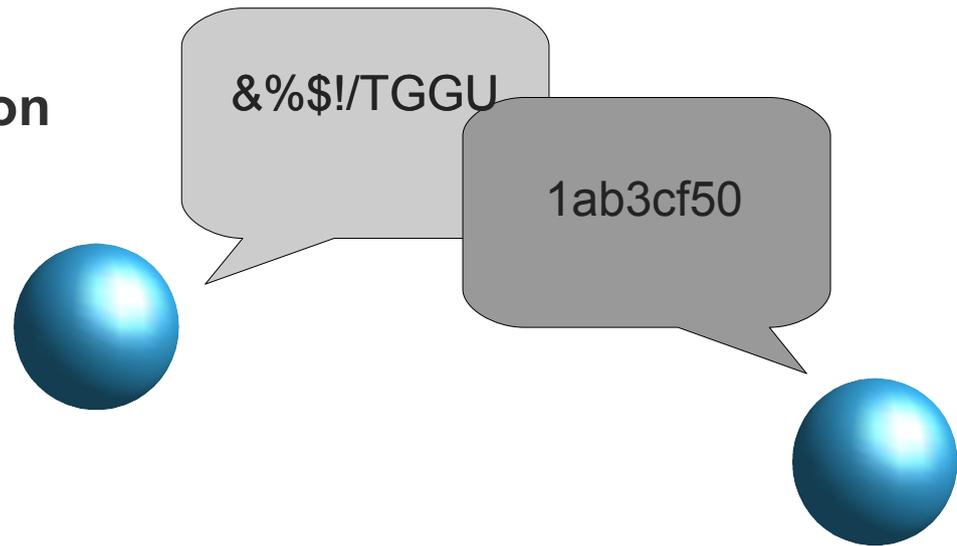


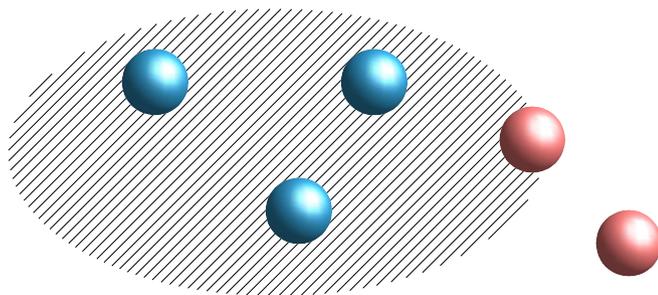
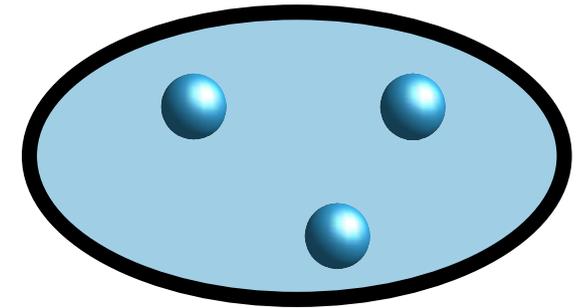
KQML

Wissensbasierte Kommunikation
zwischen Agenten



Welten

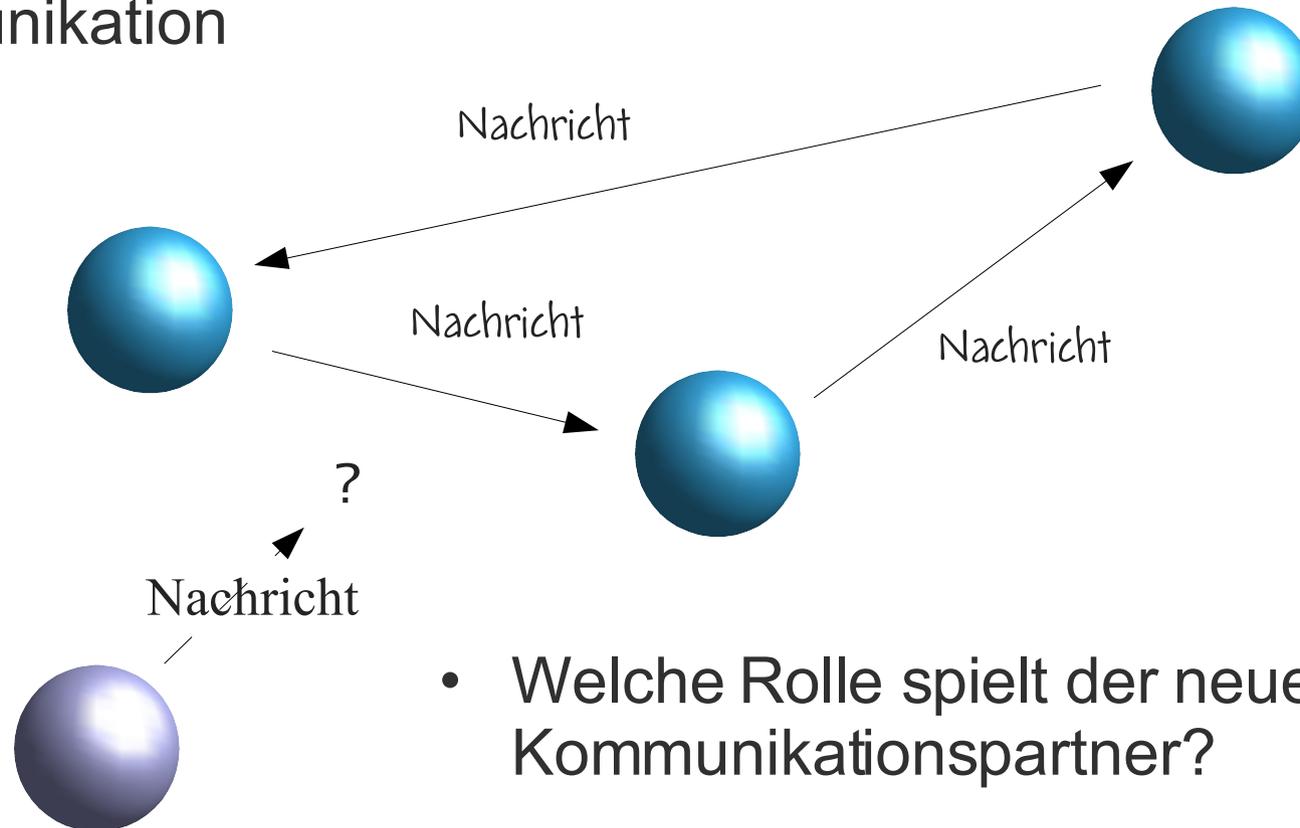
- Geschlossene Welt – geschlossene Anwendungen
 - *Closed World Assumption*
 - Fester Satz von Komponenten
- Offene Welt – offene Anwendungen
 - *Open World Assumption*
 - Ergänzung einer Anwendung jederzeit möglich



Wie nimmt man Kontakt auf?

Agentenkommunikation

- Kommunikation



- Welche Rolle spielt der neue Kommunikationspartner?
- Wie funktioniert die Kommunikation?

Geschlossene Anwendung



„Optimierte“ Verbindung
Nur innerhalb der Anwendung definiert

- Eigenschaften der geschlossenen Anwendung
 - Bekannte Kommunikationspartner
 - Bekannte Datenformate / Protokoll
- Vorteil
 - Geringes Datenaufkommen
 - Korrektheit der Kommunikation überprüfbar

Geschlossene Anwendung

- Nachteile der optimierten Kommunikation
 - Einbindung neuer Komponenten schwierig
 - Protokolle und Formate müssen zuvor festgelegt sein und dürfen sich nicht substantziell ändern
 - Formate und Protokolle implizieren bereits zu viel Struktur in den Agenten
 - Exakte Übereinstimmung der Nachrichtenformate und der Protokolle
 - Daraus folgend: festgelegte Abfolgen der Verarbeitung von Nachrichten innerhalb der Agenten
 - Wenig Spielraum für „Abwägungen“ des Agenten
 - Schlechte Erweiterbarkeit
 - Abhilfe mit strukturierten Textformaten wie XML

Standardisierte Kommunikation

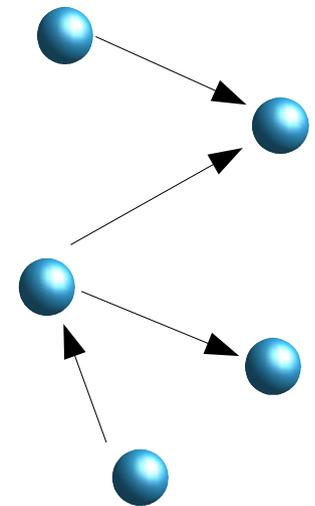
- Applikationsunabhängige Standardisierung
 - Definierte (generelle) Protokolle und Datenformate
 - daraus folgend: leichtere Wartbarkeit
 - Leichtere Integration neuer Komponenten
- Erleichterung des Datenaustauschs
 - Standardisierte Datenabfrage / Wissensabfrage über verschiedene Applikationen hinweg

Knowledge Query and Manipulation Language

Labrou, Finin (1993)

Voraussetzungen von KQML

- Agenten verfügen über gerichtete Kommunikationskanäle, die abgegrenzt definierte Datenpakete (*Nachrichten*) transportieren.
- Der Nachrichtentransport kann Zeit benötigen.
- Agenten können den Eingangskanal bestimmen, über den eine Nachricht eintraf.
- Es ist möglich, den Ausgangskanal anzugeben, über den eine Nachricht versendet werden soll.
- Das Transportsystem beachtet Reihenfolgen.
- Das Nachrichtensystem ist verlässlich.
 - Aber sind es die Agenten auch?



Voraussetzungen

- Voraussetzungen werden erfüllt durch
 - TCP-Verbindungen (Sockets)
 - Mail-Verbindungen (Agenten schicken sich E-Mail)
 - jedoch unsicher bzgl. Reihenfolge!
 - UNIX IPC (Interprozesskommunikation, Pipes)
- Auch Hardwarelösungen
 - feste Verbindungen zwischen verschiedenen Rechnern, z.B. Switches

Jede Kommunikation muss für einen Agenten wie eine Punkt-zu-Punkt-Kommunikation aussehen

Sprechakttheorie: Austin (1962)

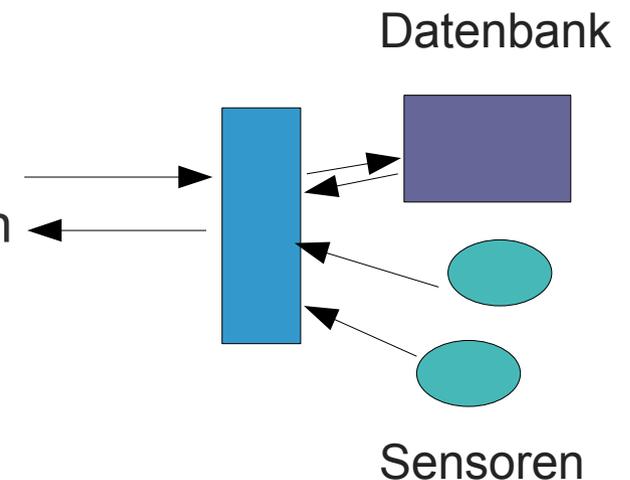
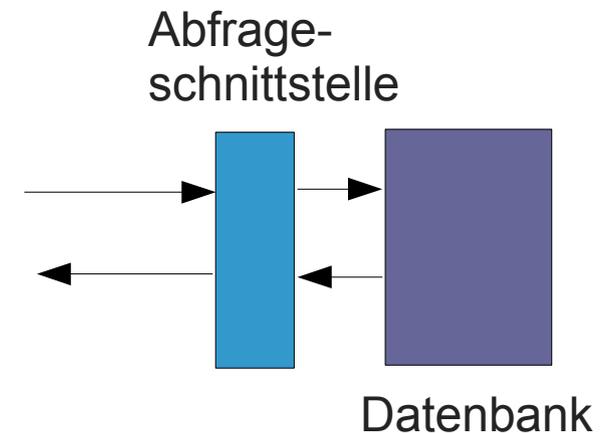
- Sprache ist nicht nur zu begreifen als Mittel zur Informationsübermittlung
 - Sprache kann Aktionen bewirken oder den Zustand der Welt ändern: „Sprechakte“
 - Beispiel: „Hiermit erkläre ich Sie zu Mann und Frau“
- Performative Verben
 - anfragen, informieren, versprechen
- Aspekte
 - Lokutionärer Akt: eine Lautäußerung hervorbringen (das Sprechen)
 - Illokutionärer Akt: eine Aktion durch eine Äußerung erwirken
 - Perlokutionärer Akt: das Ergebnis

Sprechakttheorie: Searle (1969)

- Einteilung von Sprechakten
 - Repräsentativa/Assertiva: dienen dem Informieren, Behaupten, Mitteilen
 - Direktiva: dienen dem Bitten, Fordern
 - Kommissiva: dienen dem Zusagen, Verpflichten, Drohen
 - Expressiva: dienen dem Mitteilen des eigenen Zustands (danken, grüßen, klagen)
 - Deklarativa: dienen dem Verkünden (eines institutionellen Vorgangs; ernennen, taufen, vermählen)
- Direkte und indirekte Sprechakte
 - Indirekt: die eigentliche Intention verschleiern
 - „Die Ampel ist grün“ (vom Beifahrer geäußert)

Wissensbasen

- Wissensbasis
 - beinhaltet das „Wissen“ des Agenten
 - z.B. organisiert als Datenbank mit Abfragesprache
 - im einfachsten Fall der Zustand des Agenten
- Virtuelle Wissensbasis (virtual knowledge base, VKB)
 - muss nicht alle Daten explizit beinhalten
 - muss lediglich Anfragen beantworten können

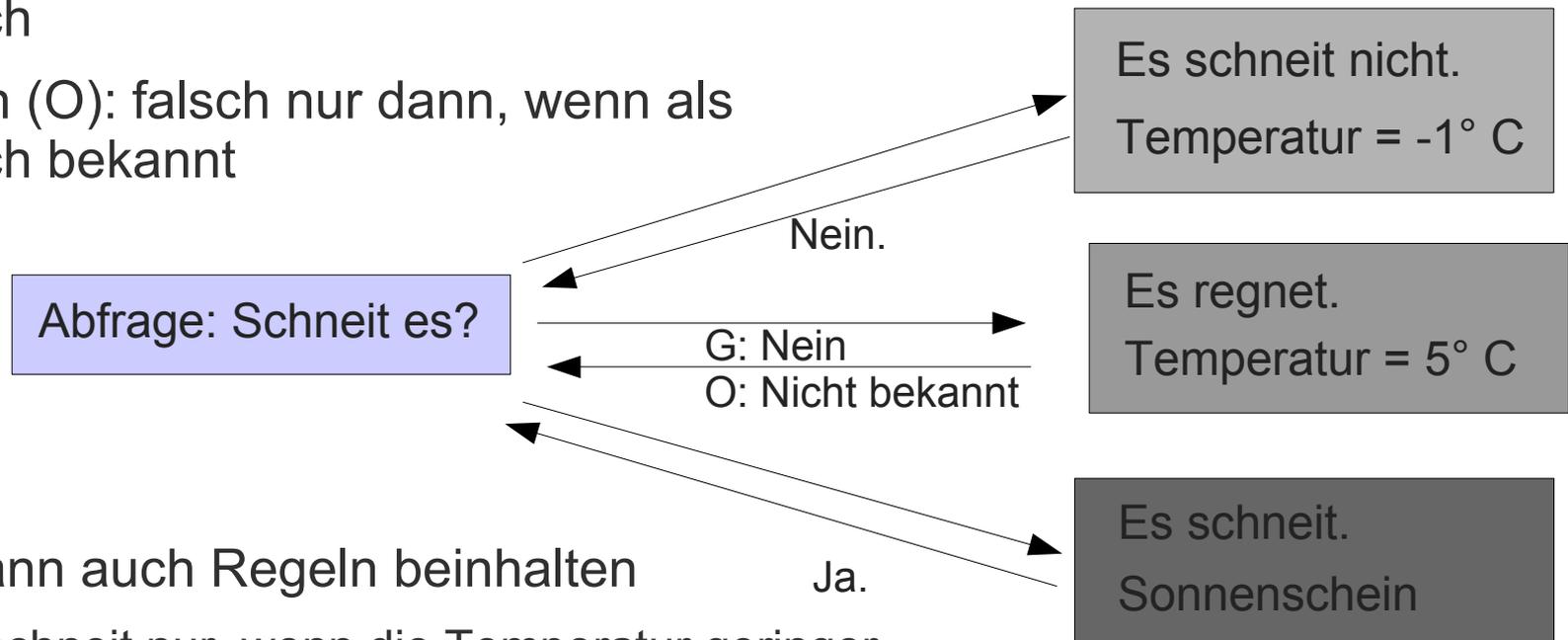


Weltmodelle

- Weltmodelle

- geschlossen (G): alles Unbekannte ist falsch
- offen (O): falsch nur dann, wenn als falsch bekannt

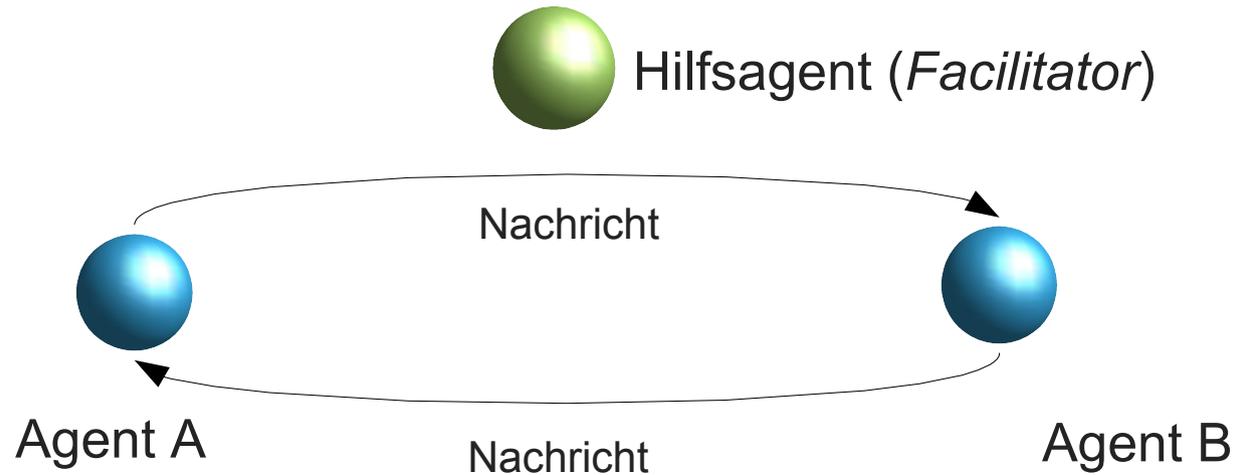
Ich weiß nicht, ob es schneit
≠ Ich weiß, dass es nicht schneit



- VKB kann auch Regeln beinhalten

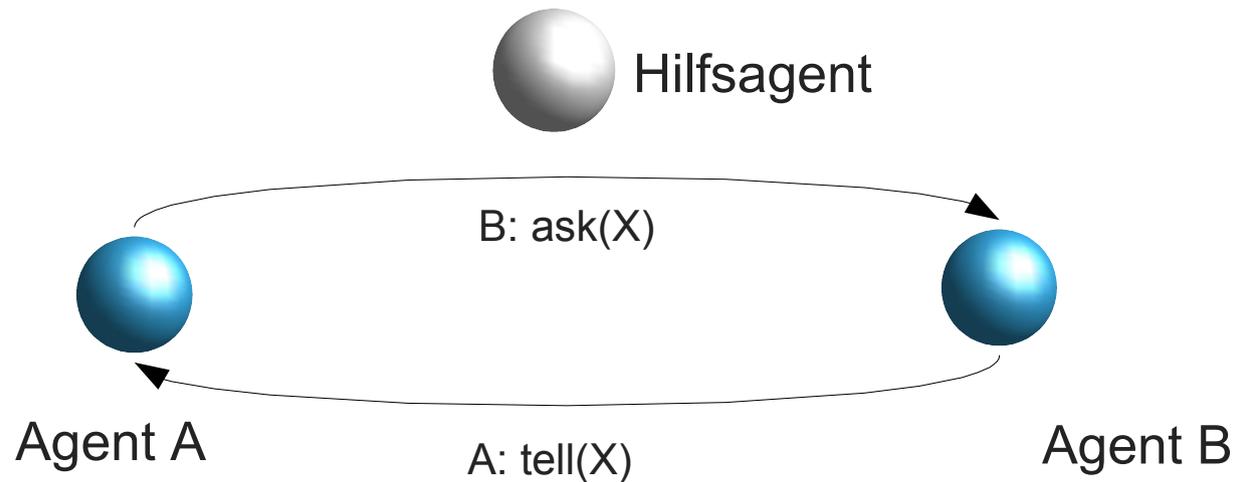
- Es schneit nur, wenn die Temperatur geringer als 1° ist
 - Aussage „Es schneit“ beweisbar / widerlegbar ohne direkte Fakten

Kommunikationsformen



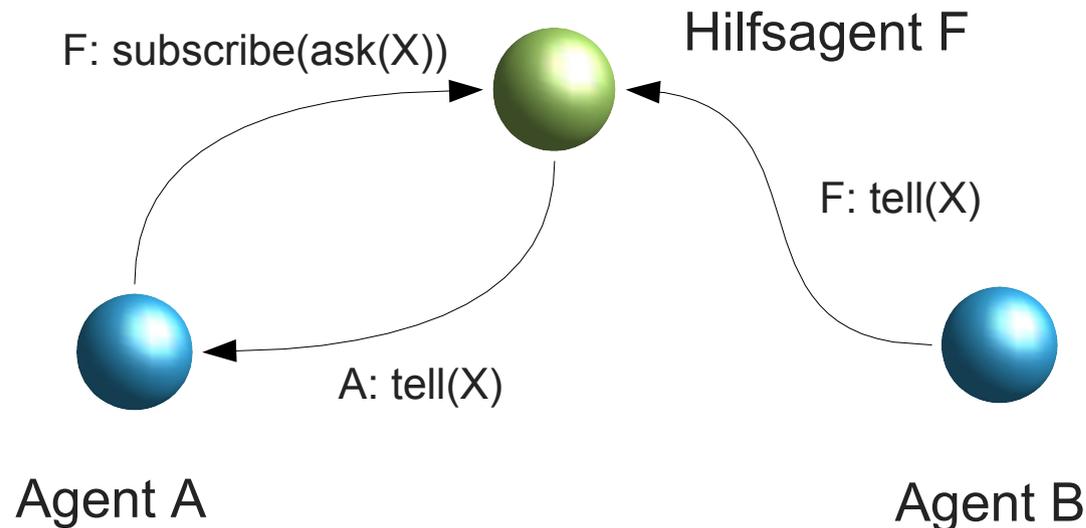
- Agenten nutzen Kanäle, um sich Nachrichten zuzusenden
 - Sender kann Kanal bestimmen (=Empfänger festlegen)
 - Empfänger kann Kanal identifizieren (=Absender feststellen)
- Kommunikationsrollen
 - im Folgenden zunächst nur 1-1-Kommunikation (kein Multicast / Broadcast)

Kommunikationsformen



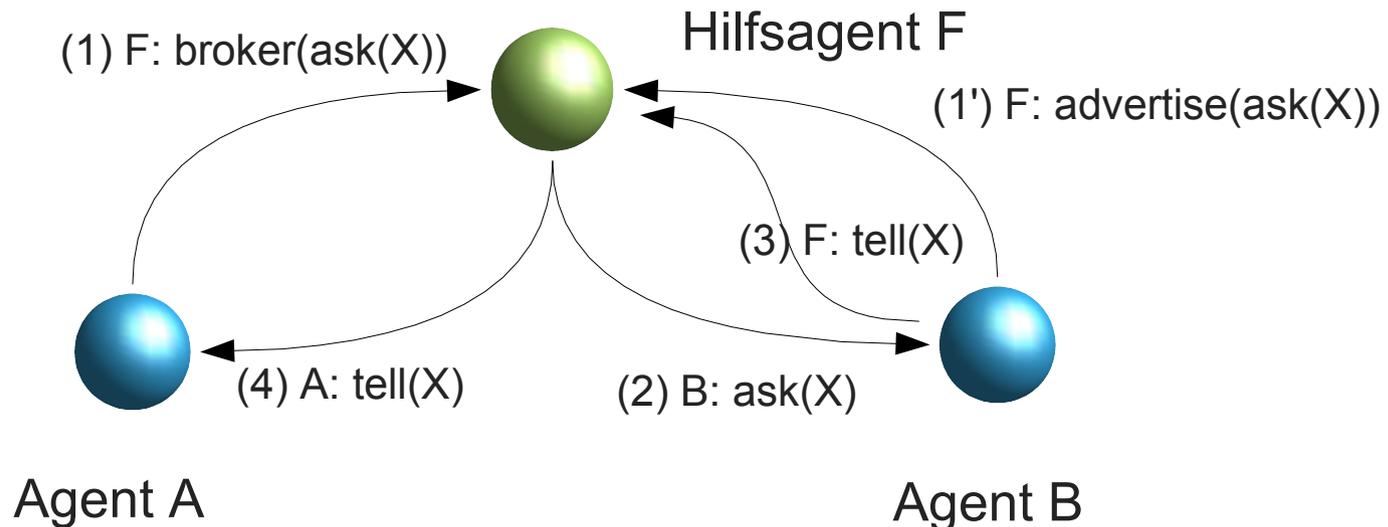
- Direkte Kommunikation
 - A kennt B und weiß, dass er B wegen X fragen kann
 - B antwortet A direkt
 - Hilfsagent unnötig

Kommunikationsformen



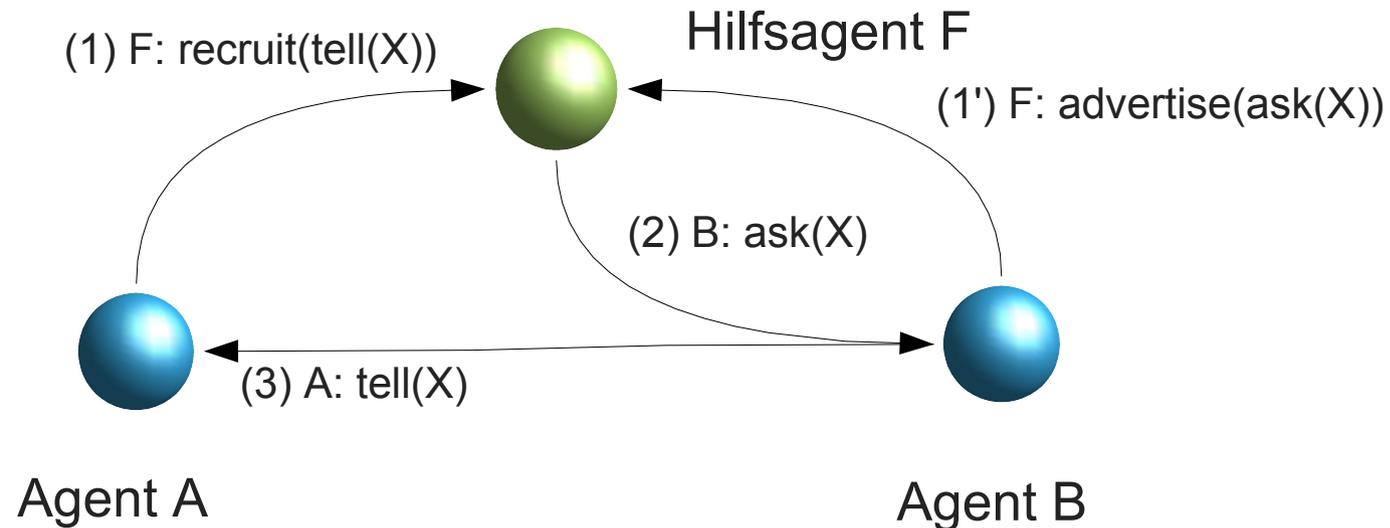
- Abonnieren von Informationen (subscribe)
 - A ist an der Wahrheit von X interessiert und bittet F um Rückmeldung, falls er Kenntnis von Änderungen bekommt
 - B informiert F, dass sich X bewahrheitet hat
 - F teilt A mit, dass X sich bewahrheitet hat.

Kommunikationsformen



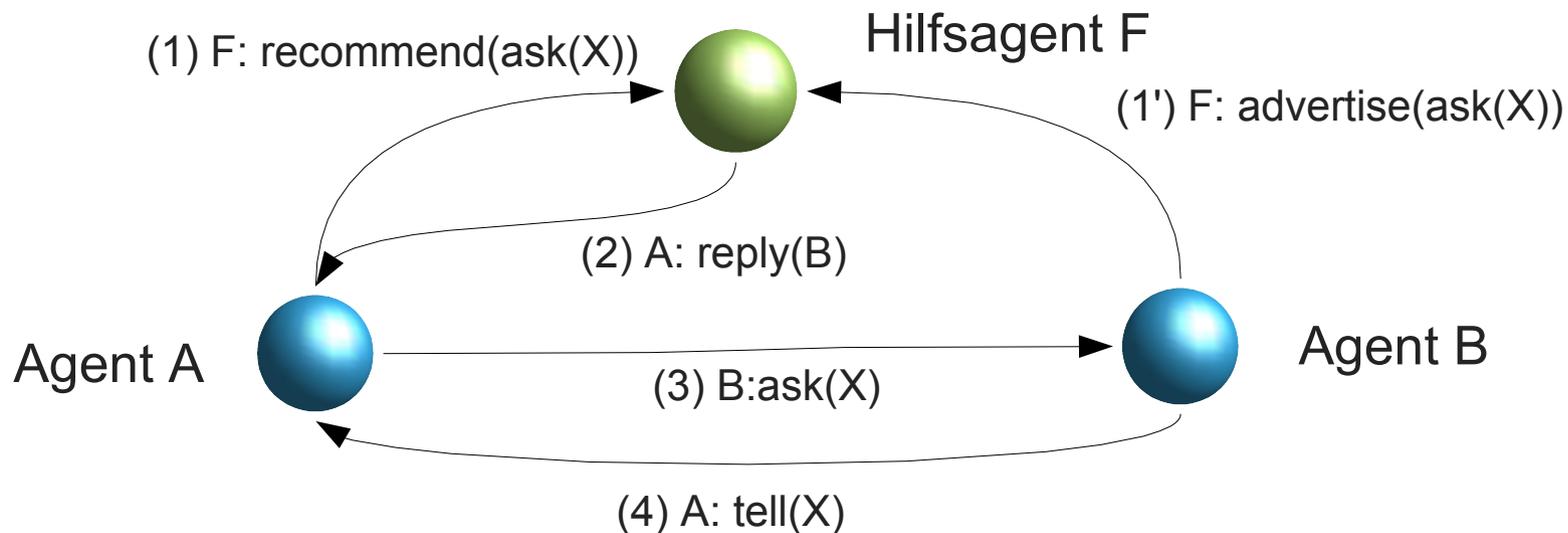
- Beschaffen von Informationen (Brokering)
 - A bittet F um Informationsbeschaffung; B bietet an, auf Fragen nach X zu antworten
 - F fragt B nach X
 - B antwortet F
 - F leitet Antwort an A weiter

Kommunikationsformen



- Vermitteln des Kommunikationspartners (Recruiting)
 - A bittet F darum, einen geeigneten Partner zu finden, der Informationen über X liefert; B bietet sich an.
 - F fragt B (mit Hinweis darauf, dass A die Antwort bekommen möchte)
 - B antwortet A

Kommunikationsformen



- Instanzvermittlung (Recommendation, Trading, Mediation)
 - A sucht einen Partner, den er nach X fragen kann
 - F liefert Adresse des Partners
 - Danach direkte Kommunikation

KQML – Genereller Aufbau

- Ausschließlich Zeichenketten-basiert (ASCII-Strings)
- KQML-Nachrichten sind Sprachäußerungen (Performative)
 - Diskurse
 - Interventionen
 - Konversationssteuerung
 - Vermittlung / Unterstützung
- Inhaltssprachen
 - KQML ist nur das Mittel zur Kommunikation; Inhalt wird anders repräsentiert
 - KQML \triangleq Sprechen/Schreiben, Inhaltssprache \triangleq Deutsch/Englisch/...
 - häufig KIF oder Prolog

Knowledge Interchange Format (KIF)

- Genesereth, Fikes (1992)
- drückt Logikprädikate erster Stufe aus
 - allerdings im LISP-ähnlichen Gewand
 - Übliche logische Konnektoren (and, or, not)
- Darüber hinaus sind Quantoren vorhanden
 - forall, exists
- Basisvokabular für Objekte und Operationen
 - Zahlen, Zeichen, Zeichenketten; kleiner als, größer als, Addition usw.
 - Listenmanipulation
- Neue Definitionen können erzeugt werden

Knowledge Interchange Format

- Nicht gedacht, um Informationen zu überbringen
- Agenten können mit KIF beispielsweise ausdrücken:
 - Eigenschaften von Objekten (Personen, Dingen) in einer gegebenen Begriffswelt (z.B. hat Michael die Eigenschaft, 1,87m groß zu sein)
 - Beziehungen zwischen Objekten (z.B. ist Michael Assistent von Professor Geihs)
 - generelle Eigenschaften (z.B. sind alle Menschen Lebewesen)

Beispiele

- Michael ist 1,87m groß:

`(= (size Michael) (scalar 1.87 meter))`

Funktion **size**
mit einem Argument

Funktion **scalar**
mit zwei Argumenten

Funktion „=
mit zwei Argumenten (vordefiniert)
Gleichheit zweier Objekte

- Definition: `(defrelation informatikstudent (?x) :=
(and (student ?x) (fach informatik)))`
- Folgerung: `(defrelation informatikstudent (?x) :=>
(kennt_sich_aus_mit ?x computer))`

Beispiele

- $(\text{if } \{ \text{Bedingung } \text{Term_wenn_ja} \}^* \text{Term_wenn_nein})$
 - Wert ist der Term nach der ersten zutreffenden Bedingung
 - $(\text{if } (> 1 2) 1 (> 2 1) 2 0)$ ergibt 2
- $(\text{defrelation natural } (?x) := (\text{and } (\text{integer } ?x) (>= ?x 0)))$
- $(\text{defrelation rational } (?x) := (\text{exists } (?y) (\text{and } (\text{integer } ?y) (\text{integer } (* ?x ?y))))))$

KQML: Allgemeines Format

- Performative

- Befehl, dann Folge von Parametername-Wert-Paare
- Parametername mit Doppelpunkt eingeleitet
- Eines davon kann der Inhalt sein (:content)

(Befehl :Param1 Wert1 :Param1 Wert1 ... :ParamN WertN)

(Befehl

:Parameter1 Arg1
:Parameter2 Arg2

...

:ParameterN ArgN)

(tell :sender A :receiver B :content ...)

KQML-Befehle (Stand 1993)

| | |
|------------|---|
| achieve | Sorge dafür, dass das Angegebene wahr wird |
| advertise | Ich kann die angegebene Performative verarbeiten |
| ask-about | Gib mir alle relevanten Informationen aus deiner Wissensbasis |
| ask-all | Gib mir alle Instanzen des Inhalts von :content aus deiner Wissensbasis |
| ask-if | Gibt es eine passende Instanz zu :content in deiner Wissensbasis? |
| ask-one | Gib mir etwas Passendes zu :content aus deiner Wissensbasis |
| break | Brich die Performative-Weiterleitung ab (→pipe) |
| broadcast | Sende die Performative an alle |
| broker-all | Gib mir alle Antworten auf meine Hilfeanfrage bezüglich der Performative |
| broker-one | Besorge dir jemanden, der dir meine Anfrage beantwortet |
| deny | Ich weiß, dass der Inhalt von :content (jetzt) falsch ist |
| delete-all | Vergiss alle Sätze in deiner Wissensbasis, die :content betreffen |
| delete-one | Vergiss eine Information deiner Wissensbasis betreffs :content |
| discard | Ich möchte keine weiteren Antworten (→subscribe, stream-all, recommend-all) |
| eos | Ich habe keine weiteren Antworten (→stream-all) |
| error | Deine Anfrage ist fehlerhaft |
| evaluate | Verarbeite (vereinfache) den Inhalt von :content |
| forward | Leite die Nachricht an den angegebenen Empfänger weiter |
| generator | gleich wie standby für stream-all |
| insert | Glaube mir, dass der Inhalt von :content wahr ist |

KQML-Befehle

| | |
|-------------------|--|
| monitor | Ich möchte Aktualisierungen zu deiner Antwort auf stream-all |
| next | Ich möchte die nächste Information im Strom (→standby) |
| pipe | Schicke die folgenden Performative direkt an den angegebenen Empfänger |
| ready | Ich kann dir (jetzt) auf deine Anfrage antworten (→standby) |
| recommend-all | Sage mir, wer alles auf eine Anfrage bezüglich :content antworten kann |
| recommend-one | Gib mir jemanden, der auf eine Anfrage zu :content antworten kann |
| recruit-all | Lasse alle passenden Agenten mir bezüglich :content antworten |
| recruit-one | Lasse einen passenden Agenten mir bezüglich :content antworten |
| register | Ich bin verfügbar unter der Adresse in :content (geht an Hilfsagent) |
| reply | Hiermit antworte ich dir |
| rest | Schicke mir den Rest deiner Antworten (→standby) |
| sorry | Ich kann dir nichts Genaueres dazu sagen |
| standby | Bereite dich auf eine Anfrage von mir vor |
| stream-about | wie ask-about, aber mit mehrteiliger Antwort |
| stream-all | wie ask-all, aber mit mehrteiliger Antwort |
| subscribe | Ich möchte über weitere Antworten auf :content informiert bleiben |
| tell | Ich weiß, dass der Inhalt von :content wahr ist (vgl. aber insert!) |
| transport-address | Ich nehme diese Adresse als meine neue symbolische Adresse |
| unregister | deny von register |
| untell | Ich weiß nicht, ob der Inhalt von :content (noch) wahr ist. |

Ausführungen zu den Performativen

- ask-all / ask-if
 - „Instanzen“ bezieht sich auf Formeln, in denen die freien Variablen mit Werten instanziiert werden
- deny
 - Es könnte der verwendeten Inhaltssprache an einer Negation mangeln, sodass man kein „tell (NOT X)“ zur Verfügung hat. Daher wurde „deny“ eingeführt.
- insert/delete
 - „delete“ wird zunächst als Gegenpart zu „insert“ definiert. Ein späterer Vorschlag führt „uninsert“ und „undelete“ ein, da dies eine einfachere Vermittlung bei „advertise“ erlaubt („insert“ impliziert „uninsert“, „delete“ impliziert „undelete“)

Ausführungen zu den Performativen

- error
 - Nachricht ist nicht interpretierbar oder verletzt ein Protokoll
- standby
 - Antworten auf die folgende Anfrage werden so lange zurückgehalten, bis der Anfrager sie fordert.
- ready/next/rest
 - zu standby: Antworten liegen vor
 - Antworten werden mit next (vom ursprünglichen Anfrager) angefragt
 - rest liefert den Rest

Neue Änderungen (1997)

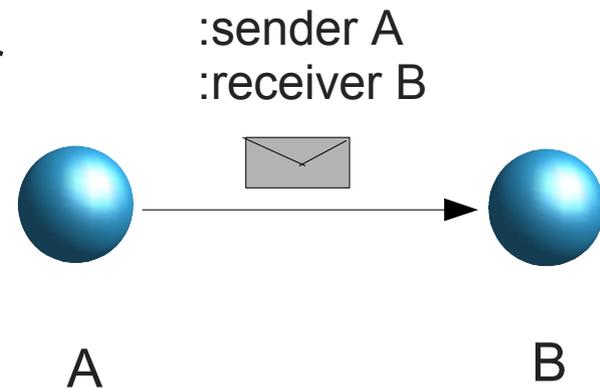
- Vorschlag für eine neue Spezifikation (Labrou/Finin, 1997)
- Weggefallen
 - ask-about, stream-about
 - break, pipe
 - evaluate, reply
 - generator, monitor
- Neu hinzugekommen
 - uninsert: Rückgängigmachen eines insert
 - unachieve: Rückgängigmachen eines achieve
 - unadvertise: Rückgängigmachen eines advertise

KQML-Format

Kennzeichnung von Sender und Empfänger

:sender *Wort*
:receiver *Wort*

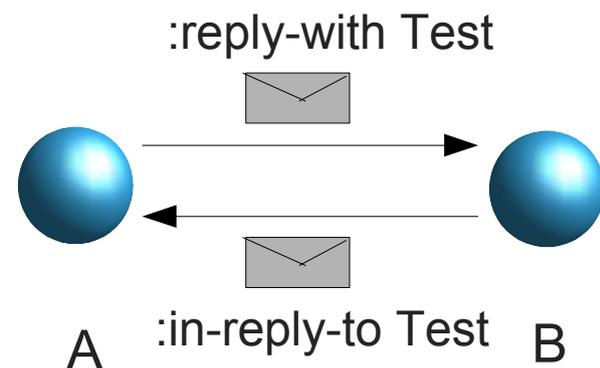
Wort: Kennzeichnung des Agenten



Nachrichten-Referenzen

:reply-with *Wort*
:in-reply-to *Wort*

Wort: Kennzeichnung einer Nachricht



KQML-Format

Interpretationshinweise

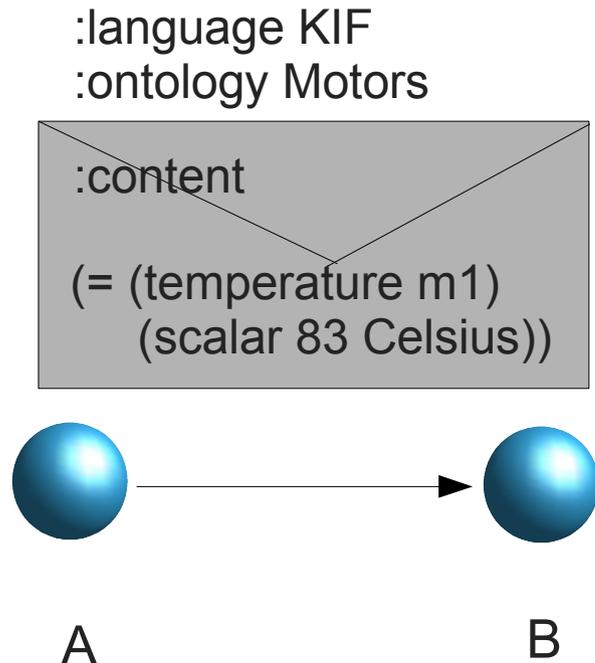
:language *Wort*
:ontology *Wort*

Die Sprache kennzeichnet die Inhaltssprache; die Ontologie parametrisiert die Auswertung (z.B. Selektion einer best. Wissensdatenbank)

Inhalt

:content *Ausdruck*

Der Ausdruck ist in der Inhaltssprache gehalten, die in **:language** angegeben ist.



KQML-Format

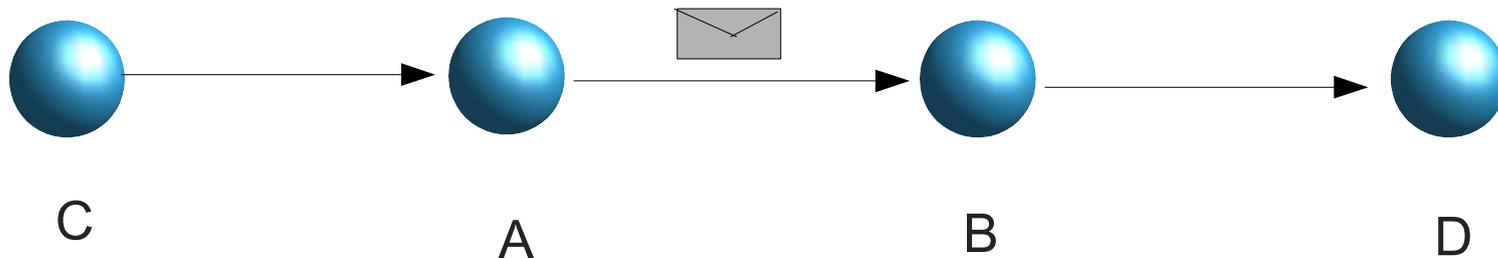
- Weiterleitung

(**forward** :sender A :receiver B :from C :to D :content ...)

:from *Wort*

:to *Wort*

Ähnlich sender/receiver, bezeichnet jedoch die originalen Absender bzw. endgültigen Empfänger



Wissensabfrage

- Diskurs-Anweisungen

(ask-if

:sender Wort

:receiver Wort

[:in-reply-to Wort]

:reply-with Wort

:language Wort

:ontology Wort

:content Ausdruck)

*Frage nach, ob der Ausdruck
für den Empfänger dieser
Nachricht als wahr gilt.*

Antwort:

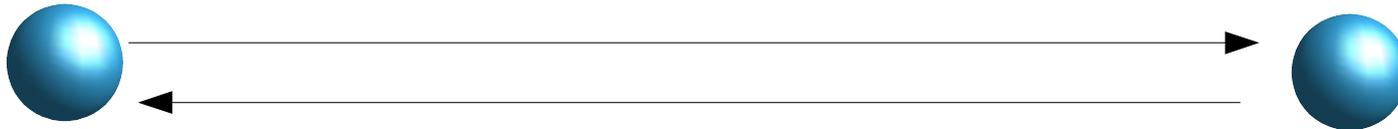
tell

deny

untell

Wissensabfrage

(**ask-if** :sender A :receiver B :reply-with m1
:language Prolog :ontology Standard
:content "arrived('AgentX')")



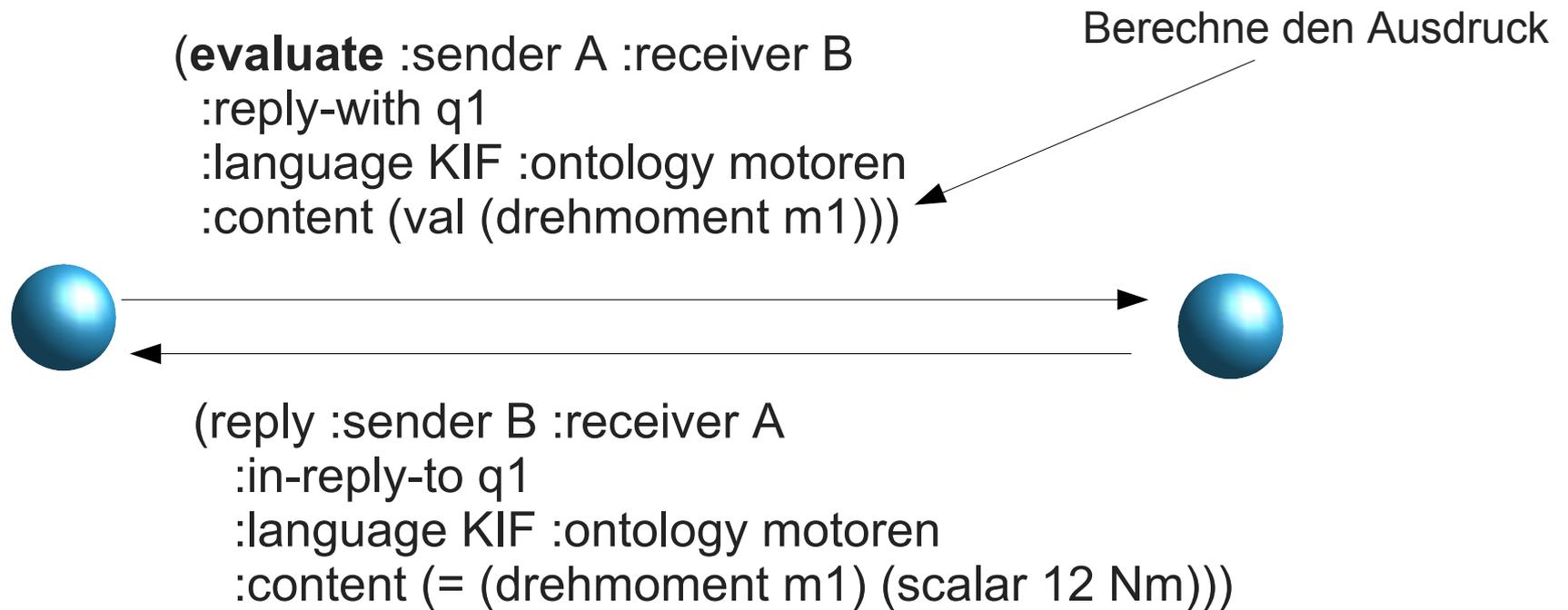
(**tell** :sender B :receiver A :reply-with m2
:in-reply-to m1 :language Prolog
:ontology Standard
:content "arrived('AgentX')")

Alternativen:

deny – Aussage ist falsch

untell – Aussage ist nicht beweisbar („keine Ahnung“)
(vgl. offene Welt – geschlossene Welt)

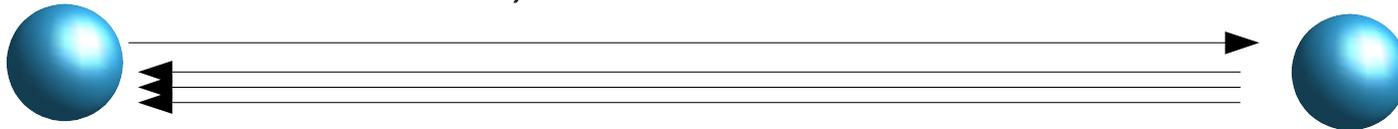
Anweisung



Beispiele aus: M. Wooldridge: Introduction to MultiAgent Systems

Einfache Frage – komplexe Antwort

```
(stream-about  
  :sender A :receiver B  
  :reply-with q1  
  :language KIF :ontology motoren  
  :content m1)
```



```
(tell :sender B :receiver A :in-reply-to q1 (...)  
  :content (= (drehmoment m1) (scalar 12 Nm)))
```

```
(tell :sender B :receiver A :in-reply-to q1 (...)  
  :content (= (status m1) (normal)))
```

```
(eos :sender B :receiver A :in-reply-to q1)
```

Dienstangebot

Agent A

```
(advertise
  :sender A
  :language KQML :ontology K10
  :content
    (subscribe
      :language KQML :ontology K10
      :content
        (stream-about
          :language KIF :ontology motoren
          :content m1)))

(tell :sender A :receiver B :in-reply-to s1
  :content (= (drehmoment m1) (scalar 12 Nm)))
(tell :sender A :receiver B :in-reply-to s1
  :content (= (status m1) normal))
(untell :sender A :receiver B :in-reply-to s1
  :content (= (drehmoment m1) (scalar 12 Nm)))
(tell :sender A :receiver B :in-reply-to s1
  :content (= (drehmoment m1) (scalar 15 Nm)))
(eos :sender A :receiver B :in-reply-to s1)
```

Agent B

```
(subscribe
  :sender B :receiver A
  :language KQML :ontology K10
  :reply-with s1
  :content
    (stream-about
      :language KIF :ontology motoren
      :content m1)))
```

Wo ist KQML geblieben?

- KQML sehr flexibel – aber warum wird es kaum benutzt?
- Wie funktioniert dieser Hilfsagent F?
 - Wie lässt sich die Suche nach X implementieren?
- Wie werden Sprachen gekennzeichnet?
- Was sind Ontologien? Wie kann man sie implementieren?

Kritik an KQML

- Befehlssatz war lange Zeit nicht festgelegt
 - inkompatible Implementierungen!
- Keine Angaben über den Transport der Nachrichten
 - wiederum Inkompatibilitäten!
- Semantik von KQML-Befehlen nur informell
 - Man kann nicht belegen, dass die Befehle korrekt implementiert sind und ausgeführt werden
- Es fehlen Kommissiva
 - wichtig für Multiagentenanwendungen: Wie soll eine Koordination funktionieren, wenn sich Agenten nicht zu Aktionen verpflichten?
- Befehlssatz ziemlich groß, mutet willkürlich an

KQML

- Weiterentwicklung: KQML ging in FIPA-ACL auf (1999)
 - Standardisierung