Dialogue Control in the Alparon System

Niels Andeweg

TUDelft

Knowledge Based Systems Group

Zuidplantsoen 4
2628 BZ Delft, The Netherlands

Alparon

26-8-99

- Information Technology and Systems
- Knowledge Based Systems
  - Artificial Intelligence
- Alparon
  - multi-media
  - automatic speech processing
  - dialogue management

Assignment

- design dialogue control for the Alparon system
- implement prototypes of the components for dialogue control

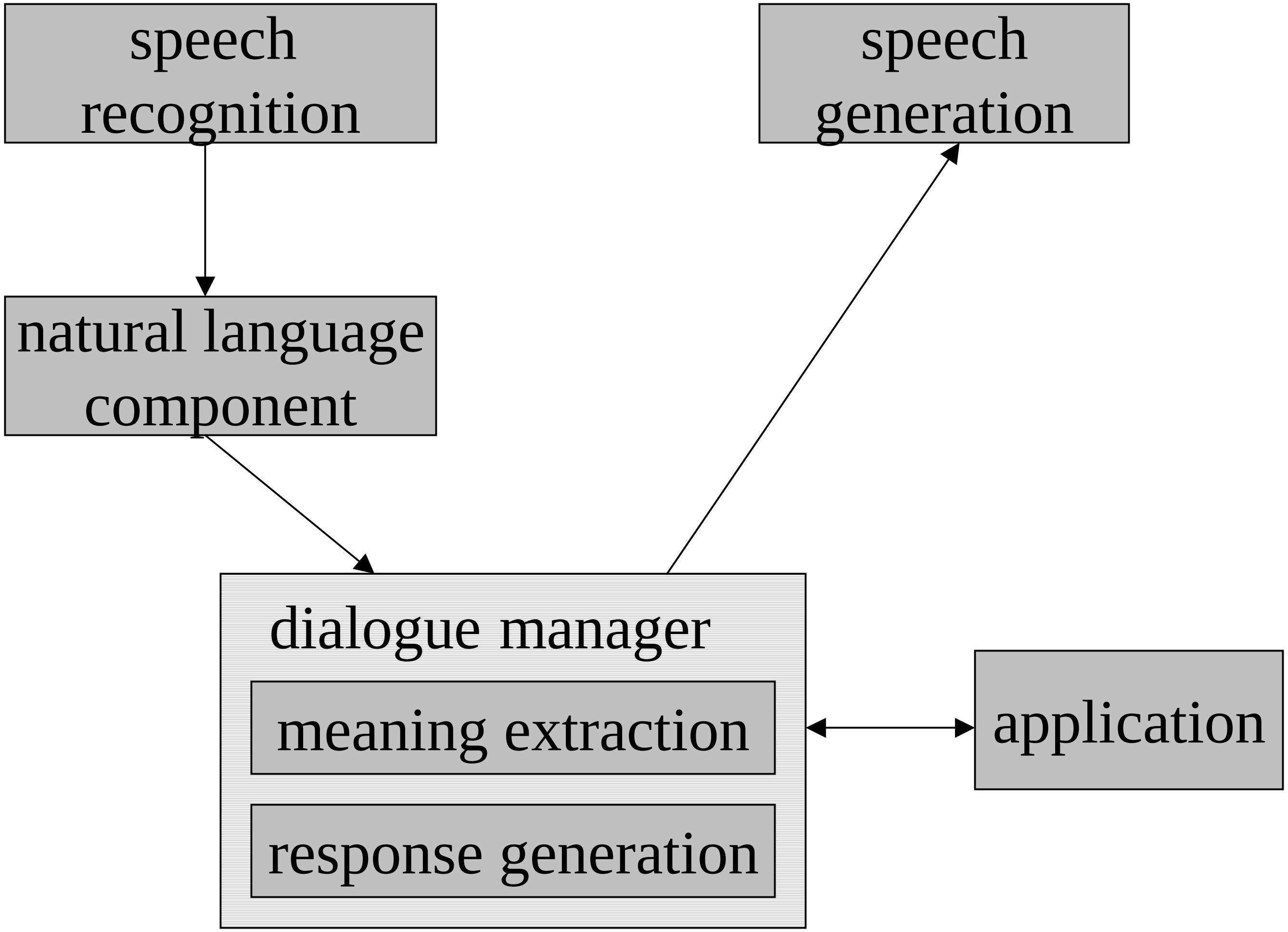
Contents

- Automatic speech processing (ASP)
- Alparon system
- Dialogue control
- Train timetable information system
- Conclusions

Contents

- Automatic speech processing (ASP)
- Alparon system
- Dialogue control
- Train timetable information system
- Conclusions

ASP system


speech recognition
natural language component
dialogue manager
meaning extraction
response generation
speech generation
application

Dialogue management

Why dialogue management?

- spread out user requirements
- discourse phenomena

Tasks:

- provide context
- problem solving
- response generation

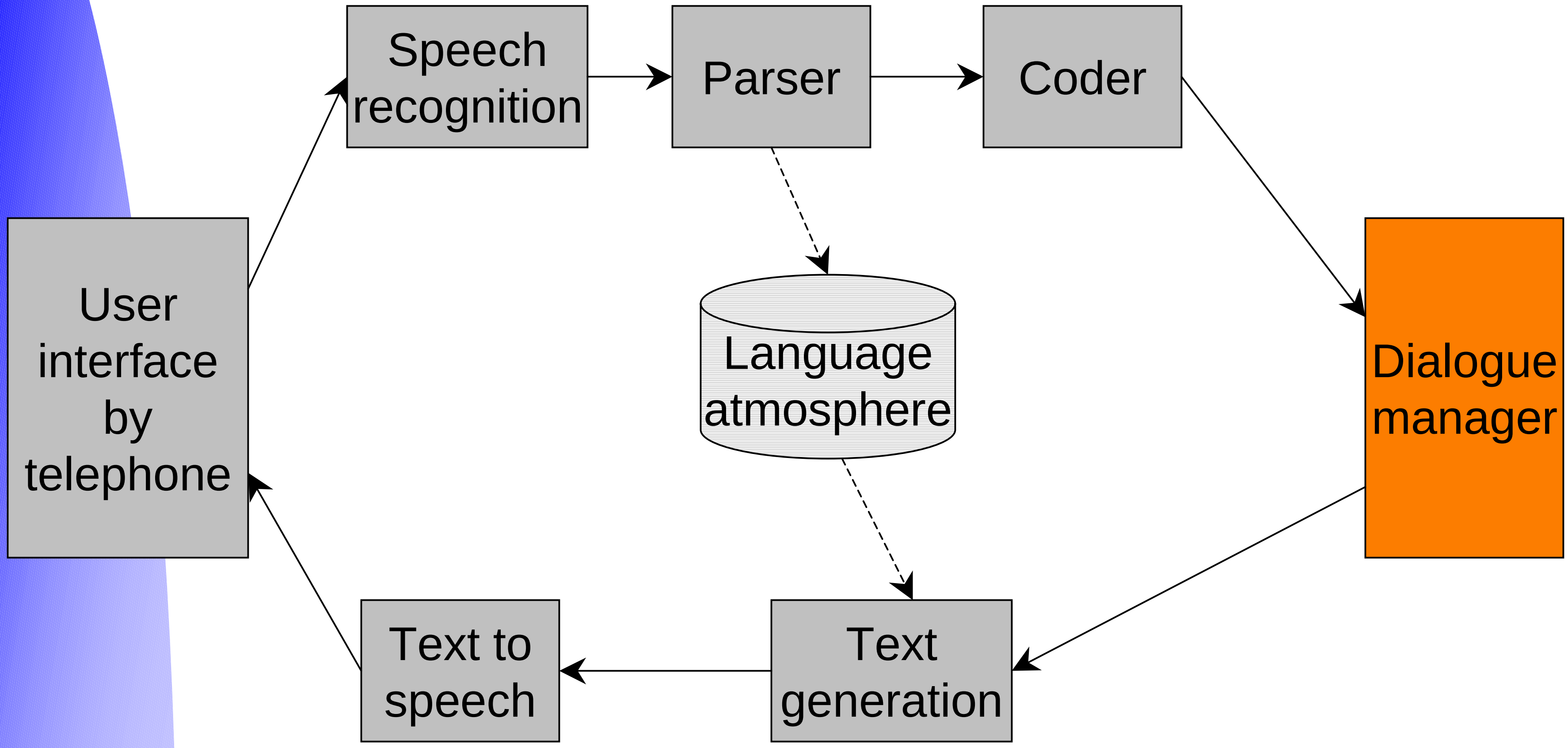
Contents

- Automatic speech processing (ASP)
- Alparon system
- Dialogue control
- Train timetable information system
- Conclusions

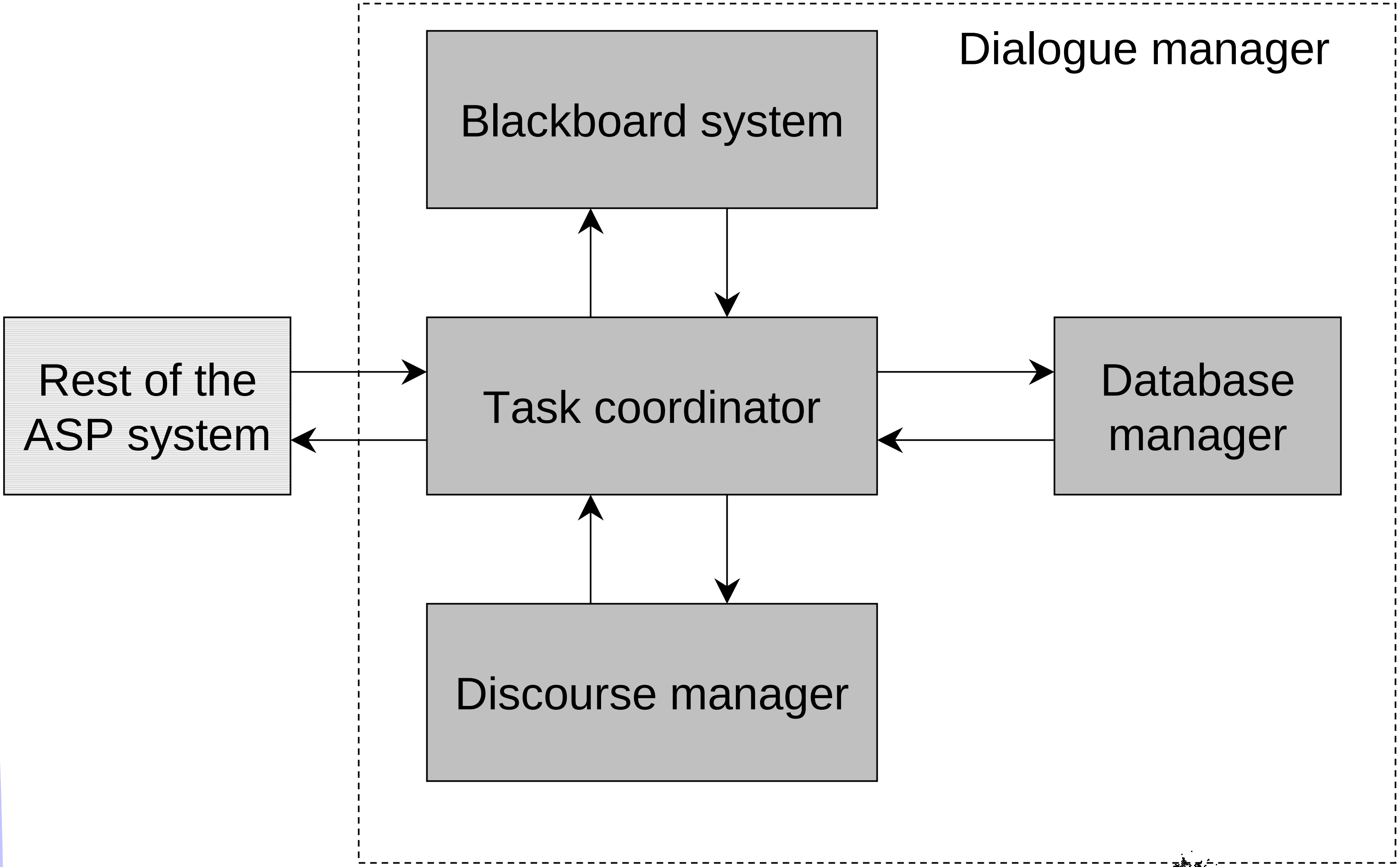
Alparon system

- information retrieval and transaction services
- natural language
- speech interface
- mixed-initiative
- focus on dialogue management
- strategies in human-human dialogues

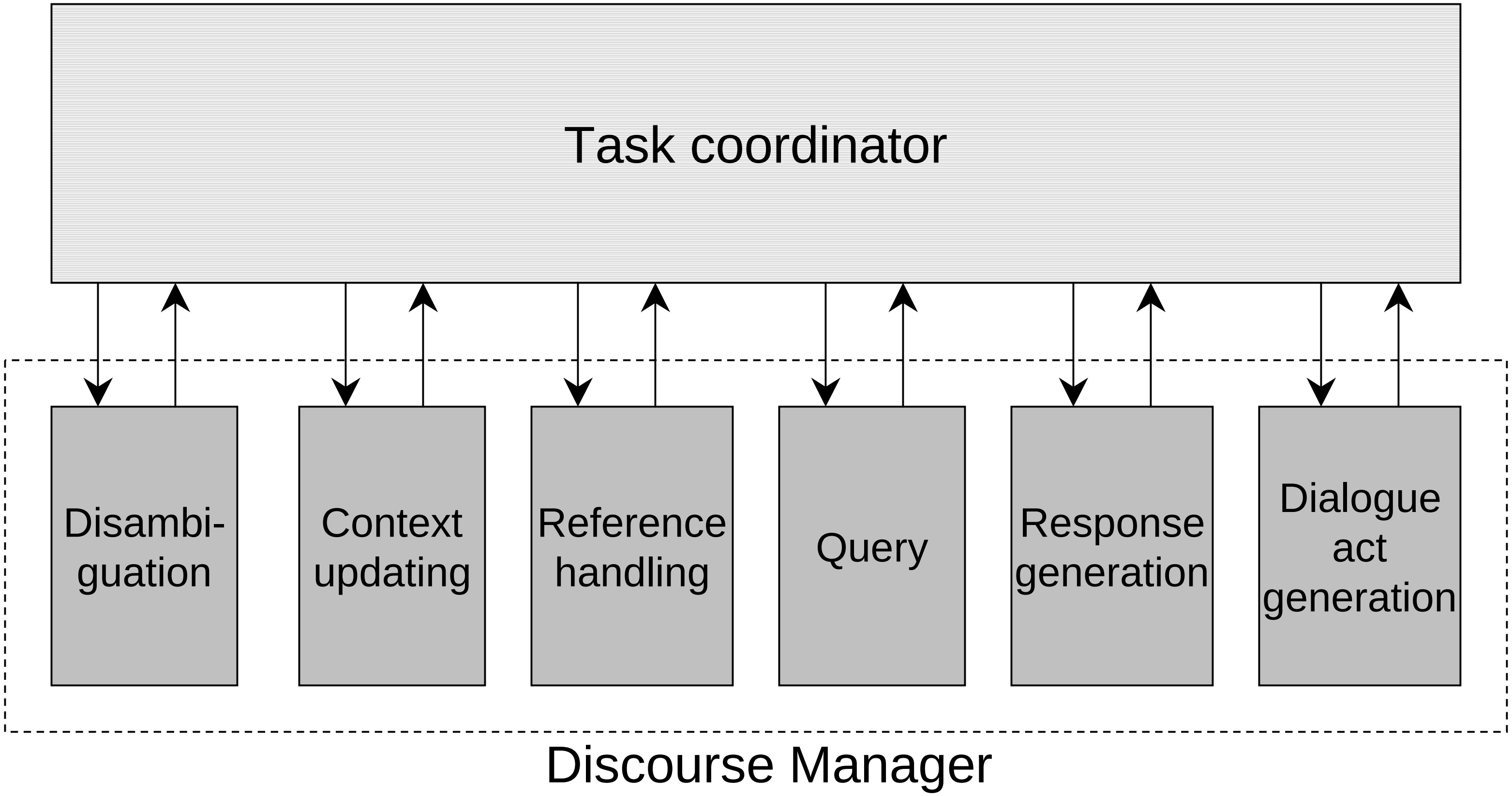
Alparon model



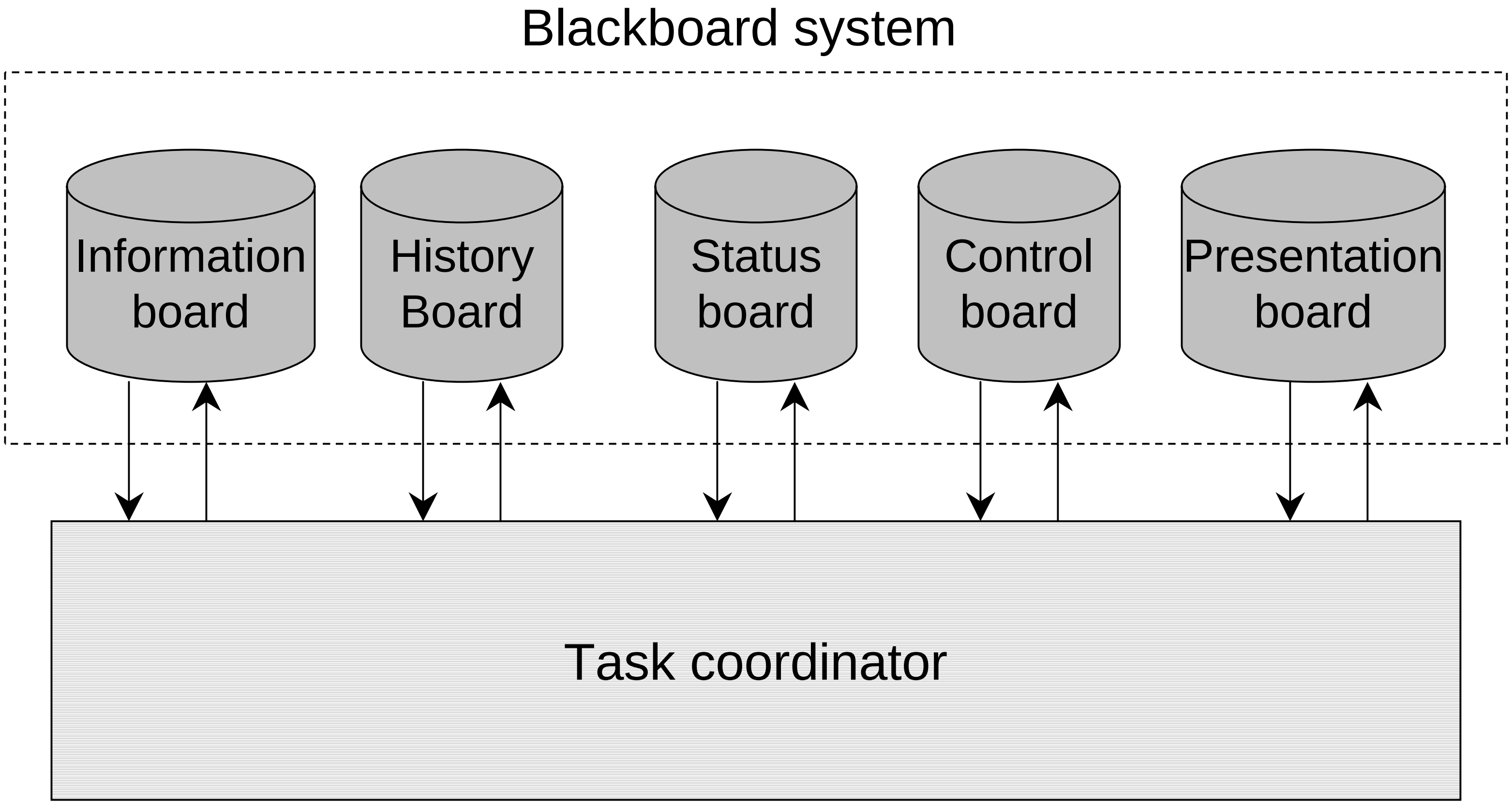
Alparon DM


Dialogue manager
Blackboard system
Rest of the ASP system
Task coordinator
Database manager
Discourse manager

Discourse Manager



Blackboard system



Contents

- Automatic speech processing (ASP)
- Alparon system
- Dialogue control
- Train timetable information system
- Conclusions

Dialogue control

Tasks

- structure dialogue
- generate responses

Two kinds of information

‘What time does the train to Amsterdam leave?’

Approach

- task-oriented dialogues
- plan-based
- goal structure
  - level of abstraction
  - task & dialogue goals
    - determine departure station
    - verify departure station
  - dynamic

Achievement

- operations
  - deductions
  - default behaviour
- interaction
  - *‘Departure at 8:30.’*
  - *‘What is your departure station?’*
- strategy

Implementation

- framework
- rulesets

Two steps

- determine consequences of user's turn
  → dialogue updating module
- generate response
  → response generation module

- Java

Dialogue updating module

Process effects of user’s turn

- new information
- conflicts
- choices
- reaction

Response generation module

Generate a system response

- actions
- candidates
- combine

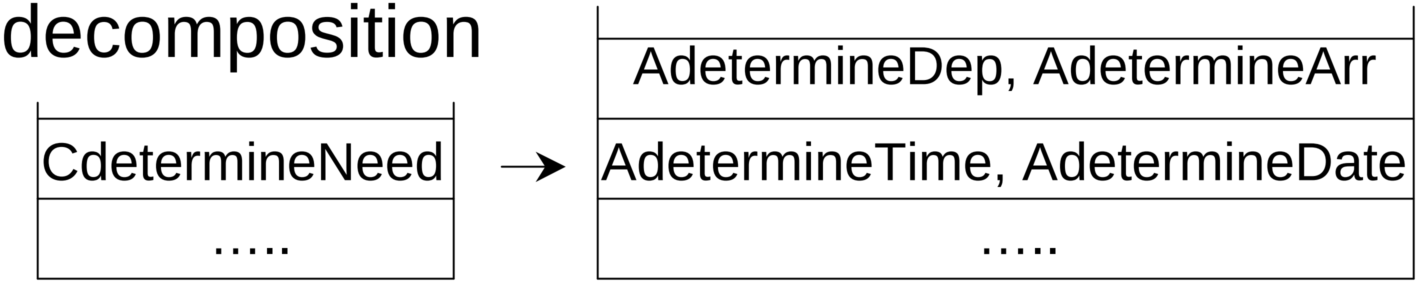
Control board

Store information for dialogue control

- goalstack
- conflicts
- choices
- response

Strategy 1

- decomposition



CdetermineNeed

.....

→

AdetermineDep, AdetermineArr

AdetermineTime, AdetermineDate

.....

- achievement
  - slot Time contains departure time
  - slot DeparturePlace is verified
  - communicated
- operations
  - city(Delft) → station(Delft)
  - → date = today

Strategy 2

- reactions

  reconfirmation of departure place → reconfirm(DepPlace)

  start dialogue → ChelpUser

- actions

  Averify(time) → [VerifyExplicit(time), VerifyImplicit(time)]

  Adetermine(DepPlace) → [Ask(DepPlace), Ask(Where), Encourage)]

- combinations
  - VerifyImplicit(*): {Ask(*)}

Contents

- Automatic speech processing (ASP)
- Alparon system
- Dialogue control
- Train timetable information system
- Conclusions

Train timetable information

- travelplan for train connections
- mixed-initiative
- information needed
  - departure station
  - arrival station
  - departure or arrival time
  - date

Phases

- greeting
- determination of user’s need
- query
- presentation
- further needs
- goodbye

Example actions

Adetermine(DepPlace): [Ask(DepPlace), Ask(Place), Encourage]

Adetermine(ArrPlace): [Ask(ArrPlace), Ask(Place), Encourage]

Adetermine(Time): [Ask(DepTime), Ask(Time), Ask(When)]

Adetermine(Date): [Ask(date), Ask(When)]

Averify(*): [VerifyExplicit(*), VerifyImplicit(*)]

Example dialogue 1

U1:	User connects.
Gs:	Cclose
	Ctravelplan
	Agreet
Res:	Greetings
S1:	*Hello, train timetable information service.*

DA Sys(Start_conversation):
send(CB, clear, dummy),
send(IB, clear, dummy),
goal(ChelpUser)

ChelpUser: [[Cclose], [Ctravelplan], [Cgreet]]

Cgreet: [[Agreet]]

Agreet: [Greetings]

Example dialogue 2

U2: *Hello, I want to travel from Baarn/Maarn to Den Haag.*

Gs: Cclose

Ctravelplan

Achoice(DepPlace, city(Baarn), city(Maarn))

Res: Choice(DepPlace, city(Baarn), city(Maarn))

S2: *Do you want to travel from Baarn or Maarn?*

Agreet: communicated

Achoice(*): [Choice(*)]

Example dialogue 3

U3: *I want to travel from Baarn.*

Gs: Cclose

CfurtherNeeds

Cpresent

Cquery

Adetermine(Time)

Adetermine(ArrPlace)

Achoice(ArrPlace, station(Den Haag cs), station(Den Haag hs))

Res: Choice(ArrPlace, station(Den Haag cs), station(Den Haag hs))

S0: *Which station in Den Haag do you want to go to, central station or Holland Spoor?*

Achoice(*): communicated

Ctravelplan: [[CfurtherNeeds], [Cpresent], [Cquery], [CdetermineNeed]]

CdetermineNeed: [[Adetermine(Date), Adetermine(Time)], [Adetermine(DepPlace), Adetermine(ArrPlace)]]

Adetermine(Date):
Date has empty : put(Date, today, default)

Adetermine(Date): Date hasnot empty

Adetermine(DepPlace): DepPlace is station(*) & DepPlace has user

Adetermine(ArrPlace):
ArrPlace is city(Den Haag) : goal(Achoice(ArrPlace, station(Den Haag cs), station(Den Haag hs))

Achoice(*): [Choice(*)]

Example dialogue 4

U4: *I want to go to Den Haag central station.*

Gs: Cclose

CfurtherNeeds

Cpresent

Cquery

Adetermine(Time)

Res: Ask(When)

S4: *When do you want to travel?*

Achoice(*): communicated

Adetermine(ArrPlace):

ArrPlace is station &
ArrPlace has user

Adetermine(Time):
[Ask(DepTime), Ask(Time),
Ask(When)]

Example dialogue 5

U5: *…???…*

Gs: Cclose

CfurtherNeeds

Cpresent

Cquery

Adetermine(Time)

Res: Ask(Time)

S5: *At what time do you want to travel?*

Adetermine(Time):
[Ask(DepTime), Ask(Time), Ask(When)]

Example dialogue 6

U6: *I want to arrive at half past nine.*

Gs: Cclose

CfurtherNeeds

ApresentPart(3)

ApresentPart(2)

ApresentPart(1)

Res: PresentPart(1)

S6: *OK. Departure at 7:35 in Baarn to Utrecht.*

Adetermine(Time):

Time is ArrTime(*)

Cquery: [[Aquery]]

Aquery:

true : send(QM, query, empty body, queryDone)

Cpresent: [[ApresentPart(*)]]

ApresentPart(*): [PresentPart(*)]

Example dialogue 7

U7: *When do I depart again?*

Gs: Cclose

CfurtherNeeds

ApresentPart(3)

ApresentPart(2)

Areconfirm(Time

(DepTime))

Res: Reconfirm(Time

(DepTime))

S7: *Departure at 7:35.*

ApresentPart(*):
communicated

DA Rec(*) :
goal(Areconfirm(*))

Areconfirm(*): [Reconfirm(*)]

Example dialogue 8

U8: *OK.*

Gs: Cclose

CfurtherNeeds

ApresentPart(3)

ApresentPart(2)

Res: PresentPart(2)

S8: *Departure at 8:32 to Den Haag cs in Utrecht.*

ApresentPart(*): communicated

ApresentPart(*): [PresentPart(*)]

Example dialogue 9

U9: *8:32 to Den Haag.*

Gs: Cclose

CfurtherNeeds

ApresentPart(3)

Res: PresentPart(3)

S9: *Arrival at 9:13.*

ApresentPart(*): communicated

ApresentPart(*): [PresentPart(*)]

Example dialogue 10

U10: *9:13 in Den Haag.*

Gs: Cclose

AfurtherNeeds

Res: AskFurtherNeeds

S10: *Do you want another travelplan?*

ApresentPart(*): communicated

CfurtherNeeds: [[AfurtherNeeds]]

AfurtherNeeds: [AskFurtherNeeds]

Example dialogue 11

U11: *No thanks.*

Gs: Aclose

Res: Bye

S11: *Have a nice day.*

AskFurtherNeeds: communicated

Cclose: [[Aclose]]

Aclose: [Bye]

Example dialogue 12

U12: *Goodbye.* and the user disconnects

Gs: ...

Res: ...

Aclose: communicated

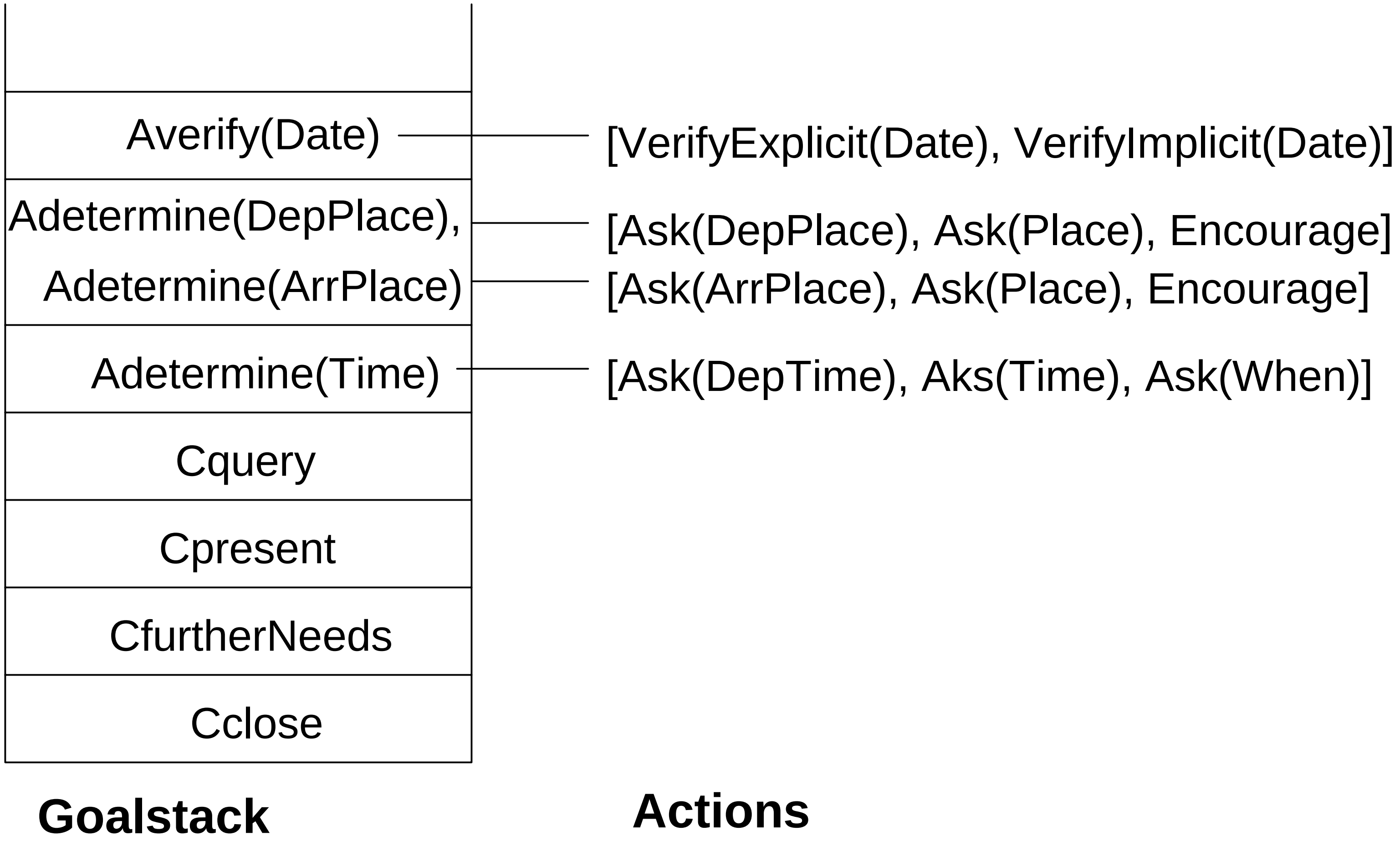
Contents

- Automatic speech processing (ASP)
- Alparon system
- Dialogue control
- Train timetable information system
- Conclusions

Conclusions

- two modules and a blackboard
- generic framework with rulesets
- mixed-initiative
- reactions
- adequate rulesets
- testing with complete system

Example goalstack & actions



Averify(Date) — [VerifyExplicit(Date), VerifyImplicit(Date)]

Adetermine(DepPlace), — [Ask(DepPlace), Ask(Place), Encourage]

Adetermine(ArrPlace) — [Ask(ArrPlace), Ask(Place), Encourage]

Adetermine(Time) — [Ask(DepTime), Aks(Time), Ask(When)]

Cquery

Cpresent

CfurtherNeeds

Cclose

Goalstack

Actions