



1/22

## David N. Chin: Intelligent Interfaces as Agents (1991)

Sebastian Marius Kirsch

skirsch@moebius.inka.de



Back

Close



# Überblick

- Agenten vs. Werkzeuge
- Beispielsessions
- Aufbau von UC
- Kodiak-Netzwerke
- UCEgo und *if-detected demons*
- Themen, Ziele, Unterziele, Metaziele
- Benutzermodellierung
- Missverständnisse
- Fazit





# Agenten und Werkzeuge

## Traditioneller Ansatz

Intelligente Benutzerschnittstellen sind Werkzeuge, die die direkte Manipulation der Maschine vereinfachen.

## Agenten

Die Benutzerschnittstelle wird als separate Entität mit eigenen Plänen und Zielen verstanden.

**autonome Agenten** ergreifen selbst die Initiative und können ohne Anleitung des Benutzers agieren.

**rationale Agenten** agieren auf der Basis von Zielen und Plänen.

**intelligente Agenten**, d.h. *autonome, rationale* Agenten benutzen einen Planner, um Pläne zu entwickeln, die die vorher bestimmten Ziele erfüllen.





# Vor- und Nachteile von Agenten

## Vorteile

- Dialog als natürliche Schnittstelle
- Schnittstelle kann selbst Initiative ergreifen
- Consultant-System weiss mehr als der Benutzer und muss deshalb eigene Entscheidungen treffen

## Nachteile

- Dialogmodelle werden noch nicht gut verstanden
- Schnittstelle kann unerwartete Aktionen durchführen
- Falsche Erwartungen der Benutzer an die Schnittstelle





# Beispielsessions

Do you know how to delete a file?

Use `rm`.

For example, to delete the file named `foo`, type `'rm foo'`.

How can I get rid of a file named `bob`?

Type `'rm bob'`.

Is `rn` used to rename files?

No, `rn` is used to read news.

To rename a file to another file, use `mv`.

For example, to relame the file named `foo` to the file named `foo1`, type `'mv foo foo1'`.



How can I crash the system?

I'm sorry, I cannot tell you.

What does 'ls -v' do?

Ls does not have a -v option.

Who is on the system?

I'm sorry, I don't know. Use users to find out who is on the system.



6/22



Back

Close



# Aufbau von UC

**ALANA** Sprachanalyse, generiert semantische Repräsentation der Eingabe.

**PAGAN** Analyse der unmittelbaren und weitergehenden Ziele des Benutzers

**UCego** autonomer, rationaler Agent

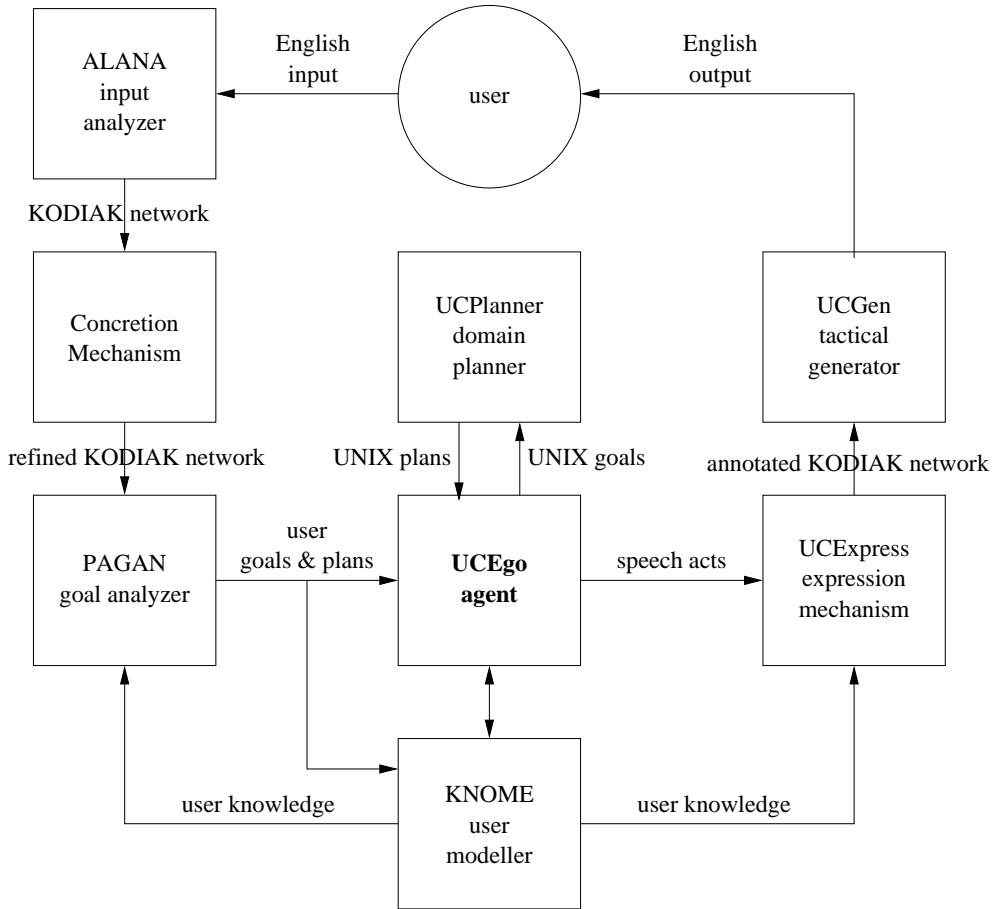
**KNOME** *knowledge base*, speichert Wissen und Überzeugungen des Benutzers.

**UCPlanner** Speichert Domänenwissen über Unix.

**UCExpress** Verändert die von UCego generierten Sprachpläne entsprechend dem Vorwissen des Benutzers.

**UCGen** Sprachgenerator.





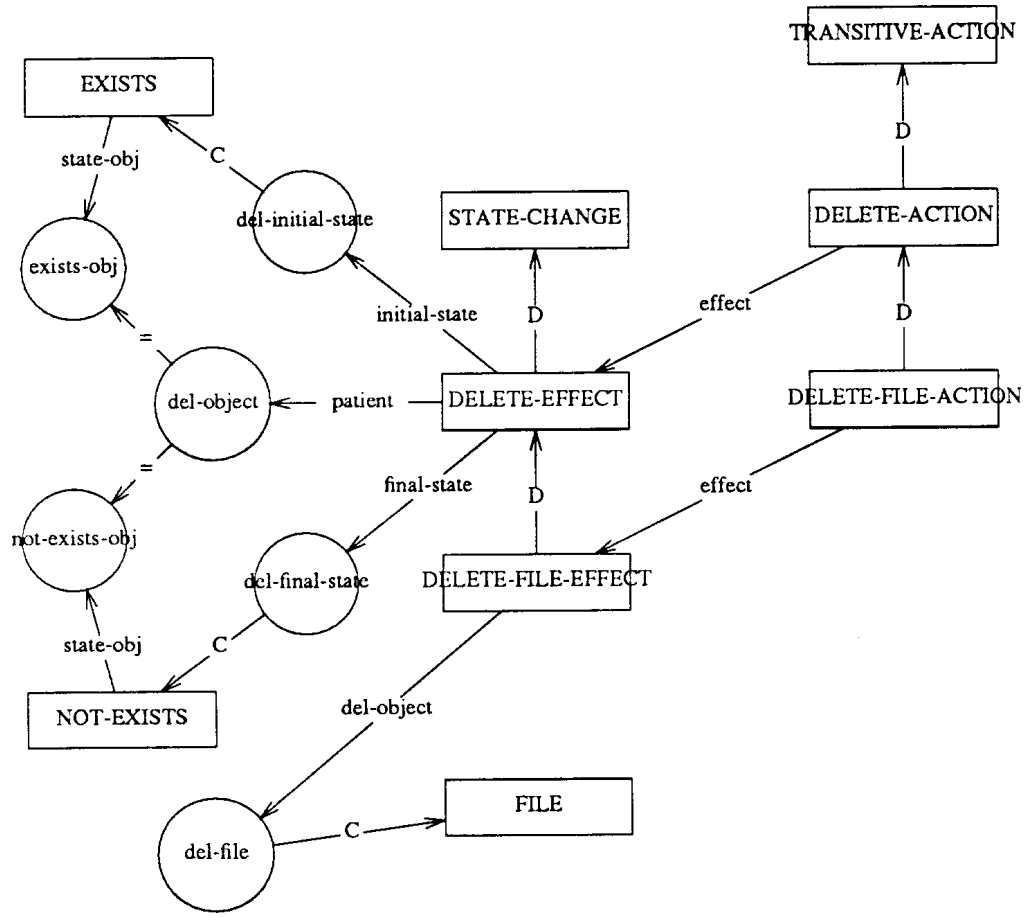




# KODIAK-Netzwerke

- Keystone to Overall Design for Interpretation and Application of Knowledge
- Wissensrepräsentationssprache
- Beziehungsorientiert (*relation-oriented*)
- strukturierte Vererbungstechnik (ähnlich KL-ONE)
- Typen von Relationen:
  - = *Equate*, Gleichheit
  - D** *Dominate*, Subtypen
  - C** *Constrain*, Einschränkungen



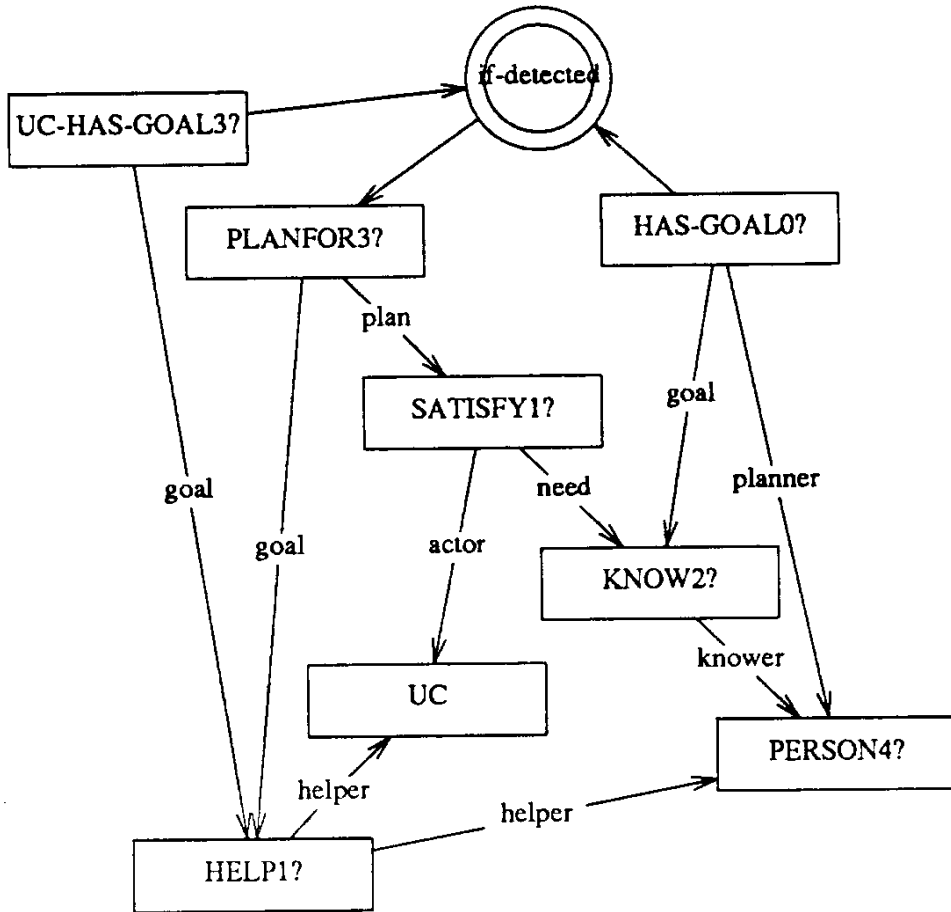


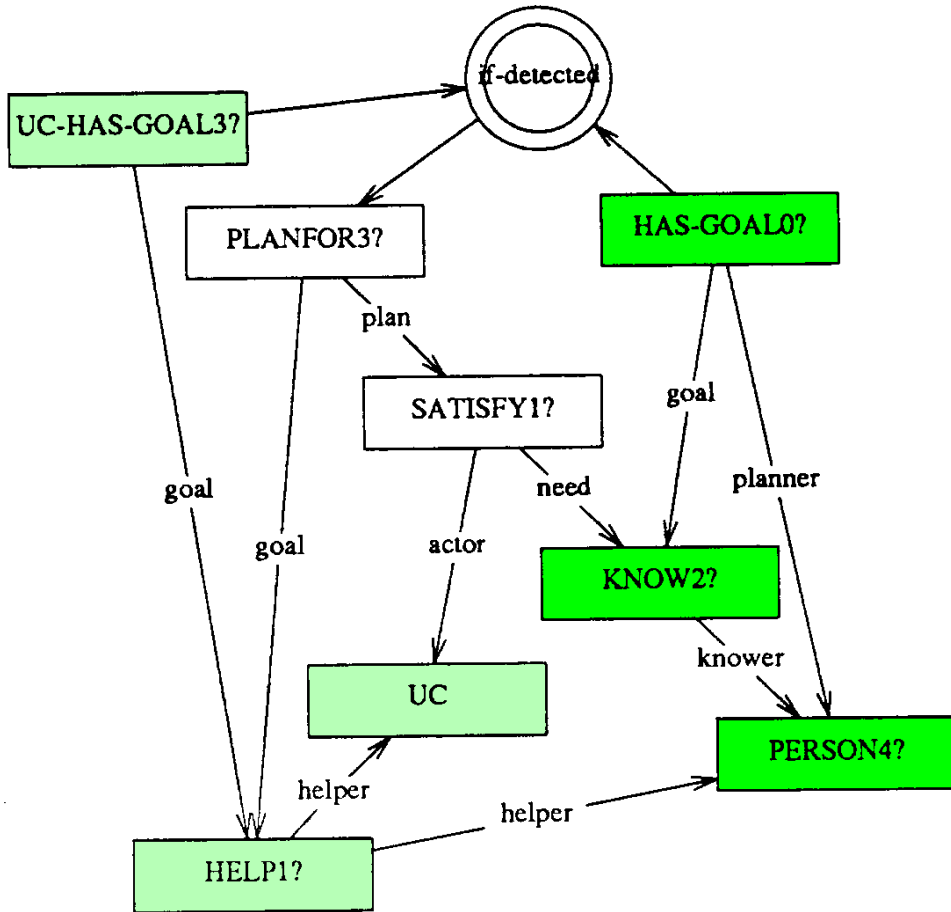


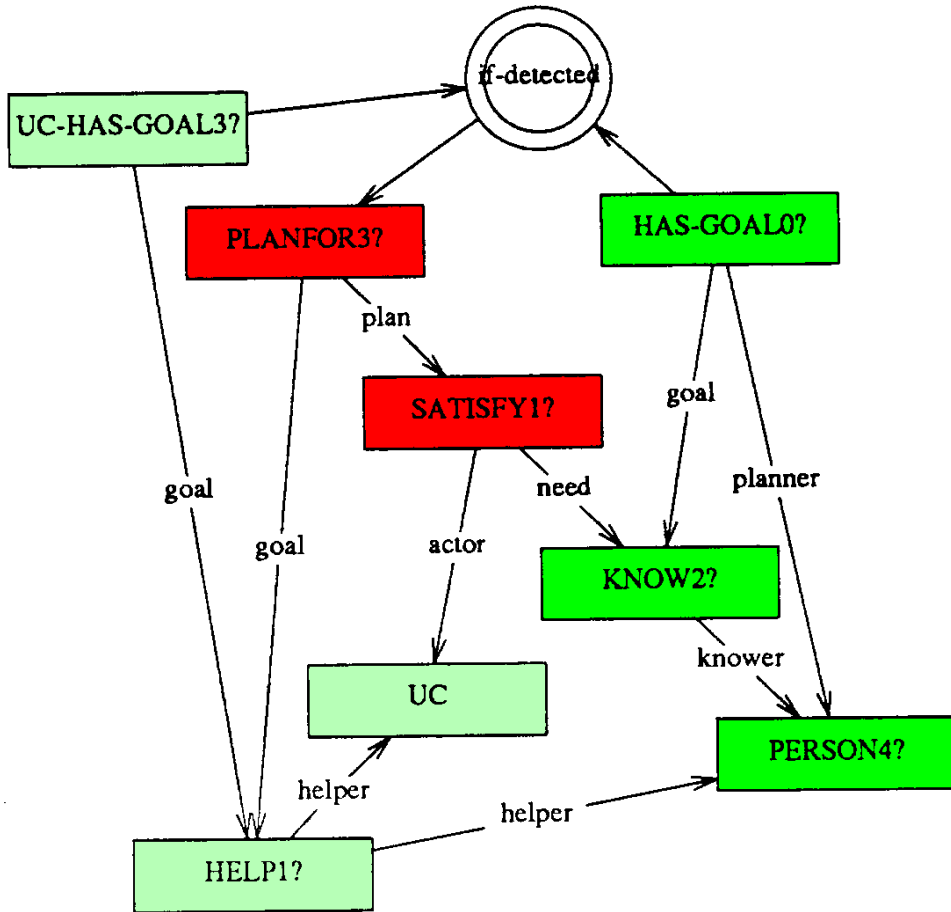
# UCego, if-detected demons

- *if-detected demons* „lauschen“ auf dem KODIAK-Netzwerk
- treten automatisch in Aktion, wenn ein Teil des Netzwerks passt
- fügen dem Netzwerk neue Objekte und Relationen hinzu.
- linke Seite (*detection net*, grün) wird erkannt → rechte Seite (*addition net*, rot) wird hinzugefügt.
- repräsentieren Situationsklassen in UC











# Ziele

- explizit repräsentiert durch *has-goal*-Relation.
- zwei Aspekte: *goal* und *planner*
- Ziele des Benutzers werden nicht blind übernommen
- UCEgo versucht, Ziele des Benutzers mit seinen eigenen Zielen zu vereinbaren und daraus Pläne abzuleiten.



Back

Close



# Themen

Allgemeine Verhaltensmassregeln, die Ziele verursachen und helfen, Konflikte zu lösen; sie sind implizit priorisiert:

1. ethisch handeln
2. selbst am Leben bleiben (d. h. das UC-Programm und das Unix-System bewahren)
3. sich wie ein Consultant verhalten (dem Benutzer helfen und höflich sein)







# Erzeugung von Zielen

Fünf Phänomene, die Ziele erzeugen:

1. Themen → Ziele
2. Pläne → Unterziele
3. Lücken im Wissen des Benutzers → Ziele
4. Interaktion zwischen Zielen → Metaziele
5. Falsche Vorstellungen des Benutzers → Ziele



Back

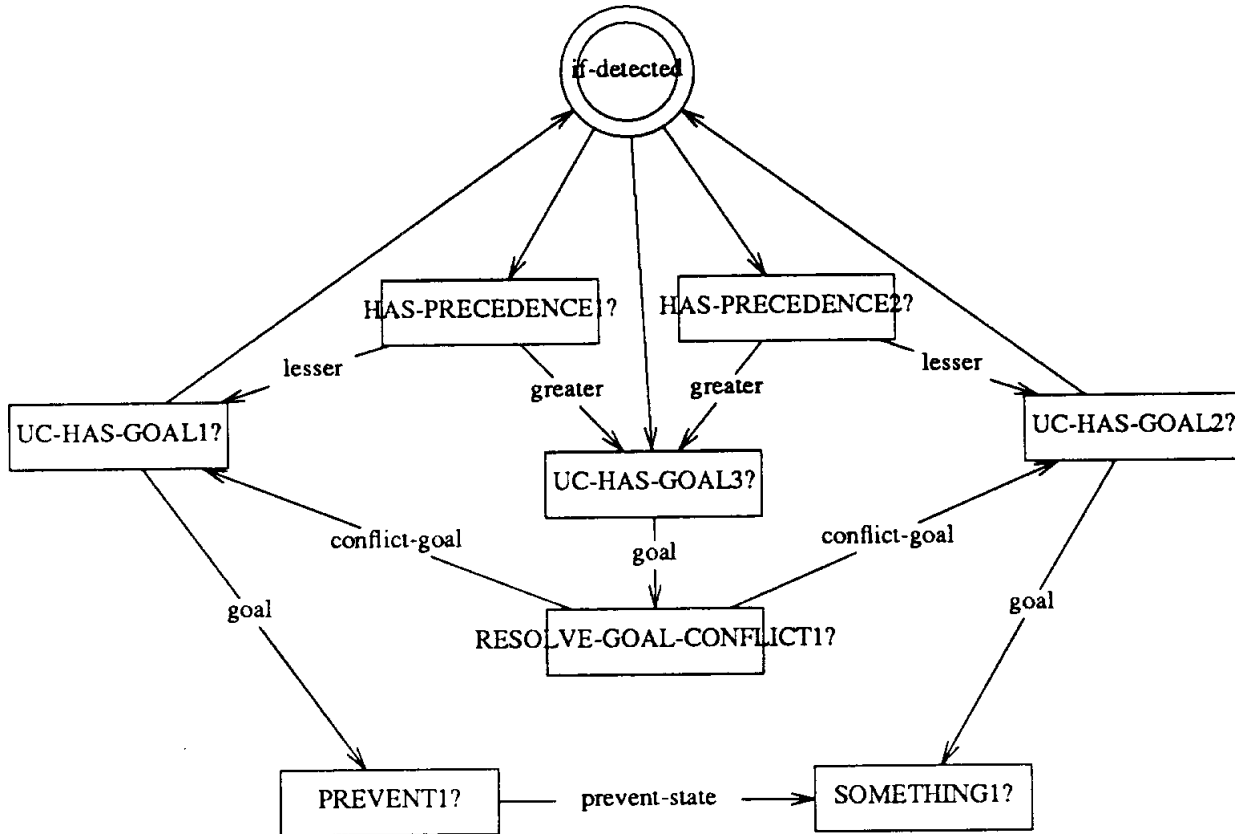
Close



# Metaziele

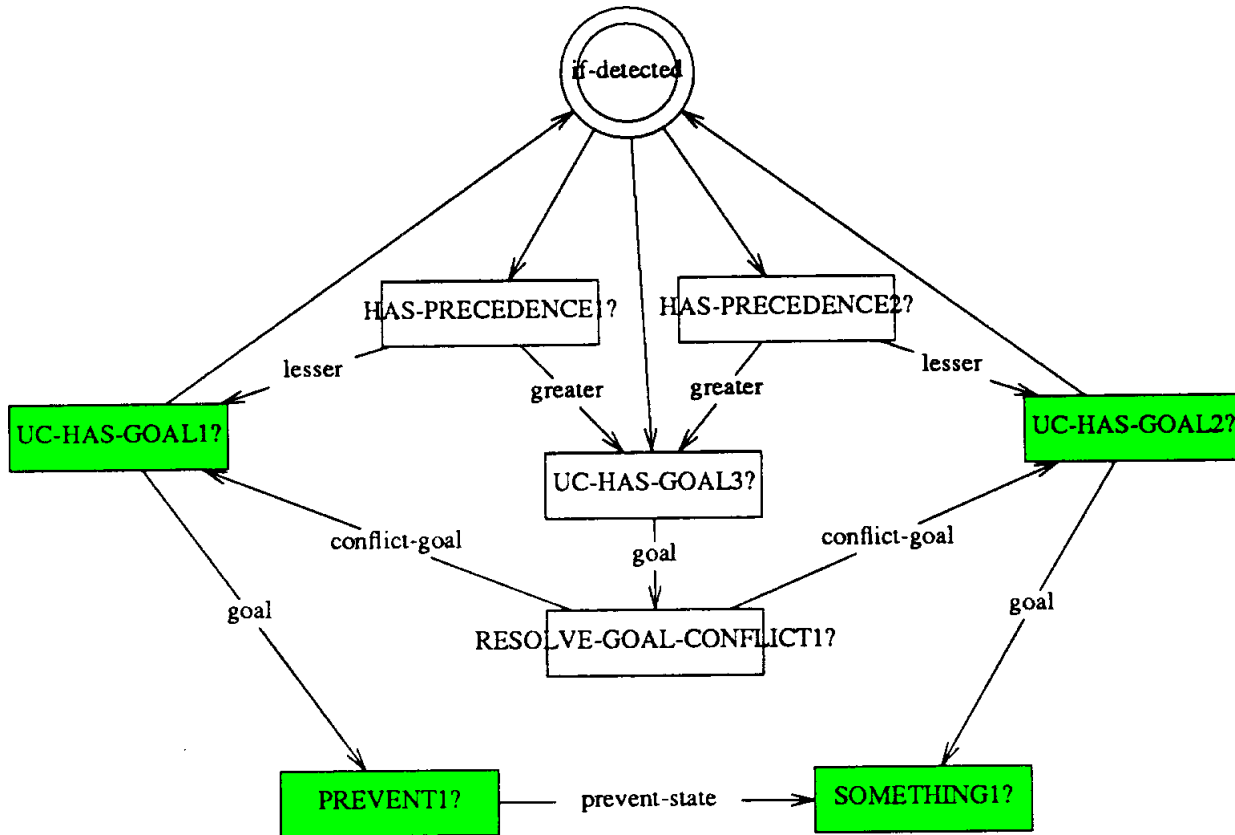
- entstehen durch Interaktion von Zielen
- werden wie normale Ziele repräsentiert
- Beispiele:
  - UC weiss etwas nicht → Metaziel: Information herausfinden
  - Konflikt zwischen zwei Zielen → Metaziel: Konflikt lösen
  - ein Ziel schliesst ein anderes ein → Metaziel: Ziele verschmelzen





Beispiel für die Erzeugung des Metaziels „Zielkonflikt lösen“



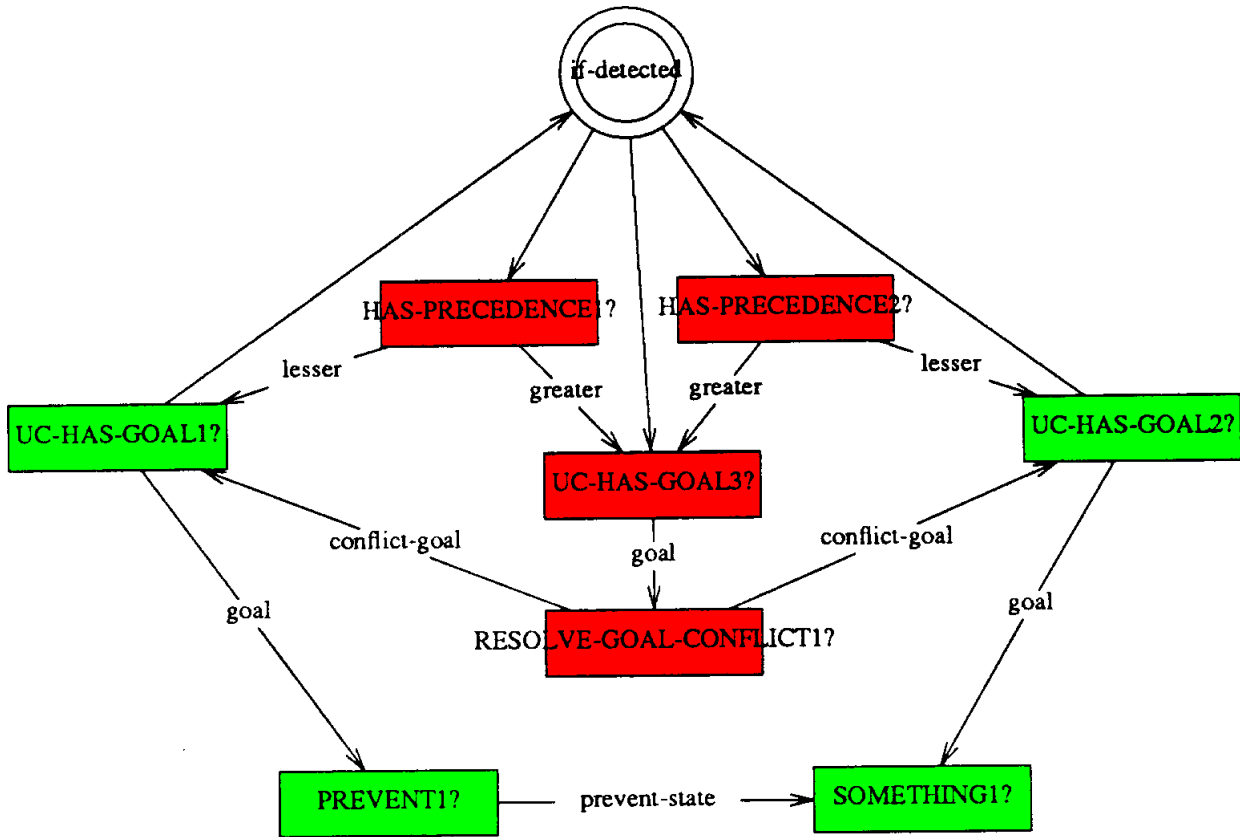


Beispiel für die Erzeugung des Metaziels „Zielkonflikt lösen“



Back

Close



Beispiel für die Erzeugung des Metaziels „Zielkonflikt lösen“





# Benutzermodellierung

- Benutzermodellierung über Stereotypen
- nur Vorwissen des Benutzers wird modelliert, nicht seine Vorlieben.
- Einteilung der Benutzer in vier Klassen
  1. Neulinge
  2. Anfänger
  3. Fortgeschrittene
  4. Experten
- Benutzermodell beeinflusst Erkennung von Plänen und Generierung der Ausgabe.





# Missverständnisse

- erkennt nur relationale Missverständnisse
- vergleicht, ob alle Relationen in der Anfrage ein Entsprechung in der Wissensbasis haben
- *open world*-Hypothese
- Metawissen: über einfache Befehle wird komplettes Wissen angenommen, bei komplexeren Lücken in der Wissensbasis.
- Missverständnisse werden nicht korrigiert, wenn dies andere Ziele (→ ethisches Handeln) beeinträchtigen würde.





20/22

## Weitere Merkmale

- *overanswering*
- Vorschläge, Warnungen, Ausführungen
- UCTeacher



Back

Close





# Fazit

- Beispiel für eine intelligente Benutzerschnittstelle auf Basis eines Agenten
- Allgemeiner Mechanismus zur Modellierung von Zielen und Plänen
- einfacher Planer für Kommunikationspläne
- komplexere Wissensbasis für Unix-Probleme
- Probleme:
  - unidirektionaler Informationsfluss
  - weitgehend künstliche Unterscheidung zwischen semantischer Analyse, Konkretisierung und Zielanalyse
  - fehlende Ausdrucksmöglichkeiten für Quantifizierung in KODIAK





# Literatur

- [1] David N. Chin: *Intelligent Interfaces as Agents*, 1991. In: Maybury, Mark T. and Wahlster, Wolfgang: *Readings in Intelligent User Interfaces*. Morgan Kaufmann, San Francisco (1998)
- [2] Wilensky, Robert, et. al: *The Berkeley UNIX Consultant Project*, 1988. <http://sunsite.berkeley.edu/TR/UCB:CSD-89-520>
- [3] Thomas Schillo: *Ausarbeitung zum Proseminar intelligente Multimediapräsentationssysteme*, Wintersemester 1998/1999. <http://w5.cs.uni-sb.de/ws98/ausarbeitung/tschillo/chin.html>

