Memory Based Reasoning

An Introduction

Deepak Khemani

Department of Computer Science & Engineering
IIT Madras

Email : khemani@iitm.ac.in

Knowledge and Memory

Knowledge	To know. Humans deal with knowledge of many kinds. We have models of the world we live in. We have models of ourselves in the world. We have knowledge of society, knowledge of facts, knowledge of how to do things etc…
Ontology	Knowledge of being. How the world is. Of concepts and the relations between them.
Epistemology	Knowledge about what is true in the world.
Knowledge Based Systems	Usually refers to systems that employ domain specific problem solving knowledge in some form.
Memory	Our repository of knowledge. In human beings memory is dynamic. We continuously learn.

Memory Based Agents

Semantic memory

Ontological knowledge in the agents repository. Organized as an *inheritance hierarchy.* The leaves are populated by facts (epistemological).

Episodic memory

Events organized in an *aggregation hierarchy.* Episodes may be actual, or an ontology of event patterns may exist (abstraction hierarchy).

Experience — The main source populating the agent's memory.

Memory Based Systems

Systems that rely on experience and knowledge. Key issues are organization, indexing, retrieval, and operationalization of the agent's knowledge.

Knowledge Based Agents

Intelligent agents acquire knowledge through various sources.

- learning from own experience
- learning from other's experience
  - via teaching
  - via media
  - via stories . . .

Knowledge structures

Refer to the form and content of knowledge in an agent's memory.

Learning

Given a finite, albeit large, memory store how how does an agent acquire new concepts and continuously improve upon problem solving skills?

Intelligent activity

There are two main kinds of intelligent activity.

Planning	Selecting and structuring actions in order to achieve a desired goal. The term *problem solving* usually refers to planning.
Understanding	Making sense of information acquired through various sense organs of the agent. Understanding is also used to refer to the activity of forming new concepts, and includes reasoning patterns. Understanding can also be posed as problem solving.

Combinatorial Explosion

Both planning and understanding have to contend with a bewildering number of possibilities. The prime adversary of intelligence.

Past vs. Future

There are two main approaches to problem solving.

Exploit experience

A memory based agent looks into the past

Knowledge essentially links the past to the present. The motto is to avoid reinventing the wheel.

The danger : "You never step into the same river twice" – the problem rarely repeats itself exactly.

Use first principles

A trial and error simulation peers into the future.

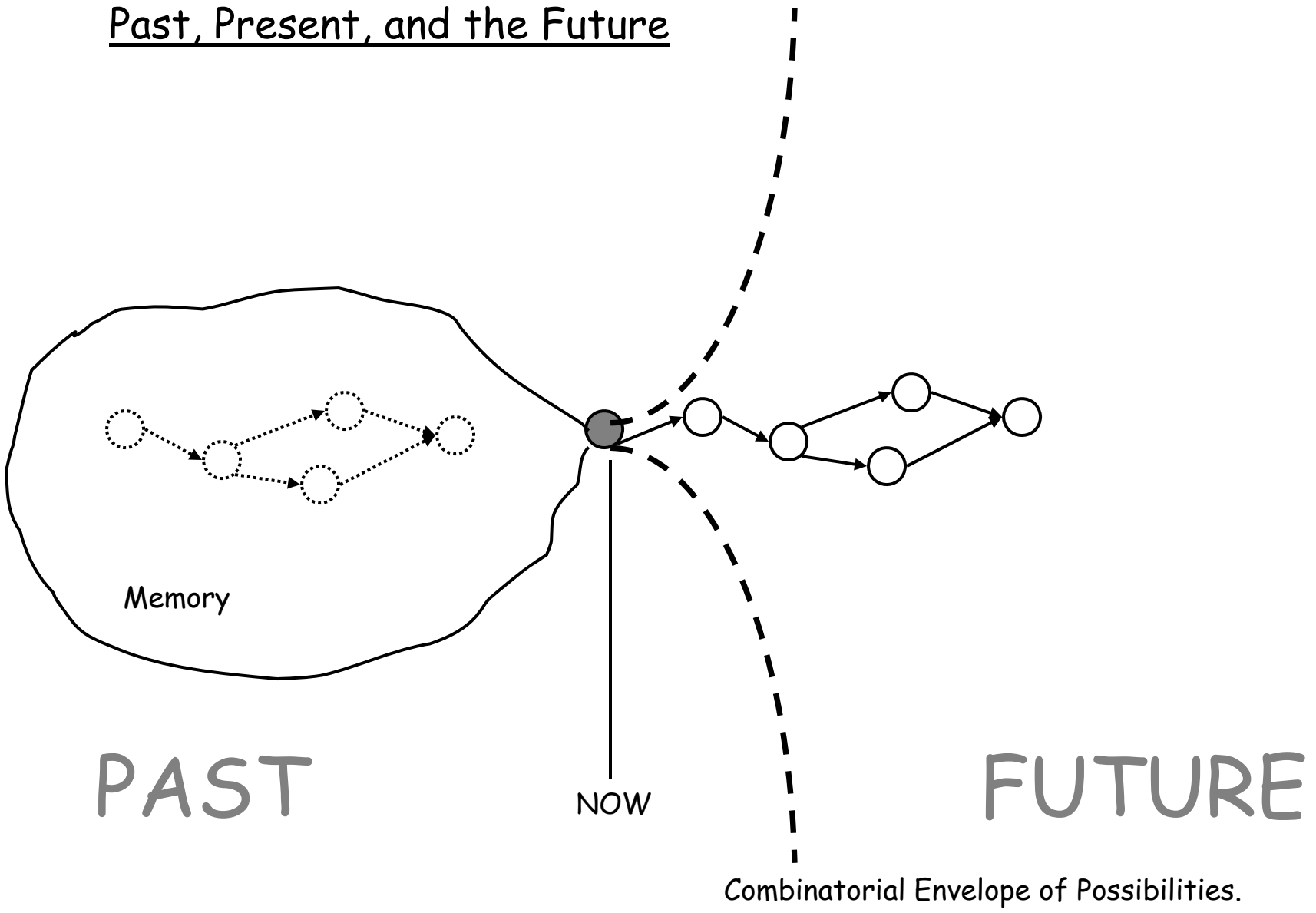
The solution is always for the problem being solved.

Drawbacks: Simulation often requires sophisticated modelling. Also the "the wheel is reinvented again and again".

Search

Both type of agents use search. One searches through possible combinations. The other searches for relevant knowledge.

Past, Present, and the Future



Knowledge = Carry Forward

> Similar problems have similar solutions.
>
> Problems *are* often similar.

Knowledge is the carry forward from the past to the future.

Knowledge based systems assume that "infinite" combinations of states and problems in the world can be categorized into a finitely manageable set.

Knowledge is the response of a resource bounded system functioning in a continuous dynamic world.

Knowledge structures are cognitive models of the world.

In *understanding* they provide the nets in which incoming information is mapped.

In *planning* they link action sequences to their effects.

Key issues
- reminding
- aggregation

Experience is stored in chunks.

Cases in Everyday Discourse

T V R Shenoy (political comment post Kargil) …permit me to quote what I wrote in this column as far back as June:

''As President Kennedy ruminated in the wake of the Bay of Pigs debacle, 'Victory has a thousand fathers, defeat is an orphan.' Very true, except That the Pakistan Army is trying to prove that the father of the 'orphan' – the Kargil crisis -- is actually Nawaz Sharief.'' …*reminding*

To which I added:

''The soldiers have been out for his head ever since last October when he forced General Jahangir Karamat out of office, and installed Pervez Musharraf as army chief of staff over the heads of six senior commanders. Students of history should note that the last Pakistani prime minister to do such a thing was Zulfiqar Ali Bhutto. He hand-picked Zia ul-Haq because he thought the man would be pliable. And everyone knows just how the protege proved his gratitude. So it was a bad omen when Nawaz Sharief too tried his hand at the fine art of supersession.''

… *case based reasoning*

Early knowledge structures

The need to structure knowledge came out of the requirements of natural language understanding. These were static structures whose slots were populated by tokens generated through parsing of stories.

Scripts	Scripts are patterns of stereotypical activity in an identifiable situation. The Script describes the expected events in the situation.
Goals	Agents in stories have goals that they strive to achieve.
Plans	The means to achieve goals are plans.
Actions	(Specific) plans result in (specific) actions. Stories refer to goals, actions and plans.
MOPs	Memory Organization Packets sought to integrate knowledge structures of different forms and levels.
TOPs	Thematic Organization Packets were attempts to capture situation independent knowledge.

MOPs

MOPs endeavor to weave together different forms and levels of episodic knowledge structures via various hierarchies.

Abstraction	An episode can be described at different levels of abstraction. For example "My trip to Palghat last week" → "Work trips to Palghat" → "Consultancy related travel".
	Operationally abstraction hierarchies are inheritance hierarchies.
Aggregation	Episodes can be described at various levels of granularity, right down to the level of individual actions.
	Chunking facilitates indexing.
Indexing	Which structure to retrieve when?
Reminding	Indexing looks at structures that facilitate retrieval. Human memory works in more interesting ways, establishing cross contextual linkages that are often useful.

Case Based Reasoning

The difficult issue is *Learning!*

Given that a *dynamic memory* is crucial to the success (and survival) of an agent in a resource limited competitive environment, how does one design and build MOPs and other forms of knowledge (for example *rules)* that are enriched by experience?

Note : Artificial Neural Networks attempt to mimic the human brain as we know it now. The aim is to acquire the right connectivity via training using a large training set. Its not clear however whether this approach will yield an *understanding* of how we use knowledge.

Case Based Reasoning (CBR) focuses on learning through a collection of experiences.

In its simplest form CBR simply uses a collection of episodes.

Provides a model for all three forms of intelligent activity

- problem solving, understanding, and learning.

CBR: simple principles

If it worked, use it again.

If it worked, do not worry about it. Corollary

If it did not work, remember not to do it again.

If it does not work, fix it

Rather than planning again, reuse plans.

Rather than projecting actions into the future, recall what they did in the past.

Assumption :	Patching a retrieved (long) plan is simpler.
Requirement:	Case library has to be dense.
Analogy:	Calculation of logarithm. Difficult to compute log(x) from scratch.

Log tables = case library. Interpolation = adaptation.

Problems Faced by a Planner

- The immediate complexity problem.
  - *Projecting forward for a safe and correct plan can be NP-hard to undecidable.*
- The asymptotic complexity problem.
  - *With a stream of conjunctive goals is the total cost of planning larger or smaller than the sum of individual planning costs?*
- The execution time failure problem.
  - *Conditions may change or assumptions may have been incorrect.*
    - Replanning, recovery, repair.
- The planning / execution crowding problem.
  - *Critical in real-time situations.*
- The costly information problem.
  - *Understanding the situation requires computational and other resources.*
- The missed opportunities problem.
  - *Changing unnoticed conditions. Lack of understanding may lead to increasing asymptotic complexity, or even execution time failure.*

Traditional Planning : assumptions

- The world will be stable; it will behave as projected.
- Time consumed in planning is independent of time available for execution.
  - *Efficiency of planner has no side effect in feasibility of the constructed plan.*
- The information available to the planner is complete, and execution will be flawless.
- Any initially correct plan will remain correct and can in fact be carried out.

In reality

- The world is not stable.
- Agents must trade off planning time against execution time
- Information is rarely complete.

Case Based Planning : assumptions

1. We do not have a closed world.
2. We have an incomplete and imperfect model of the world.
3. We are not producing analyses or proofs of soundness. We are producing plans. (Forced by 2)
4. We cannot do projection. That is, we cannot run simulations of plans to tease out problems due to step interactions. (Also forced by 2).

In many situations these assumptions are thrust upon us by the real world.

We still have to produce plans

- without projection

- without the pre and post condition information traditionally needed to string together operators

- without perfect knowledge of operators.

Even in complete information situations sometimes human prefer familiar plans.

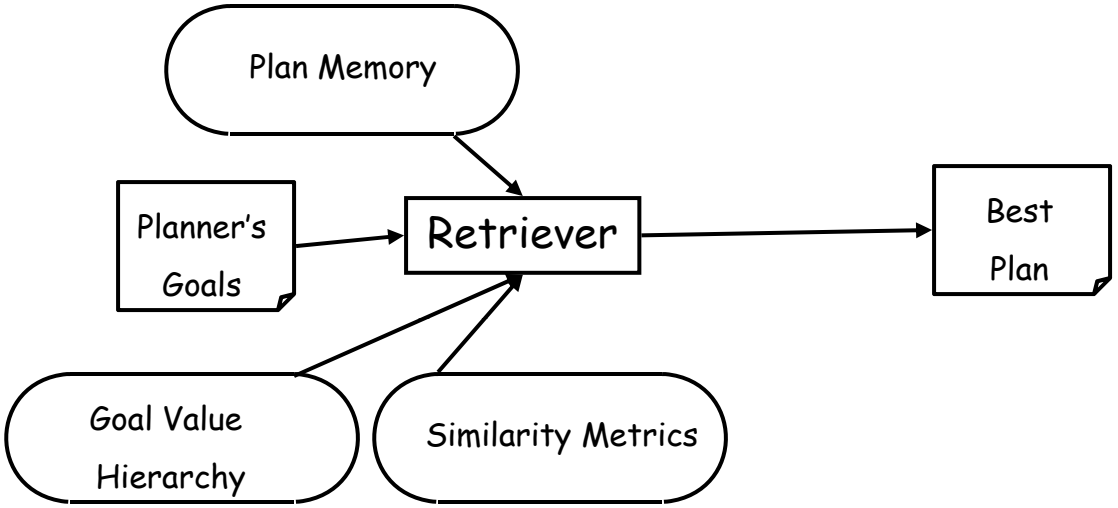
Case Based Planning : strategy

- New plans are built from old plans.
  - *Internal effects of the plans are ignored.*
- Projection is replaced with *anticipation.*
  - *Experience rather than simulation.*
- Plans are selected on the basis of –
  - The goals that they satisfy
  - The problems that they avoid
  - The features of the world that they have been associated with in the past.
- Knowledge of individual operators is not used in planning itself.
  - *Used instead to explain a failed plan for repair, recovery, and indexing.*
- Conjunctive goals are handled by modifying best fit plan.
  - *Can lead to faulty plan. Plans can be repaired and stored.*

CHEF: A Case Based Planner (Kristian Hammond –1989)

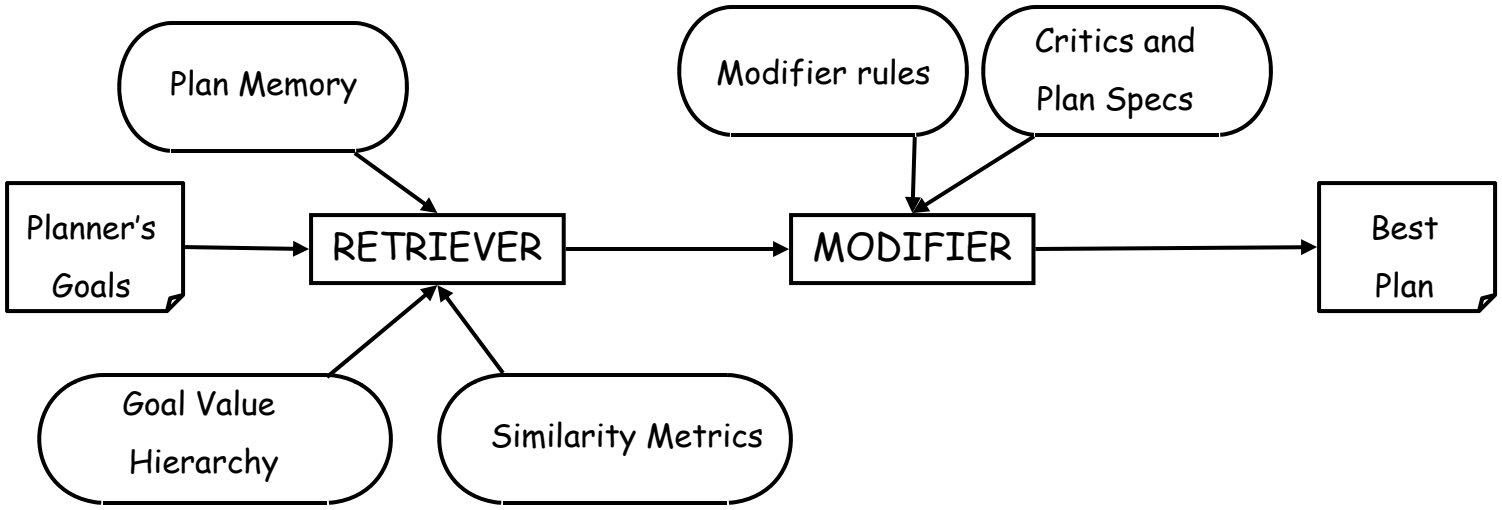
- Creates stir fried recipes.
- Uses Thematic Organization Packets (TOPs) to store rules for adaptation.

CHEF: The Retriever



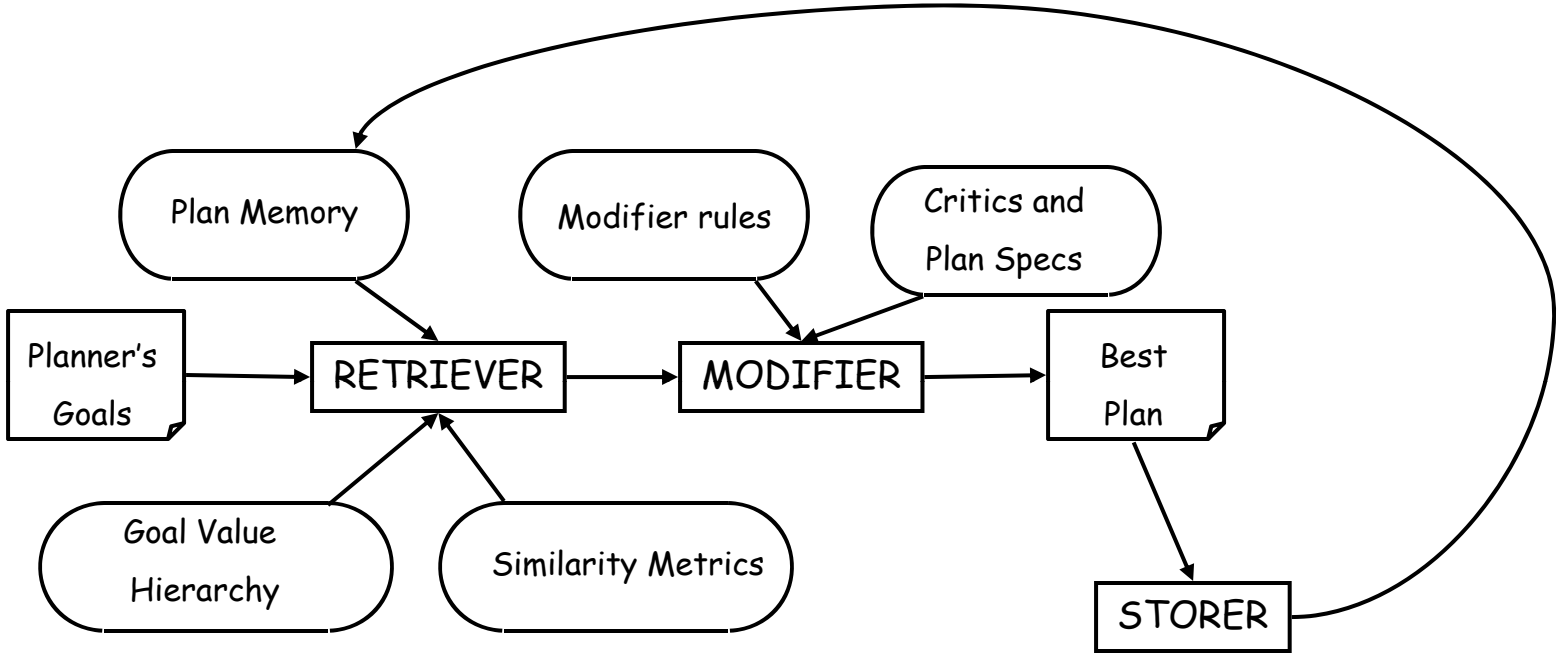
The Retriever uses the input goals to discriminate through the memory, and uses the relative value of different goals to decide between the competing plans.

CHEF: The Modifier



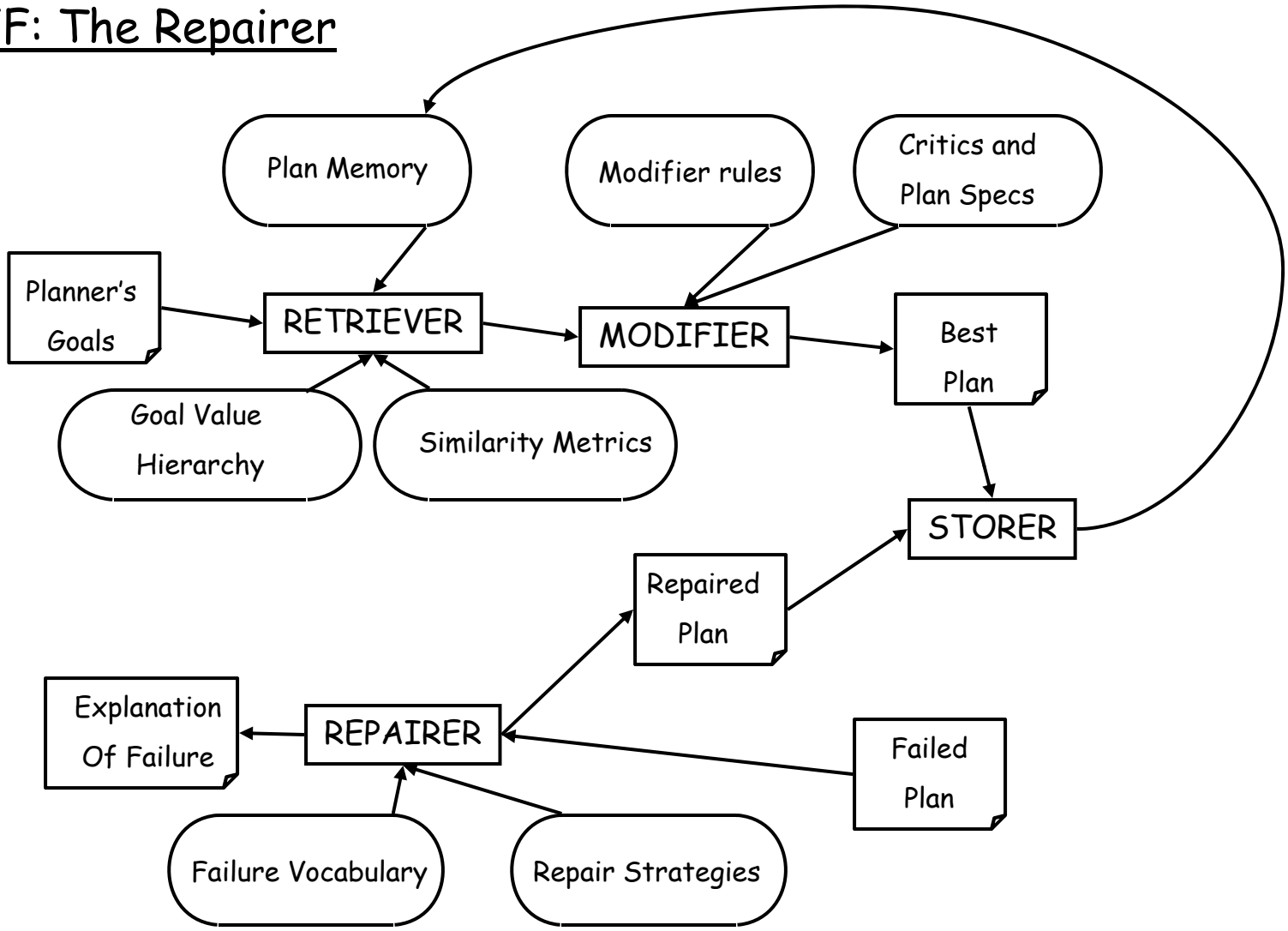
The Modifier adapts the retrieved plan to suit the planner's goals.

CHEF: The Storer



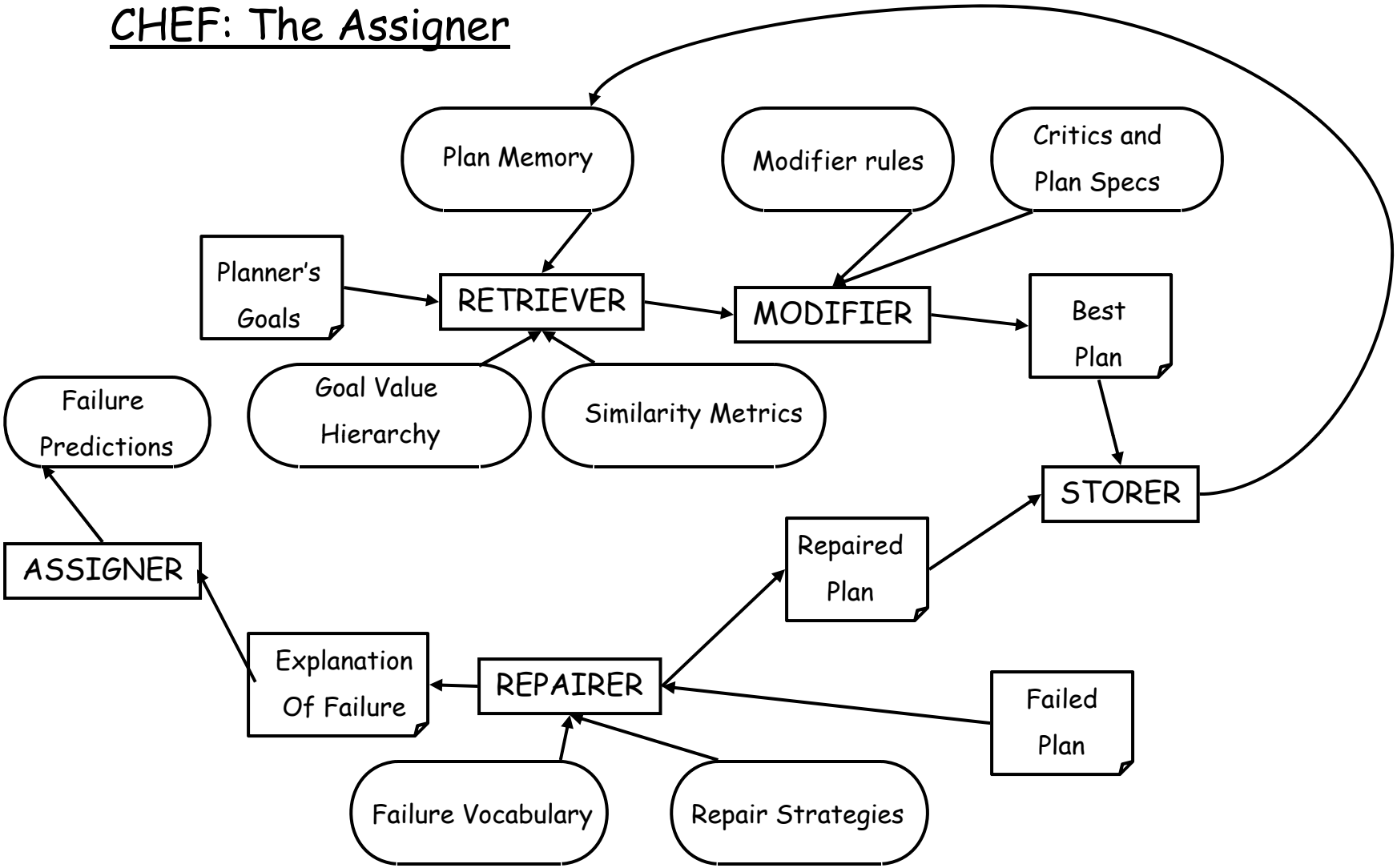
The Storer facilitates learning of new plans. It indexes the new plan under the same features that the Retriever uses to fetch them.

CHEF: The Repairer



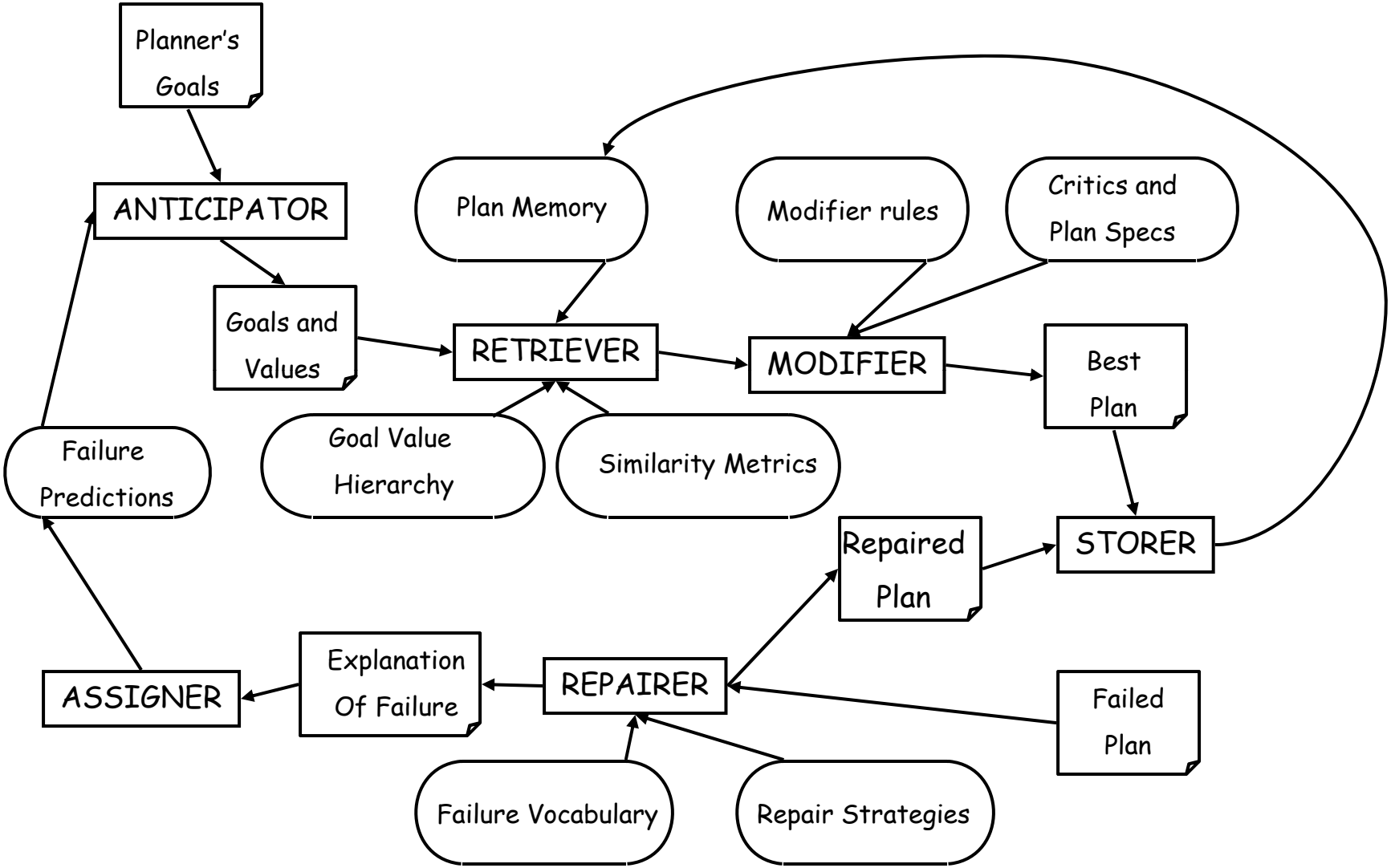
With repaired plans the Storer can index the failures that the repaired plan avoids as well.

CHEF: The Assigner



The Assigner assigns the causes in the situation that led to the failure.

CHEF: The Anticipator



The Anticipator predicts problems *before* and planning has been done.

Review

- **CHEF**: a case-based planner which can output new recipes given particular ingredients and tastes.

- **Thematic Organization Packets (TOPs)**. The idea behind TOPs is that, there are problems that are caused by *interactions* between plans and goals. These pieces of information cannot be tied to any individual goal. TOPs provide a means to store such information and strategies to deal with it.

CHEF: Beef and Broccoli

- It starts by searching for a best-matched plan, which results in a recipe BEEF-WITH-GREEN-BEANS. Goals are partially satisfied.
- The fact that there is a partial match between the target goal "to include broccoli", and an object in the existing recipe, "green beans", tells the planner that it can replace the broccoli for the green beans directly.
- **Object critics** allow it to correctly adjust the cooking time appropriate for broccoli and also suggest that broccoli has to be chopped before being stir-fried.
- All of the modifications are incorporated to produce a new recipe, BEEF-AND-BROCCOLI.
- In addition to the given set of goals, it derives more goals from its knowledge of stir-frying and its understanding of what items are important in this recipe. So it has the understanding that running this plan should result in other six goals, in addition.

The first plan

A half pound of beef, Two tablespoons of soy sauce, One teaspoon of rice wine, A half tablespoon of corn starch, One teaspoon of sugar, A half pound broccoli, One teaspoon of salt, One chunk of garlic

•Chop the garlic into pieces the size of match heads.
•Shred the beef.
•Marinate the beef in the garlic, sugar, corn starch, rice wine and soy sauce.
•Chop the broccoli into pieces the size of chunks.
•Stir fry the spices, rice wine and beef for one minute.
•Add the broccoli to the spices, rice wine and beef.
•Stir fry the spices, rice wine broccoli and beef for three minutes.
•Add the salt to the spices, rice wine, broccoli and beef.

Expected results

The beef is tender.	The dish now tastes salty.
The dish now tastes savory.	The dish now tastes sweet.
The broccoli is now crisp.	The dish now tastes like garlic

Plan Failure

CHEF runs the plan in a simulation and checks the goal. One of the goals, in particular "The broccoli is now crisp", cannot be satisfied.

- The broccoli is instead soggy.
- Not all goals are satisfied.
- Changing name of this bad recipe to BAD-BEEF-AND-BROCCOLI..

Explaining the failure

It explains the failure using a set of inference rules that tell it about the effects of each step in its domain on each object in its domain. These rules are used to chain through the steps and states of the plan to answer a set of questions that will form the explanation of why the failure has occurred.

It finds out that "The broccoli is soggy" is caused by thin liquid in the pan, which is a side effect of stir-frying all the ingredients together.

Thematic Organization Packets (TOPs)

The idea behind TOPs is that, there are problems that are caused by *interactions* between plans and goals. These pieces of information cannot be tied to any individual goal. TOPs provide a means to store such information and strategies to deal with it.

•All the TOPs are associated with causal configurations that lead to failures and strategies for fixing them.

•In this case, the causal description is: The side effect of liquid coming from "the stir-frying of the beef" disables a precondition of "the stir-frying of the broccoli" that the pan being used is dry. This is used to access the TOP SIDE-EFFECT:DISABLED-CONDITION:CONCURRENT out of the twenty TOPs known to the program.

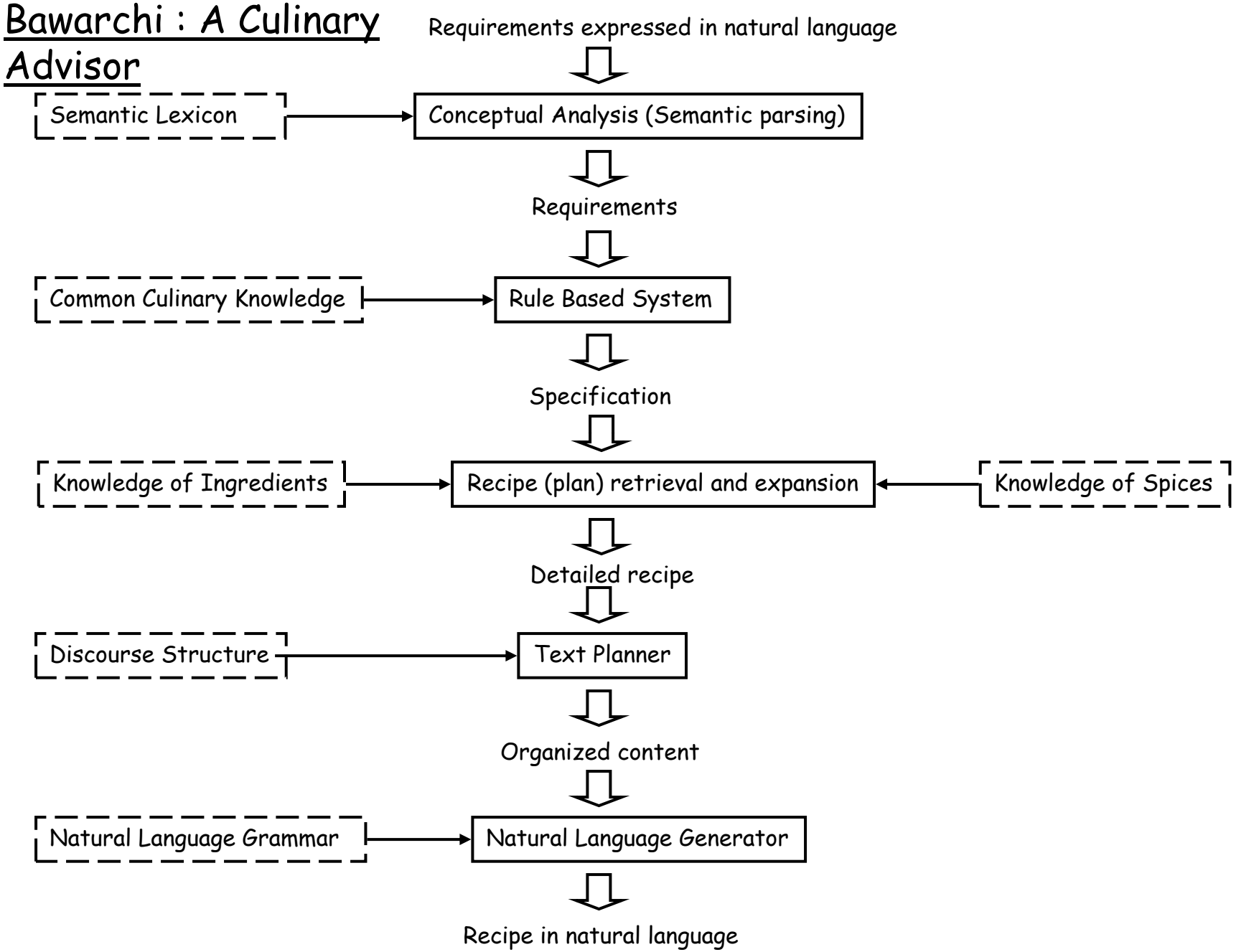
REPAIR

- The mentioned TOP, which relates to the interaction between concurrent plans in which a side effect of one violates a precondition of the other, has three strategies associated with it: SPLIT-AND-REFORM, ALTER-PLAN:SIDE-EFFECT, and ADJUNCT-PLAN.
- CHEF can only find one possible instantiation, a specialization of the strategy SPLIT-AND-REFORM (which suggests breaking the parallel plans into serial steps), so it is applied directly to the problem.

- The new plan is then:

  Stir-fry the broccoli, remove it from the plan, stir-fry everything else, add the broccoli, and stir-fry briefly.

Bawarchi : A Culinary Advisor



Conceptual Analysis

" *The objective of parsing natural language sentences is to extract the meaning*" – Roger Schank

Goal : Identification of relevant semantic representation and instantiation for the current input.

The Lexicon for Bawarchi :

- STYLES (Fry, roast, curried…)
- VEGETABLES (Potato, brinjal…)
- SPICES (Cumin, garlic, pepper…)
- FLAVOURS (Bland, spicy…)
- CONNECTIVES (With, without…)

Example

"Spicy potato with onions having low fat content"

(vegetable potato)

(with-sec-ingr onion)

(flavour spicy)

(with-less-sec-ingr oil)

Common Cooking Knowledge (recipe independent)

Rules of the form:

If you want large fried pieces and the ingredient can be boiled

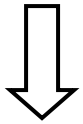
then

pre-boil the ingredient before frying.

Forward chaining inferences – (for example)

(vegetable potato) (with-sec-ingr onion)

(flavour spicy) (with-less-sec-ingr oil)



(vegetable potato)

(style fry) *default*

(small-pieces potato)

(with-sec-ingr onion)

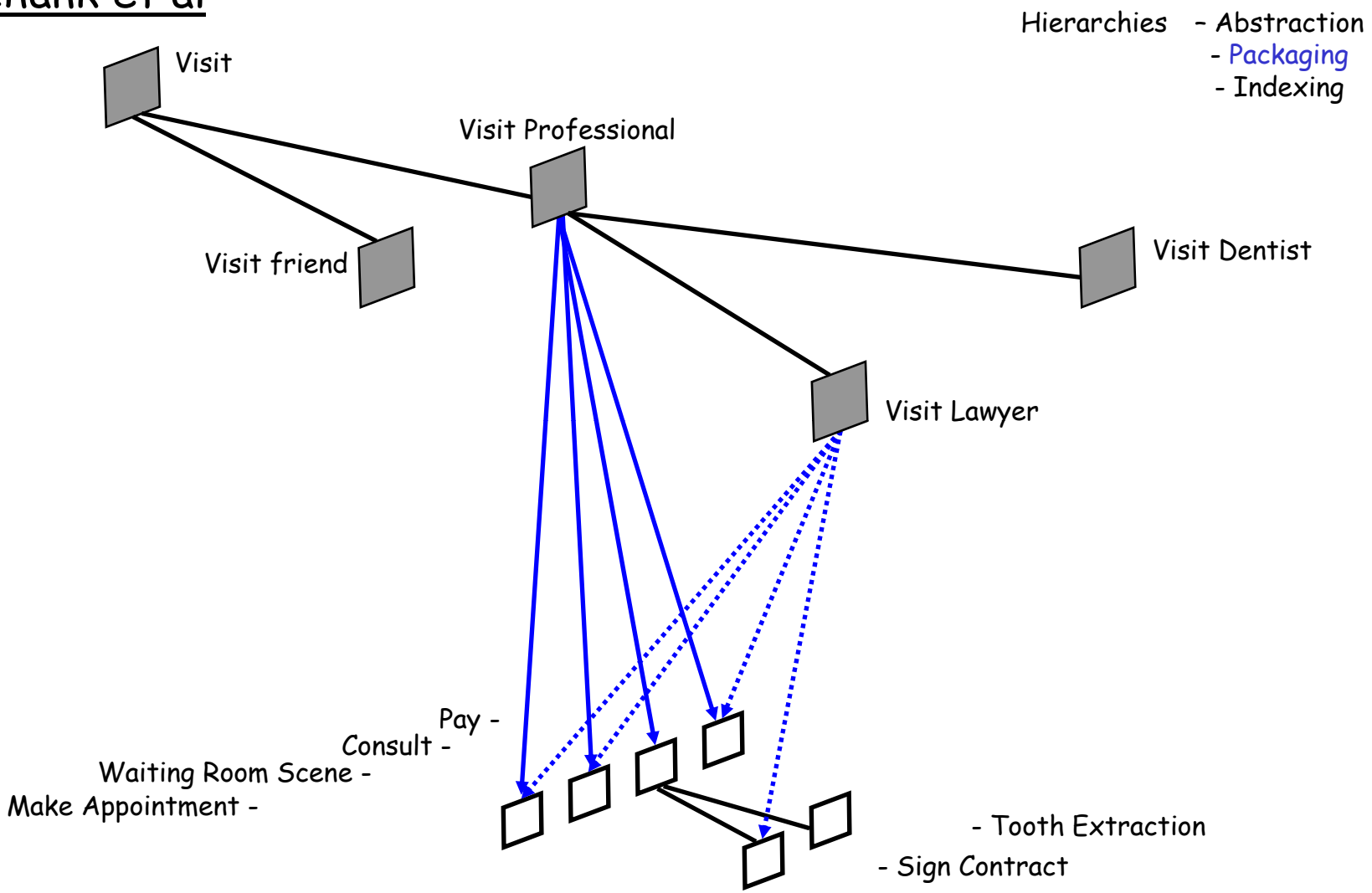
(with-less-sec-ingr oil)

(flavour spicy) *requirement*

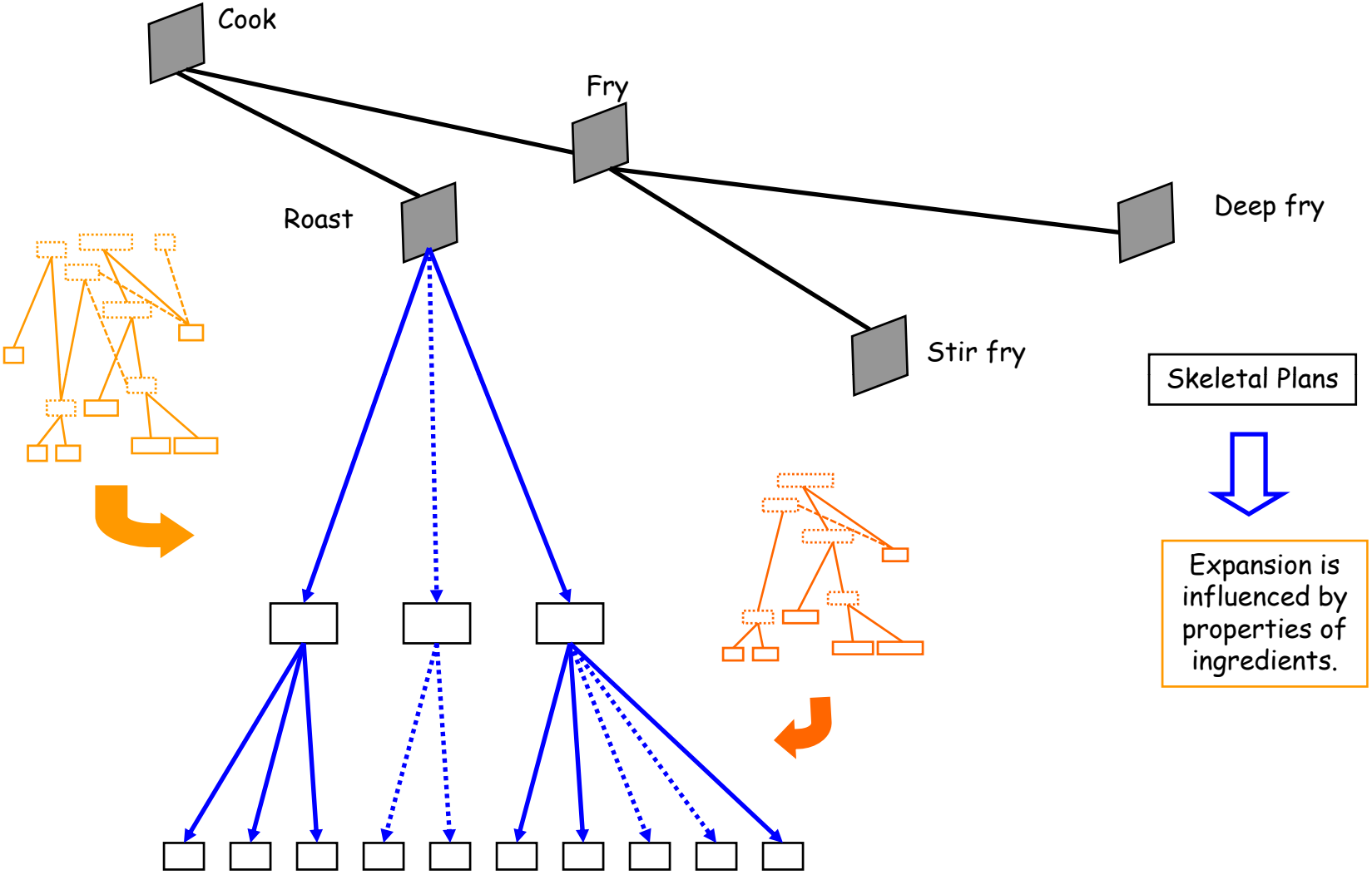
(with-sec-ingr chilly) *specification*

(with-sec-ingr garam-masala)

Memory Organization Packets (MOPs) – Schank et al



Bawarchi : Flexible Packaging



Flexible packaging : rationale

In case based reasoning, for example as in CHEF (Kris Hammond), every variation of a dish would be stored as a separate case.

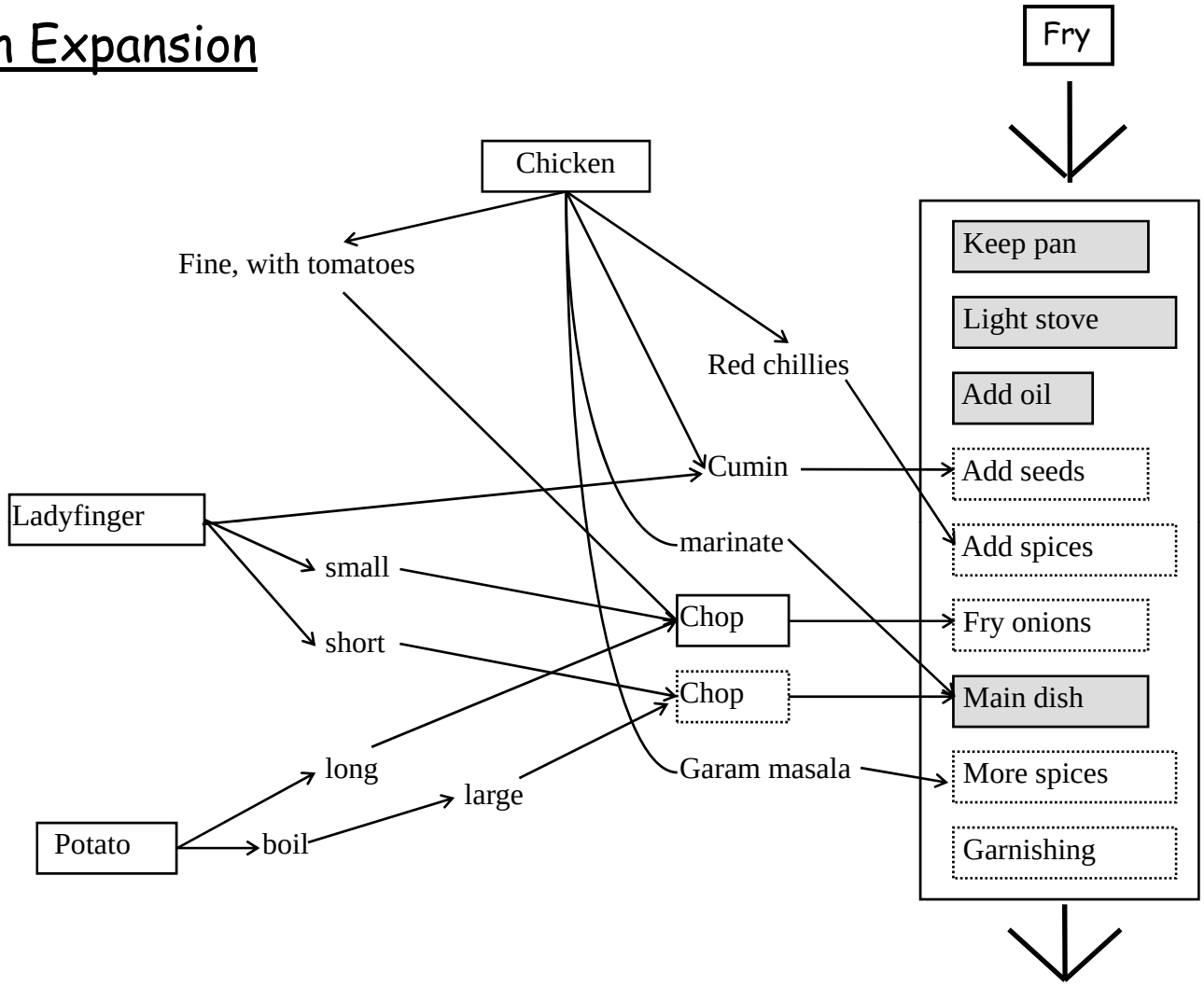
This would lead to a large and dense abstraction hierarchy creating a big indexing problem.

In Bawarchi, only a skeletal recipe is retrieved. This is based on the assumption that chunks of a "propositional" abstraction hierarchy can be collapsed into "first order" skeletal recipes.

To instantiate a specific recipe, a skeletal plan has to be expanded and filled in with the help of information derived from requirements, and ingredient specific knowledge.

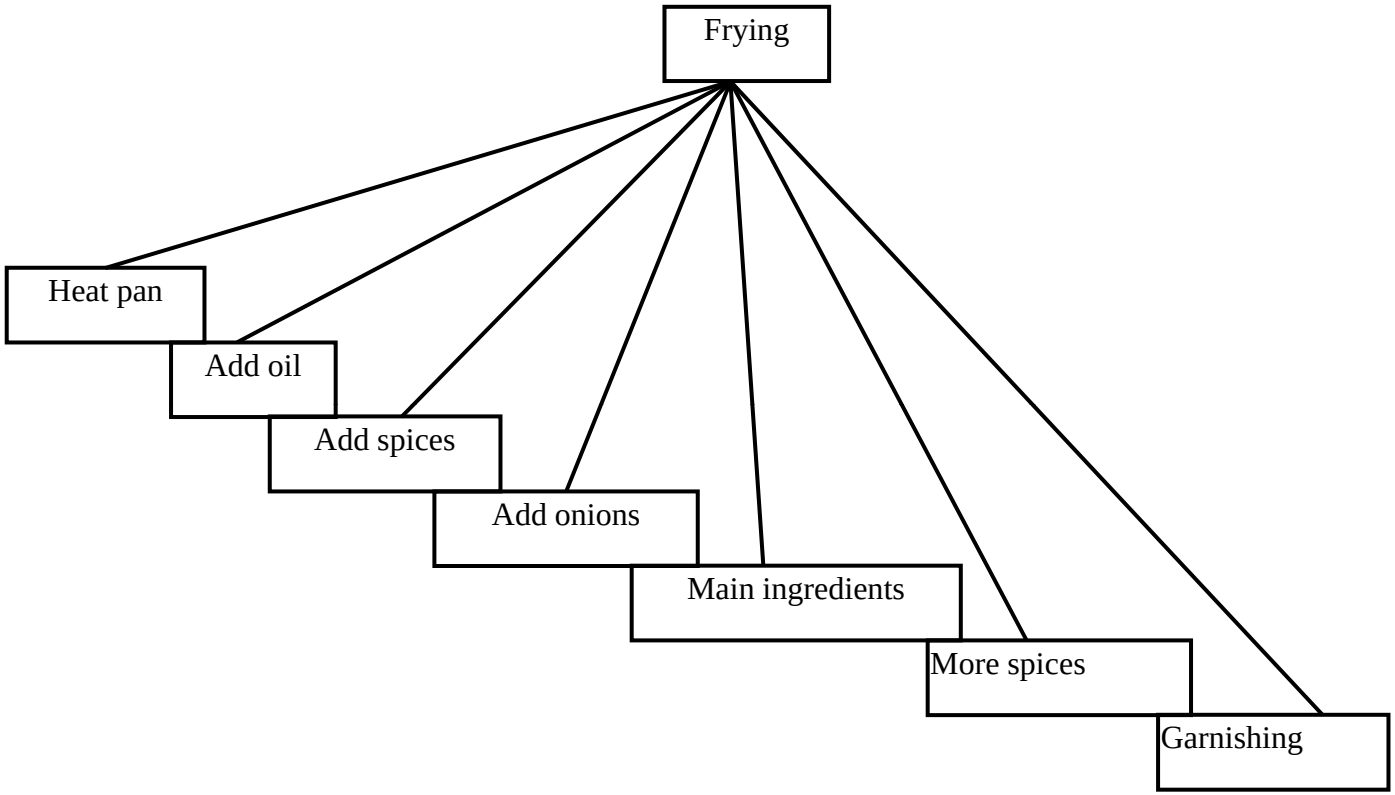
It may be observed, that one only adds to the skeletal plan. There is no modification requiring removal of actions in a plan.

Plan Expansion



A skeletal plan for fried dishes. Boxes in dotted lines are optional. The ingredients determine the optional processes to be included, and their nature.

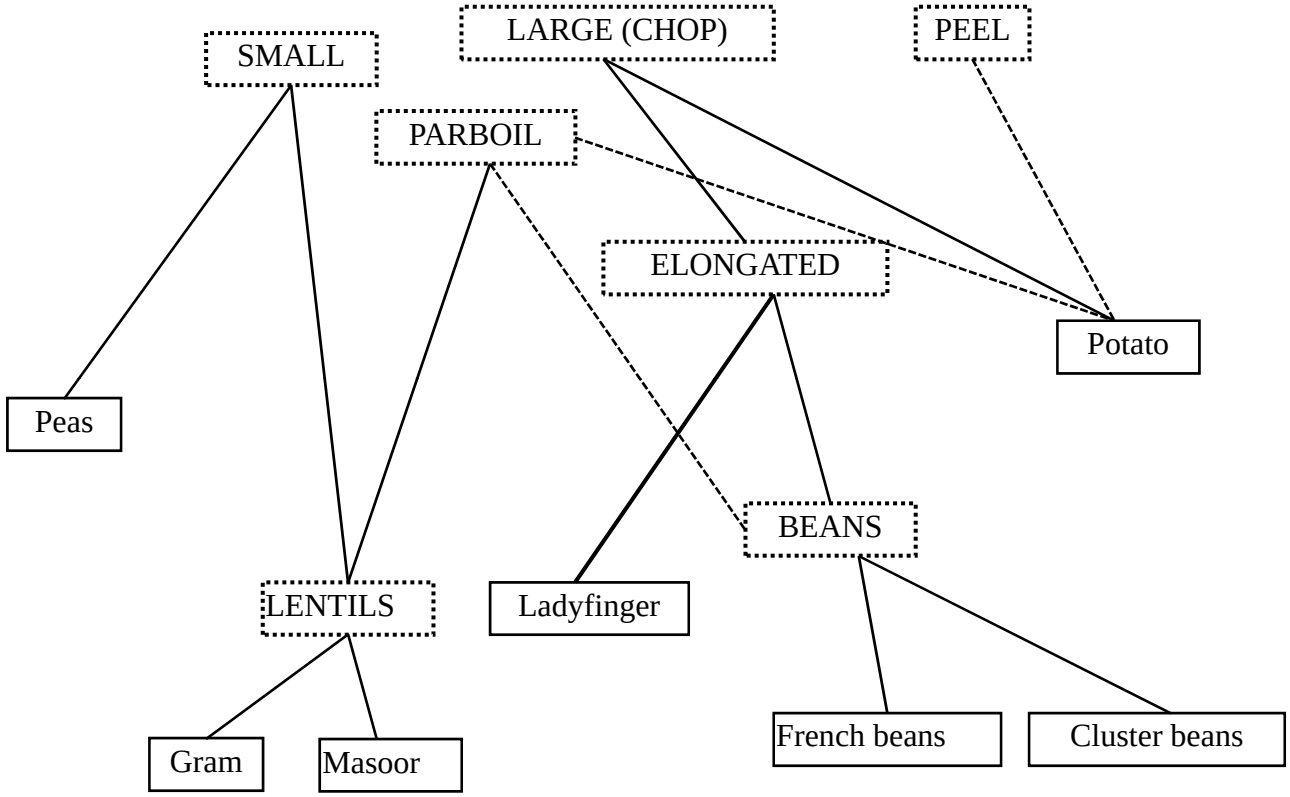
A Frying Plan



Expanding the frying plan.

Ingredients : semantic memory

Semantics : Ingredients are what you do to them!



A part of the semantic memory of ingredients. The dashed lines represented optional inheritance, depending upon theme (style) or requirements.

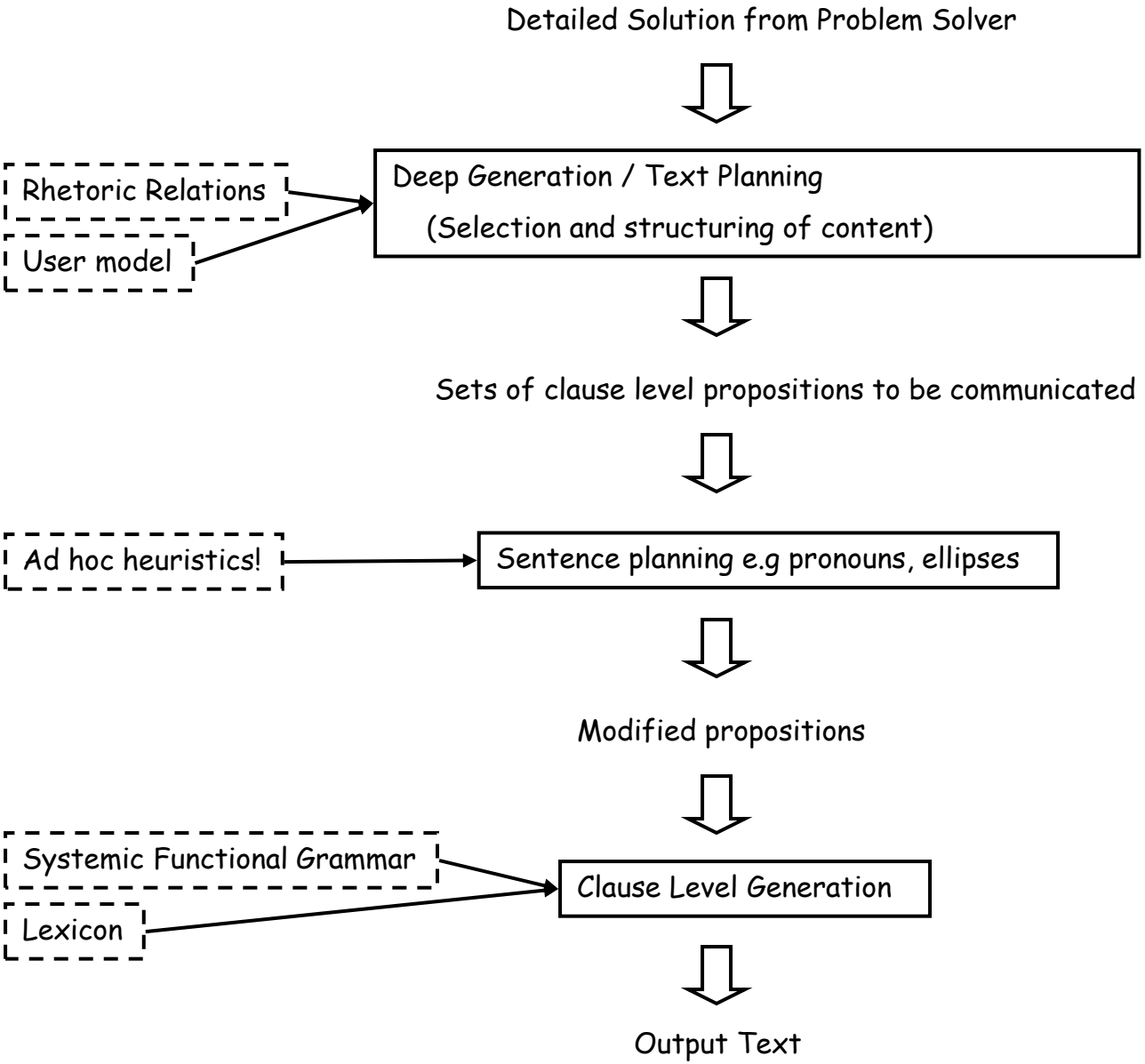
Bawarchi : raw output

Input: (A fried dish with potato and onion)

The recipe:
(take a frying dish of suitable size) (take a frying stick) (take a vessel approximately 1000 cc. Capacity) (take (1-kg.) potato) (ready the stove and a table knife) (take a peeler and remove the skin of the potato) (chop the potato into pieces of ½-inch size and keep the pieces in the appropriate vessel) (take a pressure cooker) (put the potato in the cooker) (add sufficient amount of water to the cooker and close with the lid) (light the stove and put the cooker on the stove) (wait until 1- whistle and turn-off the stove and wait for a few minutes) (transfer the potato into the vessel and pour out the water from the cooker) (take ¼ - table-spoon turmeric-powder) (take 3 – twigs curry-leaves) (take 1 – table-spoon cumin) (take 6 green-chillies) (take 1- piece green-ginger) (take 6- cloves) (take 1-piece cinnamon) (take 1 – table-spoon salt) (take ½ - garlic) (take ½ - table-spoon coriander-seeds) (make a paste of (green-chillies coriander seeds green-ginger garlic cinnamon cloves) (take 2 – coriander stalks and chop them) (take 75 - ml. oil) (take 1 – table-spoon mustard-seeds) (take 2 medium size onions and peel them and chop them) (light the stove and place the frying dish on it and wait until it becomes hot) (add the oil to the dish and wait until oil becomes hot) (add the mustard-seeds to the dish) (add the cumin to the dish) (add the curry-leaves to the dish) (add the chopped onions to the dish and mix them until they turn brown) (add the turmeric powder to the dish) (add the potato to the dish) (wait for a few minutes until the potato becomes hot) (add the paste of (green-chillies coriander seeds green-ginger garlic cinnamon cloves) to the dish and mix) (add the salt to the dish and mix) (keep on mixing the potato for every few minutes) (continue frying until the potato cease sticking to each other) (add the coriander stalks to the dish and mix) (turn-off the stove and remove the dish from the stove)

The "machine generated" templates quite obviously illustrate the need for more organized text! Also one should be able to "tune" the output to suit the user's expert level.

Part 2 : Text Generation



Text Planning

A Text is a coherent and cohesive sequence of sentences.

Rhetoric relations are relations between different pieces of text.

Examples: statement (nucleus) – justification (satellite)

premise – conclusion

question - answer

A text is coherent if it has a well defined rhetoric structure.

Issues

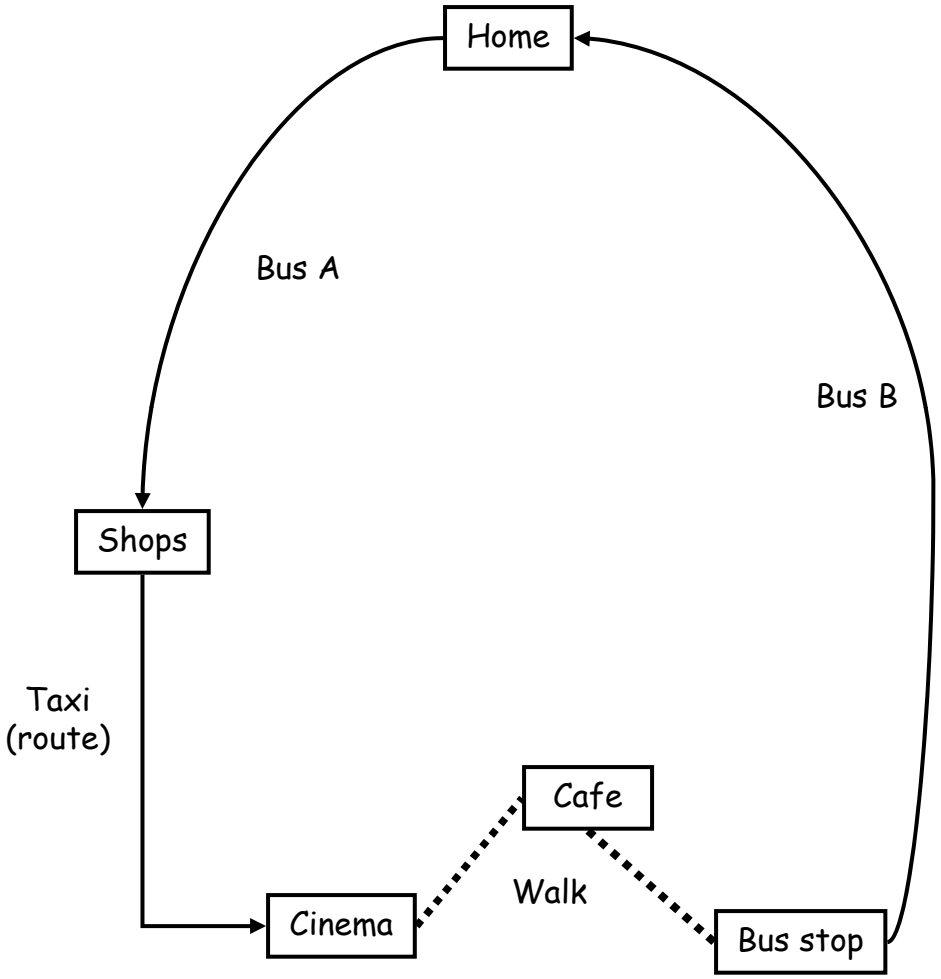
- At what level of detail should the solution be described?
- How much does the user already know?
- Who *decides* the content? Problem solver or communicator? (Our approach is that solver pipes solution to communicator via a *situation memory*)
- How much time does the speaker / listener have?
- Dialogue introduces different (truncated) text.

Note : Domain of text planning is the content (actions/events/propositions) from the problem solving domain. It *is not* linguistic.

SightSeer : Yet another route finder

Inputs : Locations (specific)
Goals (generic)

Output : Travel plan

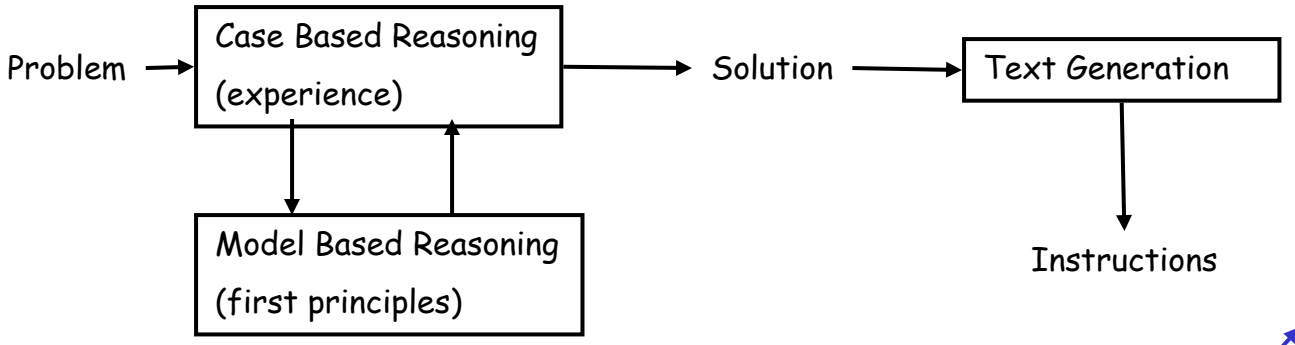


Issues:

- Time
- Distance
- Traffic
- Crowds
- Order

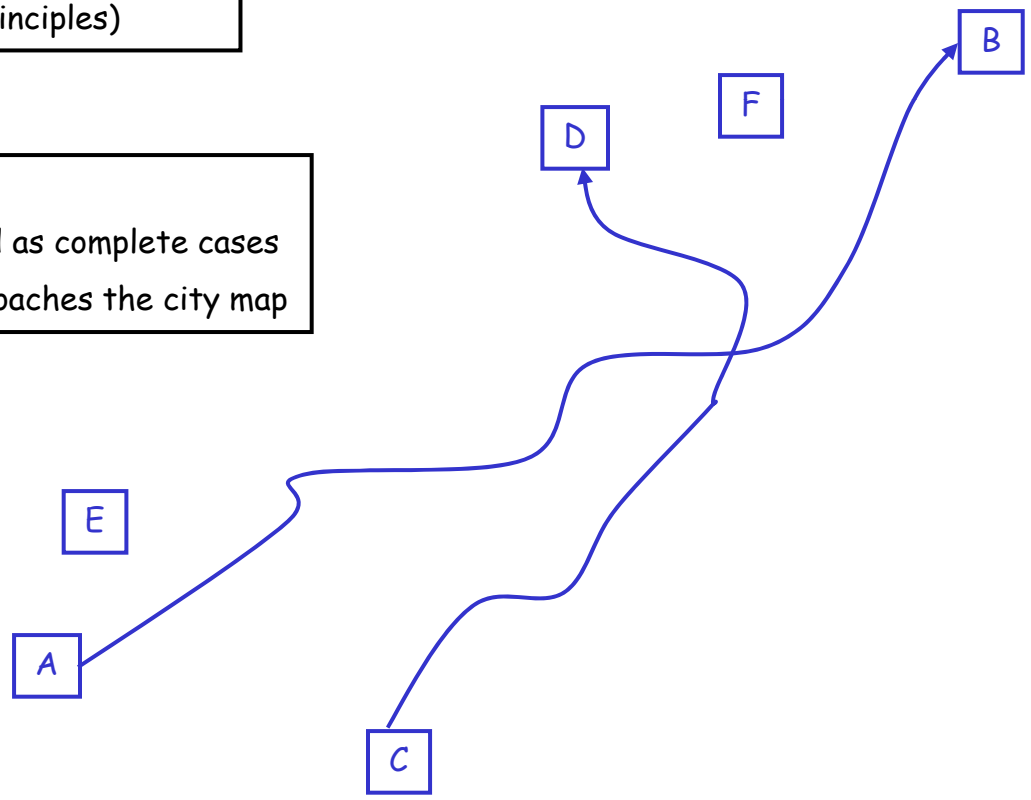
City Map in Memory?

Q: What kind of a maps do people carry in their head?



A: A Guess

- Some frequent long routes are represented as complete cases
- In memory a frequently visited region approaches the city map



Given cases A-B and C-D,

Find routes from A-D and E-F

Issue : Which cases are worth retrieving?

And storing?

Cases vs Rules

Cases are complete solutions containing a sequence of individual steps. In case based reasoning an entire case is retrieved based on some indexing scheme that takes in problem features and selects a best matching case.

If we try and store overlapping cases, which have common segments, we end up segmenting a case into macro like operators.

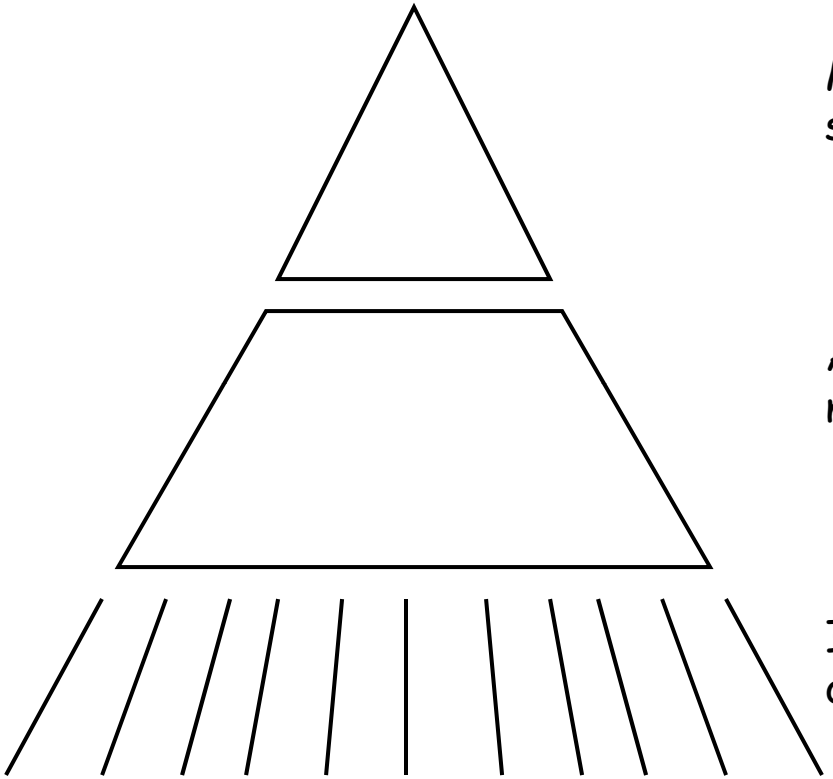
As more and more cases are added in the above manner, some solution steps will tend to become like rules – modular actions that are applicable in many situations, indexed by an antecedent describing the situation or sub-goal that triggers them.

For example, in football

If you have the ball,
and you are running down the right flank,
and opponents are closing in
then
look for a team mate to pass to.

Memory based problem solving knowledge is probably a mix of modular rules and packaged cases.

Rules and Cases



Modular compact rules embody specific lessons learnt.

Abstract full cases capture nominal scenarios

Individual episodes are instances or exceptions

RajuGuide: Example 1

This sample is generated in the task instruction mode. The contents of the situation memory, immediately after route planning, are given below:

```
Number of Intermediate goals          1
Number of journey elements in the first segment   11
Number of journey elements in the second segment                3
Source Location             IIT_GC
Intermediate Location       MALAR_HOSPITAL
Destination Location  MANDAVELI_BUS_STN
(   IIT_GC   IIT_ROAD 1 S 4.000000 )
(  IIT_GATE    SARDAR_PATEL_ROAD 1 R 0.500000  )
( 14   SARDAR_PATEL_ROAD 1 S 1.700000)
(   ADAYAR_CANAL   SARDAR_PATEL_ROAD 1 S 0.500000  )
( 30 SARDAR_PATEL_ROAD 2 S 0.900000 )
(   ADAYAR_BUS_STN    SARDAR_PATEL_ROAD  2 S 0.900000  )
( 32   SARDAR_PATEL_ROAD 2 S 0.700000 )
( 192 62 2  L 1.300000 )
( 191  60 2 R 2.200000 )
( 193   ADYAR_BRIDGE_ROAD 2 2L 0.600000 )
(   MALAR_HOSPITAL  x 2 x 0.000000  )
(   MALAR_HOSPITAL  ADYAR_BRIDGE 2 S 3.800000  )
(   AMS_SIGNAL RK_MUTT_ROAD 2 S 3.700000  )
(   MANDAVELI_BUS_STN  x 4 x 0.000000  )
```

Each element, (leg of journey) as given below, forms a frame:

(source_location road_name zone direction distance)

The relationships among the frames, for example (next #T7 #G8), are asserted in the situation memory.

Example 1 : English instructions to a known user

User model = case memory

1) First you go on the IIT road for one Km. 2) You can see the IIT gate. 3) Take a turn to the right on the Sardar Patel road. 4) Then go on that road for nearly one and a half Km. 5) On the way, you can see Adyar canal and Adayar Bus Station. 6) Then turn to the left. 7) Go for half a Km in that direction until you see location 191. 8) Take a turn to the right. 9) I suggest that you go for half a Km in that direction. 10) Then turn to the Adyar bridge road. 11) Go for a quarter Km until you see the Malar Hospital. 12) From there you know how to reach Mandaveli bus station.

Notes

- Sentences 2 and 3 come from the same frame #2.
- Sentence 5 collapses a few frames together.
- Sentence 12 – problem solver used retrieved case.
- Some phrases are juggled randomly (9 – "that direction")
- Sentence 9 also randomly uses a speech act – "recommend"

Example 2 : English narrative

1) Rama wanted to go to the Adayar Bus Station. 2) From that place he wanted to go to the Ganapathy Ram Theatre.

First a prologue describes the problem solving goals of the (fictitious) character.

3) Rama went on the Sardar Patel road for a quarter Km until he saw the location 32. 4) Then he took a turn to the right. 5) He went for half a Km in that direction. 6) He saw the location 34. 7) Then he turned to the Kasturiba main road. 7) He went on that road for three quarters of a Km until he saw the Ganapathy Ram Theatre.

In the second paragraph, the solution to the most important goal is described.

9) Earlier Rama had gone on the IIT road for 1.0 Km until he had seen the IIT gate. 10) Then he had taken a turn to the right. 11) He had gone for nearly one Km in that direction 12) He had seen the Adayar Bus Station.

Finally, the other parts are filled in. Since they occurred earlier in the chronology, in a story generation mode they become a flashback.

Addressing such information ordering issues is part of text planning. They would be important, for example, in writing mystery stories, or telling jokes.

Observation : Text planning probably makes sense *only* when one has a listener in mind!

Example 3 : Hindi and Telugu

Directions in Hindi

pehale aap IIT sadak par eek km duur tak jaaiye. aap IIT geet deekh sakate hain. tab aap Sardar Patel sadak par baayen ko mudiye. baadmein aap usi sadak par lagbhag deedh km duur tak jaaiye. raaste mein aap Kotturpuram Signal aur Gandhi Mandapam deekh sakate hain. tab aap daayen ko mudiye. baadmein aap usi disha mein lagbhag deedh km duur tak jaaiye. anthmein aap Saidapet bas adda deekh sakate hain.

Same problem : Story in Telugu

modalu raamudu IIT rooddu miida voka km duuramu varaku vellinaadu. atanu IIT geetu chuuchinaadu. appudu atanu Sardar Patel rooddu miida edama vaipuku tiriginaadu. taruvaata atanu adhe rooddu miida sumaaru vokatinnara km duuramu varaku vellinaadu. darilo atanu Kotturpuram Signal mariyu Gandhi Mandapam chuuchinaadu. appudu atanu kudi vaipuku tiriginaadu. taruvaata atanu adhe dishalo sumaaru vokatinnara km duuramu varaku vellinaadu. chivariga atanu Saidapet bus stand chuuchinaadu.