The Universal Networking Language (UNL) Specifications Version 2.0a

Restricted-access Document:
Do not copy or distribute,
except as specified under contractual conditions

UNU/IAS/UNL Center
July, 1999

Introduction

The Universal Networking Language (UNL) is an electronic language for computers to express and exchange every kind of information.
The UNL represents information, i.e. meaning, sentence by sentence. Sentence information is represented as a hyper-graph having concepts as nodes and relations as arcs. This hyper-graph is also represented a set of directed binary relations, each between two of the concepts present in the sentence.
Concepts are represented as character-strings called "Universal Words(UWs)". UWs can be annotated with attributes which provide further information about how the concept is being used in the specific sentence.
A UNL document, then, will be a long list of relations between concepts.

Relations

Binary relations are the building blocks of UNL sentences. They are made up of a relation and two UWs. This section deals with the definition and interpretation of the relations that are used as the basis 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. Conventions for syntax notation are found in Appendix 3.

Internal structure of Binary relations

Binary relations are made up as follows:

<Binary Relation> ::= <Relation Label> [":"<Compound UW-ID>]
"(" {<UW1> | ":" <Compound UW-ID1>} "," {<UW2> | ":"<Compound UW-ID2>} ")"

These elements will be defined in the paragraphs below.

Example binary relations are:
mod:01(area(icl>place):02.@indef, strategic)
obj(designate(icl>do).@entry.@may, :01)
plc(read(icl>do), home)

Relation-Labels

Relation-labels are strings of three lower-case alphabetic characters taken from the closed inventory listed below.

Examples are the elements in bold face type below:

mod:01(area(icl>place):02.@indef, strategic)
obj(designate(icl>do).@entry.@may, :01)
plc(read(icl>do), home)

Compound UW-IDs

Compound UW-IDs are strings of two upper-case alphabetic characters and digits used to identify each instance specified by Compound UWs. Compound UWs are groups of binary relations(called “Scope-Nodes”) so that they can be referred to as an UW. Examples are the elements in bold face type below. The first example is an instance of compound UW-IDs being used to define a unit; the second example is an instance of Compound UW-IDs being used to cite or refer to a Compound UW previously defined. See Compound UWs for further information.

mod**:01**(area(icl>place):02.@indef, strategic)
obj(designate(icl>do).@entry.@may, **:01**)
plc(read(icl>do), home(icl>place))

Note that the “:02” in the first example is not a Compound UW-ID but an UW-ID. A Compound UW-ID is either attached directly to Relation-Labels or appear alone, as UWs. See UW-IDs for further information.

Uws

UWs can be UWs or compound UWs. Examples are the six elements in bold face type below. Non-standard formatting has been used to make them clearer.

mod:01(	**area(icl>place):02.@indef**,	**strategic**)
plc(	**read(icl>do)**,	**home(icl>place)**)
obj(	**designate(icl>do).@entry.@may**,	**:01**)

Binary relations

In the UNL, binary relations are represented as less than three-character strings called “Relation-Labels” and are defined as specified below.
There are many factors to be considered in choosing an inventory of relations. The principles to choose relations as follows.

Principle 1) Necessary Condition
When an UW has relations between more than two other UWs, each relation label should be set as to be able to identify each relation on the premise that we have enough knowledge about a concept of each UW express.

Principle 2) Sufficient Condition
When there are relations between UWs, each relation label, we should be set as to be able to understand each role of each UW only by referring a relation label.

agt (agent)

Agt defines a thing which initiates an action.

agt (do, thing)

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

Detailed Definition
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(icl>do), John(icl>person))	John breaks
agt(translate(icl>do), computer(icl>machine))	computer translates …
agt(run(icl>do), car(icl>thing))	car runs …
agt(break(icl>do), explosion(icl>event))	explosion breaks …

Related Relations
Agent is different from **cag** in that agent initiates the action, whereas the co-agent initiates a different, accompanied action.
Agent is different from **ptn** in that agent is the focussed initiator of the action, whereas the partner is a non-focussed initiator.
Agent is different from **con** in that agent is the focussed initiator of the action, whereas condition is an indirect, usually unfocussed, influence on the action.

and (conjunction)

And defines a conjunctive relation between concepts.

and (*, *)

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

Detailed definition
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(quichly, easily)	… easily and quickly
and(dream(icl>do), think(icl>do))	… to think and to dream
and(Mary(icl>person), John(icl>person))	… John and Mary

Related Relations
Conjunction is different from **or** in that with **and** we group things together to say the same thing about both of them, whereas with **or** we separate them to say that what is true about one is not true about the other.
Conjunction is different from **cag** in that when agents are conjoined both are initiating an explicit event, whereas with **cag**, the co-agent initiates an implicit event.
Conjunction is different from **ptn** in that when agents and partners are conjoined both are in focus, whereas with **ptn**, the partner is not in focus (as compared to the agent).
Conjunction is different from **coo** and **seq** in meaning, although many times the same expressions can be used for both. 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 or not they are considered to be grouped together. In turn, **seq** means that the terms are ordered in time, one after the other.

aoj (thing with attribute)

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

aoj ((aoj>thing), thing)
aoj (thing, thing)

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

Detailed definition
Thing with attribute is defined as the relation between:
UW1 – a state or a thing which represent a state
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(icl>thing))	leaf is red
aoj(available(aoj>thing), book(icl>thing))	book is available
aoj(nice(aoj>thing), ski(icl>event))	Skiiing is nice
aoj(teacher(icl>thing), John(icl>person))	John is a teacher
aoj(have(aoj>thing,obj>thing), I)	I have a pen
obj(have(aoj>thing,obj>thing), pen(icl>thing))	
aoj(know(aoj>thing,obj>thing), John(icl>person))	John knows …
aoj(can(aoj>thing,obj>thing), I)	I can …
aoj:01(difficult(aoj>thing), it)	It is difficult for John.
aoj(:01, John(icl>person))	

Related Relations
Thing with attribute is different from **mod** in that **mod** gives some restriction, whereas **aoj** gives a state or characteristic.
Thing with attribute is different from **ben** in that a beneficiary is quite independent from an focussed event or state but this event or state can be considered to give a good or bad influence, whereas **aoj** has more close relation and can be considered to describe a state or characteristic.
Thing with attribute is different from **obj** in that **obj** gives a thing which is directly affected by action or phenomenon, whereas, **aoj** gives a thing in a state.

bas (basis for expressing degree)

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

bas (degree, thing)

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

Detailed definition
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”, “like”, “as much as”, “at least” etc., and
- UW2 is some thing used as the basis for evaluating characteristics or quantity of some other (focussed) thing.

Examples and readings

bas(more, rat(icl>thing))	…er than rat; more … than rat
bas(like, star(icl>thing))	… like star
bas(same, b(icl>thing))	… the same as b
bas(at least, 12)	… the same as b
aoj(beautiful(aoj>thing), tulip(icl>thing))	tulip is more beautiful than rose
man(beautiful(aoj>thing), more)	
bas(more, rose(icl>thing))	
aoj(:01, John(icl>person))	John is more quiet than shay
man:01(quiet(aoj>thing), more)	
bas:01(more, shy(aoj>thing))	

ben (beneficiary)

Ben defines a not directly related beneficiary or victim of an event or state.

ben (occur, thing)
ben (do, thing)
ben ((aoj>thing), thing)

Syntax

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

Detailed Definition

Beneficiary is defined as the relation between:
UW1 – an event or state, and
UW2 – a thing,
where:

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

Examples and readings

ben(give(icl>do), Mary(icl>person))	John give … for Mary.
agt(give(icl>do), John(icl>person))	
ben(good(aoj>thing), John)	It is good for John to …

Related Relations

Beneficiary is different from **aoj** in that **aoj** has close relation and can be considered to describe a state characteristic, whereas a beneficiary is quite independent from an focussed event or state but this event or state can be considered to give a good or bad influence.

cag (co-agent)

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

cag (do, thing)

Syntax

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

Detailed definition
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))	… walk with John
cag(live(icl>do),aunt(icl>person))	… lives with aunt

Related relations
Co-agent is different from **agt** in that different, independent actions occur for the agent and the co-agent. Moreover, the agent and its action are in focus, while the co-agent and its action are not in focus.
Co-agent is different from the **ptn** in that the co-agent initiates an action that is independent of the agent’s action, whereas the partner initiates the same action together with the agent.
Co-agent is different from **con** in that the co-agent initiates a non-focussed action, whereas the condition is an indirect influence on the focussed action.

cao (co-thing with attribute)

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

cao ((aoj>thing), thing)
cao (thing, thing)

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

Detailed definition
Co-thing with attribute is defined as the relation between:
UW1 – a state or a thing which represent a state
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(aoj>thing), you)	… is here with you

Related relations
Co-thing with attribute is different from **aoj** in that there is a different, independent state for the thing with attribute and co-thing with 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
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>language), universal networking language(icl>language)
UNL, Universal Networking Language
cnt(internet(icl>network), amalgamation(icl>thing)) Internet: an amalgamation
cnt(language generator(icl>tool), deconvertor(icl>tool).@double_quotation)
a language generator "deconvertor"…

cob (affected co-thing)

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

cob (occur, thing)
cob (do, thing)
cob ((aoj>thing,obj>thing), thing)

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

Detailed Definition
"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(icl>occur), Mary(icl>person)) … died with Mary
cob(have(aoj>thing,obj>thing), pencil(icl>thing)) ... have a pen with a pencil
obj(have(aoj>thing,obj>thing), pen(icl>thing))

Related concepts
Co-object is different from **obj** in that the **obj** is in focus, whereas the **cob** is related to a second, non-focussed implicit event or state.

con (condition)

Con defines an non-focused event or state which conditioned a focused event or state.

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

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

Detailed definition
Condition is defined as the relation between:
UW1 – a focussed 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) which influences whether or when UW1 can happen.

Examples and readings

aoj:01(green(aoj>thing), light (icl>thing)) con(go(icl>do), :01)	If light is green, … go

coo (co-occurrence)

Coo defines a co-occurred event or state for a focussed event or state.

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

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

Detailed definition
Co-occurrence is defined as the relation between:
UW1 – a focussed event or state,
UW2 – a co-occurred 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>do), cry(icl>do))	… run with crying
coo(red(aoj>thing), hot(aoj>thing))	… is red while … is hot

Related Relations
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.
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 an event occurs or a state exists.

dur (occur, period)
dur (do, period)
dur ((aoj>thing), period)

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

Detailed definition
Duration is defined as the relation between:
UW1 – an event or state,
UW2 – a period that the event or state continues,

Examples and readings

dur(work(icl>do), hour(icl>period))	… work nine hours
qua(hour(icl>period), 9)	
dur(talk(icl>do), meeting(icl>event)	… talk during meeting
dur(come(icl>do), absence(icl>state))	… come during … absence

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
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))	… from a to z
fmt(Osaka(icl>place), New York(icl>place))	… from Osaka to New York
fmt(Monday(icl>time), Friday(icl>time))	… from Monday to Friday

Related Relations
Range is different from **src** and **gol** in that for **src** and **gol** the initial and final states of some **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.
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
Origin is defined as the relation between:
UW1 – a thing, and
UW2 – a origin of the thing,
where:
• UW2 describes the origin such as original position of UW1.

Examples and readings

frm(man(icl>person), Japan(icl>country))	… man from Japan

gol (goal: final state)

Gol defines the final state of object or the thing finally associated with 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
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(go(gol>place), sad(aoj>person)	… go … to sad
gol(change(gol>thing), red(aoj>thing))	… change … to red
gol(transform(gol>thing), strong(aoj>thing))	… is transformed … to strong
gol(post(gol>thing), account(icl>place))	… post … to account

Related Relations
Final characteristics is different from **tmf** and **plf** in that **gol** describes qualitative characteristics and not time or place.
Final characteristics is different from **src** in that **gol** describes the characteristics of the **obj** at the final state of the event.

ins (instrument)

Ins defines the 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
Instrument is defined as the relation between:
UW1 – an event, and

UW2 – a concrete thing,
where:
• UW2 specifies the concrete thing which is used in order to make UW1 happen.

Examples and readings

ins(look(icl>do), telescope(icl>thing))	… look … with telescope
ins(solve(icl>do), pencil(icl>thing))	… solve … using pencil
ins(separate(icl>do), knife(icl>thing))	… separate … with knife

Related Relations
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.
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 the way to carry out event or characteristics of a state.

man (occur, how)
man (do, how)
man ((aoj>thing), how)

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

Detailed definition
“Manner” is defined as the relation between:
UW1 – an event or state,
UW2 – a manner,
where:
• The UWs are different, and
• UW1 is done or exist in a way characterized by UW2

Examples and readings

man(look(icl>do), quickly)	… look quickly
man(think(icl>do), often)	… think often …
man(beautiful(aoj>thing), very)	very beautiful

Related Relations
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 the means to carry out an event.

met (do, abstract thing)

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

Detailed definition
“Method or means” is defined as the relation between:

UW1 – an event, and
UW2 – an abstract thing,
where:
• UW2 specifies the abstract thing which is used or the steps carried out in order to make UW1 happen.

Examples and readings

met(solve(icl>do), dynamics(icl>abstract thing))	… solve … with dynamics
met(solve(icl>do), algorithm(icl>abstract thing))	… solve … using algorithm
met(separate(icl>do), cut(icl>do))	… separate … by cutting …

Related Relations
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.
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 which restrict a focussed thing.

mod (thing, thing)
mod (thing, (mod>thing))

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

Detailed definition
“Modification” is defined as the relation between:
UW1 – a focussed thing,
UW2 – a thing which restrict UW1 in some way

Examples and readings

mod(story(icl>thing), whole)	whole story
mod(plan(icl>thing), master)	master plan
mod(part(icl>thing), main)	main part
qua(block(icl>thing), 3))	three blocks of ice
mod(ice(icl>thing), block(icl>thing))	

Related Relations
Modification is different from **aoj** in that **aoj** describes a state or characteristic of a thing, whereas **mod** merely indicates an 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.
Modification is different from **man** in that **man** describes a way to carry out event or 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
Name is defined as the relation between:
UW1 – a thing,
UW2 – a thing used as a name,
where:
• UW2 is a name of UW1.

Examples and readings

nam(tower(icl>thing), Tokyo(icl>thing))	Tokyo tower

obj (affected thing)

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

obj (occur, thing)
obj (do, thing)
obj ((aoj>thing,obj>thing), thing)

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

Detailed Definition
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(icl>do), table(icl>thing))	table move
obj(melt(icl>occur), snow(icl>thing))	snow melt
obj(cure(icl>do), paitient(icl>person))	cure paitient
obj(have(aoj>thing,obj>thing), pen(icl>thing))	... have a pen

Related Relations
Affected thing is different from **cob** in that the **obj** is in focus, whereas the **cob** is related to a second, non-focussed implicit event or state.

opl (affected place)

Opl defines a place in focus where an event affects.

opl (do, place)

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

Detailed Definition
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>do), shoulder(icl>thing))	… pat … on shoulder
opl(cut(icl>do), middle(icl>place))	… cut … in middle

Related Relations

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

or (disjunction)

Or defines disjunctive relation between two concepts.

or (thing, thing)

Syntax

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

Detailed definition

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

Related Relations

Disjunction is different from conjunctionn that the disjunction things are grouped in order to say that something is true for one or the other, whereas in conjunction they are grouped to say that the same is true for both. Disjunction in formal logic permits three situations for a disjunction 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, 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

Proportion, rate or distribution is defined as the relation between:
UW1 – a quantity,
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(2, day(icl>period))	… two … per day
per(time(icl>unit), week(icl>period))	… twice a week
qua(time(icl>unit), 2)	

plc(place)

Plc defines the place an event occurs or a state is true or a thing exists.

plc (occur, thing)
plc (do, thing)
plc ((aoj>thing), thing)
plc (thing, thing)

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

Detailed definition
Place is defined as the relation between:
UW1 – an event , state, or thing,
UW2 – a place or thing understood as a place,

Examples and readings

plc(cook(icl>do), kitchen(icl>thing))	… cook … in kitchen
plc(sit(icl>do), beside(icl>relative place))	… sit beside …
plc(red(aoj>thing), bottom(icl>thing))	… red on bottom

Related Relations
Place is different from **plf** and **plt** or **src** and **go**l in that **plc** describes a place with respect to an event as a whole, whereas these other relations describe position with respect to parts of an event.
Place is different from **opl** in that **plc** is not seen as being modified by an event, merely a reference point for characterizing it, whereas **op**l is seen as being modified.

plf (initial place)

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

plf (occur, thing)
plf (do, thing)
plf ((aoj>thing), thing)

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

Detailed definition
“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(go(icl>do), home(icl>place))	… go from home …
plf(call(icl>do), New York(icl>place))	… call from New York
plf(cut(icl>do), edge(icl>place))	… cut … from edge …
plf(beautiful(aoj>thing), side(icl>place))	… is beautiful from side …

Related Relations

Initial place is different from **plc** in that **plc** describe events or states taken as wholes, whereas **plf** describes only the initial part of an event or state.
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.
Initial place is different from **src** in that **plf** describes the place where the event began, whereas **src** describes the initial state of the **obj**.

plt (final place)

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

plt (occur, thing)
plt (do, thing)
plt ((aoj>thing), thing)

Syntax

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

Detailed definition

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(talk(icl>do), Boston(icl>place))	… talk … until Boston
plt(cut(icl>do), edge(icl>place))	… cut … to edge
plt(beautiful(icl>state), fence(icl>thing))	… is beautiful up to fence

Related Relations

Final place is different from **plc** in that **plc** describe events or states taken as wholes, whereas **plt** describes only the final part of an event.
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.
Final place is different from **gol** in that **plt** describes the place where an event or state ended, whereas **gol** described the final state of the **obj**.

pof(part-of)

Pof defines a concept of which a focussed 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 part thing, and
UW2 – a whole thing,
where:
• UW1 is the part of UW1.

Examples

pof(wing(icl>body), bird(icl>animal))	Bird’s wing.

pos (possessor)

Pos defines possessor of a thing.

pos (thing, volitional thing)

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

Detailed definition
Possessor is defined as the relation between:
UW1 – a thing or a place, and
UW2 - a human or non-human, seen-as-volitional thing
where:
• UW2 is a possessor of UW1.

Examples and redings

pos(dog(icl>thing), John(icl>person))	John’s dog
pos(book(icl>thing), I)	my book

ptn (partner)

Ptn defines 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
Partner is defined as the relation between:
UW1 - an action, and
UW2 - a human or non-human, seen-as-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 redings

ptn(compete(icl>do), John(icl>person))	… compete with John
ptn(share(icl>do),poor(icl>person))	… share … with poor
ptn(collaborate(icl>do), machine(icl>thing))	… collaborate with machine

Related Relations
Partner is different from **agt** in that the agent and its event are in focus, while the partner and its event are not in focus.
Partner is different from **cag** in that the co-agent initiates an event that is independent of the agent's event, whereas the partner initiates the same event together with the agent.
Partner is different from **con** in that the partner initiates the same event as the agent does whereas the condition is only an indirect influence on that event.

pur (purpose or objective)

Pur defines the purpose or objectives of agent of an event or the purpose of a thing exist.

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
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 objectives, 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))	… come to see
pur(work(icl>do), money(icl>do))	… work for money
pur(budget(icl>money), research(icl>do))	… budget for research

Related Relations
Purpose or objective is different from **gol** in that **pur** describes the desires of the agent, whereas **gol** describes the state of the **obj** at the end of the event.
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 quantity of a thing or unit.

qua (thing, quantity)

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

Detailed definition
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(block(icl>thing), 3))	three blocks of ice
mod(ice(icl>thing), block(icl>thing))	
qua(kilo(icl>unit), many(aoj>thing))	many kilos …
qua(truckload(icl>unit), 7)	seven truckload …

Related Relations
Quantity is different from **per** in that quantity is absolute number or amount, whereas **per** is number or amount relative to some unit of reference (time, distance, etc.).
Quantity is also used to express iteration, or number of times an event or state occurs.

rsn (reason)

Rsn defines a reason that 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, (aoj>thing))
rsn (do, (aoj>thing))
rsn ((aoj>thing), occur)
rsn ((aoj>thing), do)
rsn ((aoj>thing), thing)
rsn ((aoj>thing), (aoj>thing))

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

Detailed definition
Reason is defined as the relation between:
UW1 – an event or state,
UW2 – a reason of an event or state,
where:
• UW2 is a reason that UW1 happens.

Examples and readings

rsn(go(icl>do), illness(icl>thing))	go because of illness
agt:01(arrive(icl>occur), Mary(icl>person))	Because Mary arrive, team collaborate …
agt:02(collaborate(icl>do), team(icl>person)	
rsn(:02, :01)	

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 ((aoj>thing), thing)
scn (thing, thing)

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

Detailed definition
Scene is defined as the relation between:
UW1 – an event or state or thing,
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), competition(icl>event))	… win … in competition
scn(apear(icl>occur), program(icl>thing))	… appear on TV program
mod(program(icl>thing),TV(icl>thing))	
scn(play(icl>do), movie(icl>thing))	… play in movie

Related Relations
Scene is different from **plc** in that the reference place for **plc** is in real world, whereas for **scn** it is 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, (aoj>thing))
seq (do, state)
seq ((aoj>thing), occur)
seq ((aoj>thing), do)

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

Detailed definition
“Sequence” is defined as the relation between:
UW1 – a focussed 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>dot), look(icl>do))	… look before leaping
seq(green(aoj>thing), red(aoj>thing))	… was red before … was green

Related Relations

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.
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 object or the thing initially associated with 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

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 **obj** of UW1 at the beginning of UW1, or
- UW2 is a thing that is associated with the **obj** of UW1 at the beginning of UW1.

Examples and readings

src(go(icl>change), sad(aoj>thing))	… go from sad …
src(change(icl>occur), red(aoj>thing))	… change from red
src(transform(icl>do), weak(aoj>thing))	… is transformed from weak …
src(steal(icl>do), bank(icl>thing))	… steal … from bank

Related Relations

Initial state is different from **tmf** and **plf** in that src describes qualitative characteristics and not time or place.
Initial characteristics is different from **gol** in that **gol** describes the characteristics of the **obj** 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 ((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,
UW2 – a time,
where:

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

Examples and readings

tim(look(icl>do), Tuesday(icl>time))	… look on Tuesday
tim(cut(icl>do), o'clock(icl>time))	… cut … at … o'clock
tim(start(icl>do), come(icl>do))	… start when … come

Related Relations

Time is different from **tmf** and **tmt** in that time characterized 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)

Time-from defines a time an event starts or a state become true.

tmf (occur, time)
tmf (do, time)
tmf ((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 started, or
• UW2 specifies the time at which UW1 became/become true.

Examples and readings

tmf(look(icl>do), morning(icl>time))	… look since morning
tmf(full(aoj>thing), noon(icl>time))	… is full at noon

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 a 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 **obj** 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 the end of the event.

tmt (final time)

Time-to defines the time an event ends or a state becomes false.

tmt (occur, time)
tmt (do, time)
tmt ((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 ended, or
• UW2 specifies the time at which UW1 became/becomes false.

Examples and readings
tmt(think(icl>do), morning(icl>time)) … think until moning
tmt(cut(icl>do), noon(icl>time)) … cut until noon
tmt(full(aoj>thing), tomorrow(icl>time)) … be full until tomorrow

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 a 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 **obj** 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 **tmt** expresses the time at the beginning of the event.

to (destination)

To defines a destination of a thing.

to (thing, thing)

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

Detailed definition
Destination is defined as the relation between:
UW1 – a thing, and
UW2 – a destination of the thing,
where:
• UW2 describes the destination such as final position of UW1.

Examples and readings
to(train(icl>thing), London(icl>city)) … train for London

via (intermediate place or state)

Via defines a 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
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 **obj** of UW1 at some time in the middle of UW1,
• UW2 is a thing that describes a place or state that the **obj** of UW1 passed by or through during UW1.

Examples and readings

via(go(icl>do), New York(icl>place))	… go … via New York
via(bike(icl>do), Alps(icl>place))	… bike … through the Alps
via(drive(icl>do), tunnel(icl>thing))	… drive … by way of tunnel

Related Relations

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.
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.

Universal Words

A UW(Universal Word) represent 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.

Uws

Syntax

Informally, UWs are made up of a character string (an English-language word) followed by a list of constraints and a list of attributes. These can also be followed by an Instance ID. 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. See Appendix 3 for notational conventions.

```
<UW> ::=              <Head Word> [<Constraint List>] [ “:” <IUW-ID>] [ “.” <Attribute List>]
<Head Word> ::=   <character>…
<Constraint List> ::= “(“ <Constraint> [ “,” <Constraint>]… “)”
<Attribute List> ::= <Attribute Label> [ “.” <Attribute Label>]…
<UW-ID> ::=          {<upper case alphabetical character> |<digit>}
                      {<upper case alphabetical character> |<digit>}
<Constraint> ::=     <Relation Label> { “>” | “<” } <UW> [<Constraint List>] |
                      <Relation Label> { “>” | “<” } <UW> [<Constraint List>]
                      [ { “>” | “<” } <UW> [<Constraint List>] ] …
<Attribute Label> ::= “@volitional” | “@reason” | “@past” | ….
<Relation Label> ::= “agt” | and” | “aoj” | “obj” | “icl” | ...
<digit> ::=           0 | 1 | 2 | ... | 9
<upper case alphabetical character> ::=    “A” | ... | “Z”
<character> ::=      “a” | ... | “z” | “_” | ” “ | “#” | “!” | “$” | “%” | “=” | “^” | “~” | “|” | “@” | “+” | “-“ |
                      “<” | “>” | “?”
```

Interpretation

HeadWord

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. An 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 headword 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”, 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.
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”. Constraints can use Relation Labels, as they are defined in Appendix 3. Each constraint in the Constraint List should be sorted in alphabetical order.
When relation label is omitted, it is assumed that left most relation is omitted. For example, xxx(icl>change(icl>do)) can be replaced xxx(icl>change>do).

Attributes

The Constraint List can be followed by a list of attributes defined in Appendix 2, which provide information about how the concept is being used in a particular sentence.

UW ID

A UW can include an UW ID. The UW ID is simply used to indicate some referential information: that there are two 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>person)” and we could distinguish one from the other with UW IDs:

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

Types

UWs, then, 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:

- Restricted UWs, which are Head Words with a Constraint List, for example:
  state(agt>person,obj>information)
  state(equ>nation)
  state(icl>situation)
  state(icl>government)
- Extra UWs, which are a special type of Restricted UW, for example:
  ikebana(icl>activity,obj>flowers)
  samba(icl>dance)
  souflé(icl>food,pof>egg)
  murano(icl>glass,aoj>colorful)
- and Basic UWs, which are bare Head Words with no Constraint List, for example:
  go
  take
  house
  state

Restricted Uws

Restricted UWs are by far the most important. Each Restricted UW represents a more specific concept, or subset of concepts.

Consider again the examples of Restricted UWs given above:

state(agt>person,obj>information) is more specific concept (arbitrarily associated with the English word "state") that denotes situations in which humans produce some information, or state something.

state(equ>nation) is more specific sense of "state" that denotes a nation.

state(icl>situation) is more specific sense of "state" that denotes a kind of situation.

state(icl>government) is 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, that's why these 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 for a given language there is a wide variety of different words for these concepts and not, coincidentally, all the same word, as in English.
Notice that by organizing these senses around the English words, we can simplify the task of making a new UW/Specific Language dictionary: we can use a bilingual English/Specific Language dictionary and proceed from there, specifying the number different concepts necessary for each English word.
This of course does not mean that we're translating English words; we're just using the English dictionary to remind us of the concepts that we will want to deal with and thus to organize work more efficiently.

Extra Uws

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

ikebana(icl>activity, obj>flower)	"something you do with flowers"
samba(icl>dance)	"a kind of dance"
soufflé(icl>food, pof>egg)	"a kind of food made with eggs"
murano(icl>glass, aoj>colorful)	"a kind of colorful glass"

To the extent that these concepts exist for English speakers, they are expressed with foreign-language loanwords and don't always appear in English dictionaries. So, they simply have to be added if we are going to be able to use these specific concepts in the UNL system. Notice that the Constraint List or restrictions already give some idea of what concept is associated with these Extra UWs and the Constraints binary relation this concept to other concepts already present (activity, flower, egg, food, etc.).

Basic Uws

Basic UWs are character strings that correspond to an English word. They are used to structure the knowledge base and as a fall-back method for establishing correspondences between different language words when more specific correspondences cannot be found.

Compound UWs

Compound UWs are a set of binary relations that are grouped together to express a concept . A sentence itself is considered a compound UW. This allows us to deal with situations like:
[Women who wear big hats in movie theaters] should be asked to leave.
Without Compound UWs, we wouldn't be able to build up complex ideas like "women who wear big hats in move theaters" and then relate them to other concepts.

Syntax

Compound UWs are indicated by Compound UW-IDs, which are a colon ":" followed by two digits. Compound UW-IDs can also be followed by an AttributeList.

More formally, their syntax can be described as follows:

```
<Compound UW> ::= ":" <Compound UW-ID> ["."<Attribute List>]
<Compound UW-ID> ::= {<upper case alphabetical character> |<digit>}
                     {<upper case alphabetical character> |<digit>}
<Attribute List> ::= <Attribute Label> ["." <Attribute Label>]…
<Attribute Label> ::= "@imperative" | "@may" | "@past" | ...
<digit> ::=   0 | 1 | 2 | ... | 9
<upper case alphabetical character> ::=   "A" | ... | "Z"
```

Interpretation

Compound UWs denote complex concepts that are to be interpreted as unit-concepts, understood as a whole so that we 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 example does not mean that [women] or [women who wear big hats] should be asked to leave. Only when we group the structure together and talk about it as a whole unit do we get the correct interpretation.
Just as we can relate such complex units to other concepts with conceptual relations, we can attach Attributes to them to express, negation, speaker attitudes, etc. which are usually interpreted as modifying the main predicate within the Compound UW.

How to define Compound Uws

Compound UWs are 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), woman(icl>person).@pl)
obj:01(wear(icl>do), hat(icl>thing))
aoj:01(big(aoj>thing), hat(icl>thing))
plc:01(wear(icl>do, theater(icl>place))
mod:01(theater(icl>place), movie(icl>thing))
```

After this group has been defined, wherever ":01" is used as an UW, it means that the UW should be understood as all of these Binary relations.
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.

How to cite Compound UWs

Once defined, Compound UWs can be cited or refered to by simply using the Compound UW-ID as an UW. To complete the example above, we could continue with:

```
agt(ask(icl>do).@should, :01)
obj(ask(icl>do), leave(icl>do))
```

Again, ":01" is interpreted as the whole set of binary relations defined above. Compound UWs can be cited within other Compound UWs.

Attributes

Attribute of UWs are used to describe 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. Conceptual relations and UWs are used to describe objectively things, events and states-of-affairs in the world. Attributed of UWs enrich this description with more information about how the speaker views these states-of-affairs and his attitudes toward them.

Types of Attributes

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 present time, but as "independently of specific times".
Consider the example: The earth is round.
This sentence is true in the past, in the present and in the future, independently of speaker time, so although the tense is "present" it is not interpreted as present time.

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

Speaker's view of Aspect

A speaker can emphasize or focus on a 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.
He can focus on the beginning of the event, looking forward to it (@begin-soon), or backward to it (@begin-just).
He can focus on the middle of the event (@progress).
He can also focus on the end of the event, looking forward to it (@end-soon) or backward to it from nearby (@end just) or from farther away (@complete).
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).

@begin-soon
@begin-just
@progress
@end-soon
@end-just
@complete
@state
@repeat

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 anti-lope.
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
@def	already referred
@indef	non-specific class
@not	complement set
@order	ordinal number

These Attributes are usually attached to UWs that denote things.

Speaker’s Focus

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

@emphasis	
@entry	entry point or main UW of whole UNL expression or in hyper(scope) node
@qfocus	
@theme	instantiates an object from different class
@title	
@topic	ex) He(@topic) was killed by her.

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

Speaker’s attitudes

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

@affirmative	
@confirmation	
@exclamation	feeling of exclamation
@imperative	imperative
@interrogative	
@invitation	inducement to do
@politeness	polite feeling considered as the way to speak. “Please, could you...”
@respect	feeling of respect considered as a particle to show the respect (“Dear sir: ...”).
@vocative	ex) Dear(@vocative) , please hurry up !

Speaker’s viewpoint

The variety of possibilities reflects degrees of belief, emphasis, and the extent to which what is said should be interpreted as a suggestion or order, as well as many other social factors such as the relative status of the speakers.
The following labels are used to clarify the speaker’s viewpoint information.

@ability	ability, capability of doing things ex) He can speak English but he can't write it very well.
@ability	
@ability-past	ability in the past ex) I never could play the banjo.
@apodosis-real	apodosis: reality in the first person ex) We should (would) love to go abroad if we had the chance.
@apodosis-unreal	apodosis: A supposed result from a supposition contrary to reality ex) If we had more money, we could buy a car.
@apodosis-cond	apodosis: A supposed result from an assumed condition ex) He would smoke too much if I did not stop him.
@conclusion	He is her husband ; she is his wife.
@custom	habitual action: habit in the past ex) Every morning he would go for a long walk. John would make a mess of it.
@expectation	expectation to other's ex) He'll help you if you ask him. Will you have another cup of coffee? Will you (please, kindly, etc.) open the window? Would you excuse me ?
@grant	to give consent to do ex) Can I smoke in here? Could I smoke in here ? You may borrow my car if you like.
@grant-not	to not give consent to ex) You {mustn't/are not allowed to/may not} borrow my car.
@insistence	strong will to do ex) You shall do as I say. He shall be punished. It's your own fault; you would take the baby with you.
@intention	will, intention to do ex) He shall get this money. You shall do exactly as you wish. I'll write as soon I as can.. We won't stay longer than two hours. He will do it, whatever you say. He will keep interrupting me.
@inevitability	supposition that something is inevitable ex) They should be home by now. The game will (must / should) be finished by now. Oil will float (floats) on water. He'll (always) talk for hours if you give him the chance. There must be a mistake. Mustn't there be another reason for his behavior? They ought to be here by now.
@may	supposition of actual possibility ex) We could go to the concert. The road may be blocked. We might go to the concert. What you say might be true.
@obligation	to oblige someone ex) The vendor shall maintain the equipment in good repair.
@obligation-not	forbid to do ex) You must be back by 10 o'clock. Yesterday you had to be back by 10 o'clock. Yesterday you said you must {had to} be back by 10 o'clock.

	You {needn't/don't have to/are not obliged to} be back by 10 o'clock.
@possibility	assume reasonable possibility ex) Anybody can make mistakes. The road can be blocked. The road could be blocked.
@probability	assume probability ex) That would be his mother.
@should	to feel duty ex) You should do as he says.
@unexpected-presumption	presumption contrary to a wish or expectation ex) It is odd that you should say this to me. I am sorry that this should have happened.
@unexpected-consequence	consequence contrary to a wish or expectation ex) I made a draft, but it still needs another work.
@will	will to do ex) I shall not be long. We shall let you know our decision. We shall overcome.

Convention

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

@angle_bracket	< > is used
@double_parenthesis	(( )) is used
@double_quotation	" " is used
@parenthesis	( ) is used
@pl	plural
@single_quotation	' ' is used
@square_bracket	[ ] is used

Format of UNL

UNL Document

The structure of UNL document is expressed using following tags.

[D] Beginning of document
[/D] End of document
[P] Beginning of paragraph
[/P] End of paragraph
[S] Beginning of sentence
[/S] End of sentence

UNL documents are generally constructed in the following manner.

[D] Beginning of document
[P] Beginning of paragraph
[S] Beginning of sentence
… UNL expressions
[/S] End of sentence
… Repetition of [S]…[/S]
[/P] End of paragraph
… Repetition of [P]…[/P]
[/D] End of document
… Repetition of [D]…[/D]

UNL Expression

UNL expression is identified with the following tags:

{unl} Beginning of UNL expression
{/unl} End of UNL expression

There are two kinds of UNL expression, one is table form and another is list form. Table form of UNL expression is more readable than list form, but list form of UNL expression is more compact than table form.
In UNL expression, there are three types of information, such as binary relations, Uws, and encoded binary relations. The following tags are used to distinguish this information.

[W] Beginning of UW set
[/W] End of UW set
[R] Beginning of binary relations
[/R] End of binary relations

<Binary Relation> ::= <Relation Label> [“:”<Compound UW-ID>]
“(“ {<UW1>] | “:” <Compound UW-ID1>} “,”
{<UW2>] |“:” <Compound UW-ID2>} “)”

<UW> ::= <Head Word> [<Constraint List>] [“:” <UW-ID>] [“.” <Attribute List>]

<Encoded Binary Relation> := {<UW-ID> | <Compound UW-ID>}
<Relation Label> [“:” <Compound UW-ID>]
{<UW-ID> | <Compound UW-ID>}

Table form of UNL expression

Table form of UNL expression consist of binary relations or only one UW.

{unl}
<Binary Relation> …
{/unl}

or

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

List form of UNL expression

List form of UNL expression consists of UWs and encoded binary relations.

{unl}
[W]
<UW> …
[/W]
[R]
<Encoded Binary Relation> …
[/R]
{/unl}

Each tag, binary relation, UW, and encoded binary relation should be separated with carriage return (0x0a, or 0x0d 0x0a).

Sample of UNL expression in list form.

ex1) Monkey eats bananas.

```
[S]
{unl}
[W]
eat(icl>do).@present.@entry:00
monkey(icl>animal).@generic:01
banana(icl>food).@generic:02
[/W]
[R]
00agt01
00obj02
[/R]
[/S]
```

ex2) UNL is a common language that would be used for network communications.

```
[S]
```

```
{unl}
[W]
language(icl>abstract thing).@present.@entry:00
UNL(icl>language).@topic:01
common(aoj>thing):02
use(icl>do).@present:03
language(icl>abstract thing).@present.@entry:04
communication(icl>action).@pl:05
network(icl>thing):06
[/W]
[R]
00aoj01
00mod02
03obj04
03pur05
05mod06
[/R]
[/S]
```

Appendix 1: Relation Labels

agt	a thing in forcus which initiates an event
and	a conjunctive relation between concepts
aoj	a thing which has an attribute
bas	a thing used as the basis of comparison for focussed thing
cag	a thing not in focus which initiates an event
cao	a thing not in focus which is in a state in parallel
cnt	an equivalent concept
cob	a thing not in focus which is directly affected by an implicit event which is done in parallel
con	a non-focused event or state which influences on an focused event or state
coo	a co-occurred event or state for a focussed event or state
dur	a period of time during an event or state occurred
fmt	a range between two things
frm	an origin of a thing
gol	a final state of object or the thing finally associated with object of an event
ins	an instrument to carry out an event
man	a way to carry out event or characteristics of a state
met	a means to carry out an event
mod	a thing which restrict a focussed thing
nam	a name of a thing.
obj	a thing in focus which is directly affected by an event
or	disjunctive relation between two concepts
per	a basis or unit of proportion, rate or distribution
plc	a place an event occurs or a state is true or a thing exists
plf	a place an event begins or a state becomes true
plt	a place an event ends or a state becomes false
pof	a thing of which a focussed thing is a part.
pos	a possessor of a thing or a place
ptn	an indispensable non-focused initiator of an event
pur	a purpose or objectives of agent of an event or the purpose of a thing exist
qua	a quantity of a thing or unit
rsn	a reason that an event or a state happens
scn	a environment where an event occurs
seq	a prior event or state of a focused event or state
src	an initial state of object or the thing initially associated with object of an event
tim	a time an event occurs or a state is true
tmf	a time an event starts or a state become true
tmt	a time an event ends or a state becomes false
to	a destination of a thing
via	an intermediate place or state of an event

Appendix 2: Attribute Labels

@ability
@affirmative
@angle_bracket
@apodosis-cond
@apodosis-real
@apodosis-unreal
@begin-just
@begin-soon
@complete
@conclusion
@confirmation-req
@custom
@def
@double_parenthesis
@double_quotation
@emphasis
@end-just
@end-soon
@entry
@exclamation
@expectation
@future
@generic
@grant
@grant-not
@indef
@imperative
@inevitability
@insistence
@intention
@interrogative
@invitation
@may
@not
@obligation
@obligation-not
@order
@parenthesis
@past
@pl
@politeness
@possibility
@present
@probability
@progress
@qfocus
@repeat
@respect
@single quotation
@should
@square_bracket
@state
@theme

@thought
@title
@topic
@unexpectation
@will

Appendix 3: Conventions for syntax notation

Symbol	Definition
::=	...is defined as...
|	disjunction, “or”
[ ]	optional element
{ }	alternative element
…	one or more occurences
“ ”	encloses string of literal characters
<xxx>	variable name