



Vortrag

MicroKosmos

Nach
Ontology Development for MT: Ideology and Methodology,
Kavi Mahesh, 1995

IfI Universität Leipzig

30. Mai 2002

Inhaltsverzeichnis

1 Was ist MikroKosmos (μK)?	2
1.1 Aufbau von μK	2
1.2 Probleme bei der Übersetzung	3
2 Die Ontologie	3
2.1 Concepts	4
2.1.1 Concept als Datenstruktur	4
2.1.2 Concept als ontologischer Begriff	5
2.2 Slots und fillers	5
2.2.1 Facets	5
2.2.2 Spezielle slots	5
2.3 Relations	6
2.4 Attribute	8
2.5 Literale	8
2.6 TMR als erweiterter Instanzengraph	8
2.7 Beispiel	9
2.8 Und-Graph	9
3 Prinzipien und Probleme der Ontologieentwicklung	11
3.1 Anforderungen an die Ontologie	11
3.2 Strukturelle Prinzipien	14
3.2.1 Allgemeine Prinzipien von Ontologien	14
3.3 Redundanz	15



3.4 Erfassung von Concepts	15
3.5 Integration anderer Ontologien	17
4 Richtlinien	17
4.1 Welche Konzepte in die Ontologie	17
4.2 Namenskonventionen	17
5 Qualitätssicherung	18

1 Was ist MikroKosmos (μK)?

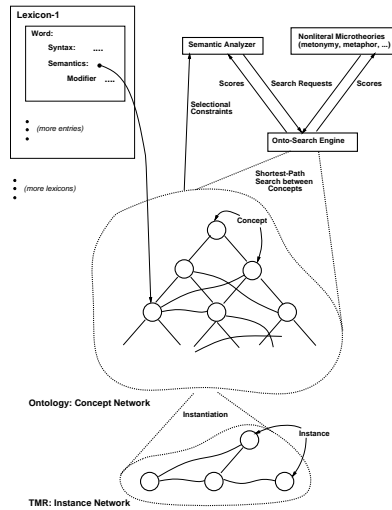
- System zur automatischen Übersetzung von spanischen Nachrichtenartikeln ins Englische
- Wurde an 400 Seiten Nachrichtenmeldungen und Artikeln über Firmenaufkäufe und Firmenfusionen getestet.
- Entwickelt an der New Mexico State University in Zusammenarbeit mit dem US Department of Defense
- Konzeptuell sprachunabhängig
- Erweiterung um andere Sprachen (Lexikas für Japanisch und Russisch sind in Arbeit)
- Es sind keine Weiterentwicklungen im Internet nach 1997 zu finden.

1.1 Aufbau von μK

Das System besteht aus den folgenden Bestandteilen.

- | | |
|-----------------------|---|
| Syntaktische Analyse: | <ul style="list-style-type: none">• Erster Schritt bei der Übersetzung• Speichert Zeitform, Satzbau, Geschlecht und Fall |
| Semantische Analyse: | <ul style="list-style-type: none">• Löst Bezüge zwischen Sätzen auf• Bedeutung von Metaphern herausfinden |
| Ontologie: | <ul style="list-style-type: none">• Notwendig um Bedeutungen zwischen den Sprachen zu übertragen.• Sprachgeneratoren der Zielsprache können Wissen teilen.• Wegen Aufgabenstellung ist breite Abdeckung von Themen der realen Welt notwendig.• Beinhaltet momentan 4000 Konzepte |
| Lexikas: | <ul style="list-style-type: none">• Für die verschiedenen Sprache möglich.• Bindeglied zwischen Ontologie und Text.• 7000 im Spanischen |
| Sprachgeneratoren: | <ul style="list-style-type: none">• Erzeugt aus TMRs (Text Meaning Representation) die Sätze in der Zielsprache.• Bisher nur für Englisch |

In Abbildung Abb. 1 ist die Struktur und das Zusammenwirken der Komponenten von μK dargestellt.

Abbildung 1: Struktur von μK

1.2 Probleme bei der Übersetzung

Bei der Übersetzung von Texten natürlicher Sprache treten zahlreiche Probleme auf, die durch den Einsatz einer Ontologie besser gelöst werden können. Folgende Probleme sollen hier erwähnt werden:

- Auflösen von Bezügen (z.B. “Klaus geht in das Seminar. Dort lernt er spannende Sachen.” Dort muß jetzt für die Lokalität des Raumes stehen, in dem das Seminar abgehalten wird.)
- Auflösen von Mehrdeutigkeiten: Mit Hilfe von Ontosearch, das den kürzesten Pfad zwischen 2 Kategorien findet.
Bsp.: lernen (etwas lernen, jmd. kennenlernen) wird durch Typen (**Abstract-Object** und **Social-Object**) aufgelöst
- Methapher richtig zuordnen: Die Bedeutung des Worts wird über die Kategorie ermittelt, die semantisch verwandt mit dem restlichen Kontext ist. Die semantische Verwandtschaft wird über die Struktur der Ontology herausgefunden.
- Lücken mit Defaults füllen: Bsp.: “John geht schwimmen, auch wenn das Draußen kalt ist” SUBSTRATE von SWIM ist WATER

Da es uns in dieser Arbeit vorrangig um die Wissensmodellierung geht, wollen wir zuerst die Unterschiede und Ideen der hier verwendeten Ontologie beleuchten.

2 Die Ontologie

Die Ontologie ist ein gerichteter Graph. Die *concepts* bilden die Knoten und die Kanten werden durch *slots* und *fillers* umgesetzt.

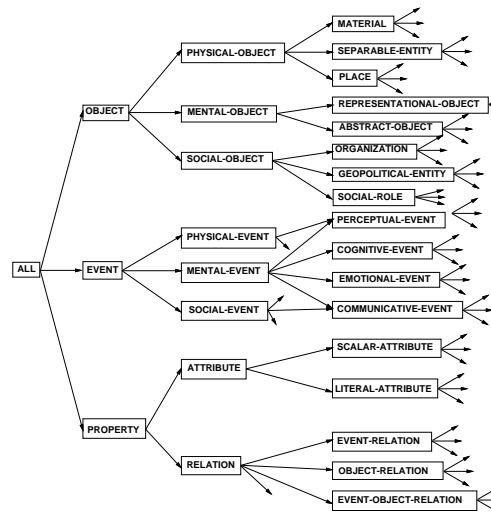


Abbildung 2: Top-Level Hierarchie

2.1 Concepts

Es ist hier von der Datenstruktur *concept* die Rede und es wird von *concepts* als bekannte Kategorien gesprochen.

2.1.1 Concept als Datenstruktur

Alle Knoten des Ontologigraphen sind von dieser Struktur. Um keine Verwirrung hervorzurufen haben wir uns entschieden im Folgenden für die Datenstruktur *node* zu sagen.

- Namen (eindeutig)
- Menge von *slots*

Folgende Strukturen sind *nodes*:

Object: Kategorie für ein Objekt. Kann alleine existieren und ist instanzierbar.

Event: Kategorie für ein Ereignis. Kann auch alleine existieren und ist instanzierbar.

Property: Eine Eigenschaft ist nicht instanzierbar. Es wird unterschieden in:

- **Relation**
- **Attribute**

Alle Attribute und Relationen werden durch das Belegen der *slots* mit *nodes* verwirklicht. In der Abb. 2 sind die ersten Ebenen der Ontologie dargestellt.



2.1.2 Concept als ontologischer Begriff

Bei μK wird der Begriff *concept* im Sinne von Kategorie benutzt und steht für *objects* oder *events*.

2.2 Slots und fillers

Wie schon erwähnt sind die Beziehungen der *nodes* ausschliesslich über *slots* und *fillers* realisiert. Ein *filler* ist der *node*, der einen *slot* belegt. Die genaue Rolle der *filler* wird im nächsten Abschnitt erläutert. Den *node* der einen *slot* belegt heißt *filler*.

- Ein *slot* ist entweder ein *relation-slot* oder ein *attribute-slot*.
- Ein *slot* hat dann den Namen der *relation* oder des *attribute*s. (Typisierung)
- Es gibt eine abgeschlossene Menge von Spezial- *slots*.
- Spezielle *slot*-Namen: sind nicht als *property*s gespeichert.

2.2.1 Facets

Genau genommen ist ein *slot* ein 6-dimensionaler Vektor, denn jeder *slot* hat sechs verschiedene feste *facets* die unabhängig mit *fillern* belegt werden können. Eine Ausnahme bilden hier die spezielle *slots*, die wir später noch behalgen werden. Ein *facet* schreibt vor welche *nodes* diesen Teil-*slot* belegen dürfen.

Value: Wird normalerweise mit Instanz eines *concepts* oder mit einem Skalar belegt.

Sem: *Filler* ist ein anderes *concept* oder ein skalarer Bereich

Default: Wie *value*. Wird benutzt bei Instanziierung wenn *value-facet* fehlt.

Measuring-Unit: Quantisiert *value*. Maßeinheit-*concept* als *filler*.

Saliency: Bezeichnet die Wichtigkeit eines *slots* für den *node*.

Relaxable-to: Wird nur in Lexikon benutzt. Bezeichnet wie hoch bei Metaphern gesucht werden soll. Sollte mit dem Top-*concept* aller Möglichkeiten belegt werden.

2.2.2 Spezielle slots

Es gibt eine feste Menge von speziellen *slots*, die eine gesonderte Rolle einnehmen. Sie tauchen nicht als *node* in der Ontologie auf und haben meistens nur ein *facet*.

Pflicht *slots* für alle *nodes*:

Definition: Nur *value-facet* mit Englischer Text, der nur für den Menschen oder bestimmt ist.

Is-A: Nur *value-facet* mit Liste der direkten Eltern. (Außer dem ALL *concept*)

Subclasses: Nur *value-facet* mit Liste der direkten Kinder. (Außer den Blättern)

Slots für *property*s:

Domain: Pflicht für alle *property*s. Hat *sem-facet* mit der Liste der *concepts* die als Domain für die *relation* oder das *attribute* möglich sind.

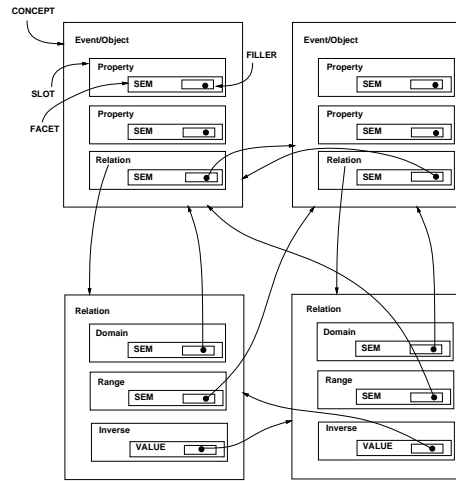


Abbildung 3: abstraktes Beispiel einer Relation

Range: Pflicht für alle *property*s und hat auch nur *sem-facet*.

- **Relations**: Liste der *concepts* die als Wertebereich möglich sind.
- **Attributes**: Liste der Literale oder ein numerischer Wertebereich.

Inverse: Nur für *relations*. Hat nur *value-facet* mit der inversen Relation.

Weitere *slots*:

Instances: Hat nur *value-facet* mit der Liste der Instanzen dieses *concepts*

Instance-Of: Pflicht für alle Instanzen. Hat nur *value-facet* mit dem parent *concept*

Bei allen anderen *slots* sind meistens *sem-facets* in *concepts* und *value-facets* in Instanzen belegt. Kein *facet* ist Pflicht.

2.3 Relations

Jede *relation* ist als *node* in der Ontologie gespeichert. *Property*s sind nicht “stand-alone” und bilden einen Teilgraph der Ontologie (Abb. 4). Wenn zwei *concepts* mit einer *relation* verbunden sind, so gibt es in der Ontologie zwei *relations* die diesen Sachverhalt ausdrücken. Beide sind inverse zueinander. In Abb. 3 ist beispielhaft dargestellt wie eine *relation* präsentiert wird.

Ein Ausschnitt aus der Ontologie unter **RELATION** zeigt Abb. 4.

Es kann zu komplexen Strukturen kommen, wenn die Relationspartner verwandt sind.

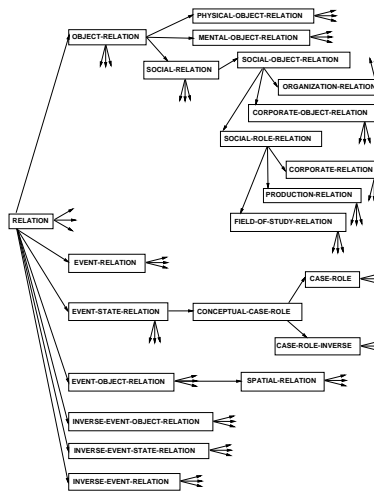


Abbildung 4: Ausschnitt aus der Ontologie

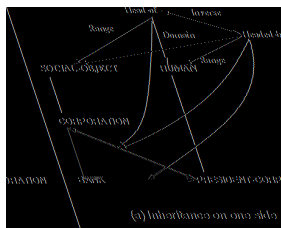


Abbildung 5: Vererbung auf einer Seite

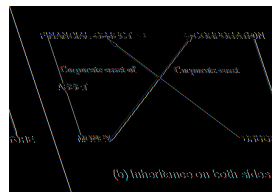


Abbildung 6: Vererbung auf einer Seite

2.4 Attribute

Attribute von *concepts* werden durch einen *attribute-slot* realisiert. Es gibt zu jedem *attribute* ein *attribute-node* mit den festen *slots Domain* und *Range*. (Siehe Abb. 7)

2.5 Literale

Literale sind wichtig, um unendliche Ketten bei der Bedeutungsanalyse zu vermeiden.

- Sind entweder binär oder Werte auf einer beinhaltenden Skala.
- Binäre literale werden Attributspezifische Werte genutzt anstatt von “yes” und “no”. Werden genutzt wenn keine numerische Skala besteht.
- Auch wenn es eine Skala gibt wird sie nicht unbedingt genutzt. Bsp: Farbe.
- Es wird nicht *light-red* eingeführt sondern eine Relation **GREATER-THAN** und **LESS-THEN** eingeführt.

2.6 TMR als erweiterter Instanzengraph

TMRs werden als Eingabe für den Sprachgenerator genutzt. Sie beinhalten neben linguistischen Informationen auch einen Instanzengraph. Es wird versucht den sematischen Inhalt eines Ausgangssatzes oder Teilsatzes durch Instanzen der *concepts* wiederzuspiegeln.

- Instanzen sind auch *nodes*
- Unterscheiden sich nur in der Belegung der *slots*

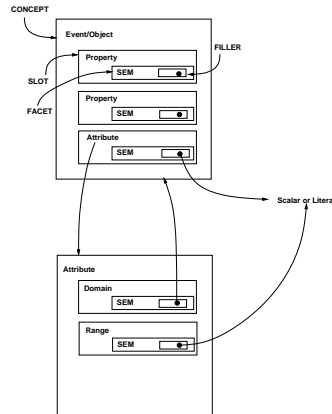


Abbildung 7: abstraktes Beispiel eines Attributes

- *Value-facet* sind mit konkreten Skalaren, Literalen oder Instanzen von *concepts* belegt.

2.7 Beispiel

Es soll an Hand eines Beispiels die bisher eingeführten Strukturen verdeutlicht werden. Wir können nicht garantieren, daß das folgende Beispiel in μK genau so verarbeitet wird. “I eat a green Apple.” soll der Ausgangssatz sein.

conzepts:

- HUMAN (*object*)
- APPLE (*object*)
- EAT (*event*)

2.8 Und-Graph

Die Ontologie von μK ist ein Und-Graph. Das heißt bei Mehrfachvererbung ist das Kind immer die Konjunktion aller Eltern. Diese mengentheoretische Beziehung besteht zwischen den Modellen (Menge der Instanzen) nicht zwischen den Eigenschaften. Disjunktionen der Modelle zweier Konzepte ist nicht möglich. Ein Beispiel soll dies verdeutlichen: Es gibt Organisationen, die profitorientiert sind und welche, die nichtprofitorientiert sind. Labore sind entweder das Eine oder das Andere, sind aber auf alle Fälle Organisationen. In einem Oder-Graph wäre das jetzt klar: Labor ist Kind von “profitorientierten” und “nicht profitorientierten” Organisationen, mit der Eigenschaft, das eine Instanz entweder das Eine oder das Andere ist. Im Und-Graph ist das eine Kontradiktion und ist demzufolge nicht gültig. Die Lösung in μK ist in Abb. 8 zu sehen.

Problem: Mögliche Fehlinterpretation und keine disjunkte Zerlegung des Elternkonzepts.

Die eigentliche Bedeutung und die Fehlinterpretation ist in 9 zu erkennen.

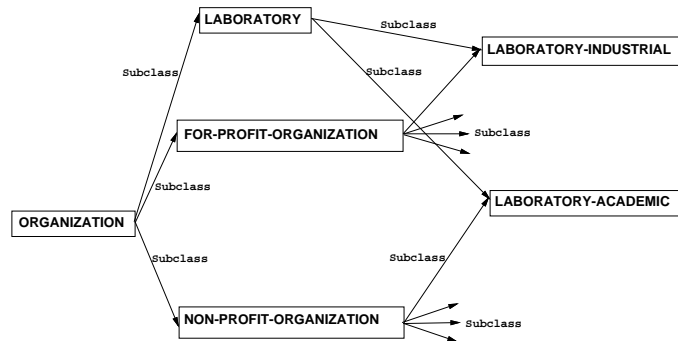


Abbildung 8: Muster zur Generierung überlappender Teilklassen

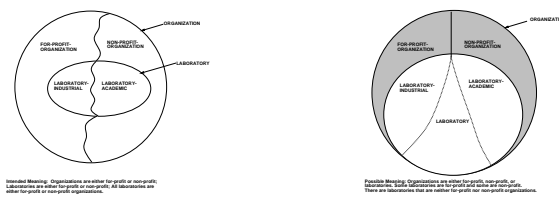


Abbildung 9: links: eigentliche Bedeutung. rechts:mögliche Fehlinterpretation



3 Prinzipien und Probleme der Ontologieentwicklung

3.1 Anforderungen an die Ontologie

Die Ontologie wurde einer Reihe von aufgaben-spezifischen Regeln unterworfen, die im näheren Abschnitt näher beleuchtet werden.

- Sprachunabhängigkeit

(keine 1:1-Abbildungen, Idealfall: 2 Lexikas werden neben der Ontologie entwickelt)

Um Sprachunabhängigkeit in der Ontologie zu erreichen, müssen Wörter mit gleicher Bedeutung auch auf das selbe Konzept verweisen. Dadurch fungiert die Ontologie als Meta-Sprache für die Übersetzung und darf nicht nur auf den Bedeutungen der Wörter aus nur einer Sprache basieren, da sonst keine sinnvolle Übersetzung zustandekommt.

- Entwicklung der Ontologie darf sich nicht der der Lexikas unterordnen

Die Erfasser der Ontologie müssen den Blick auf die verschiedenen Bedeutungen in den einzelnen Sprachen richten

- Wollgeformtheit entsprechend den axiomatischen Spezifikation

Der Aufbau der Ontologie soll gewissen Regeln folgen, die die Beziehung zwischen *concept*, *event*, *property* und *slots*, *facets* sowie *filler*n definieren.

- Konsistenz und Kompatibilität mit den anderen Teilen von μK Nur durch die Zusammenarbeit und die systematische Verwendung der Ontologie in allen Teilen von μK wird maschinelle Übersetzung erst möglich.

- reich an Inhalt, hochgradige Vernetzung der Konzepte

keine bloße Hierarchie von Konzeptnamen; Konzept nicht nur als Bezeichner

Durch den Aufbau einer Hierarchie von Klassen von Objekten kann man noch lange keine Sätze von einer Sprache in eine andere übersetzen. Erst durch Ontosearch und die Verknüpfung von Konzepten auch quer zur Hierarchie wird ein vernünftiges Übersetzen durch die Suche von kürzesten Pfaden möglich.

- Verständlichkeit

Eine gute Handhabbarkeit der Ontologie durch leichte Verständlichkeit, eine einfache Suche und die Möglichkeit des Browsens der Hierarchie ist entscheidend für die Qualität der Lexikas und somit auch der Übersetzungen. Diese Forderung erhält ihre Wichtigkeit dadurch, daß die Mehrheit der Mitarbeiter keine Experten in der Wissensrepräsentation sind. Für Linguisten oder Experten in einem Spezialgebiet muß es ein leichtes sein, sich in das Konzept der Ontologie einzuarbeiten und die richtigen Konzepte in den Lexikas zu verwenden.

- Nutzen für NLP steht im Vordergrund

Mehrdeutigkeiten auflösen und notwendige Ableitungen in beliebige Tiefe durchführen

Laut Mahesh ist es enorm wichtig, das Anwendungsgebiet der Ontologie nicht aus den Augen zu verlieren. Er meint, daß eine Ontologie nur nach der Anwendung und den ihren Bedürfnissen modelliert werden kann.

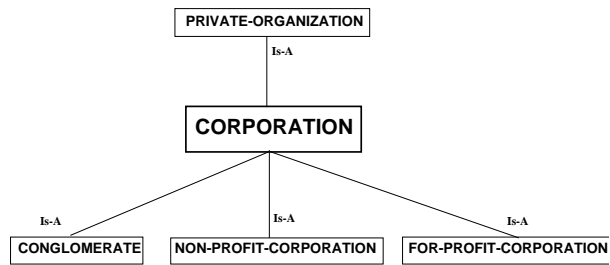


Abbildung 10: Taxonomische Relationen

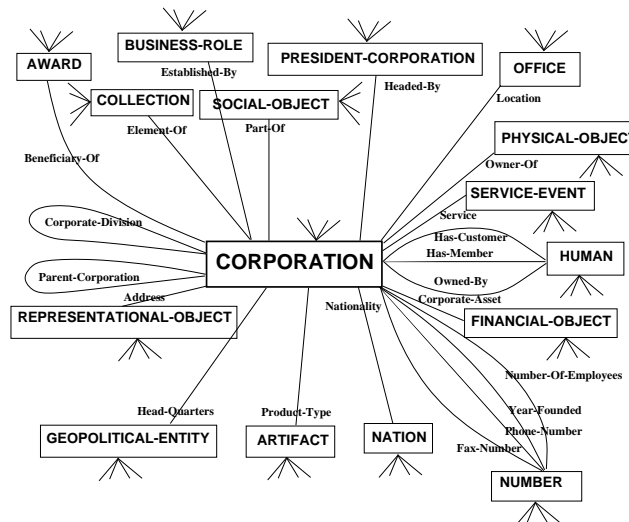


Abbildung 11: Nicht-taxonomische Relationen



- beschränktes Themengebiet Ziel der Ontologie ist es nicht eine umfassende Enzyklopädie zu werden, sondern auf einem gewissen G
- kein episodisches Wissen Es werden keine Instanzen in der Ontologie erfaßt. Die TMRs - Instanzen der Ontologie - stehen aber in Verbindung mit den Konzepten.
- Automatisierung der Entwicklung: Graphische Browser, Halbautomatisierte Benutzerinterfaces, Programme zum Erkennen von Inkonsistenzen und Richtliniendurchsetzung

Durch eine Reihe von automatisierten Hilfsmitteln soll die Erfassung, die Qualitätssicherung, Abfrage und das Browsing ermöglicht werden. Dadurch wird die Ontologie erst einer breiten Nutzergruppe (Lexikographen, Experten auf verschiedenster Fachbereiche, Tester etc.) zugänglich.

3.2 Strukturelle Prinzipien

Entwicklung der Ontologie nicht mit vorgefertigter Struktur

→ Modellierung der Ontologie nach der Welt und den Bedürfnissen

Mahesh strukturiert die Ontologie nicht nach festen in anderen Papern vorgestellten Prinzipien, sondern orientiert sich ausschließlich an den der Aufgabe förderlichen Elemente. Deshalb werden folgende Prinzipien aufgeweicht.

3.2.1 Allgemeine Prinzipien von Ontologien

- **Gleichartigkeit:** Ein Kindkonzept muß die Bedeutung des Oberkonzeptes teilen.

μK : Man kann *properties* bei der Vererbung mit ***nothing*** blocken → Nicht-Monotonie.

Durch das Einführen des Blockens, erreicht man eine Nicht-Monotonie. Diese erleichtert die Wartung erheblich, wenn z.B. fast alle Kinder eines Konzeptes eine gewisse Eigenschaft haben, ist es möglich diese in den Ausnahmen zu "blocken". Dadurch erspart man sich die explizite Aufzählung der Kinder, die die Eigenschaft tragen.

Bsp: eine *event* wird gestartet mit **ACTIVATE**: startet man ein *event*, braucht dieses einen *agent*. Deshalb braucht *event* einen *slot* für *agent*.

Es gibt aber zwei *events* die keinen Agent benötigen. **PASSIVE-COGNITIVE-EVENT** und **INVOLUNTARY-PERCEPTUAL-EVENT**

- **Spezifität:** Ein Kind-Konzept muß sich von seinen Eltern unverkennbar unterscheiden.

μK : Diskriminator wird nicht repräsentiert.

Beispiel: Unterschied von **ANIMAL** und **INVERTEBRATE** ist nicht dargestellt in der Ontologie.

Ein Diskriminator zwischen Konzepten und Kind-Konzepten vereinfacht die Suche nach einem verwandten Konzept nicht, da diese von Ontosearch nur durch Eltern-Konzepte, Kind-Konzepte, Relationen, Attribute gefunden werden. Deshalb erfaßt man ihn nicht extra mit, da er keinen praktischen Nutzen für μK bietet.

- **Gegensatzprinzip** Ein Konzept muß sich von seinen Geschwistern unterscheiden und der Unterschied muß repräsentiert werden.

μK : nicht notwendig, wir brauchen keine Vollständigkeit.



Zwischen WALK und RUN wird keine Abgrenzung vorgenommen.

In μK wird die Abgrenzung von einem Nachbarn meist nur durch den Knoten in der Hierarchie vorgenommen. Diese Unterscheidung reicht für μK meist aus, da es in jeder Sprache meist ähnliche Wörter gibt.

- **eindeutige semantische Achse:** Alle Geschwister müssen sich in einer Art und Weise unterscheiden.

μK : Zu restriktiver Ansatz. Mehrere semantische Achsen mittels *property*s und *concept*s möglich.

Beispiel: ORGANIZATION mit dem Kind NON-PROFIT-ORGANIZATION und LABORATORY und der *property* CUSTOMER-CONTACT-ATTRIBUTE (repräsentiert eine Dritte Achse)

3.3 Redundanz

Der Abwägungsprozess in der Modellierung hat uns schon in den vorangegangenen Vorträgen beschäftigt. Wir möchten nun ein wenig auf die Entscheidungsprozesse und die Redundanz, die entsteht eingehen.

- Dualität zwischen *objects* und *property*s,

Wie modelliert man “headquarter of a corporation”?

- a) *concept* ORGANIZATION noch in HEADQUARTER und BRANCHES unterteilen oder
- b) als *attribute*

- Dualität zwischen *events* und *property*s,

Oft gibt es bei *events* mehrere Phasen:

- Beginphase, FALL-ASLEEP oder ACQUIRE
- fortlaufende Phase, *state*, FALL-ASLEEP oder ACQUIRE
- Endphase, WAKE-UP oder RELINQUISH

Möglichkeiten: OWN als *Event* oder eine OWNED-BY und OWNED-OF *property*

ACQUIRE ist ein zentrales Konzept von μK wird als *event* modelliert.

- Dualität zwischen *objects* und *events*

Das *event* (OWN) und *object* (LABORATORY) müssen mit einer *property* OWNED-BY verbunden werden. Außerdem muß ACQUIRE mit OWNED-BY verbunden werden, damit im TMR eine Verbindung zwischen den beiden existiert.

Die Dualität zwischen den beiden kommt dadurch zustande, daß OWN alle Beteiligten *concepts* verlinken muß und aber trotzdem eine Verbindung von den Beteiligten *concepts* existieren muß. Dadurch entsteht eine notwendige Redundanz.

3.4 Erfassung von Concepts

Bei der Erfassung von neuen *concepts* ist es notwendig, daß Themengebiet hinter dem Konzept näher zu betrachten, um eine sinnvolle Modellierung zu erreichen. Dabei wird das betrachtete Gebiet in orthogonale

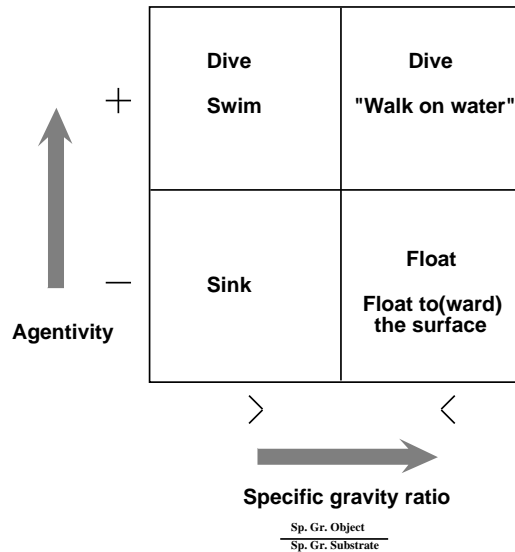


Abbildung 12: Die verschiedenen semantischen Achsen AGENTIVITY und GRAVITIVY

semantische Achsen zerlegt und die für das Anwendungsgebiet notwendigen Achsen in den neuen *concepts* eingebaut.

- gegeben: zwei englische Worte *swim* und *float*
- Bedeutung der Wörter verdeutlichen
 - English: Swim → SWIM, Float → FLOAT
 - Spanish: Nadar → SWIM, Flotar → FLOAT
 - Russian: Plyt → SWIM or FLOAT, Plabat → SWIM or FLOAT

Bedeutung der Wörter in verschiedenen Sprachen

Durch die Vergegenwärtigung der Verwendung von Wörtern in verschiedenen Sprachen erreicht man ein besseres Verständnis für das Themengebiet.

- Dimension finden, in der sich die Worte unterscheiden... AGENTIVITY (aktive Rolle im Geschehen haben vs. passiv teilhaben)

In diesem Schritt muß der Erfasser seine Fachkenntnis und seinen Verstand einsetzen um die sprachbezogenen Achsen von den sprachunabhängigen zu unterscheiden.

- ähnliche Konzepte gleich miterfassen (*dive* und *sink*)

Diese Konzepte werden später meist eh gebraucht und es hat sich als effektiv herausgestellt, diese gleich miterfassen. Deshalb lohnt sich auch diese Mehrererfassung

- weitere Analysen sind nicht sinnvoll bei dieser Domäne



3.5 Integration anderer Ontologien

Bei der Erweiterung der Ontologie möchte man sich unnötige Arbeit sparen, deshalb ist es manchmal sinnvoll andere Ontologien mit zu integrieren. Dabei sind einige Punkte zu beachten.

Diese Punkte müssen bei der Integration sichergestellt werden, um die Qualität zu sichern.

- a) breite Themenbasis, d.h. Fachbegriffe herauslassen (keine Terminologie aus der Nerven Chirurgie)
- b) viele *Properties* und ein hoher Verbindungsgrad

Durch diese Forderungen wird es erst möglich in μK auch weiterhin Mehrdeutigkeiten notwendig um zu prüfen wie gut Auswahlbedingungen (in *Ontosearch*) erfüllt ist \rightarrow am besten passenstes Konzept instantiieren

- c) Leichte Verständlichkeit, einfache Suche und Browsing
- d) Wirtschaftlichkeit Aufgrund der beschränkten Ressourcen des Projektes, muß bei möglicherweise auf die Integration von nicht anpassbaren Ontologien verzichtet werden.

4 Richtlinien

4.1 Welche Konzepte in die Ontologie

- *concepts* nicht hinzufügen nur weil es möglich ist.
- bei *concepts* mit ähnlicher Bedeutung, daß allgemeine nehmen und ein *Attribute* für die Abstufungen einführen
- keine *events* mit vielen Argumenten, d.h. `walk-to-airport-terminal` und `walk-to-parking-lot` vermeiden.
- englische Worte, keinen Plural, Konzept

4.2 Namenskonventionen

Es sind nur alphanummerische Zeichenketten mit Bindestrichen erlaubt.

- normale Regeln: kein Plural, keine Umgangssprache, zusammengesetzte Wörter nicht zweideutig wählen (`unit-of-time` statt `time-unit`), Namensdopplung ausschließen (bei *properties* und *objects*: `employee` als *object* und `employed-by` für *property*)
- *concept* Namen: englisch, max. vier Wörter mit Bindestrichen getrennt.
- *instance* Namen: Namen der Instanz + *concept* Name + Integer (jeweils mit – getrennt)
- Literale: englisch, meist ein Wort.
- Weiter Symbole sind: mathematische Symbole, Lexikonische Symbole, TMR-Symbole



5 Qualitätssicherung

Während der Entwicklung der Ontologie auf Grundlage von anderen Ontologien war zu Anfang eine große Bereinigung von *concepts* nötig, die keine waren. Außerdem wurden die Namen besser gewählt und *concepts* zusammengefaßt. Danach wurden zahlreiche neue Konzepte erfaßt und die Ontologie gemäß den aufgestellten Richtlinien erweitert. Für den Erfolg der Ontologie und die Sicherung der Qualität waren nun folgende Punkte entscheidend.

- Erfassung bleibt lenkbar (man kann sich am Ziel orientieren; keine Automatisierung)
- Waisen entfernen, da jedes Konzept mit jedem auf Ähnlichkeit vergleichbar bleiben muß
- Computerunterstützung: fehlende Erklärungstexte, falsch geschriebene Bezeichner, Waisenerkennung, Relationen ohne inverse Relationen
- Nutzer finden die tieferliegenden Redundanzen und Inkonsistenzen

Resourcen

Paper:

Ontology Development for MT: Ideology and Methodology,
Kavi Mahesh, 1995

Internet:

<http://crl.nmsu.edu/users/mahesh/onto-intro-page.html>