2>|“:<Compound UW-ID>” “)”

Detailed definition

An instrument is defined as the relation between:

UW1 – an event, and

UW2 – a concrete thing,

where:

- UW2 specifies the concrete thing that is used in order to make UW1 happen.

Examples and readings

ins(look(agt>thing,obj>thing),
telescope(icl>optical instrument)
ins(write(icl>express(agt>thing,obj>thing)),
pencil(icl>stationery))
ins(cut(agt>thing,obj>thing,opl>thing),
scissors(icl>cutley))

look at stars through [with] a telescope

write [draw] with a pencil

He cut the string with a pair of scissors

Related Relations

An instrument is different from **man** in that **man** describes an event as a whole, whereas **ins** characterizes one of the components of the event: the use of the instrument.

An instrument is different from **met** in that **met** is used for abstract things (abstract means or methods), whereas “ins” is used for concrete things.

man (manner)

Man defines a way to carry out an event or the characteristics of a state.

man (occur, how)

man (do, how)

man (uw(aoj>thing), how)

Syntax

man [“:<Compound UW-ID>] “(“ {<UW1>|“:<Compound UW-ID>” “,” {<UW2>|“:<Compound UW-ID>” “)”

Detailed definition

A “manner” is defined as the relation between:

UW1 – an event or state, and

UW2 – a manner,

where:

- The UWs are different, and
- UW1 is done or exists in a way characterized by UW2.

Examples and readings

man(move(agt>thing, gol>place, src>place), quickly)	<u>move quickly</u>
man(visit(agt>thing, obj>thing)), often)	I <u>often visit</u> him.
man(beautiful, very(icl>how))	it is <u>very beautiful</u> .

Related Relations

A manner is different from **ins** or **met** in that **met** describes how an event is carried out in terms of the instruments or component steps of the event, whereas **man** describes other quantitative or qualitative characteristics of the event as a whole.

met (method or means)

Met defines a means to carry out an event.

met (do, abstract thing)

met (do, do)

Syntax

met [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “”)

Detailed definition

A “method or means” is defined as the relation between:

UW1 – an event, and

UW2 – an abstract thing or an action,

where:

- UW2 specifies the abstract thing used or the steps carried out in order to make UW1 happen.

Examples and readings

met(solve(icl>resolve(agt>thing, obj>thing)), dynamics(icl>science))	... <u>solve</u> ... with <u>dynamics</u>
met(solve(icl> resolve(agt>thing, obj>thing)), algorithm(icl>method))	... <u>solve</u> ... using ... <u>algorithm</u>
met(separate(agt>thing, obj>thing, src>thing)), cut(agt>thing, obj>thing, opl>thing))	... <u>separate</u> ... by <u>cutting</u> ...

Related Relations

A method or means is different from **man** in that **man** describes an event as a whole, whereas **met** characterizes the component steps, procedures or instruments of the event.

A method or means is different from **ins** in that **met** is used for abstract things (abstract means or methods), whereas **ins** is used for concrete things.

mod (modification)

Mod defines a thing that restricts a focused thing.

mod (thing, thing)

mod (thing, uw(mod>thing))

Syntax

mod [“:”<Compound UW-ID>] “(“ {<UW1>|“:”<Compound UW-ID>} “,” {<UW2>|“:”<Compound UW-ID>} “)”

Detailed definition

A “modification” is defined as the relation between:

UW1 – a focused thing, and

UW2 – a thing that restricts UW1 in some way.

Examples and readings

mod(story(icl>tale), whole(mod<thing))	the <u>whole story</u>
mod(plan(icl>idea), master(mod<thing))	a <u>master plan</u>
mod(part(pof>thing), main(aoj>thing))	the <u>main part</u>
qua(block(icl>concrete thing), 3)	... three <u>blocks of ice</u> ...
mod(ice(icl>solid), block(icl>concrete thing))	

Related Relations

A modification is different from **aoj** in that **aoj** describes a state or characteristic of a thing, whereas **mod** merely indicates a restriction, which might indirectly suggest some characteristics of the thing described. Most **mod** relations require a paraphrase introducing some implicit event to become clearer, and even then many possibilities are usually available.

A modification is different from **man** in that **man** describes a way to carry out an event or the characteristics of a state.

nam (name)

Nam defines a name of a thing.

nam (thing, thing)

Syntax

nam [“:”<Compound UW-ID>] “(“ {<UW1>|“:”<Compound UW-ID>} “,” {<UW2>|“:”<Compound UW-ID>} “)”

Detailed definition

A name is defined as the relation between:

UW1 – a thing, and

UW2 – a thing used as a name,

where:

- UW2 is a name of UW1.

Examples and readings

Nam(tower(icl>building), Tokyo(icl>city))	<u>Tokyo tower</u>
---	--------------------

obj (affected thing)

Obj defines a thing in focus that is directly affected by an event or state.

obj (occur, thing)

obj (do, thing)

obj (be, thing)

obj (uw(aoj>thing,obj>thing), thing)

Syntax

obj [“:”<Compound UW-ID>] “(“ {<UW1>|“:”<Compound UW-ID>} “,” {<UW2>|“:”<Compound UW-ID>} “)”

Detailed Definition

An affected thing is defined as the relation between:

UW1 – an event or state, and

UW2 – a thing,

where:

- UW2 is thought of as directly affected by an event or state.

Examples and readings

obj(move(gol>place,obj>thing,src>place), table(icl>furniture))	the <u>table moved</u> .
obj(melt(gol>thing,obj>thing), sugar(icl>seasoning))	the <u>sugar melts</u> into ...
obj(cure(agt>thing,obj>thing), patient(icl>person))	to <u>cure the patient</u> .
obj(have(aoj>thing,obj>thing), pen(icl>writing instrument))	I <u>have a pen</u> .

Related Relations

An affected thing is different from **cob** in that **obj** is in focus, whereas **cob** is related to a second, non-focused implicit event or state.

opl (affected place)

Opl defines a place in focus affected by an event.

opl (do, place)

Syntax

opl [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “”

Detailed Definition

An affected place is defined as the relation between:

UW1 – an event, and

UW2 – a place or thing defining a place,

where:

- UW2 is the specific place where the change described by UW1 is directed, or
- UW2 is a place that is seen as being affected during the event.

Examples and readings

opl(pat(icl>touch(agt>thing,obj>thing,opl>thing)), shoulder(pof>trunk))	... pat ... on shoulder
opl(cut(agt>thing,obj>thing,opl>thing), middle(icl>place))	... cut ... in middle

Related Relations

An affected place is different from **obj** and **cob** in that what is affected by the event is a place rather than other kinds of things.

An affected place is different from **plc** in that an affected place is characterized by the event, while the physical and logical place defines the environment in which the event happens.

or (disjunction)

Or defines a disjunctive relation between two concepts.

or (uw, uw)

Syntax

or [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “”

Detailed definition

A disjunction is defined as the relation between:

UW1 – a thing, and

UW2 – a concept,

where:

- The UWs are different, and

- Some description is true for either UW1 or UW2 (but not both), or
- Some description is true for either UW1 or UW2 (and perhaps both).

Examples and readings

or(stay(icl>do), leave(icl>do))
 or(red(icl>color), blue(icl>color))
 or(John(icl>person), Jack(icl>person))

Will you stay or leave?
 Is it red or blue?
 Who is going to do it, John or Jack?

Related Relations

A disjunction is different from a conjunction in that the items of disjunction are grouped in order to say that something is true for one or the other, whereas in a conjunction they are grouped to say that the same is true for both. A disjunction in formal logic permits three situations for it to be true: 1) it is true for UW1, 2) it is true for UW2, 3) it is true for both. On the other hand, a conjunction only permits the third situation.

per (proportion, rate or distribution)

Per defines a basis or unit of proportion, rate or distribution.

per (thing, thing)

Syntax

per [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “)”

Detailed definition

A proportion, rate or distribution is defined as the relation between:

UW1 – a quantity, and

UW2 – a quantity, or a thing seen as a quantity,

where:

- UW1 and UW2 form a proportion, where UW1 is the numerator and UW2 is the denominator, or
- UW2 is the basis or unit for understanding UW1, or
- Each UW expresses a different dimension, of size, for example.

Examples and readings

per(hour(icl>period), day(icl>period))
 qua(hour(icl>period), 8)
 per(time(icl>frequency), week(icl>period))
 qua(time(icl>frequency), 2)

eight hours a day

... twice a week

plc (place)

Plc defines a place where an event occurs, or a state that is true, or a thing that exists.

plc (occur, thing)

plc (do, thing)

plc (be, thing)

plc (uw(aoj>thing), thing)

plc (thing, thing)

Syntax

plcl [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “)”

Detailed definition

A place is defined as the relation between:

UW1 – an event, state, or thing, and

UW2 – a place or thing understood as a place.

Examples and readings

plc(cook(icl>do), kitchen(pof>building))
 plc(sit(icl>do), beside(icl>place))

... cook ... in the kitchen
 ... sit beside me

plc(cool(icl>cold), here(icl>place))

It's cool here.

Related Relations

A place is different from **plf** and **plt** or **src** and **gol** in that **plc** describes a place with respect to an event as a whole, whereas these other relations describe the position with respect to parts of an event.

A place is different from **opl** in that **plc** is not seen as being modified by an event but merely as a reference point for characterizing it, whereas **opl** is seen as being modified.

plf (initial place)

Plf defines a place where an event begins or a state that becomes true.

plf (occur, thing)

plf (do, thing)

plf (uw(aoj>thing), thing)

Syntax

plf ["<Compound UW-ID>"] (" {<UW1>|<Compound UW-ID>} ", " {<UW2>|<Compound UW-ID>} ")

Detailed definition

An "initial place" (or "place-from") is defined as the relation between:

UW1 – an event or state, and

UW2 – a place or thing defining a place,

where:

- UW2 is the specific place where UW1 started, or
- UW2 is the specific place from where UW1 is true.

Examples and readings

plf(come(icl>do), home(icl>place))

plf(deep(aoj>thing), there(icl>place))

... come from home

The sea is deep from there to here.

Related Relations

An initial place is different from **plc** in that **plc** describes events or states taken as a whole, whereas **plf** describes only the initial part of an event or state.

An initial place is different from **plt** in that **plt** describes the final part of an event or state, whereas **plf** describes the initial part of an event or state.

An initial place is different from **src** in that **plf** describes the place where the event began, whereas **src** describes the initial state of the object.

plt (final place)

Plt defines a place where an event ends or a state that becomes false.

plt (occur, thing)

plt (do, thing)

plt (uw(aoj>thing), thing)

Syntax

plt ["<Compound UW-ID>"] (" {<UW1>|<Compound UW-ID>} ", " {<UW2>|<Compound UW-ID>} ")

Detailed definition

A final place is defined as the relation between:

UW1 – an event or state, and

UW2 – a place or thing defining a place,

where:

- UW2 is the specific place where UW1 ended, or
- UW2 is the specific place where UW2 becomes false.

Examples and readings

plt(travel(icl>do), Boston(icl>city))
plt(deep(aoj>thing), here(icl>place))

I'm travelling up to Boston
The sea is deep from there to here

Related Relations

A final place is different from **plc** in that **plc** describes events or states taken as a whole, whereas **plt** describes only the final part of an event.

A final place is different from **plf** in that **plt** describes the final part of an event or state, whereas **plf** describes the initial part of an event.

A final place is different from **gol** in that **plt** describes the place where an event or state ended, whereas **gol** describes the final state of the object.

pof(part-of)

Pof defines a concept of which a focused thing is a part.

pof (thing, thing)

Syntax

pof [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “”

Detailed definition

Part-of is defined as the relation between:

UW1 – a partial thing, and

UW2 – a whole thing,

where:

- UW1 is a part of UW2.

Examples and readings

Pof(wing(icl>limb), bird(icl>animal))

Bird's wing.

pos (possessor)

Pos defines the possessor of a thing.

pos (thing, volitional thing)

Syntax

pos [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “”

Detailed definition

A possessor is defined as the relation between:

UW1 – a thing or a place, and

UW2 - a human or non-human, seen as a volitional thing

where:

- UW2 is a possessor of UW1.

Examples and readings

pos(dog(icl>animal), John(icl>person))
pos(book(icl>concrete thing), I)

John's dog
my book

ptn (partner)

Ptn defines an indispensable non-focused initiator of an action

ptn (do, thing)

Syntax

ptn [“.”<Compound UW-ID>] (“ {<UW1>|“.”<Compound UW-ID>} “,” {<UW2>|“.”<Compound UW-ID>} “”

Detailed definition

A partner is defined as the relation between:

UW1 - an action, and

UW2 - a human or non-human, seen as a volitional thing

where:

- UW2 is thought of as having a direct role in making an indispensable part of UW1 happen, and
- UW1 is the same, collaborative event as that initiated by the agent, and
- UW2 is seen as not being in focus (as compared to the agent).

Examples and readings

ptn(compet(e)(icl>do), John(icl>person))	... <u>compete with John</u>
ptn(share(icl>do(obj>thing)), poor(icl>person))	... <u>share ... with the poor</u>
ptn(collaborate(icl>do), he)	... <u>collaborate with him</u> ...

Related Relations

A partner is different from **agt** in that an agent and its event are in focus, while a partner and its event are not in focus.

A partner is different from **cag** in that a co-agent initiates an event that is independent of an agent's event, whereas a partner initiates the same event together with an agent.

A partner is different from **con** in that a partner initiates the same event as an agent does, whereas a condition only has an indirect influence on that event.

pur (purpose or objective)

Pur defines the purpose or objective of an agent of an event or a purpose of a thing that exists.

pur (occur, occur)
pur (occur, do)
pur (do, occur)
pur (do, do)
pur (occur, thing)
pur (do, thing)
pur (thing, occur)
pur (thing, do)
pur (thing, thing)

Syntax

pur [“:”<Compound UW-ID>] (“ {<UW1>|“:”<Compound UW-ID>} “,” {<UW2>|“:”<Compound UW-ID>} “)”

Detailed definition

A purpose or objective is defined as the relation between:

UW1 – a thing or an event, and

UW2 – a thing or an event,

where:

- The UWs are different, and

When UW1 is an event:

- UW2 specifies the agent's purpose or objective, or

- UW2 specifies the thing (object, state, event, etc.) that the agent desires to attain by carrying out UW1, or

When UW1 is not an event:

- UW2 is what UW1 is to be used for.

Examples and readings

pur(come(icl>do), see(icl>do(obj>thing)))	... <u>come to see you</u>
pur(work(icl>do), money(icl>do))	... <u>work for money</u>
pur(budget(icl>expense), research(icl>do))	our <u>budget for research</u>

Related Relations

A purpose or objective is different from **gol** in that **pur** describes the desires of an agent, whereas **gol** describes

the state of the object at the end of the event.

A purpose or objective is different from **man** and **met** in that **pur** describes the reason why the event is being carried out, while **man** and **met** describe how it is being carried out.

qua (quantity)

Qua defines the quantity of a thing or unit.

qua (thing, quantity)

Syntax

qua [": "<Compound UW-ID>] "(" {<UW1>|": "<Compound UW-ID>} ", " {<UW2>|": "<Compound UW-ID>} ")

Detailed definition

A quantity is defined as the relation between:

UW1 – a thing, and

UW2 – quantity,

where:

- UW2 is the number or amount of UW1.

Examples and readings

qua(cup(icl>tableware), 2))

mod(coffee(icl>beverage), cup(icl>tableware))

qua(kilogram(icl>unit), many(aoj>thing))

qua(dog(icl>animal), 2)

Two cups of coffee

many kilograms

two dogs

Related Relations

A quantity is different from **per** in that a quantity is an absolute number or amount, whereas **per** is a number or amount relative to some unit of reference (time, distance, etc.).

A quantity is also used to express iteration, or the number of times an event or state occurs.

rsn (reason)

Rsn defines a reason why an event or a state happens.

rsn (occur, thing)

rsn (do, thing)

rsn (occur, occur)

rsn (occur, do)

rsn (do, occur)

rsn (do, do)

rsn (occur, uw(aoj>thing))

rsn (do, uw(aoj>thing))

rsn (uw(aoj>thing), occur)

rsn (uw(aoj>thing), do)

rsn (uw(aoj>thing), thing)

rsn (uw(aoj>thing), uw(aoj>thing))

Syntax

rsn [": "<Compound UW-ID>] "(" {<UW1>|": "<Compound UW-ID>} ", " {<UW2>|": "<Compound UW-ID>} ")

Detailed definition

A reason is defined as the relation between:

UW1 – an event or state, and

UW2 – a reason for an event or state,

where:

- UW2 is a reason why UW1 happens.

Examples and readings

rsn(go(icl>do), rain(icl>weather))	... didn't <u>go because of the rain</u>
agt:01(arrive(icl>do), Mary(icl>person))	They can <u>start because Mary arrived.</u>
rsn(start(icl>do(obj>thing)), :01)	
rsn(known(aoj>thing), beauty(icl>abstract thing))	a city <u>known for its beauty</u>
mod(city(icl>region), known(aoj>thing))	
mod(beauty(icl>abstract thing), city(icl>region))	

scn (scene)

Scn defines a virtual world where an event occurs, or state is true, or a thing exists.

scn (do, thing)
scn (occur, thing)
scn (uw(aoj>thing), thing)
scn (thing, thing)

Syntax

scn ["<Compound UW-ID>"] (" {<UW1>|<Compound UW-ID>} ", " {<UW2>|<Compound UW-ID>} ")

Detailed definition

A scene is defined as the relation between:

UW1 – an event or state or thing, and

UW2 – an abstract or metaphorical thing understood as a place,
where:

- The UWs are different, and
- UW1 is or happens in a place characterized by UW2.

Examples and readings

scn(win(icl>do(obj>thing)), contest(icl>event))	... <u>win a prize in a contest</u>
scn(appear(icl>occur), program(icl>thing))	... <u>appear on a TV program</u>
scn(play(icl>do), movie(icl>entertainment))	... play in movie

Related Relations

A scene is different from **plc** in that the reference place for **plc** is in the real world, whereas for **scn** it is an abstract or metaphorical world.

seq (sequence)

Seq defines a prior event or state of a focused event or state.

seq (occur, occur)
seq (occur, do)
seq (do, occur)
seq (do, do)
seq (occur, uw(aoj>thing))
seq (do, state)
seq (uw(aoj>thing), occur)
seq (uw(aoj>thing), do)

Syntax

seq ["<Compound UW-ID>"] (" {<UW1>|<Compound UW-ID>} ", " {<UW2>|<Compound UW-ID>} ")

Detailed definition

A “sequence” is defined as the relation between:

UW1 – a focused event or state,

UW2 – a prior event or state,

where:

- The UWs are different, and

- UW1 occurs or is true after UW2.

Examples and readings

seq(Leap(icl>do), look(icl>do))	<u>Look before</u> you <u>leap</u> .
seq(red(aoj>thing), green(aoj>thing))	It was <u>green and then red</u> .
seq(take off(icl>do(obj>thing)), come in(icl>do))	She <u>came in and took her coat off</u> .

Related Relations

A sequence is different from **coo** in that **seq** describes events or states that do not occur at the same time, but one after the other, whereas **coo** describes events that occur simultaneously.

A sequence is different from **bas** in that **seq** describes events or states in terms of order in time, whereas **bas** describes things or states in terms of qualitative differences or similarities.

src (source: initial state)

Src defines the initial state of an object or thing initially associated with the object of an event.

src (occur, thing)
src (do, thing)

Syntax

src [":"<Compound UW-ID>] "(" {<UW1>|":"<Compound UW-ID>} "," {<UW2>|":"<Compound UW-ID>} ")"

Detailed definition

An initial state is defined as the relation between:

UW1 – an event, and

UW2 – a state or thing,

where:

- UW2 is the specific state describing the object of UW1 at the beginning of UW1, or
- UW2 is a thing that is associated with the object of UW1 at the beginning of UW1.

Examples and readings

src(change(icl>occur), red(aoj>thing))	The lights <u>changed from green</u> to red.
src(withdraw(icl>do(obj>thing)), stove(icl>furniture))	I quickly <u>withdrew</u> my hand <u>from the stove</u> .

Related Relations

An initial state is different from **tmf** and **plf** in that **src** describes qualitative characteristics and not time or place.

An initial state is different from **gol** in that **gol** describes the characteristics of the object at the final state of the event.

tim (time)

Tim defines the time an event occurs or a state is true.

tim (occur, time)
tim (do, time)
tim (be, time)
tim (uw(aoj>thing), time)

Syntax

tim [":"<Compound UW-ID>] "(" {<UW1>|":"<Compound UW-ID>} "," {<UW2>|":"<Compound UW-ID>} ")"

Detailed definition

Time is defined as the relation between:

UW1 – an event or state, and

UW2 – a time,

where:

- UW1, taken as a whole, occurs at the time indicated by UW2.

Examples and readings

tim(leave(icl>do), Tuesday(icl>time))	... <u>leave on Tuesday</u>
tim(do(obj>thing), o'clock(icl>time))	... <u>do ... at ... o'clock</u>
tim(start(icl>do), come(icl>do))	Let's <u>start when ... come</u>

Related Relations

Time is different from **tmf** and **tmt** in that time characterizes the event or state as a whole, whereas **tmf** and **tmt** describe only parts of the event.

Time is different from **coo** and **seq** in that time does not describe states and events relatively, with respect to each other, but with respect to certain points in time.

tmf (initial time)

Tmf defines the time an event starts or a state becomes true.

tmf (occur, time)
tmf (do, time)
tmf (uw(aoj>thing), time)

Syntax

tmf [":"<Compound UW-ID>] "(" {<UW1>|":"<Compound UW-ID>} "," {<UW2>|":"<Compound UW-ID>} ")"

Detailed definition

Initial time is defined as the relation between:

UW1 – an event or state, and

UW2 – a time,

where:

- UW2 specifies the time at which UW1 starts, or
- UW2 specifies the time at which UW1 became/becomes true.

Examples and readings

tmf(work(icl>do), morning(icl>time))	... <u>work from morning</u> to [till] night
tmf(change(icl>occur), live(icl>do))	... <u>has changed</u> ... <u>since I have lived</u> here.

Related Relations

Initial time is different from **tim** in that **tmf** expresses the time at the beginning of the event or state whereas **tim** expresses the time for the event taken as a whole.

Initial time is different from **src** in that **tmf** expresses the time at the beginning of the event or state whereas **src** expresses characteristics of the object at the beginning of the event.

Initial time is different from **tmt** in that **tmf** expresses the time at the beginning of the event or state whereas **tmt** expresses the time at its end.

tmt (final time)

Tmt defines a time an event ends or a state becomes false.

tmt (occur, time)
tmt (do, time)
tmt (uw(aoj>thing), time)

Syntax

tmt [":"<Compound UW-ID>] "(" {<UW1>|":"<Compound UW-ID>} "," {<UW2>|":"<Compound UW-ID>} ")"

Detailed definition

Final time is defined as the relation between:

UW1 – an event or state, and

UW2 – a time,

where:

- UW2 specifies the time at which UW1 ends, or

- UW2 specifies the time at which UW1 became/becomes false.

Examples and readings

tmt(work(icl>do), night(icl>time))	... <u>work</u> from moning to <u>[till] night</u>
tmt(full(aoj>thing), tomorrow(icl>time))	... <u>be full till tomorrow</u>

Related Relations

Final time is different from **tim** in that **tmt** expresses the time at the end of the event or state, whereas **tim** expresses the time for the event taken as a whole.

Final time is different from **gol** in that **tmt** expresses the time at the end of the event or state, whereas **gol** expresses characteristics of the object at the end of the event.

Final time is different from **tmf** in that **tmt** expresses the time at the end of the event or state, whereas **tmf** expresses the time at the beginning of the event.

to (destination)

To defines the destination of a thing.

to (thing, thing)

Syntax

to [“:<Compound UW-ID>] (“ {<UW1>|“:<Compound UW-ID>} “,” {<UW2>|“:<Compound UW-ID>} “)”

Detailed definition

A destination is defined as the relation between:

UW1 – a thing, and

UW2 – a destination of the thing,

where:

- UW2 describes the destination such as the final position of UW1.

Examples and readings

to(train(icl>vehicle), London(icl>city))	<u>a train for London</u>
to(letter(icl>message), you)	<u>a letter to you</u>

via (intermediate place or state)

Via defines an intermediate place or state of an event.

via (occur(gol>thing,src>thing), thing)

via (do(gol>thing,src>thing), thing)

Syntax

via [“:<Compound UW-ID>] (“ {<UW1>|“:<Compound UW-ID>} “,” {<UW2>|“:<Compound UW-ID>} “)”

Detailed definition

An intermediate place or state is defined as the relation between:

UW1 – an event, and

UW2 – a place or state,

where:

- UW2 is the specific place or state describing the object of UW1 at some time in the middle of UW1,
- UW2 is a thing that describes a place or state that the object of UW1 passed by or through during UW1.

Examples and readings

via(go(icl>do), New York(icl>city))	... <u>go</u> ... <u>via New York</u>
via(bike(icl>do), Alps(icl>place))	... <u>bike</u> ... <u>through the Alps</u>
via(drive(icl>do), tunnel(icl>topography))	... <u>drive</u> ... <u>by way of the tunnel</u>

Related Relations

An intermediate place or state is different from **src**, **plf** and **tmf** in that these all refer to the beginning of an event,

whereas **via** describes the middle of an event.

An intermediate place or state is different from **gol**, **plt** and **tmt** in that these all refer to the end of an event, whereas **via** describes the middle of an event.

Chapter 3: Universal Words

A UW (Universal Word) represents simple or compound concepts. There are two classes of UWs:

- simple, unit concepts called “UWs” (Universal Words), and
- compound structures of binary relations grouped together and called “Compound UWs”. These are indicated with Compound UW-IDs, as described below.

3.1 UWs

3.1.1 Syntax of UW

A UW is made up of a character string (an English-language word) followed by a list of constraints. The meaning and function of each of these parts is described in the next section, on Interpretation.

The following expressions provide a more formal statement of the syntax of UWs.

<UW>	::= <Head Word> [<Constraint List>]
<Head Word>	::= <character>...
<Constraint List>	::= (“<Constraint> [“,” <Constraint>]... “”)
<Constraint>	::= <Relation Label> { “>” “<” } <UW> [<Constraint List>] <Relation Label> { “>” “<” } <UW> [<Constraint List>] [{ “>” “<” } <UW> [<Constraint List>]] ...
<Relation Label>	::= “agt” “and” “aoj” “obj” “icl” ...
<character>	::= “A” ... “Z” “a” ... “z” 0 1 2 ... 9 “_” “ ” “#” “!” “\$” “%” “=” “^” “~” “ ” “@” “+” “-” “<” “>” “?”

3.1.2 Interpretation

Head Word

The Head Word is an English word/compound word/phrase/sentence that is interpreted as a label for a set of concepts: the set made up of all the concepts that may correspond to that in English. A Basic UW (with no restrictions or Constraint List) denotes this set. Each Restricted UW denotes a subset of this set that is defined by its Constraint List. Extra UWs denote new sets of concepts that do not have English-language labels.

Thus, the Head Word serves to organize concepts and make it easier to remember which is which.

Constraints or Restrictions

The Constraint List restricts the interpretation of a UW to a subset or to a specific concept included within the Basic UW, thus the term “Restricted UWs”.

The Basic UW “drink”, without a Constraint List, includes the concepts of “putting liquids in the mouth”, “liquids that are put in the mouth”, “liquids with alcohol”, “absorb” and others.

The Restricted UW “drink(icl>do,obj>liquid)” denotes the subset of these concepts that includes “putting liquids in the mouth”, which in turn corresponds to verbs such as “drink”, “gulp”, “chug” and “slurp” in English.

The restrictions of Restricted UWs, their Constraint Lists, are Constraints. The Constraints that use the Relation Labels defined above can be seen as an abbreviated notation for full binary relations: drink(icl>do,obj>liquid) is the same as obj(drink(icl>do),liquid) which means something like “cases of drinking where the ‘obj’ is a liquid”. Every constraint in the Constraint List should use the Relation Labels listed in Appendix 2 and each of them should be sorted in alphabetical order.

The relation label “icl” can be omitted when it is repeated to restrict the upper concept. For instance, a UW like “xxx(icl>change(icl>occur))” can be simply defined as “xxx(icl>change>occur)”.

3.1.3 Types of UW

UWs, therefore, are character strings (words or expressions) that can be given specifications, attributes and Instance-IDs. Their function in the UNL system is to represent simple concepts. The three types of UWs, in order of practical importance, are:

- Basic UWs, which are bare Head Words with no Constraint List, for example:

go
take
house
state

- Restricted UWs, which are Head Words with a Constraint List, for example:

state(icl>express)
state(icl>country)
state(icl>abstract thing)
state(icl>government)

- Extra UWs, which are a special type of Restricted UW, for example:

ikebana(icl>flower arrangement)
samba(icl>dance)
soufflé(icl>food)

Basic UWs

Basic UWs are character strings that correspond to an English word. A basic UW denotes all the concepts that may correspond to those in English. They are used to structure the knowledge base and as a fallback method for establishing correspondences between different language words when more specific correspondences cannot be found.

Restricted UWs

Restricted UWs are by far the most important. Each Restricted UW represents a more specific concept, or subset of concepts. The Constraint List restricts the range of the concept that a Basic UW represents.

The Basic UW “drink”, with no Constraint List, includes the concepts of “putting liquids in the mouth”, “liquids that are put in the mouth”, “liquids with alcohol”, “absorb” and others.

The Restricted UW “drink(icl>do(obj>liquid))” denotes the subset of these concepts that includes “putting liquids in the mouth”, which in turn corresponds to verbs such as “drink”, “gulp”, “chug” and “slurp” in English.

Consider again the examples of Restricted UWs given above:

state(icl>express) is a more specific concept (arbitrarily associated with the English word “state”) that denotes an action in which humans express something.

state(icl>country) is a more specific sense of “state” that denotes a nation or country.

state(icl>abstract thing) is a more specific sense of “state” that denotes a kind of condition that persons or things are in. This UW is defined as a more general concept that can be referred to when defining other synonymous Uws, such as “situation” or “condition”.

state(icl>government) is a more specific sense of “state” that denotes a kind of government.

The information in parentheses is the Constraint List and it describes some conceptual restrictions; this is why they are called Restricted UWs. Informally, the restrictions mean “restrict your attention to this particular sense of the word”. Thus, the focus is clearly the idea and not the specific English word.

It often turns out that in a given language there is a wide variety of different words for these concepts and not, coincidentally, all the same word, as in English.

It should be noted that by organizing these senses around the English words, the task of making a new UW/Specific Language dictionary is simplified. A bilingual English/Specific Language dictionary can be used, and proceeding from there, the number of different concepts necessary for each English word can be specified.

This, of course, does not mean that English words are translated; the English dictionary is simply used as a reminder of the concepts that will be dealt with so that the work can be organized more efficiently.

Extra UWs

Extra UWs denote concepts that are not found in English and therefore have to be introduced as extra categories. Foreign-language words are used as Head Words using English (Alphabetical) characters. Consider again the examples given above:

ikebana(icl>flower arrangement) is “a kind of flower arrangement” for the meaning of “something you do with flowers”,
samba(icl>dance) is “a kind of dance”, and
soufflé(icl>food) is “a kind of food”.

To the extent that these concepts exist for English speakers, they are expressed with foreign-language loanwords and do not always appear in English dictionaries. So they simply have to be added to be able to use these specific concepts in the UNL system. The Constraint List or restrictions give the idea of what kind of concept is associated with these Extra UWs and the constraints provide the binary relations between this concept and other, more general, concepts already present (action, dance, food, etc.).

3.2 Compound UWs

Compound UWs are a set of binary relations that are grouped together to express a complex concept. A sentence itself is considered as a compound UW. This makes it possible to deal with situations like:

Women who wear big hats in movie theaters should be asked to leave.

Without Compound UWs, it would be impossible to build up complex ideas like “women who wear big hats in movie theaters” and then relate them to other concepts.

Compound UWs denote complex concepts that are to be interpreted as unit concepts, understood as a whole so that one can talk about their parts all at the same time. Consider again the example given above.

[Women who wear big hats in movie theaters] should be asked [to leave].

The part of the sentence within square brackets is what should be asked. Only when they are grouped together and considered as a whole unit can the correct interpretation be obtained.

Just as such complex units can be related to other concepts with conceptual relations, attributes can be attached to them to express negation, speaker attitudes, etc., which are usually interpreted as modifying the main predicate within the Compound UW.

3.2.1 The way to define a Compound UW

A Compound UW is defined by placing a Compound UW-ID immediately after the Relation Label in all of the binary relations that are to be grouped together. Thus, in the example below, “:01” indicates all of the elements that are to be grouped together to define Compound UW number 01.

```
agt:01(wear(icl>do(obj>thing)), woman(icl>person)).@pl)
obj:01(wear(icl>do(obj>thing)), hat(icl>clothes))
aoj:01(big(aoj>thing), hat(icl>clothes))
plc:01(wear(icl>do(obj>thing), theater(icl>facilities))
mod:01(theater(icl>facilities), movie(icl>entertainment))
agt:01(leave(icl>do).@entry, woman(icl>person)).@pl)
```

After this group has been defined, wherever the Compound UW-ID is, for instance “01” in the above example, it can be used to cite the Compound UW. The way to cite a Compound UW is explained in the next section.

A Compound UW is considered as a sentence or sub-sentence, so in the definition of a Compound UW one entry node marked by @entry is necessary.

3.2.2 The way to cite a Compound UW

Once defined, a Compound UW can be cited or referred to by simply using the Compound UW-ID as an UW. The method is to indicate the Compound UW-ID following a colon “:”. The reference to a Compound UW is also called a Scope-Node. The Scope-Node has the following syntax:

```
<Scope-Node>      ::= “:” <Compound UW-ID> [ <Attribute List> ]
<Compound UW-ID>  ::= two digits of a number 00 – 99
```

<Attribute List> ::= { “.” <Attribute Label> } ...
<Attribute Label> ::= “@entry” | “@may” | “@past” | ...

To complete the example above, it could be continued with:

obj(ask(icl>do(obj>thing)).@should, :01)
gol(ask(icl>do(obj>thing)).@should, woman.@pl)

Again, “:01” is interpreted as the whole set of binary relations defined above. It means that “:01” should be understood as comprising all of these binary relations. Compound UWs can be cited within other Compound UWs.

Chapter 4: Attributes

Attributes of UWs are used to describe the subjectivity of sentences. They show what is said from the speaker's point of view: how the speaker views what is said. This includes phenomena technically called "speech acts", "propositional attitudes", "truth values", etc. Relations and UWs are used to describe the objectivity of sentences. Attributes of UWs enrich this description with more information about how the speaker views these states-of-affairs and his attitudes toward them. Such attributes play the role of bridging the conceptual world represented by UWs and relations, and the real world. In other words, such attributes bring the concept defined by UWs and relations into the real world.

4.1 Time with respect to the Speaker

Where does the speaker situate his description in time, taking his moment of speaking as a point of reference? A time before he spoke? After? At approximately the same time? This is the information that defines "narrative time" as past, present or future. These Attributes are attached to the main predicate.

Although in many languages this information is signaled by tense markings on verbs, the concept is not tense, but "time with respect to the speaker". The clearest example is the simple present tense in English, which is not interpreted as the present time, but as "independently of specific times".

Consider the example: The earth is round.

This sentence is true in the past, present and future, independently of the speaker's time, so although the tense is "present" it is not interpreted as the present time.

@past	happened in the past
@present	happening at present
@future	will happen in future

ex) It <u>was</u> snowing yesterday
ex) It <u>is raining</u> hard.
ex) He <u>will arrive</u> tomorrow

4.2 The Speaker's View of Aspect

A speaker can emphasize or focus on part of an event or treat it as a whole unit. This is closely linked to how the speaker places the event in time. These Attributes are attached to the main predicate.

The speaker can focus on the beginning (@begin) of the event, looking forward to it (@begin.@soon), or backward to it (@begin.@just).

He can also focus on the end (@end) or completion (@complete) of the event, looking forward to it (@end.@soon or @complete.@soon), or backward to it (@end.@just or @complete.@just).

He can focus on the middle (@progress) or continuation (@continue) of the event.

The speaker can choose to focus on the lasting effects or final state of the event (@state) or on the event as a repeating unit (@repeat), experience (@experience) or custom (@custom).

He can also focus on the incompleteness or the fact that it has not yet happened, by using @yet.

@begin	beginning of an event or a state
@complete	finishing/completion of a (whole) event.

@continue	continuation of an event
------------------	--------------------------

@custom	customary or repetitious action
----------------	---------------------------------

@end	end/termination of an event or a state
-------------	--

@experience	experience
--------------------	------------

@progress	an event is in progress
------------------	-------------------------

@repeat	repetition of an event
----------------	------------------------

@state	final state or the existence of the object on which
---------------	---

Ex) It <u>began</u> to work again.
Ex) I've <u>looked through</u> the script.
look.@entry.@complete
Ex) He <u>went on talking</u> .
talk.@continue.@past
Ex) I <u>used to visit</u> [I <u>would often go</u>] there when I was a boy.
<u>visit.@custom.@past</u>
Ex) I <u>have done</u> it.
do.@end.@present
Ex) Have you ever visited Japan?
visit.@experience.@interrogation
Ex) I have been there.
visit.@experience
Ex) I <u>am working</u> now.
work.@progress.@present
Ex) It is so windy that the tree branches <u>are knocking</u> against the roof.
knock.@entry.@present.@repeat
Ex) It <u>is broken</u> .

an action has been taken

break.@state

These attributes are used to modify the attributes above, to express a variety of aspects of natural languages.

@just	Expresses an event or a state that has just begun or ended/been completed	Ex) He has just come. come.@complete.@just
@soon	Expresses an event or a state that is about to begin or end/be completed	Ex) The train is about to leave. <u>leave.@begin.@soon</u>
@yet	Expresses the feeling of something not yet begun, ended or completed, or expresses an event or a state that has not yet started or ended/been completed, together with @not.	Ex) I have not yet done it. do.@complete.@not.@yet

4.3 The Speaker's View of Reference

Whether an expression refers to a single individual, a small group or a whole set is often not clear. The expression "the lion" is not sufficiently explicit for us to know whether the speaker means "one particular lion" or "all lions". Consider the following examples:

The lion is a feline mammal.

The lion is eating an antelope.

In the first example, it seems reasonable to suppose that the speaker understood "the lion" as "all lions", whereas in the second example as "one particular lion".

The following Attributes are used to make explicit what the speaker's view of reference seems to be.

@generic	generic concept	Ex) The <u>dog</u> is a faithful animal.
@def	already referred	Ex) <u>the</u> book you lost
@indef	non-specific class	Ex) There is <u>a</u> book on the desk.
@not	complement set	Ex) <u>Don't</u> be late!
@ordinal	ordinal number	Ex) the <u>2nd</u> door

These attributes are usually attached to UWs that denote things.

4.4 The Speaker's View of Emphasis, Focus and Topic

The speaker can choose to focus or emphasize parts of a sentence to show how important he thinks they are in the situation described. This is often related to sentence structure.

@contrast	Contrasted UW	For instance, "but" in the examples below is used to introduce a word or phrase that contrasts with what was said before. Ex) It wasn't the red one <u>but the blue one</u> . Ex) He's poor <u>but happy</u> . Ex) I do <u>like</u> it.
@emphasis	Emphasized UW	Ex) He <u>promised</u> (entry of the sentence) that he would <u>come</u> (entry of the scope)
@entry	Entry or main UW of a sentence or a scope	Ex) Are you painting the <u>bathroom</u> blue? To this question, the answer will be "No, I'm painting the LIVING-ROOM blue"
@qfocus	Focused UW of a question	Ex)
@theme	Instantiates an object from a different class	Ex)
@title	Title	Ex)
@topic	Topic	Ex) He(@topic) was killed by her. Ex) The girl(@topic) was given a doll. Ex) This doll(@topic) was given to the girl.

One UW marked with "@entry" is essential for each UNL expression or in a Compound UW.

4.5 The Speaker's Attitudes

The speaker can also express, directly or indirectly, what his attitudes or emotions are towards what is being said or who it is being said to. This includes respect and politeness towards the listener and surprise toward what is being said.

@affirmative	Affirmation Ex)
@confirmation	Confirmation Ex) You won't say that, will you? Ex) It's red, <u>isn't it</u> ? Ex) Then you won't come, right?
@exclamation	Feeling of exclamation Ex) kirei na! ("How beautiful (it is)!" in Japanese) Ex) Oh, look out!
@imperative	Imperative Ex) Get up! Ex) You will please leave the room.
@interrogative	Interrogation Ex) Who is it?
@invitation	Inducement to do something Ex) Will / Won't you have some tea? Ex) Let's go, shall we?
@polite	Polite feeling. Puts emphasis on a way of talking. Ex) Could you (please)... Ex) If you could ... I would ...
@request	Request Ex) Please don't forget...
@respect	Respectful feeling. In many cases, some special words are used. Ex) o taku ("your house" in Japanese) Ex) Good morning, sir.
@vocative	Vocative Ex) Boys, be ambitious!

4.6 The Speaker's Feelings, Judgement and Viewpoint

These attributes express the speaker's feelings or how the speaker views or judges what is said.

This sort of subjective information is very much dependent on the type of language. It should be possible to express every kind of subjective information from all languages. Thus, the development of the attributes is open to the developers of each language, who can introduce a new attribute when no current attribute expresses its meaning. The new attribute must be also introduced in the same way.

The following attributes are used to clarify the speaker's viewpoint information.

Ability	
@ability	Ability, capability of doing something Ex) The child <u>can't</u> walk yet. Ex) He <u>can speak</u> English but he <u>can't write</u> it very well.

Admiration	
@admire	Admiring feeling of the speaker about something Ex)

Conclusion	
@conclusion	Logical conclusion due to a certain condition

	Ex) He is her husband; <u>she is his wife.</u>
@consequence	Logical consequence Ex) He was angry, <u>wherefore</u> I left him alone.

Blames	
@blame	Blameful feeling of the speaker about something Ex) A sailor, <u>and</u> afraid of the sea!

Consent and dissent	
@dissent	Dissenting feeling of the speaker about something Ex) <u>But</u> that's not true.
@grant	To give/get consent/permission to do something Ex) <u>Can I smoke</u> in here? Ex) You <u>may borrow</u> my car if you like.
@grant-not	Not to give consent to do something Ex) You { <u>mustn't/are not allowed to/may not</u> } borrow my car.

Expectation	
@although	Something follows against [contrary to] or beyond expectation Ex) <u>Although</u> he didn't speak, I felt a certain warmth in his manner.
@discontented	Discontented feeling of the speaker about something Ex) (I'll tip you 10 pence.) <u>But</u> that's not enough!
@expectation	Expectation of something Ex) Children <u>ought</u> to be able to read by the age of 7. Ex) If you leave now, you <u>should get</u> there by five o'clock.
@wish	Wishful feeling, to wish something is true or has happened Ex) <u>If only</u> I could remember his name! (~I do wish I could remember his name!) Ex) You <u>might have just let me know.</u>

Intention	
@insistence	Strong determination to do something Ex) He <u>will do</u> it, whatever you say.
@intention	Intention about something or to do something Ex) He <u>shall get</u> this money. (Speaker's intention) Ex) We <u>shall let you know</u> our decision.
@will	Determination to do something Ex) I <u>'ll write</u> as soon as I can. Ex) We <u>won't stay</u> longer than two hours.

Necessity, obligation	
@need	Necessity to do something Ex) You <u>need to finish</u> this work today.
@obligation	Obligation to do something according to (quasi-) law, contract, or ... Ex) The vendor <u>shall maintain</u> the equipment in good repair.
@obligation-not	Obligation not to do something, forbid to do something according to (quasi-) law, contract or ... Ex) Cars <u>must not park</u> in front of the entrance. Ex) <u>No smoking</u>
@should	To do something as a matter of course Ex) You <u>should do</u> as he says. Ex) You <u>ought to start</u> at once.

Possibility	
@certain	Certainty that something is true or happens Ex) If Peter had the money, he <u>would have bought</u> a car.
@inevitable	Logical inevitability that something is true or happens Ex) There <u>must</u> be a mistake. Ex) They <u>should</u> be home by now.
@may	Practical possibility that something is true or happens Ex) It <u>may be</u> true. Ex) It <u>could be</u> .
@possible	Logical possibility that something is true or happens Ex) Anybody <u>can make</u> mistakes. Ex) If Peter had the money, he <u>would buy</u> a car.
@probable	(Practical) probability that something is true or happens Ex) That <u>would be</u> his mother. Ex) He <u>must</u> be lying.
@rare	Rare logical possibility that something is true or happens Ex) If such a thing <u>should</u> happen, what shall we do? Ex) If I <u>should</u> fail, I will [would] try again.
@unreal	Unreality that something is true or happens Ex) If we had enough money, we <u>could buy</u> a car. Ex) If Peter had the money, he <u>could buy</u> a car.

Regret	
@regret	Regretful feeling of the speaker about something Ex) It's a pity that he <u>should miss</u> such a golden opportunity.

Surprises	
@surprised	Surprised feeling of the speaker about something Ex) (He has succeeded!) <u>But</u> that's great!

4.7 Convention

Typical UNL structures can be expressed by attributes to avoid the complexity of enconverting and deconverting. These attributes do not express the speaker's information.

@pl	Plural	These (this.@pl) are the wrong size.
@angle_bracket	< > are used	
@brace	{ } are used	
@double_parenthesis	(()) are used	
@double_quote	“ ” are used	
@parenthesis	() are used	UNL (Universal Networking Language) cnt(UNL, Universal Networking Language.@parenthesis)
@single_quote	' ' are used	
@square_bracket	[] are used	

Chapter 5: Format of UNL

5.1 UNL Document

Information is provided in UNL documents. The UNL document has the following format.

<UNL document>	::= "[D:" <dinf> "]" { "[P:" <number> "]" { "[S:" <number> "]" <sentence> "[/S]" }... "[/P]" }... "[/D]"
<dinf>	::= <document name> "," <owner name> ["," <document id> "," <date> "," <mail address>]
<document name>	::= "dn=" <character string>
<owner name>	::= "on=" <character string>
<document id>	::= "did=" <character string> /* defined by system */
<date>	::= "dt=" <character string> /* defined by system */
<mail address>	::= "mid=" <character string> /* defined by system */
<sentence>	::= "{org:" <l-tag> ["=" <code>] }" <source sentence> "{/org}" "{unl" [":" <uinf>] }" <UNL expression> "{/unl}" "{" <l-tag> ["=" <code>] [":" <sinf> "]" <generated sentence> "{/" <l-tag> }" /* necessary information about one sentence */
<l-tag>	::= "ab" "cn" "de" "el" "es" "fr" "id" "hd" "it" "jp" "lv" "mg" "pg" "ru" "sh" "th" /* language flag */
<code>	::= <character code name>
<character name>	code ::= <character string>
<source sentence>	::= <character string>
<generated sentence>	::= <character string>
<uinf>	::= <system name> "," <post editor name> "," <reliability> ["," <date> "," <mail address>]
<sinf>	::= <system name> "," <post editor name> "," <reliability> ["," <date> "," <mail address>]
<system name>	::= "sn=" <character string>
<post editor name>	::= "pn=" <character string>
<reliability>	::= "rel=" <digit>
<number>	::= <digit> /* sentence number */

The tags used in the above definition are the following.

[D:<dinf>]	indicates the Beginning of a document and the necessary information about the document
[/D]	indicates the End of a document
[P:<number>]	indicates the Beginning of a paragraph
[/P]	indicates the End of a paragraph
[S:<number>]	indicates the Beginning of a sentence and the sentence number
[/S]	indicates the End of a sentence
{org:<l-tag>=<code>}	indicates the Beginning of an original/source sentence, language and character code, “=<code>” can be omitted.
{/org}	indicates the End of an original sentence
{unl:<uinf>}	indicates the Beginning of the UNL expressions of a sentence and necessary information, “:<uinf>” can be omitted.
{/unl}	indicates the End of the UNL expressions of a sentence

See the following section about <UNL expression>.

5.2 UNL Expression

A UNL expression of a sentence is identified with the following tags: {unl} and {/unl}.

There are two forms for expressing UNL expressions, one is the table form and the other is the list form. The table

form of a UNL expression is more readable than the list form, but the list form of a UNL expression is more compact than the table form.

Any component, such as a word, phrase or title and, of course, a sentence of a natural language can be represented with UNL expressions. A UNL expression therefore consists of a UW or a (set of) binary relation(s). In UNL documents, a UNL expression for a sentence is enclosed by the tags {unl} and {/unl} inside [S] and [/S]. If a UNL expression consists of a UW, this UW should be enclosed further by the tags [W] and [/W]. If necessary, the whole sentence can also be expressed as a scope. In this case, the Compound UW-ID of the scope should be enclosed by [W] and [/W].

Thus, a UNL expression of a sentence is the following:

```
{unl}
<Binary Relation>
...
{/unl}

or,

{unl}
[W]
<UW><Attribute List>
[/W]
{/unl}

or,

{unl}
[W]
":.<Compound UW-ID><Attribute List>
[/W]
<Binary Relation>
...
{/unl}
```

Each tag and binary relation should end with a return code: "0x0a".

5.2.1 The table form of UNL expression

Syntax of binary relation

<Binary Relation>	::= <Relation Label> [":.<Compound UW-ID>] "(" {{ <UW ₁ > [":." <UW-ID ₁ >]} { ".<Compound UW-ID ₁ > }}[<Attribute List>] "," {{ <UW ₂ > [":." <UW-ID ₂ >]} { ".<Compound UW-ID ₂ > }}[<Attribute List>] ")"
<Relation Label>	::= a relation, see "Chapter 2: Relations"
<UW>	::= an UW, see "Chapter 3: Universal Words"
<Attribute List>	::= { ".<Attribute> } ...
<Attribute>	::= an attribute, see "Chapter 4: Attributes"
<UW-ID>	::= two characters of '0' – '9' and 'A' – 'Z'
<Compound UW-ID>	::= two-digit decimal number (00 – 99) 00 is used for representing the main sentence, which can be omitted.

Compound UW-IDs are strings of two digits used to identify each instance specified by Compound UWs. Compound UWs are groups of binary relations (so-called "Scope-Nodes") that can be referred to as a UW.

For instance, the following shows an example of a UNL expression of the sentence "I can hear a dog barking outside".

```

{unl}
aoj(hear(icl>perceive(agt>thing,obj>thing)).@entry.@ability, I)
obj(hear(icl>perceive(agt>thing,obj>thing)).@entry.@ability, :01)
agt:01(bark(agt>dog).@entry, dog(icl>mammal))
plc:01(bark(agt>dog).@entry, outside(icl>place))
{/unl}

```

In the above UNL expression, “**aoj**”, “**agt**” and “**obj**” are the **relation labels**, “**I**”, “**bark(agt>dog)**”, “**dog(icl>mammal)**”, “**hear(icl>perceive(agt>thing,obj>thing))**” and “**outside(icl>place)**” are the **UWs**, and “**:01**”, which appears three times in the example, shows the **Compound UW-ID**. The Compound UW-ID appears in the position of a UW, the so-called “**scope-node**”, and is used to cite or refer to a Compound UW previously defined. Binary relations indicated by the Compound UW-ID define the contents of the scope. A scope-node always begin with “:” followed by the two digits of a Compound UW-ID.

UW-IDs are omitted from the above UNL expression. When a UW is unique in a UNL expression, the UW-ID can be omitted.

The UW-ID is used to indicate some referential information: that there are two or more different occurrences of the same concept (they are not co-referent). Normally, if the same UW occurs more than once, it is in all cases understood to refer to the same entity or occurrence. For example, if one man greeted another man, the same UW would be used twice -- “man(icl>male person)” to distinguish one from the other with UW-IDs:

```

man(icl>male person):01 for the first and
man(icl>male person):02 for the other, to make it clear that the first man did not greet himself.

```

5.2.2 The list form of UNL expression

The list form of a UNL expression consists of a set of UWs and encoded binary relations of a sentence. In case a whole sentence is treated as a scope, the Compound UW-ID of the scope for the sentence can be included in the UW list between [W] and [/W].

```

{unl}
[W]
{<UW> | {“:”<Compound UW-ID>}}”<Node-ID> /* node identifier */
...
[/W]
[R]
<Encoded Binary Relation>
...
[/R]
{/unl}

```

The tags used above have the following meanings.

[W]	indicates the Beginning of the Node identifier
[/W]	indicates the End of the Node identifier
[R]	indicates the Beginning of the encoded binary relations
[/R]	indicates the End of the encoded binary relations

Each tag, encoded binary relation and node identifier should end with a return code: “0x0a”.

Syntax of an encoded binary relation

```

<Encoded Binary Relation> := <Node1-ID><Relation Label>[“:”<Compound UW-ID>]<Node2-ID>
<Node-ID>                 := two characters of ‘0’ – ‘9’ and ‘A’ – ‘Z’

```

For instance, the following shows an example of the list form of a UNL expression of the sentence “I can hear a dog barking outside”.

```
{unl}  
[W]  
I:01  
hear(icl>perceive(agt>thing,obj>thing)).@entry.@ability:02  
dog(icl>mammal):03  
bark(agt>dog).@entry:04  
outside(icl>place):05  
:01:06  
[/W]  
[R]  
02aoj01  
02obj06  
04agt:0103  
04plc:0105  
[/R]  
{/unl}
```

Appendix 1: Syntax Definition Notation

<u>Symbol</u>	<u>Definition</u>
::=	to indicate the left is defined as the right
	to indicate two disjunctive elements: “or”
[]	to indicate an optional element
{ }	to indicate an alternative element
...	to indicate repetition of the previous element, 0 or more than 1 time
“ ”	to enclose a string of literal characters
< >	to indicate a variable name

Appendix 2: [List of Relation Labels](#)

Appendix 3: [List of Attribute Labels](#)