Lecture 4: Plan and Agent-based Approaches to Dialogue

Dialogue Control as Theorem-Proving

- Examples
  - Smith, Hipp & Biermann
  - Sadek et al
  - Midas (Bos & Gabsdil)
  - Active Logic (Perlis, Traum, Purang,…)

Smith, Hipp, & Biermann

- Target Behaviors for dialogue system
  - Convergence to goal (intentional structure
  - Subdialogues (segments, linguistic structure)
  - User model (updated as dialogue progresses)
  - Mixed initiative
  - Linguistic focus (attentional structure)
    - Expectations for Speech recognition/parsing
    - Expectations for tracking topic shifts

SHB: Initiative Modes

- Directive
- Suggestive
- Declarative
- Passive

SHB: Sample directive mode Circuit-fixit dialogue

1. Computer: Put the knob to one zero.
2. User: Okay.
3. Computer: What is the voltage between connector one two one and connector three four?
4. User: I do not know.
5. Computer: Locate the voltmeter.
6. User: Done.
7. Computer: Add a wire between the minus com hole on the voltmeter and the connector one two one.
8. User: How do I do that?
9. Computer: Connect the end of the black wire with the small plug to the minus com hole on the voltmeter.
10. User: Done.
11. Computer: Connect the end of the black wire with the large plug to connector one two one.
12. User: Which knob?
13. Computer: Near the top boundary and in the center is a green region. Now, in the middle right of this green region is the control knob.
14. User: Okay.
15. Computer: Put the knob to zero one.
16. User: Done.
17. Computer: Connect the end of the black wire with the large plug to connector one two one.
18. User: Where is connector three four?
19. Computer: Near the left boundary and toward the middle is a green region. Now, in the bottom center of this green region is connector three four.
20. User: Okay.
21. Computer: What is the voltage between connector one two one

SHB: circuit fix-it declarative mode

1. Computer: This is the circuit-fixit-shop. How may I help you?
2. User: RS111 is not working.
3. Computer: I am familiar with that circuit. The LED is supposed to be displaying alternately flashing one and seven.
4. User: There is no wire on one zero four.
5. Computer: There is supposed to be a wire between connector one zero four and connector one zero two.
6. User: The circuit is working.
7. Computer: Glad to have been of assistance. Good-bye.

SHB: subdialogue structure

- Sub-dialogue is all the interactions related to a sub-goal
- Subgoals always accessible (even when completed)
- Subgoals can be entered by
  - Theorem prover creates new subgoal to prove
  - Dialogue Controller moves to a new subgoal
  - User initiaties new subgoal

SHB: Circuit Fixit

- Domain: Radio shack circuit board
- Goal: create circuits to achieve some objective (e.g., light 1, and flashing 7)
- System capabilities:
  – Knowledge of how to build circuits
  – Knowledge of how to diagnose situations
- Human capabilities:
  – Can report circuit status
  – Can modify circuits

SHB: Missing Axiom Theory

- Dialogue as Proof process (a la prolog)
  - Goal of dialogue is goal of proof
  - When proof is completed, dialogue is finished
  - Interactions with user to supply "missing axioms" to help complete the proof
  - Example: goal of observeposition(sw1,X)
    - If this goal is in KB, can proceed, otherwise backward chain:
    - Inference rule:

    observeposition(sw1,X) <- find(sw1), reportposition(sw1,X)
    - If both clauses in KB, then can prove goal with no dialogue
    - Some subgoals can be vocalized to get info from user

SHB: IPSIM theorem proving

- Prolog depth-first search too limited to support all the types of user interaction
- IPSIM implementation (interruptible prolog simulator)
- IPSIM operations
  - Normal theorem proving
  - Pass control to dialogue controller to get a missing axiom
  - Accept queries from dialogue controller about proof status

SHB: User Model input inferences

(1) If the input indicates that the user has a goal to learn some information, then conclude that the user does not know about the information.

(2) If the input indicates that an action to achieve or observe a physical state was completed, then conclude that the user knows how to perform the action.

(3) If the input describes some physical state, then conclude that the user knows how to observe this physical state. In addition, if the physical state is a property, then infer that the user knows how to locate the object that has the property.

(4) If the input indicates that the user has not performed some primitive action, make the appropriate inference about the user's knowledge about how to perform this action.

(5) If the user has completed an action by completing each substep, then conclude that the user knows how to do the action.

(6) Infer that the user has intensional knowledge about a physical state if the user has knowledge on how to observe or achieve the physical state.

(7) Infer that the user has knowledge on how to observe a physical state if he or she has knowledge on how to achieve the physical state.

SHB: Initiative mode and subdialogue selection

- Directive mode: system chooses all subdialogues (except clarification)
- Suggestive mode: system chooses, but user can interrupt with related goals
- Declarative mode: user chooses but computer can mention relevant facts
- Passive mode: system never takes initaitve, only responds to user

SHB: Input Processing with expectations

- Hierarchy of expectations based on the subdialogue relationship
- Expectation types related to task attempt:

(1) A statement about missing or uncertain background knowledge necessary for the accomplishment of *S*.

(2) A statement about a subgoal of *S*.

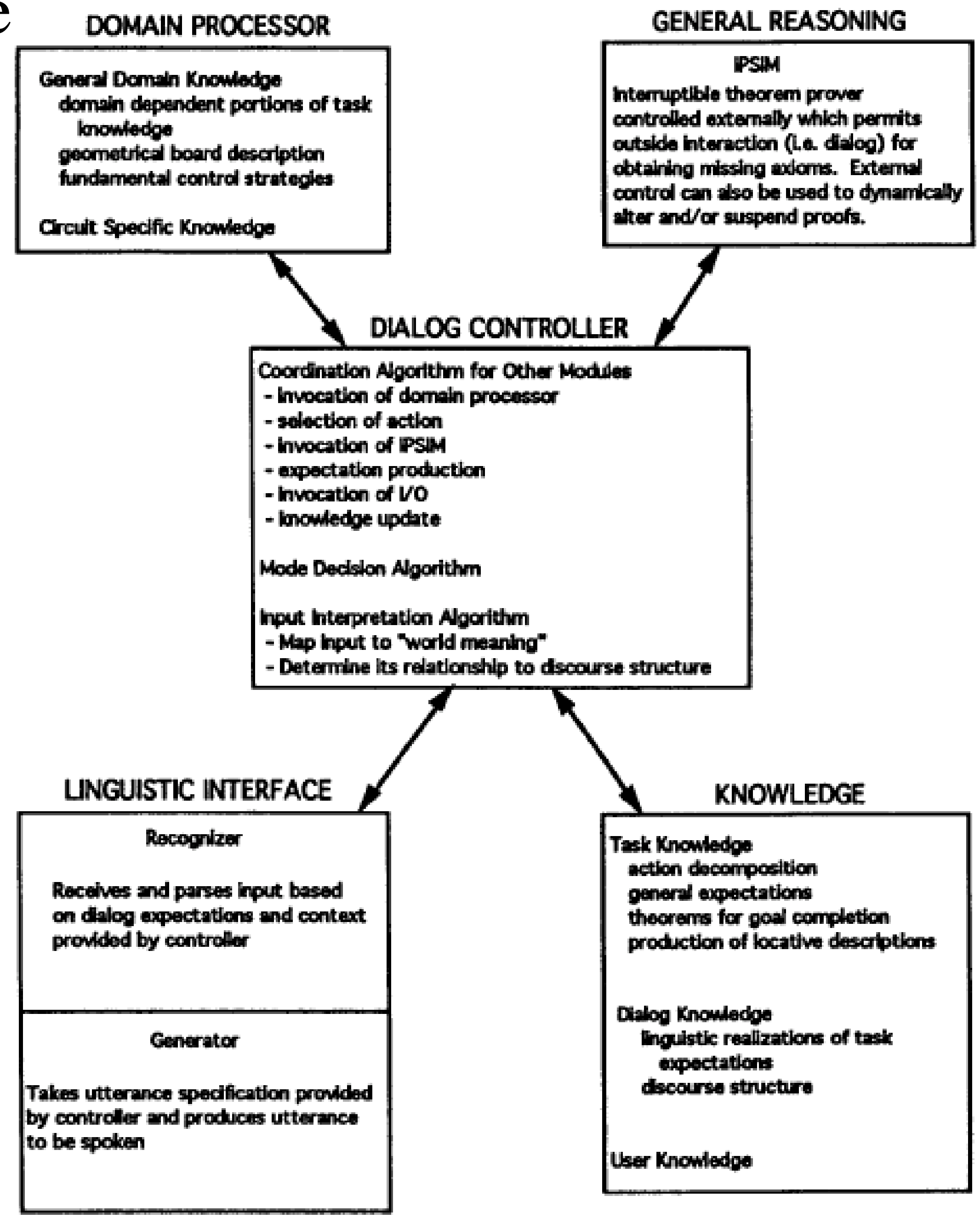
(3) A statement about the underlying purpose for *S*.

(4) A statement about ancestor task steps of which accomplishment of *S* is a part.

(5) A statement about another task step which, along with *S*, is needed to accomplish some ancestor task step.

(6) A statement indicating accomplishment of *S*.

SHB: System Architecture



SHB: Dialogue Control Algorithm

(1) Obtain suggested goal from the domain processor.

(2) Based on the suggested goal and the current state of the dialog, select the next goal to be pursued by the computer and determine the expectations associated with that goal. (The goal may thus be selected from one of the active subdialogs. The choice is partially dependent on the current level of initiative.)

(3) Attempt to complete the goal using the IPSIM system, possibly invoking voice interactions.

(4) Update system knowledge based on efforts at goal completion.

(5) Determine next operations to be performed by the domain processor in providing a suggested goal.

(6) Go to step 1.

Input Processing (Hipp’s parser)

- Output in GADL meaning representatin
- Syntactic/semantic paired grammer rules
- Input is
  - ASR output word lattice(Verbex 6000 speech recognizer)
  - Weighted Expectations from dialogue controller
- Cost minimization function for best parse

Plan-based dialogue: Roots

- Speech Act theory (50s-60s: Austin, Searle, Gordon&Lakoff,…)
- AI Planning (early 70s: Fikes & Nilsson, Tate, Sacerdoti,…)
- Plan-based theory of SAs (Perrault, Cohen and Allen: late 70s)
- Theory of rational action (80s, 90s)
- Theory of collaborative action (late 80s, 90s)

Speech Acts

- Austin: doing things with words
  - Explicit performatives
  - Locutionary, Illocutionary, perlocutionary acts
- Searle
  - Felicity conditions
- Gordon & Lakoff
  - Conversational postulates

AI Planning

- Actions as plan operators
  - Preconditions
  - Body (decomposition)
  - Effects
- Plan construction
  - Find a sequence of actions to lead from current state to goal state
  - Backward chaining - find action with goal as effect then use preconditions of action as new goal, until no unsatisfied preconditions
- Plan recognition (inference)
  - From action to preconditions (before action)
  - From action to effect (after action)

Theory of Rational Action

- Basic Attitudes
  - Belief
  - Desire
  - intention

Plan-based account

- Speech acts as AI Planning operators (Perrault, Cohen and Allen)
  - Plan construction (Cohen)
  - Plan recognition (Allen)

Perrault and Allen

- Logic of Beliefs and Wants
- Plan operators for speech acts
  - 2 levels:
    - Illocutionary
    - surface
- Inference rules for construction
- Heuristics for plan expansion

Perrault and Allen: Hypotheses

1. Language users are rational agents
2. Rational agents can identify actions and goals of others (and sometimes adopt them)
3. To successfully perform a speech act, speaker must intend hearer recognize intention to achieve effects of act
4. Language users know that others are rational agents
5. Speakers can perform one act by performing another, along with expectations of cooperative and rational behavior of others

Perrault & Allen: Speech Acts

Illocutionary

- Inform(S,H,P)
  - Pre: K(S,P) & W(S,Inform(S,H,P)
  - Effect: K(H,P)
  - Body: B(H,W(S,K(H,P))),
  - DecideToBelieve(A,O,P)
- Informif, informref
- Request(S,H,P)
  - Pre: W(S,A(H))
  - Effect: W(H,A(H))
  - Body: B(H,W(S,A(H)))
  - CauseToWant

Surface

- S.Inform(S,H,P)
  - Effect: B(H,W(S,K(H,P)))
  - Body: declarative utterance "that P"
- S.Request(S,H,A)
  - Effect: B(H,W(S,A(H)))
  - Body: imperative (or interrogative if A is an inform)

Perrault and Allen: Inference

- S performed IA by uttering x to H if S intends that H recognize
  1. X is an instance of surface act SA
  2. S intended H to infer from S having performed SA that S wants to achieve the effects of IA

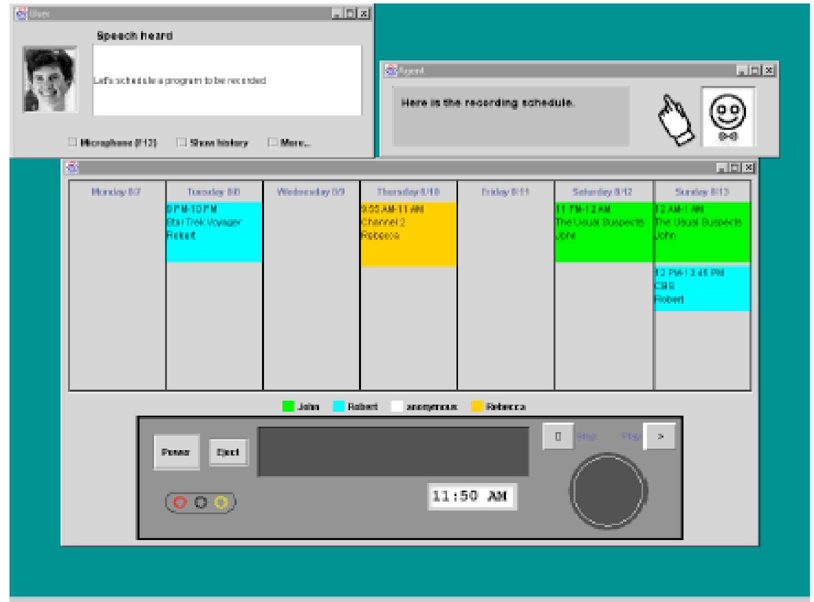
Collagen

- Rich, Sidner, Lesh (and others on applications)
- Theoretical Foundations:
  - Grosz & Sidner 86: Discourse Structure
  - Grosz & Sidner 90,Grosz and Krauss: Shared Plans
  - Sidner 94: Collaboration Language
  - Lochbaum: Discourse Interpretation
- Viewpoint:
  - Problem-solving layer as User-interface “middleware”
    - Facility for observing, recording, and organizing collaborative action
    - Not full agent, making decisions and acting
  - Advancing over GUIs/WIMP
    - Context-sensitive menus, based on current collaboration state
  - Focus on dialogue modelling, NOT NL interpretation/generation

Collagen: GUI use

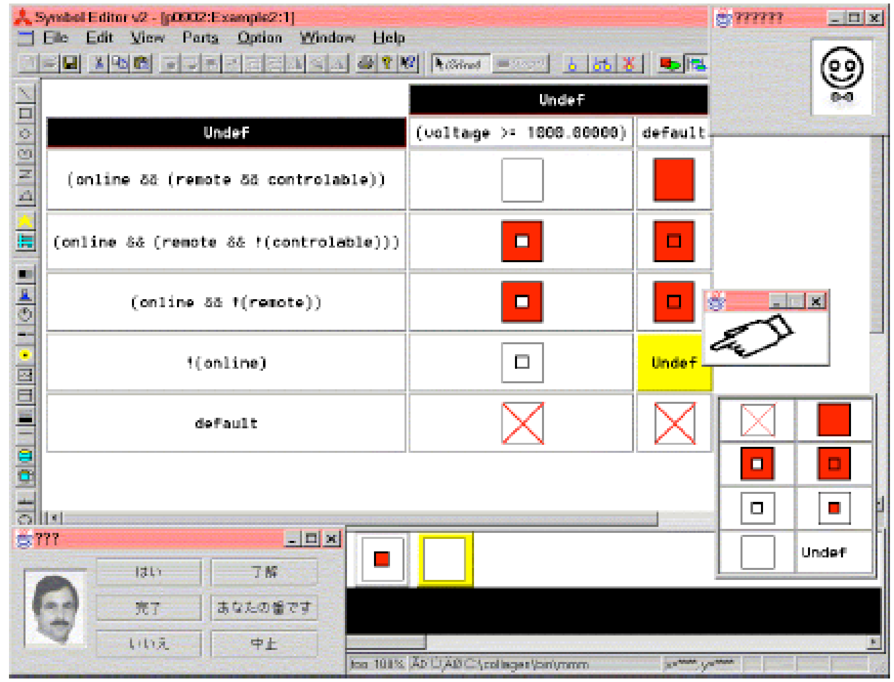
- Application GUI (as could be used without collagen)
- User “home”
  - Menu selection
- Agent “home”
  - Pointer and attention
  - Output text

Sample Collagen System: VCR help



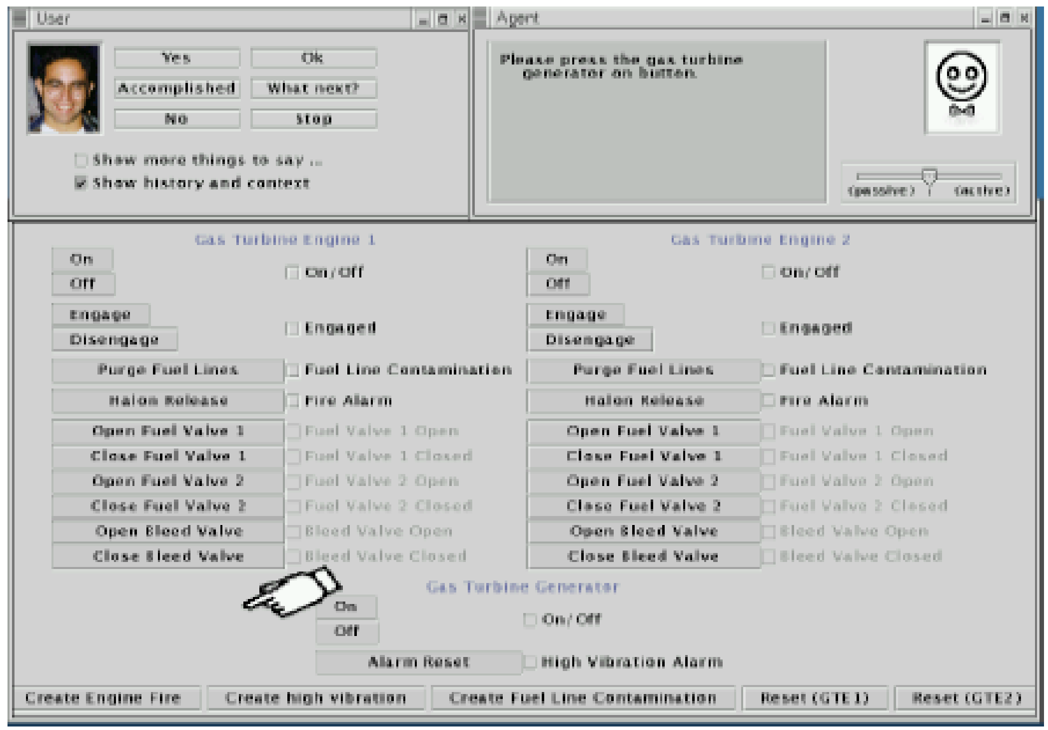

```
 1 User says "Hello."
 2 Agent says "Hello, what's your name?"
 3 User says "Candy."
 4 Agent says "Hello, Candy!"
 5 User says "Let's set the clock."
 6 Agent says "What is the time?"
 7 User says "The time is 11:00 AM."
 8 Agent sets the clock to 11:00 AM.
   ...
 9 User says "How do I play back a video from my camcorder?"
10 Agent says "First, plug in the camcorder cables."
11 User says "How do I plug in the camcorder cables?"
12 Agent points to where you plug in the camcorder cables.
13 Agent says "The red cable plugs into the red terminal, the
   black cable plugs into the black terminal, and the yellow cable
   plugs into the yellow terminal.  Make sure the other ends of the
   cables plug into the corresponding colors on the camcorder."
14 User says "What next?"
15 Agent says "Now press Play on your camcorder.  You should
   see a picture on the TV screen."
16 User says "Ok."
```

Sample Collagen System: Symbol Editor



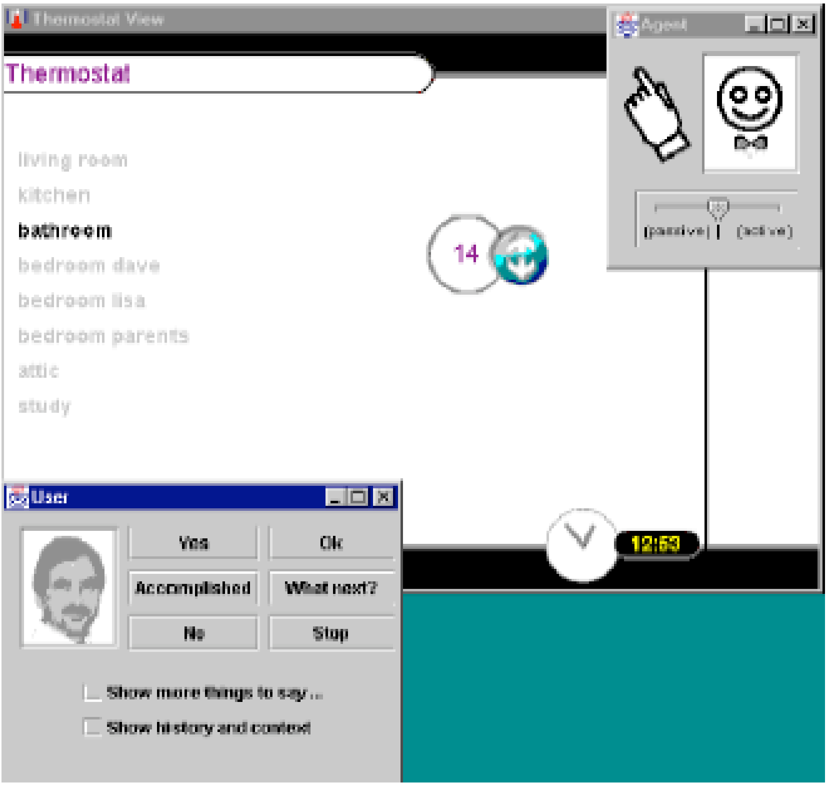
1 Agent says "Please list the other (non-boolean) attributes used in the columns of the logic table."
2 User says "The attributes are: voltage."
3 Agent creates a new user-defined attribute.
4 Agent enters "voltage" as the name of the new attribute.
5 Agent says "Please select the data type of the new attribute."
6 User selects float as the data type of the new attribute.
7 User clicks ok to accept the definition of voltage.
8 Agent says "Please edit the initial setting for the voltage attribute (if necessary) and then click ok."
9 User clicks ok to accept 0.0 as the initial setting for voltage.
10 Agent shows the expression editor for column 1.
11 Agent enters (voltage > 0.0) as the title for column 1.
12 Agent says "Please define an expression for column 1."
...[user goes to lunch]...
14 User says "Where are we?"
15 Agent says "You are defining an expression for column 1, as part of making a logic table."

Sample Collagen System: Turbine Agent



1 Agent says "Let's start the generator."
2 User says "Ok."
3 Agent says "You take it from here."
4 User presses the alarm reset button.
5 User says "<u>What next?</u>"
6 Agent says "Please press the gas turbine generator on button."
7 Agent points to the gas turbine generator on button.
8 User presses the gas turbine generator button.
9 Agent says "Good. We have successfully started the generator."

Sample Collagen System: Thermostat Agent

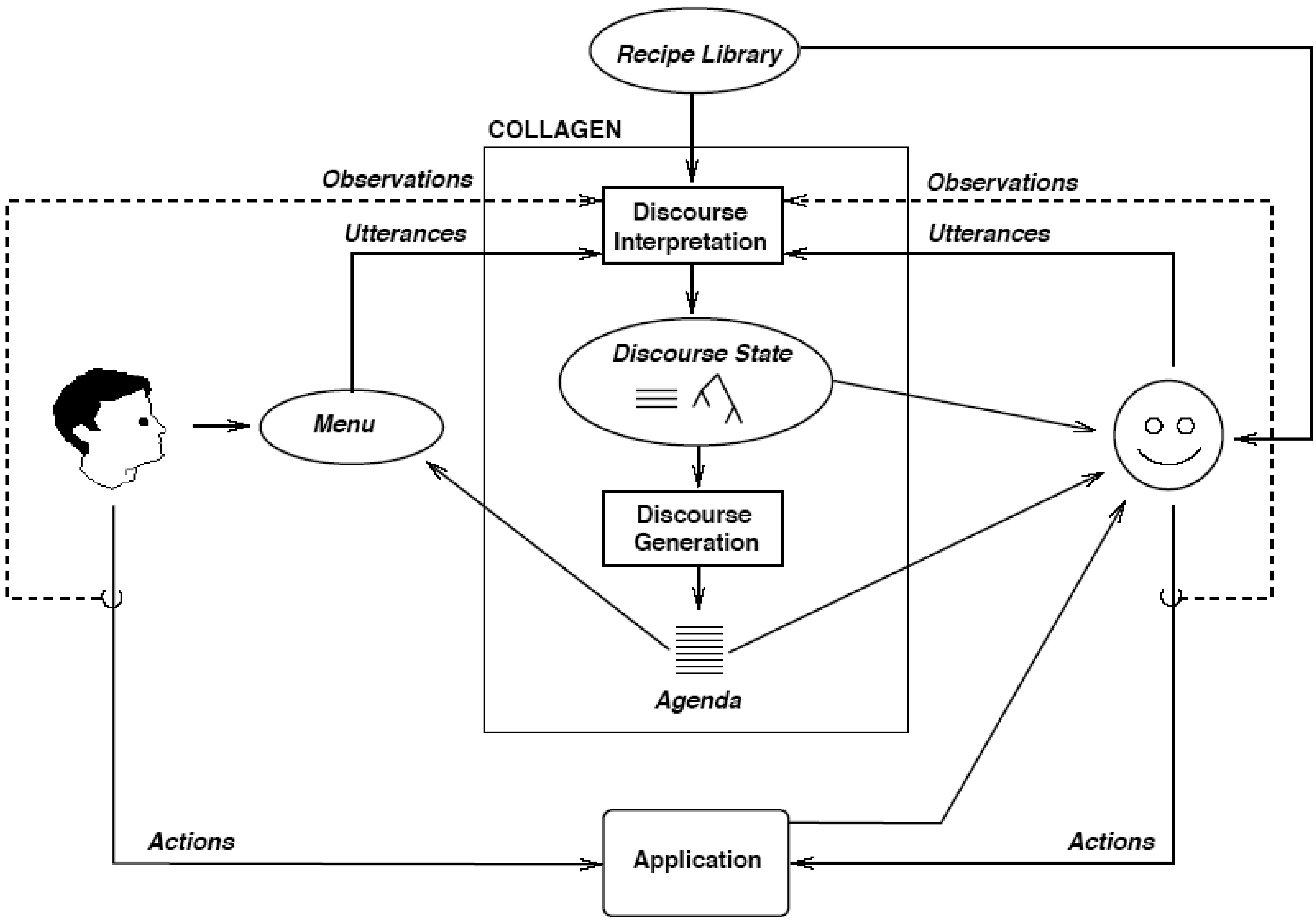


1 Agent says "Let's set the temperature in a room."
2 User selects the bathroom.
3 User says "What next?"
4 Agent says "What should the temperature setting be?"
5 User says "18 degrees Centigrade."
6 Agent sets the temperature in the bedroom to 18 degrees.

Collagen System components

- User
- Application
- Agent
- Collagen “middleware”
- Plan library
- Speech and Natural Language interface

Collagen System Architecture



Collagen: Discourse Structure

- Intentional Structure: Plan Tree
- Linguistic Structure: Hierarchical segments
  - Linear history of interactions
  - History list of closed segments
- Attentional Structure: focus stack

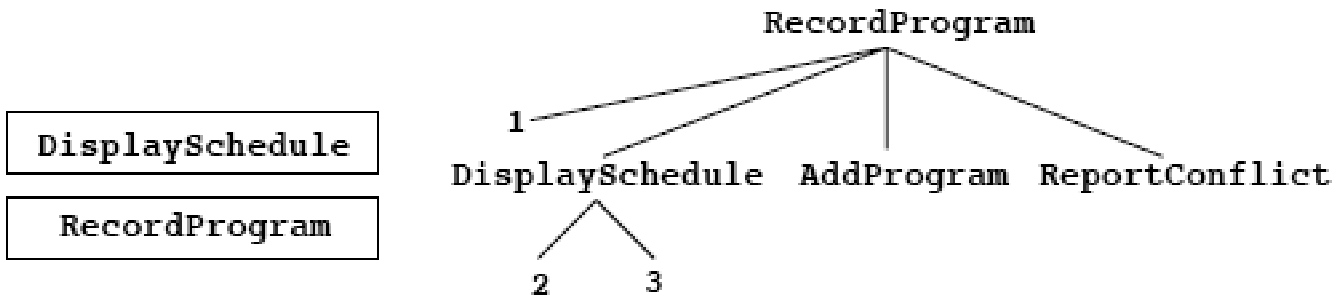
SharedPlans (Grosz and Sidner 1990)

- Shared plan requires
  - Common goal
  - Agreed recipe to accomplish the goal
  - Each can perform her actions
  - Each intend her actions
  - Committed to overall success
- Partial Shared Plan:
  - Some of the above requirements missing, but working on filling them out
- (partial) Shared Plans are composed of other (partial) shared plans

Discourse Segments (G&S 1986)

- Segment is contiguous sequence of communication serving the same purpose
- Segments have hierarchical structure
- Phenomena related to segments
  - Reference resolution
  - Cue words & tense
  - Initiative
  - prosody

Collagen: Example Discourse structure



Focus Stack ***Plan Tree***

Scheduling a program to be recorded.

1 User says "I want to record a program."

Done successfully displaying the recording schedule.

2 Agent displays recording schedule.

3 Agent says "Here is the recording schedule."

4 User says "Ok."

Done identifying the program to be recorded.

5 Agent says "What is the program to be recorded?"

6 User says "Record 'The X-Files'."

Next expecting to add a program to the recording schedule.

Expecting optionally to say there is a conflict.

Collagen: Discourse Interpretation

- Based on Lochbaum's Dissertation work
- Each discourse event is either
  - Starting a new segment (contributing to current purpose) (push)
  - Continuing the current segment (contributing to current purpose) (no-op)
  - Completing the current purpose (pop)
  - Unrelated to current purpose (interruption) (push)
- An act or utterance contributes to a purpose if:
  1. Directly achieves the purpose (goal)
  2. Is a step in a recipe for achieving the purpose
  3. Identifies the recipe to be used
  4. Identifies the actor of the step or recipe
  5. Identifies a parameter of the purpose or step

Sadek et al 96

- France Telecom Research (formerly CNET)
- AGS demonstrator (built using Artemis Agent technology)
  - Spoken telephone weather servers and job info
  - Rational Unit
  - NL Input (non-logical)
    - Island parsing
    - Semantic completion
  - NL Generation
    - Surface speech acts
    - Referring acts
  - Constraint relaxation engine (approximate database match)

Sadek: Dialogue Requirements

- Negotiation ability
  - Underspecified requests
  - Clarification on constraints to zoom in on answer set
- Contextual interpretation
  - Ellipsis
  - Anaphora
  - Deixis
- Mixed Initiative
  - Flexible interaction patterns
- Cooperative reactions
  - Information desired rather than literal meaning
  - Extra information (to help the user's goals0
  - Corrections (to implicatures)
  - Abstractions (intensional answers)

Sadek: Approach

- Rational Balance
  - Basic attitudes
    - Formal definitions
  - Rationality principles
- Communication is special case of rational action

Sadek: Rationality Principles

- Acts
  - Feasibility preconditions (FPs)
  - Rational Effects (RE) [intended perlocutionary effect]
- Principles
  - I(RE) -> I(plan)
  - I(Plan) -> B(FP) || I(FP)
  - Consistency of beliefs: B(a) -> -B(-a)
  - Purpose for intention: I(a) -> B(-a)

Sadek: Cooperation

- Recognizing plan of other
- Intention adoption principle
- Cooperative operations
  - Find reasons for failure of request
  - Compute a solution to a similar request
  - Find information to add
  - Find information to negotiate (when answer set too large)

Sadek: example Speech Acts

$< i, INFORM(j, \phi) >$

FP: $B(i, \phi) \wedge \neg B(i, B(j, \phi))$

PE: $B(j, \phi)$

$< i, INFORMIF(j, \phi) >$

FP: $Bif(i, \phi) \wedge \neg B(i, Bif(j, \phi))$

PE: $Bif(j, \phi)$

TRAINS Project (1990-1994)

- Platform for integrated research on
  - Natural language dialogue
  - Mixed-initiative planning

Trains Example Dialogue

```
Manager: We have to ship a boxcar of oranges to Bath by 8 AM          (1.1)
         and it is now midnight                                        (1.2)
Assistant: Okay                                                        (2.1)
Manager: Okay                                                          (3.1)
         Um ... all right                                              (3.2)
         So, there are two boxcars at Bath and one at Dansville and ... (3.3)
Assistant: and there's                                                 (4.1)
Manager: Wait                                                          (5.1)
         I've forgotten where the oranges are                          (5.2)
         Where are the oranges                                         (5.3)
Assistant: The oranges are in the warehouse at Corning                 (6.1)
Manager: Okay                                                          (7.1)
         So we need to get a boxcar to Corning                         (7.2)
Assistant: Right                                                       (8.1)
Manager: Alright                                                       (9.1)
         So, why don't we take one of the ones from Bath               (9.2)
Assistant: Okay                                                        (10.1)
Manager: So                                                            (11.1)
Assistant: We need                                                     (12.1)
           Okay                                                        (12.2)
           Which engine do you want to bring it                        (12.3)
Manager: Oh. Um. How about                                             (13.1)
         Well let's see                                                (13.2)
         what's shorter                                                (13.3)
         the distance between Avon and                                 (13.4)
         Avon and um Bath or                                           (13.5)
         Elmira                                                        (13.6)
         It looks like it's shorter from Elmira to Corning             (13.7)
         so why don't you send E2                                      (13.8)
         to Corning                                                    (13.9)
Assistant: Okay
```

(non-communicating)
Deliberative Agent

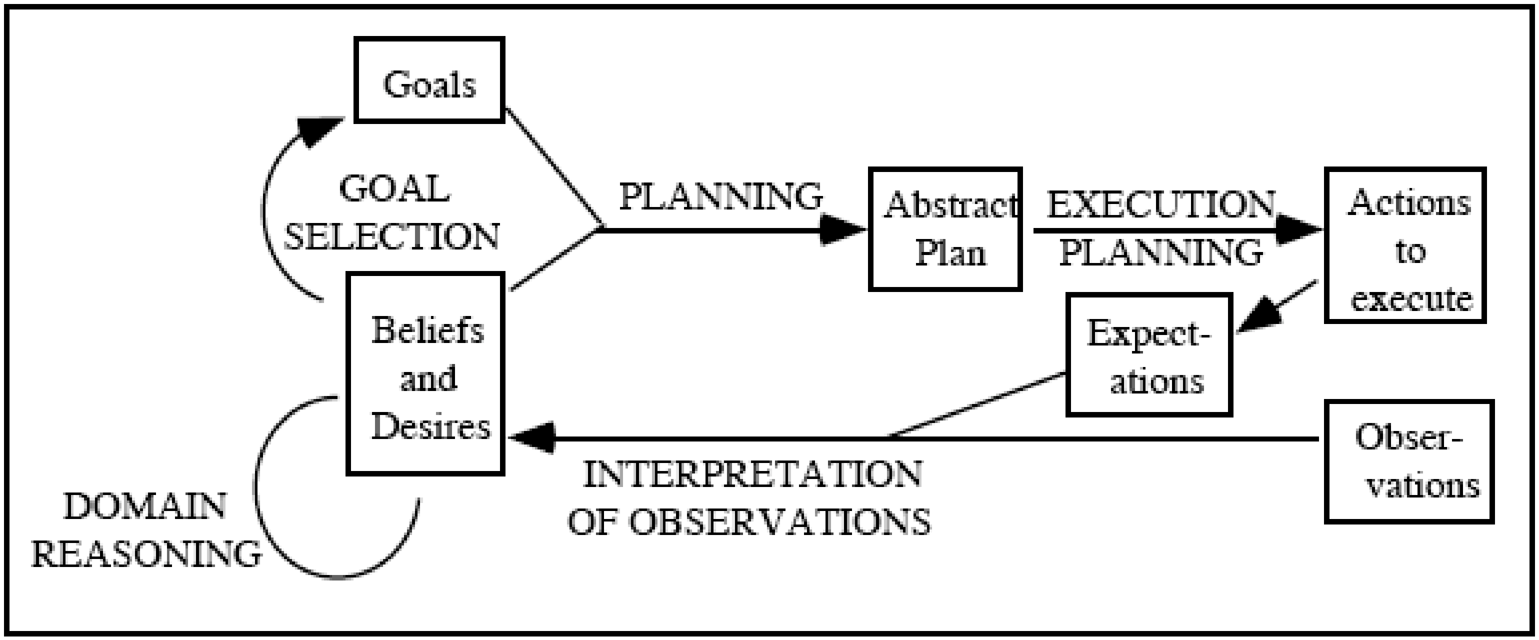


Figure 3: A model of a simple deliberative agent

Trains-95 Agent

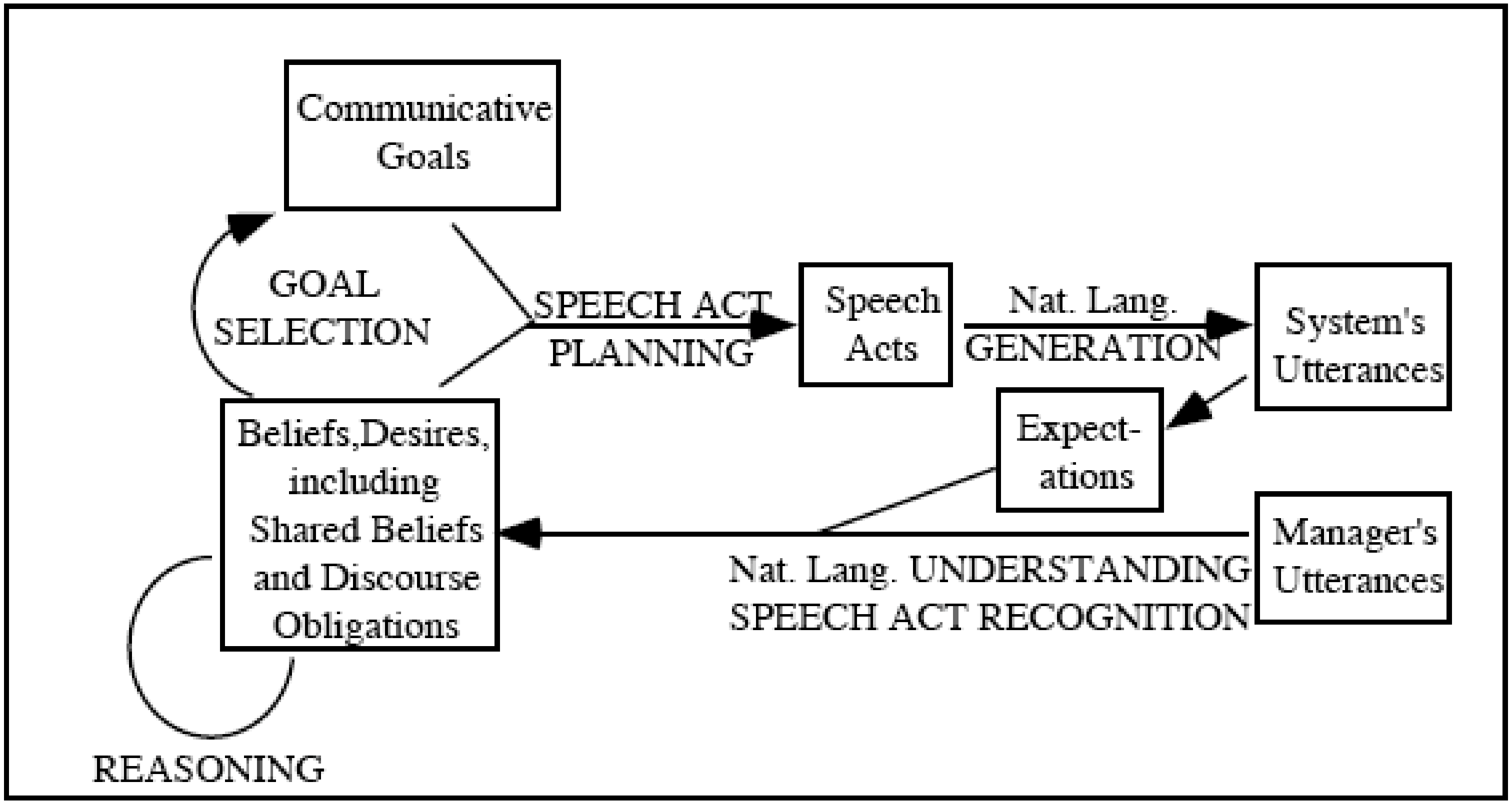


Figure 4: A model of TRAINS as a conversational agent

TRAINS-90-91 System Dialogue

1.1 M: We have to make OJ

1.2 M There are oranges at I

1.3 M and an OJ Factory at B.

1.4 M: Engine E3 is scheduled to arrive at I at 3PM

1.5 M: Shall we ship the oranges?

2.1 S: Yes

2.2 S: Shall I start loading the oranges in the empty car at I?

3.1 M: Yes,

3.2 M: and we'll have E3 pick it up.

3.3 M: Ok?

4.1 s: OK

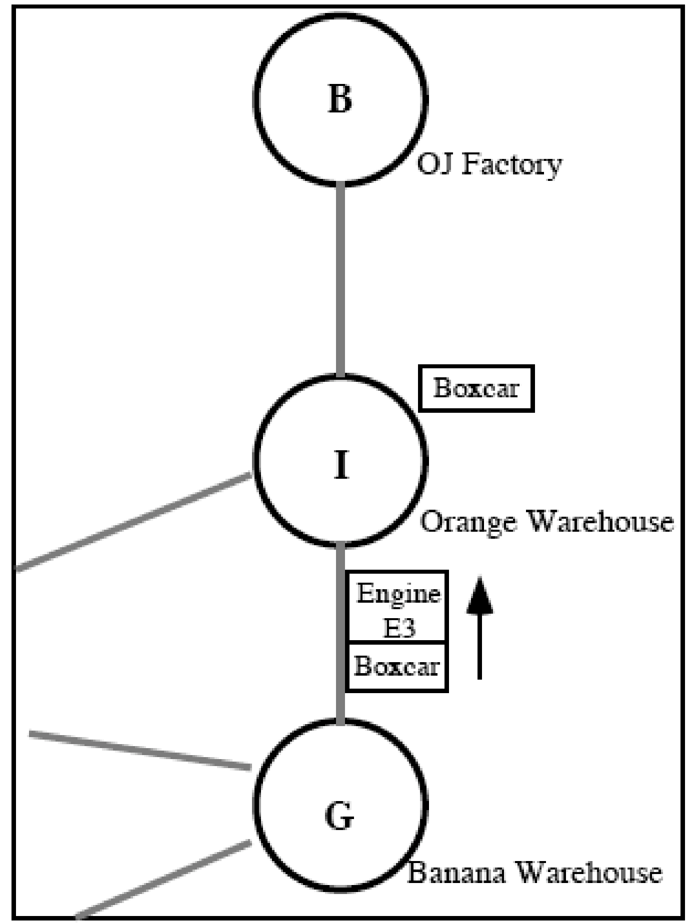


Figure 8: Part of the map for the TRAINS-91 Dialogue

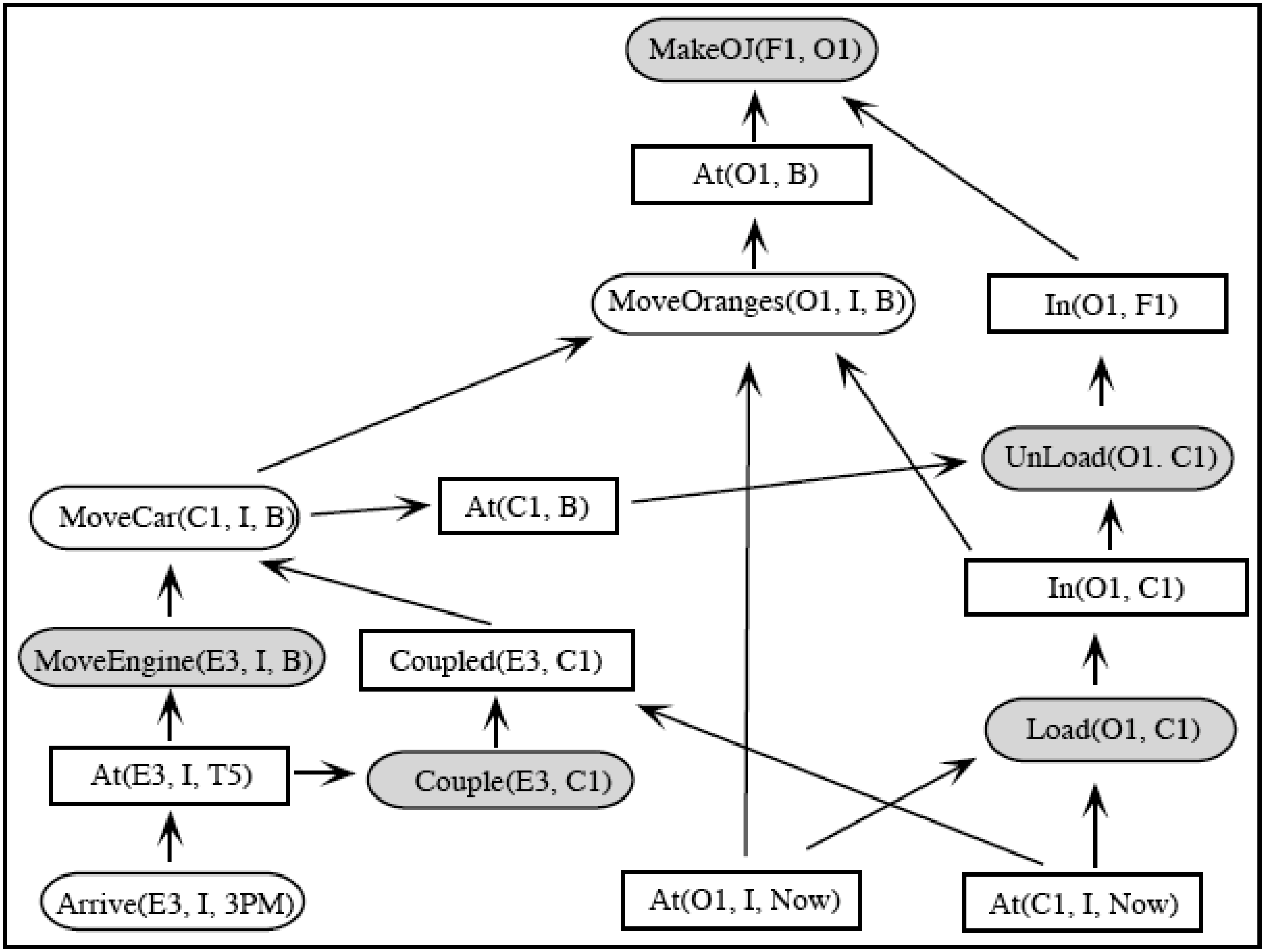


Figure 9: The plan constructed for the TRAINS-91 dialogue

```
Manager: We better ship a boxcar of oranges to Bath by eight a.m.         (1.1)
System: OK                                                                (2.1)
Manager: So we need to get a boxcar to Corning, where there are oranges. (3.1)
         There are oranges at Corning.                                    (3.2)
         Right?                                                           (3.3)
System: Right                                                             (4.1)
Manager: So we need an engine to move the boxcar.                         (5.1)
         Right?                                                           (5.2)
System: Right                                                             (6.1)
Manager: So there's an engine at Avon.                                    (7.1)
         Right?                                                           (7.2)
System: Right                                                             (8.1)
Manager: So we should move the engine at Avon, engine E1, to Dansville
         to pick up the boxcar there                                      (9.1)
System: Okay                                                              (10.1)
Manager: and move it from Dansville to Corning                            (11.1)
         load up some oranges in the boxcar                               (11.2)
         and then move it on to Bath                                      (11.3)
System: Okay                                                              (12.1)
Manager: How does that sound?                                             (13.1)
System: That's no problem                                                 (14.1)
Manager: Good                                                             (15.1)
```

Figure 10: The TRAINS-93 demonstration dialogue

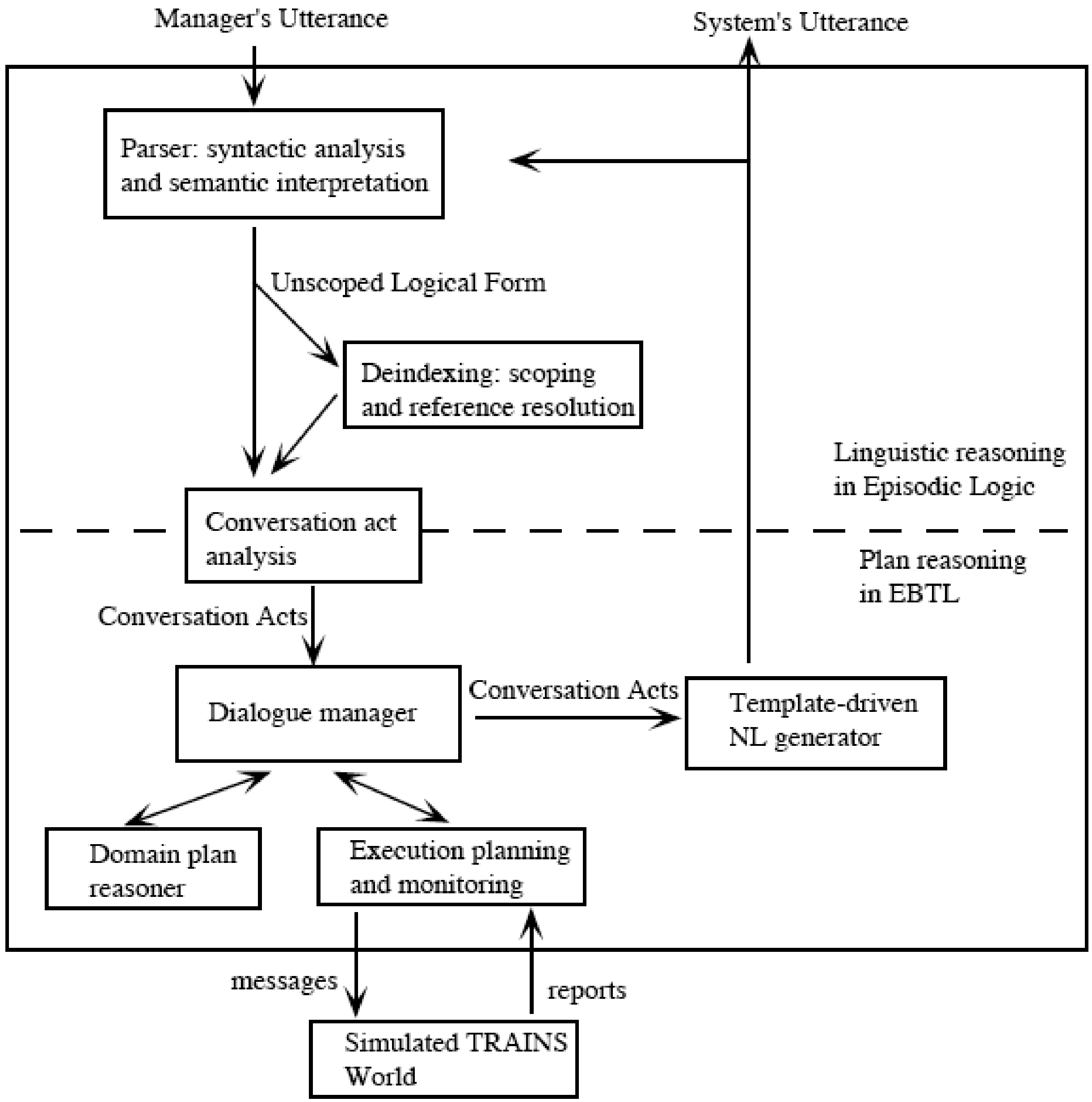


Figure 11: The TRAINS-93 System Architecture

Traum and Hinkelman: Conversation Acts

Level	Act Type	Sample Acts
<UU	**Turn-taking**	take-turn keep-turn
UU	**Grounding**	Initiate Repair Ack Continue
DU	**Core Speech Acts**	Inform YNQ Accept Request
>DU	**Argumentation**	Elaborate Q&A

Cohen Perrault Request Plan

$$
\begin{tikzcd}
\alpha\,(\text{JOHN}) \arrow[d, "\text{want.pr}"] \\
\text{JOHN WANT} \quad \alpha\,(\text{JOHN}) \\
\text{CAUSE-TO-WANT(S,JOHN,} \ \alpha\,(\text{JOHN})) \arrow[u, "\text{effect}"'] \arrow[d, "\text{cando.pr}"] \\
\text{JOHN BELIEVE} \ \ \text{S WANT} \ \alpha\,(\text{JOHN}) \\
\text{REQUEST(S,JOHN,} \ \alpha\,(\text{JOHN})) \arrow[u, "\text{effect}"']
\end{tikzcd}
$$

Traum Allen: Request Plan

$$\begin{tikzcd}
\alpha\,(\text{JOHN}) \\
\text{JOHN INTEND } \alpha\,(\text{JOHN}) \arrow[u] \\
\text{OBLIGED(JOHN, S, ADDRESS REQUEST(...))} \arrow[u, "\text{Deliberation}"'] \\
\text{REQUEST(S,JOHN, } \alpha\,(\text{JOHN})) \arrow[u, "\text{effect}"']
\end{tikzcd}$$

Trains-93 Obligation Rules (Traum & Allen 94)

source of obligation	**obliged action**
S_1 Accept or Promise A	S_1 achieve A
S_1 Request A	S_2 address Request: accept **or** reject A
S_1 YNQ whether P	S_2 Answer-if P
S_1 WHQ P(x)	S_2 Inform-ref x
utterance not understood or incorrect	repair utterance
S_1 Initiate DU	S_2 acknowledge DU
Request Repair of P	Repair P
Request Acknowledgement of P	acknowledge P

Trains-93 Illocutionary Acts

T-INFORM	The speaker aims to establish a shared belief in the proposition asserted
T-YNQ	The speaker asks a yes-no question, creating an obligation for the hearer to respond
T-CHECK	The speaker is verifying that a certain proposition is true (that the speaker already suspects is true)
T-SUGGEST	The speaker proposes a new item (action, proposition) as part of the plan
T-REQUEST	The speaker aims to get the hearer to perform some action. In the TRAINS domain, this is treated like a suggest, with the addition of an obligation on the hearer to respond.
T-ACCEPT	The speaker agrees to a prior proposal by the hearer.
T-REJECT	The speaker rejects a prior proposal by the hearer.
T-SUPP-INF	The speaker provides additional information that augments, or helps the hearer interpret some other accompanying speech act.

Interpretation

- E.g: So we need an engine to move the boxcar
- EL:

```
(DECL
  (UTT-IMP SO-COORD
    (WE1
      ((ADV-A (IN-DISCOURSE-RELATION
                (TO1 (MOVE <THE BOXCAR>))))
       (NEED-REQUIRE <A ENGINE>))))))
```

- Speech Act Hypotheses:

1. An inform of a need for an engine to move the boxcar
2. A check whether there is a need for an engine to move the boxcar
3. A question whether there is a need for an engine to move the boxcar
4. A suggestion that an engine be used in the plan, with a supplementary suggestion of moving the boxcar.

Trains-93 Belief spaces: dynamics

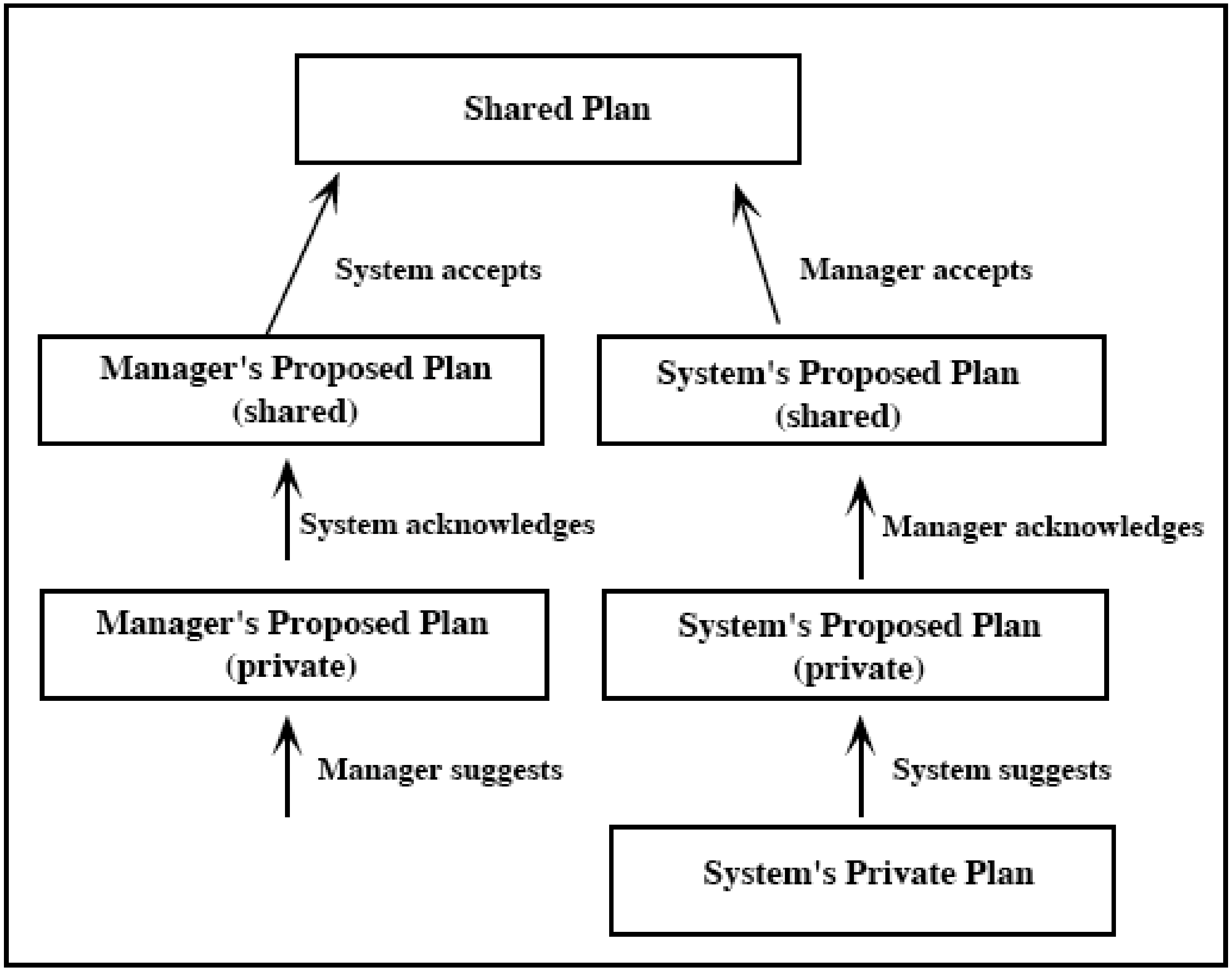
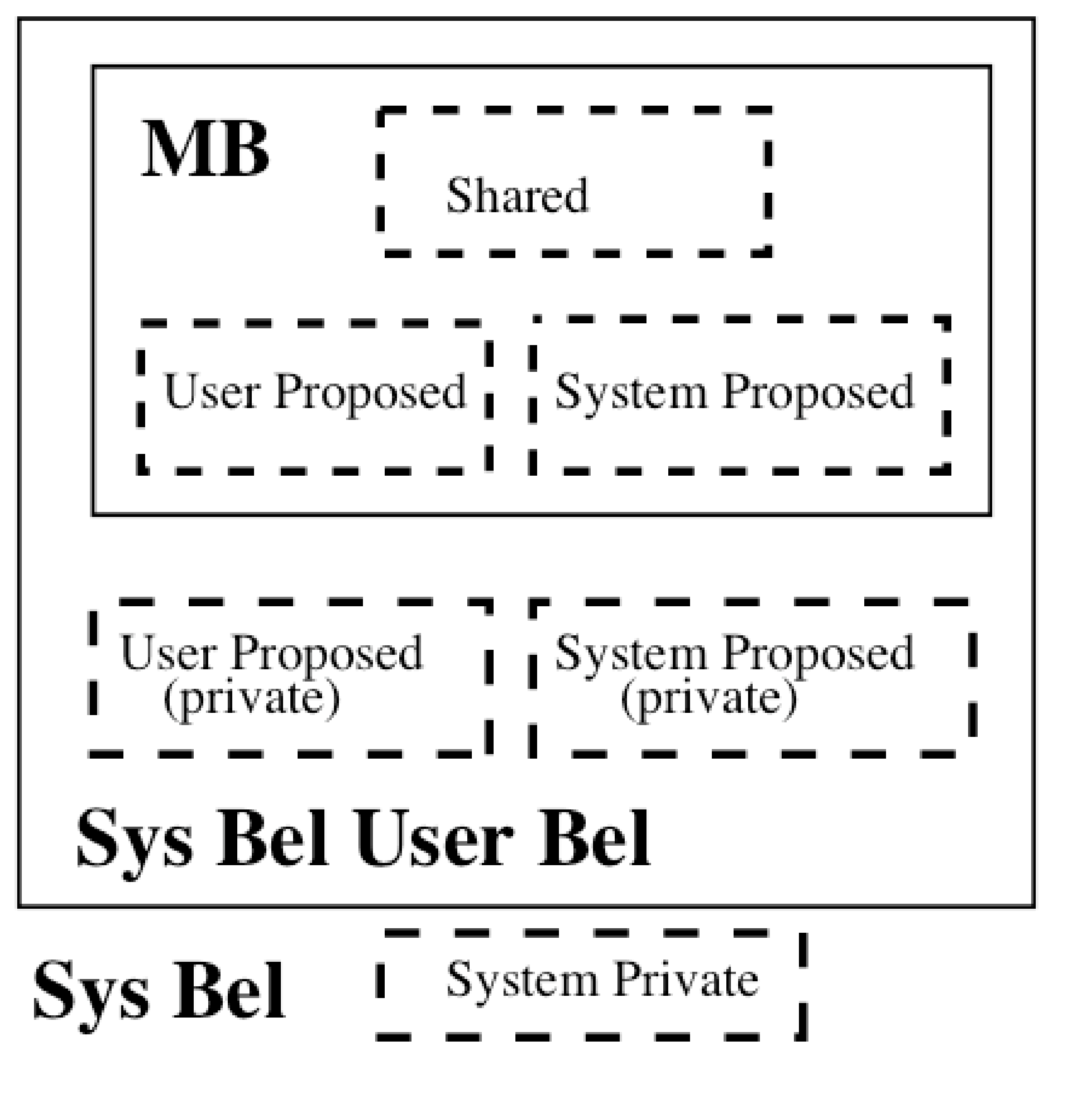


Figure 5: The different plan modalities in handling suggestions

Trains-93 Belief spaces: beliefs and plans



Trains-93 DM algorithm

```
while conversation not finished
    if system has obligations
        address obligations
    else if system has turn
        if system has intended conversation acts
            call generator to produce NL utterances
        else if some DU is ungrounded
            address grounding situation
        else if some proposal is not accepted
            consider proposals
        else if some discourse goals are unsatisfied
            address discourse goals
        else release turn or attempt to end conversation
    else if no one has turn
        take turn
    else if long pause
        take turn
    else wait for user
```