Dialogue management using Finite State Models

Hagen Böhm

hagen@net.uni-sb.de

SS 2002

Categorizing dialogue management systems

- System-led control

System asks questions to elicit required parameters of task

- User-led control

User asks questions to obtain information

- Mixed initiative control

User and System elicit information, clarify unclear information, develop a common plan

Approaches to dialogue management

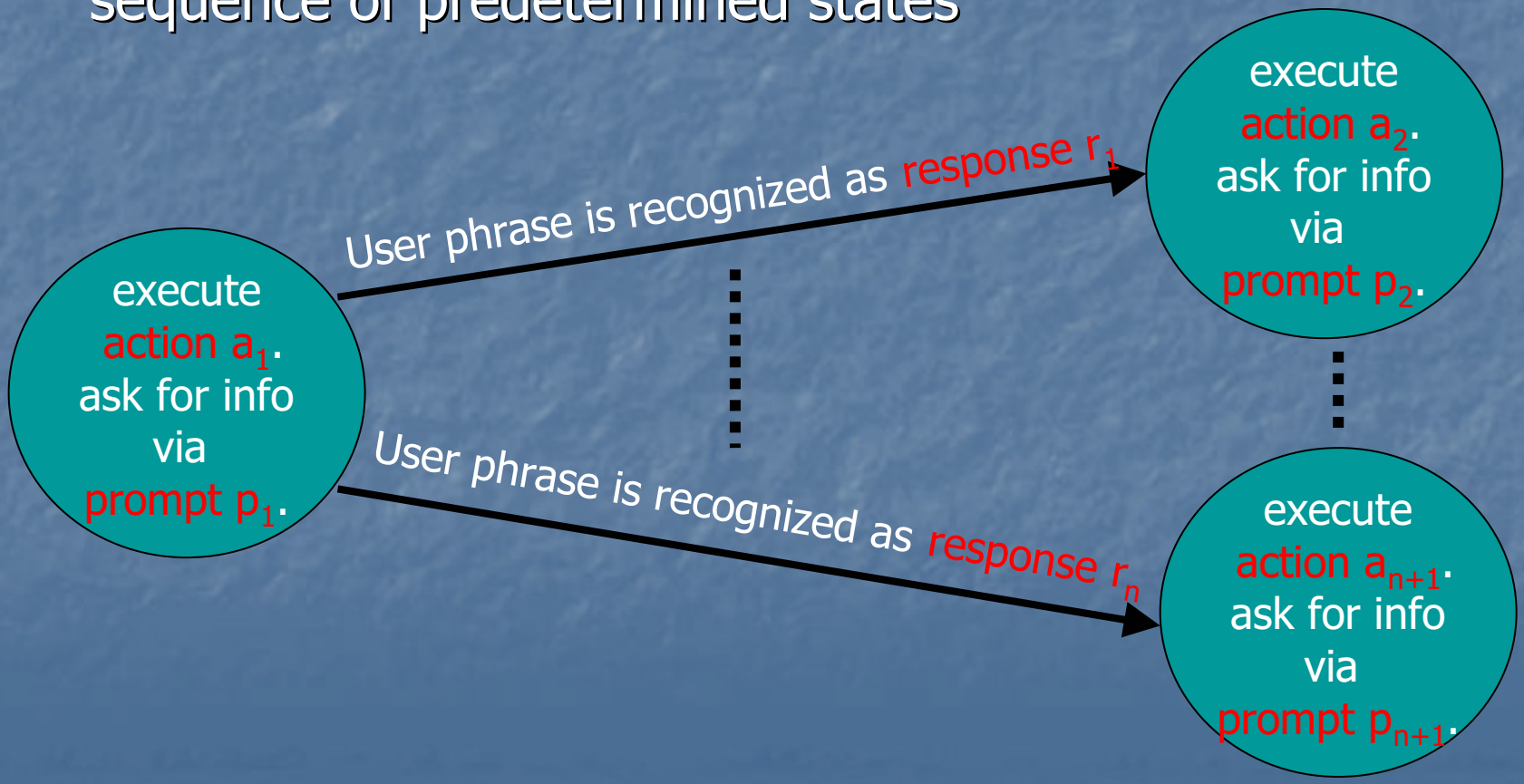
- Finite state methods
  - dialogue structure = state transition network
  - nodes = system's questions, network paths = legal dialogues
  - subdialogues in form of loops (supporting modular approach and libraries)
- Self-organizing
  - dynamically evolving structure
  - based on computation of next dialogue act
  - Several variants: plan-based, event-driven, frame-based …

Finite state automata (short introduction)

- Quintupel (Q,Σ,δ,q_0,F)
- Q = finite set of states
- Σ = finite input alphabet
- $q_0 \in Q$ and q_0 initial state
- F is
  - subset of Q
  - set of finite states
- $\delta : Q \times \Sigma \rightarrow Q$ d.h. $\delta(q,s) = p \quad (q,p \in Q, s \in \Sigma)$

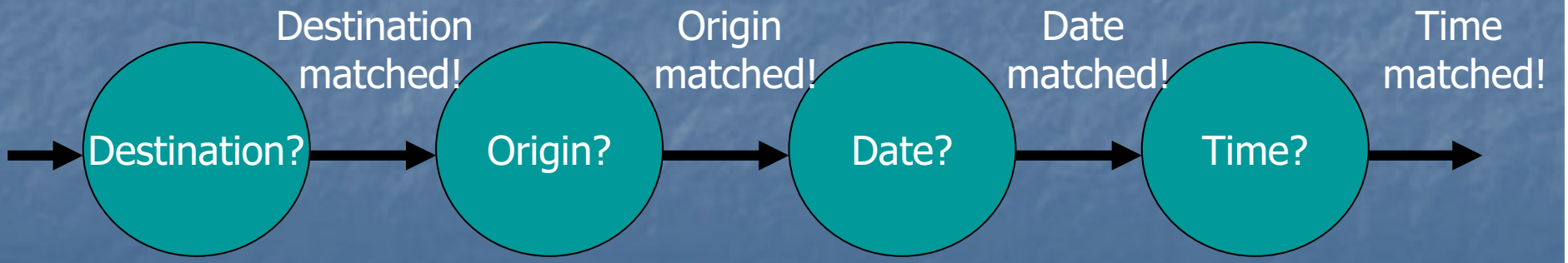
Finite state-based system

- User is taken through the dialogue via following a sequence of predetermined states



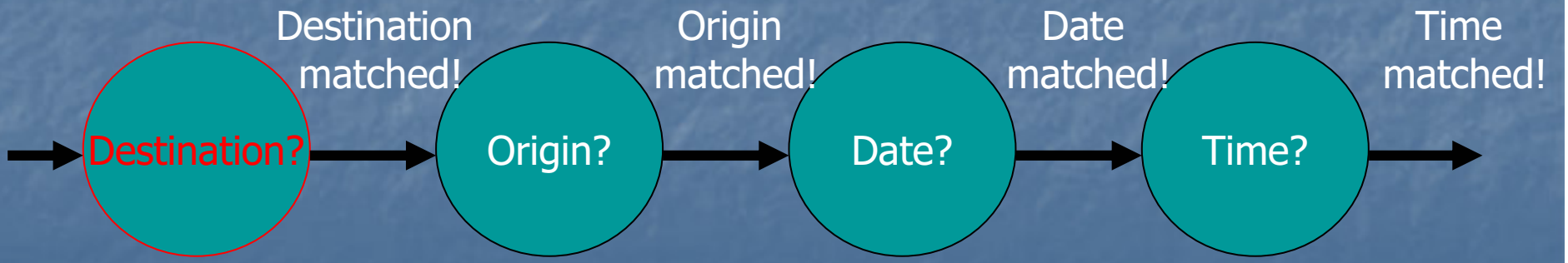
Finite state dialogue model example

- Simple travel inquiry system



Finite state dialogue model example

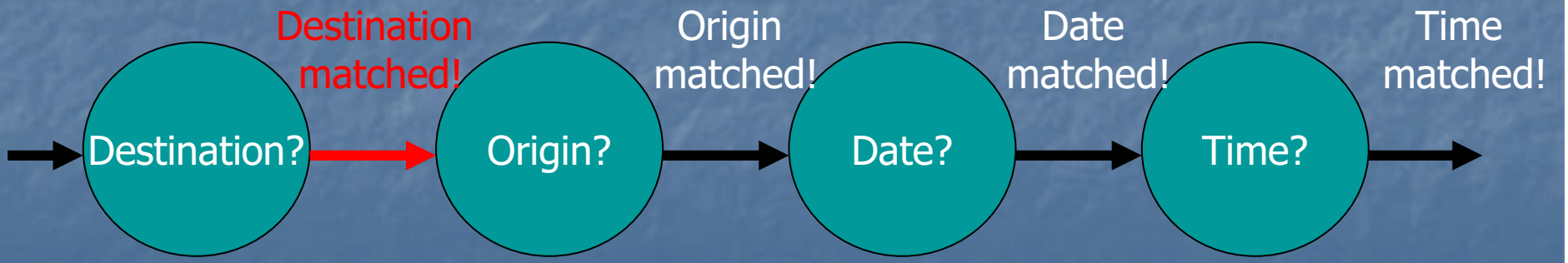
System: What is your destination?



Finite state dialogue model example

System: What is your destination?

User: London.

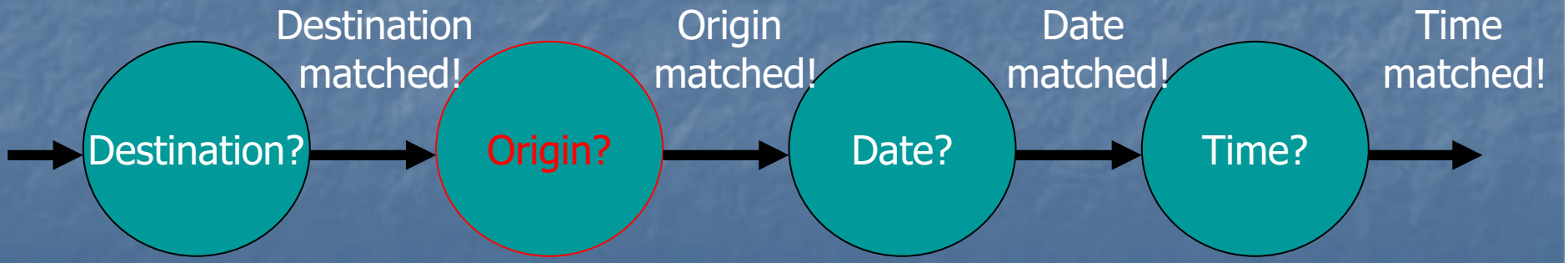


Finite state dialogue model example

System: What is your destination?

User: London.

System: What is your origin?



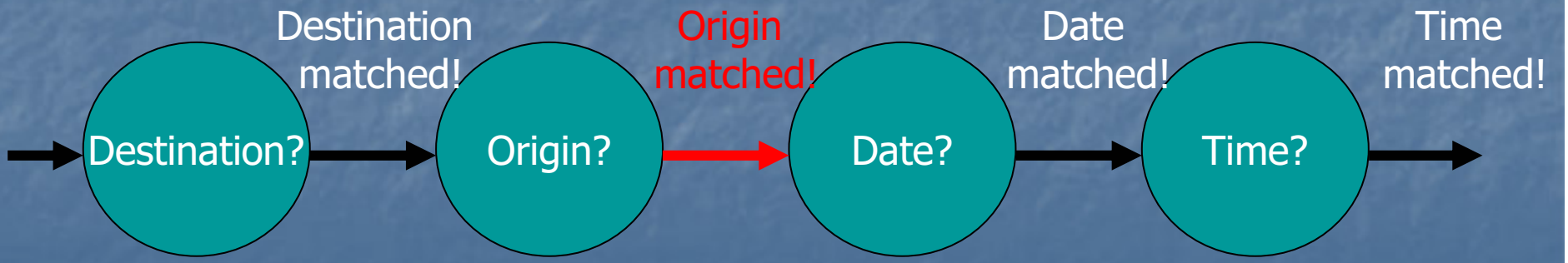
Finite state dialogue model example

System: What is your destination?

User: London.

System: What is your origin?

User: Hamburg.



Finite state dialogue model example

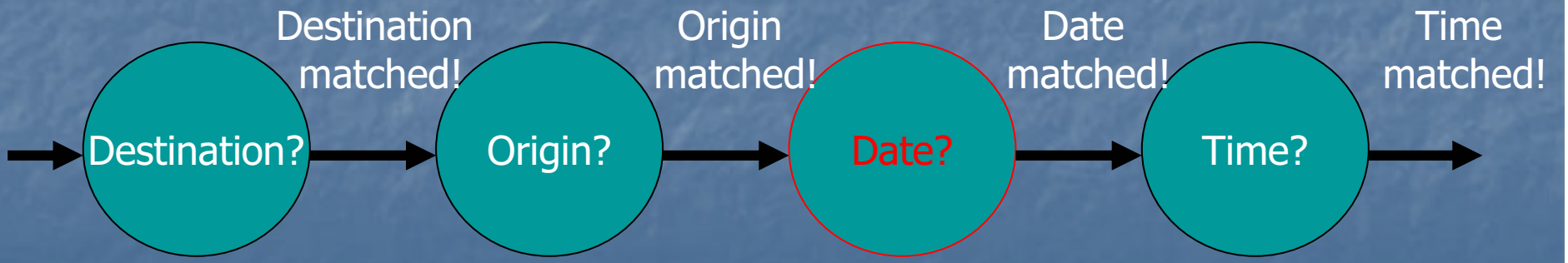
System: What is your destination?

User: London.

System: What is your origin?

User: Hamburg.

System: What day do you want to travel?



Finite state dialogue model example

System: What is your destination?

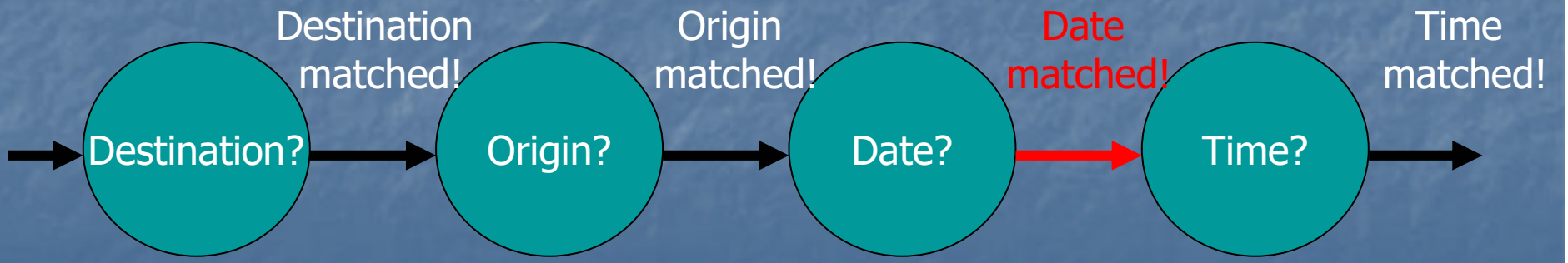
User: London.

System: What is your origin?

User: Hamburg.

System: What day do you want to travel?

User: Saturday.



Finite state dialogue model example

System: What is your destination?

User: London.

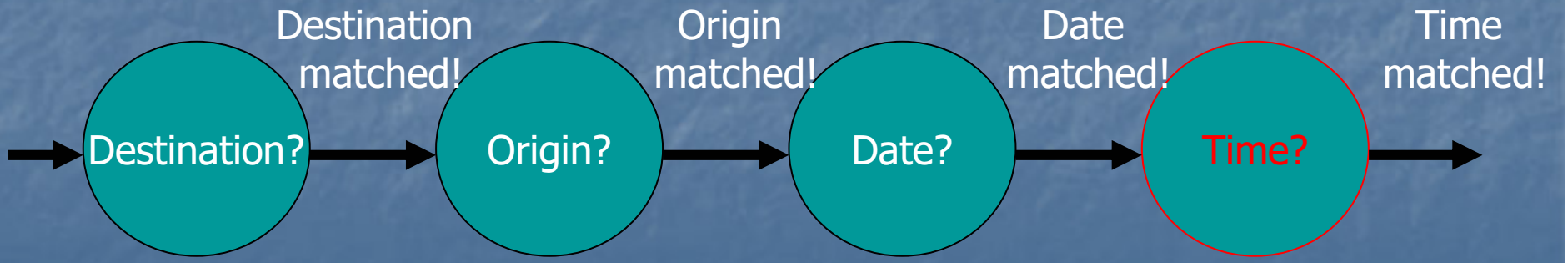
System: What is your origin?

User: Hamburg.

System: What day do you want to travel?

User: Saturday.

System: What is the departure time?



Finite state dialogue model example

System: What is your destination?

User: London.

System: What is your origin?

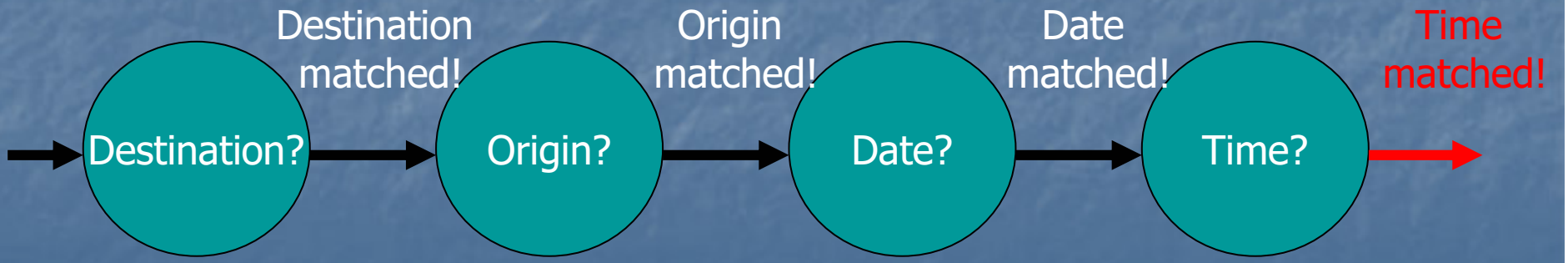
User: Hamburg.

System: What day do you want to travel?

User: Saturday.

System: What is the departure time?

User: 6pm.



Advantages

- User's input limited to single predefined words or phrases
  - Simplified speech recognition
  - Full natural language processing not necessary
- Structured dialogue
  - Simplified dialogue management component (user is directed through dialogue)
  - More reliable performance (quite robust!)
  - Simple to develop
  - Suitable for well-structured tasks

Disadvantages

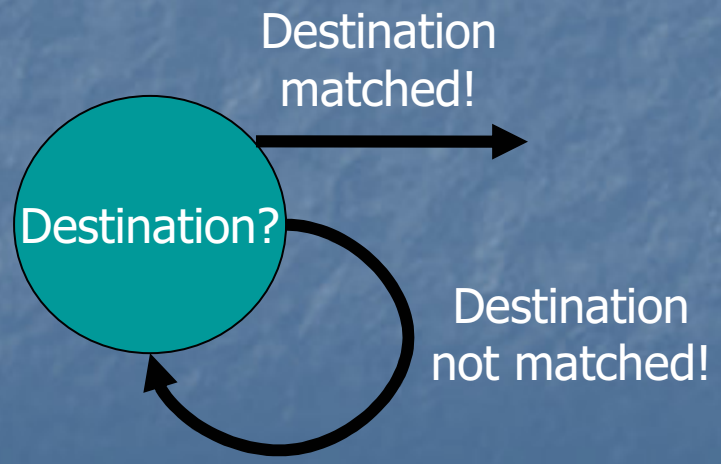
- User's input **limited** to **single predefined** words or phrases
  - Unable to cope with more complex dialogues. In particular:
    - Verification of speech recognition
    - Deviations from dialogue structure
    - Error recovery and dialogue repair
    - Dealing with non-atomic structures
- User can't take initiative
  - Grounding behavior must be hard-coded (verification)
  - Negotiation cannot be modeled (constraints may be unknown by system and user at outset)

Verification of speech recognition

- Every piece of information is elicited by a single question and matched against possible answers
- Interpretation of User's answer could fail due to
  - Speech recognition failure
  - Match failure

Verification of speech recognition Prompt repetition

- User's answer could not be recognized
- 1st approach: For every state add an extra transition-arc handling the "no match!" case by repeating the prompt
- Example:



Verification of speech recognition Prompt repetition example

System: What is your destination?

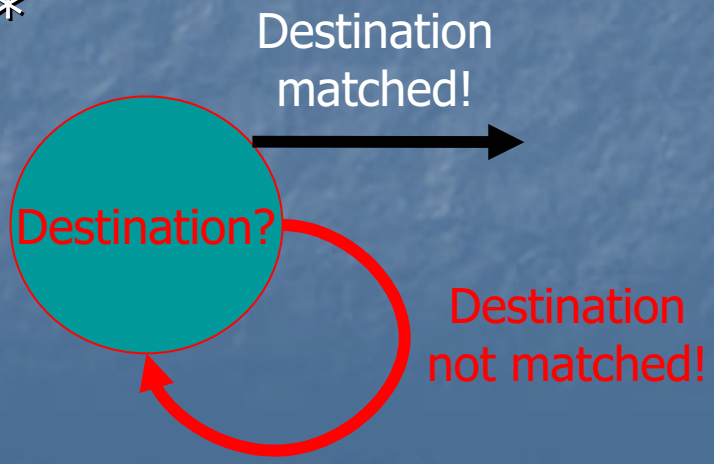
User: London.

System: What is your destination?

User: London!

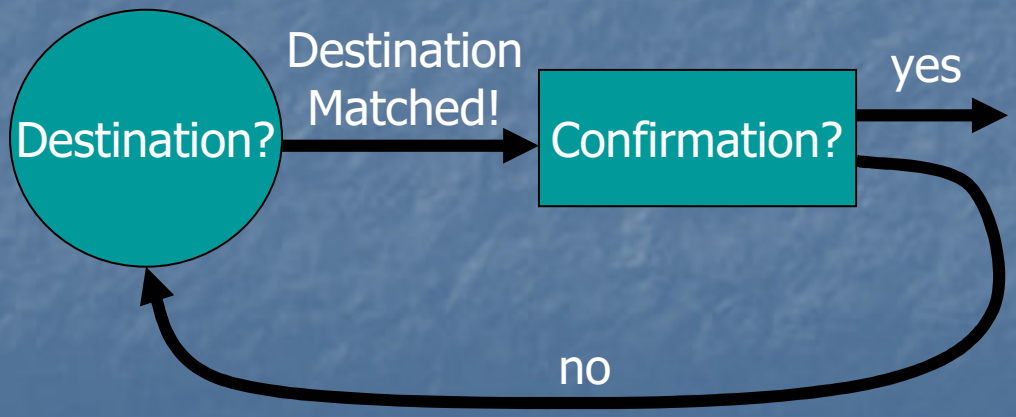
System: What is your destination?

User: … *ARGH!*



Verification of speech recognition Explicit confirmation

- User's answer could not satisfactorily be recognized
- 2nd approach: For every state add an extra "confirmation"-state, confirming the proposed match
- Example:



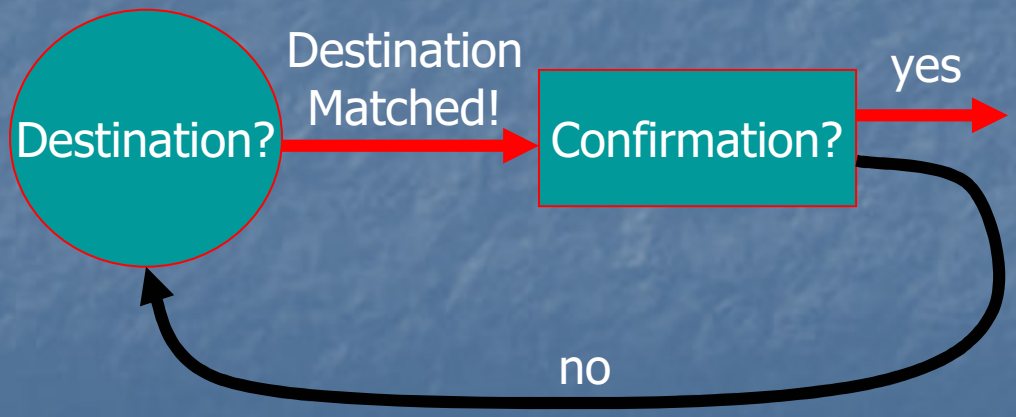
Verification of speech recognition Explicit confirmation example

System: What is your destination?

User: London.

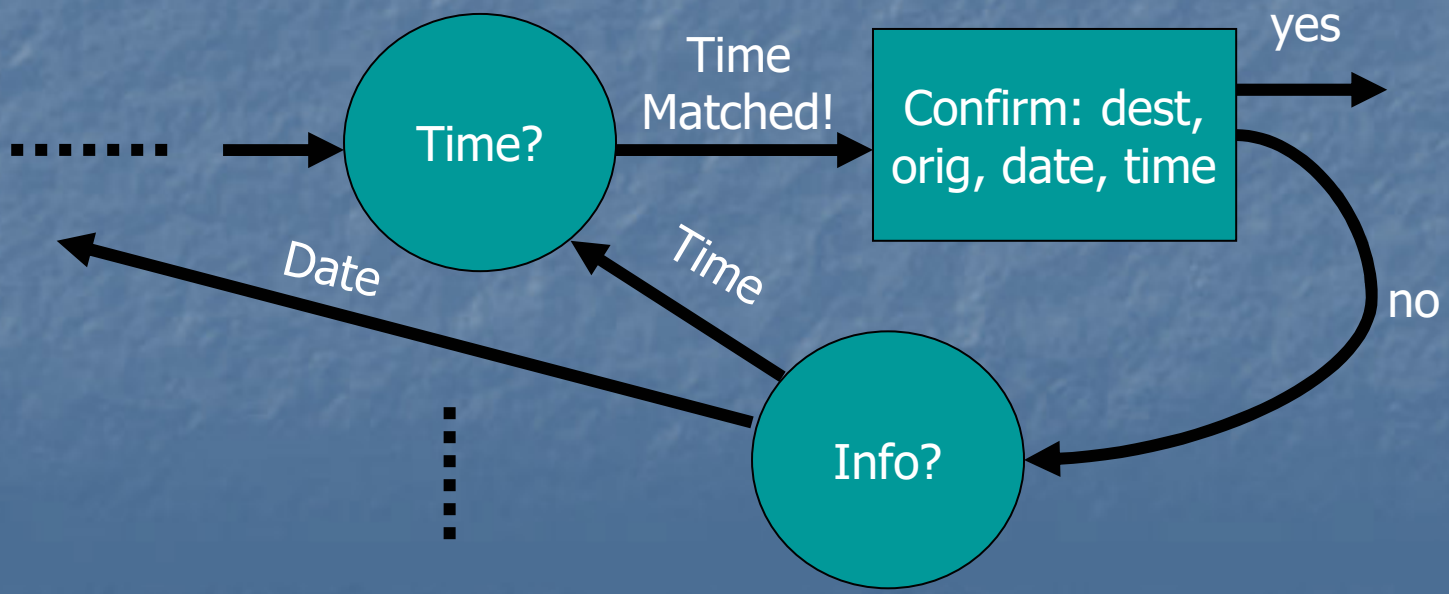
System: Was that London?

User: Yes.



Verification of speech recognition Delayed confirmation

- User's answer could be matched with high probability
- 3rd approach: dispense with confirmation for all but the most important items of info
- Example:



Verification of speech recognition Delayed confirmation example

……

System: What is the departure time?

User: 6pm.

System: You want to travel from Hamburg to Landau on Sat May 8th 2002 at 6pm. Is that right?

User: No.

System: Would you like to change the destination, origin, time or date?

User: Destination please.

System: What is your destination?

……

Verification

- Depends on the confidence score for the user's response
  - Low confidence: Repeat prompt
  - Medium confidence: Confirm answer
  - High confidence: Delayed confirmation of info
- Add more natural behavior to dialogue

Deviation from dialogue structure

- User's answer is over-informative and deviates from dialogue path

- System ignores or fails to process additional information
- System asks irrelevant questions

Managing over-informative answers

- Constrain user's input to the required responses (more restricted, simple)
- Provide additional questions/transitions to cope with range of possibilities (more flexible, more complex)
- Keep track of additional information by using a frame or form. More complex natural language input can be accepted. (more flexible, more natural, deviates from system architecture)

Example: Banking system

System: What company would you like to pay?
User: *Abbey National.*
System: How much would you like to pay?
User: *One hundred pounds next Monday.*
System: What date would you like the payment to be made on?
User: *Next Monday.*
System: You want to pay Abbey National one hundred pounds and sixty-one pence on April 5th? Is that correct?
User: *No.*
System: Would you like to change the company, amount or date?
User: *Amount.*
System: How much would you like to pay?
User: *One hundred pounds.*
System: You want to pay Abbey National one hundred pounds on April 5th? Is that correct?
User: *Yes.*

Does this system have track keeping (info storage)?

Error recovery + dialogue repair

- Errors originate from
  - Misinterpretation of input
  - Ill-formed, incomplete or inaccurate user input
  - Getting into stuck configuration (dependencies of information on different levels)

Error recovery + dialogue repair Adding natural behavior

- Augment finite state system with appropriate transitions and states to…
  - undo the last operation (“scratch”-command)
  - change the last input (“change”-command)
  - completely reset the system (“clear history”-command)
- System’s complexity grows fast!!!
- Problems remaining:
  - How to undo a whole branch?
  - How to deal with complex dependencies?

Problem: non-atomic structures

Do you want to go from Trento to Milano? Yes or no?

- One value is incorrect → user may correct it
- Both values are incorrect → ???

Alternative: Confirm each value separately

Do you want to go from Trento? Do you want to go to Milano?

- More robust method for confirming values
- But increases the number of turns required to complete the dialogue

Conclusion
Strengths (1)

Finite state models are ...

- simple to develop

- suitable for modeling simple tasks with flat menu structure and less options like
  - travel inquiries
  - weather forecasts
  - ordering items
  - simple bank transactions

Conclusion
Strengths (2)

Finite state models have ...

- clear and intuitive semantics
  - System retaining control
  - System decides which question to ask next
- fewer technological demands (?)
  - Less flexibility and naturalness (trade-off against their demands) leads to less errors (?)
  - More robust and save (explicit confirmation/repair, greater use of isolated word recognition)

Conclusion
Weaknesses (1)

Finite state models are …

- not suitable for tasks having subtasks in unpredictable order
- inflexible. Problems occur if …
  - user must correct items
  - user must introduce information not foreseen
- bound to predictable dialogue courses

Conclusion
Weaknesses (2)

- FSM cannot deal with complex dependencies
  - Clash between **dependencies** results in a stuck which can only be repaired by expensive **backtracking**
  - Example: Flight reservation system giving the opportunity of starting a dialogue on a particular "level" (discounted fares) then reaching another "level" (flight availability) which results in a clash with the former one.