

Yale University
Department of Computer Science

Question-driven understanding:
An integrated theory of story understanding,
memory and learning

Ashwin Ram
YALEU/CSD/RR #710
May 1989

This research was supported by the Defense Advanced Research Projects Agency and the Office of Naval Research under contract N00014-85-K-0108, and by the Air Force Office of Scientific Research under contracts F49620-88-C-0058 and AFOSR-85-0343.

**Question-driven understanding:
An integrated theory of story understanding,
memory and learning**

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Ashwin Ram
May 1989

© Copyright by Ashwin Ram 1989
All Rights Reserved

Acknowledgements

My advisor, Roger Schank, has been my *guru* in the original sense of the word. He has been my teacher, my friend, my benefactor and my guide. His enthusiasm and insights have led to many of the ideas presented in this dissertation, and his guidance and encouragement have led to my being able to write them down. Roger's *joie de vivre* cannot help but rub off on people around him; he has stimulated me into thinking about AI and, indeed, about food, wine, politics and pretty much everything else under the sun.

I would also like to thank the other members of my committee, Drew McDermott and Larry Horn, for their detailed comments about this work. Drew patiently sat through what must have been several frustrating meetings to make sure we understood each other. His comments have had a great deal of influence on this dissertation. Larry Birnbaum helped me formulate many of my ideas; his comments have been especially helpful.

Thanks also to the other faculty members for making Yale a fun place to be. Chris Riesbeck was always there when I needed him, and his help has been invaluable throughout my years at Yale. I learned a lot from Mike Fischer, Paul Hudak and Richard Gerrig. To all of you, thanks for being around.

Almost everyone who was part of the AI lab had some influence on me. Many of the ideas presented here were developed through countless discussions with Alex Kass, Eric Jones, Rob Farrell, and Eric Domeshek. I got into more arguments with them than I could hope to remember; I can only thank them for their relentlessness. Special thanks to Eric Jones who wrote much of the frame representation language used in AQUA. I would also like to thank Ed Hovy, David Leake, Chris Owens, Larry Hunter, Charlie Martin, and Dave Littman for their company as well as for their help in formulating my ideas.

Jerry DeJong and David Waltz first introduced me to AI. They convinced me that it was fun, and I haven't regretted it since. Alberto Segre was responsible for introducing me to my first AI project, and to fine coffee.

My friends and colleagues have provided much more than just an academic environment. I would like to thank Steve Hanks, Jim Firby, Steve Slade, Rick Mohr, Lucian Hughes, Diane Schank, and many others too numerous to list here for making my life interesting and enjoyable.

My parents, though far away, have managed to perform the incredible feat of being here with me. They believed I could do it, and waited patiently while I did. A most special thanks to Preetha for her love and affection. I couldn't have climbed the infamous thesis hill without her support.

Finally, I would like to thank the agencies that provided financial support for the research reported here, the Defense Advanced Research Projects Agency and the Office of Naval Research under contract N00014-85-K-0108, and by the Air Force Office of Scientific Research under contracts F49620-88-C-0058 and AFOSR-85-0343.

Abstract

Question-driven understanding: An integrated theory of story understanding, memory and learning

Ashwin Ram
Yale University
1989

Story understanding is a goal-directed process. The memory of an understanding system is never quite complete: knowledge structures may have "gaps" in them, or they may not be indexed correctly in memory. When one reads a story, these gaps give rise to questions about the input. The point of reading is find answers to these questions, to learn by filling in the gaps in one's world model.

Questions, therefore, represent the "knowledge goals" of the understander, things that the understander wants to learn about. Understanding is viewed as the opportunistic pursuit of these questions, as a process of seeking answers to these questions in the input, which in turn raises new questions while answering old ones. This is a dynamic model; after reading a story, an understander would read a similar story, or even the same story again, a little differently since different questions would be raised the next time around. We discuss where these questions come from (i.e., what kinds of knowledge goals an understanding system has), how they affect the understanding process, and how an understander can learn from novel answers to its questions. The theory is implemented in a program called AQUA (Asking Questions and Understanding Answers), which learns about terrorism by reading newspaper stories about terrorist incidents in the Middle East.

Contents

List of Figures	vii
1 The nature of questions	1
1.1 Question-driven understanding	2
1.1.1 Cognitive motivations: Goal-based understanding	4
1.1.2 Computational motivations: What are the knowledge goals of an understanding program?	6
1.1.3 Knowledge structures aren't perfect	7
1.1.3.1 Mis-indexed knowledge structures	7
1.1.3.2 Gaps in knowledge structures	8
1.1.3.3 Incremental refinement of knowledge structures	8
1.1.4 What is the output of the understanding process?	9
1.1.5 Understanding is subjective	10
1.1.6 Understanding is dynamic	11
1.1.7 Controlling inferences	12
1.1.8 The problem of interestingness	13
1.1.9 Opportunism and the memory model	13
1.2 The AQUA program	14
1.3 Issues addressed in this thesis	15
2 Previous work in story understanding	17
2.1 Integrated processing	17
2.2 The nature of stories	18
2.3 The nature of predictions	19

2.4	The source of predictions	20
2.5	Inference control	21
2.6	Interestingness	21
2.7	Subjective interpretation	22
2.8	Questions and opportunistic memory	23
2.9	Learning and explanation	24
2.10	Conclusion	25
3	Theory of explanation	26
3.1	What is an explanation?	27
3.1.1	Introspective or meta-explanations	28
3.1.2	Domain explanations	28
3.1.2.1	Physical explanations	29
3.1.2.2	Volitional explanations	30
3.2	Decision models: Representing volitional explanations	30
3.2.1	The outcome of an action	31
3.2.2	The decision process	31
3.2.3	Considerations in decisions	33
3.2.4	Components of explanation patterns	33
3.2.5	Example: The religious fanatic explanation	34
3.2.6	Example: The depressed teenager explanation	36
3.3	Elaborating explanations	38
3.3.1	Additivity of explanations	38
3.3.2	Compositionality of explanations	38
3.3.3	Levels of explanation	39
3.4	Indexing explanations in memory	39
3.4.1	Category index	41
3.4.2	Situation index	41
3.4.3	Stereotype index	41
3.5	The explanation cycle	42
3.5.1	Anomaly detection	43

3.5.2	XP retrieval	44
3.5.3	XP application	45
3.5.4	Hypothesis verification	45
3.6	Evaluating explanations	45
3.7	Conclusion	46
4	Theory of questions	47
4.1	A taxonomy of questions	48
4.1.1	Explanation questions	49
4.1.1.1	Anomaly detection questions	50
4.1.1.2	XP retrieval questions	56
4.1.1.3	XP application questions	62
4.1.1.4	Hypothesis verification questions	63
4.1.2	Text-level questions	64
4.1.2.1	Actor-action identification	64
4.1.2.2	Attachment questions	65
4.1.2.3	Reference questions	66
4.1.2.4	Disambiguation and specialization questions	67
4.1.2.5	Questions or requests?	67
4.1.3	Memory-level questions	68
4.1.3.1	Journalism questions	68
4.1.3.2	Reference questions	68
4.1.3.3	Attachment questions	68
4.1.3.4	Stereotype activation questions	68
4.1.3.5	MOP and scene inference questions	70
4.1.3.6	Reminding and memory search questions	71
4.1.3.7	Analogy and generalization questions	72
4.1.4	Relevance questions	72
4.2	Conclusion	72

5	Theory of understanding	73
5.1	Processing issues	73
5.1.1	Integration of higher-level reasoning processes	73
5.1.2	Subjective understanding	75
5.2	Controlling inferences	77
5.2.1	Bottom-up vs. top-down processing	77
5.2.1.1	Inferences required for matching	78
5.2.1.2	Inferencing after a question is answered	79
5.2.2	Theory of inference control	79
5.3	Memory representations	84
5.3.1	Representation of explanations	84
5.3.2	Representing hypotheses	86
5.3.3	The TRUTH slot	87
5.3.4	The STATUS slot	87
5.4	Opportunistic memory architecture	87
5.4.1	The understanding cycle	88
5.4.1.1	The READ step	88
5.4.1.2	The EXPLAIN step	88
5.4.1.3	The GENERALIZE step	89
5.4.2	Process model for running understanding tasks	89
5.4.3	Representation of questions	90
5.4.3.1	Concept specification	90
5.4.3.2	Task specification	91
5.4.4	Indexing questions	92
5.4.5	Retrieving questions	92
5.4.5.1	Type retrieval	92
5.4.5.2	Relation retrieval	92
5.4.5.3	Specialization retrieval	93
5.4.6	Answering questions	93
5.4.7	Advantages of the opportunistic memory model	95

6	Theory of learning	96
6.1	Incremental theory formation: Where do XPs come from?	96
6.2	Incremental learning in AQUA	98
6.3	Learning indices for explanation patterns	99
6.3.1	Learning situation indices	100
6.3.2	Learning stereotype indices	101
6.3.2.1	Learning from successful explanations	105
6.3.2.2	Learning from failed explanations	106
6.4	Modifying existing explanation patterns	107
6.4.1	Associating new questions with XPs	107
6.4.2	Incremental refinement of XPs by answering questions	110
6.5	Conclusion: Learning from explanation questions	111
6.6	Extensions	113
6.6.1	Extension: Learning new XPs	113
6.6.2	Extension: Learning by case adaptation	114
6.6.2.1	XP retrieval	114
6.6.2.2	XP application	115
6.6.2.3	Inferencing for HVQ verification	116
6.6.2.4	Subjective interpretation	117
6.7	Conclusion: Memory and understanding	117
7	Theory of interestingness	119
7.1	Interestingness heuristics	120
7.1.1	Interestingness from relevance goals	122
7.1.1.1	What could be relevant to a program?	122
7.1.1.2	Goal relevance	124
7.1.1.3	Vicarious goals	125
7.1.1.4	Actor relevance	126
7.1.1.5	Object relevance	127
7.1.1.6	Location relevance	127
7.1.2	Interestingness from explanation goals	128

7.1.2.1	Anomaly detection: What needs to be explained	128
7.1.2.2	XP application	132
7.1.2.3	Hypothesis verification	133
7.1.3	Interestingness from memory goals	134
7.2	Computing interestingness by combining heuristics	135
7.2.1	Interestingness of questions	135
7.2.2	Interestingness of new concepts	136
7.2.3	Interestingness of concepts in memory	137
7.3	Examples	138
7.3.1	Example: A stereotypical story with personal relevance	139
7.3.2	Example: A novel story with no personal relevance	139
7.3.3	Example: Another novel story	140
7.4	Conclusion: Interest-producing conditions	141
8	Program design: Putting it all together	144
8.1	Control structure: The understanding cycle	144
8.2	Memory representations	146
8.2.1	The frame system	146
8.2.2	Representing concepts	146
8.2.2.1	Frames, slots and facets	146
8.2.2.2	Types of role fillers	147
8.2.2.3	Relations	148
8.2.2.4	The MOP hierarchy	148
8.2.2.5	Ontology	148
8.2.3	Representing explanation patterns	154
8.2.3.1	Causal representations	157
8.3	Program example	159
9	Conclusions	189
9.1	Current implementation of AQUA: Capabilities and limitations	190
9.2	Boundary conditions: Capabilities and limitations	191
9.2.1	No stereotypical XP in memory	191

CONTENTS

vii

9.2.2	Story too bizarre	191
9.2.3	Story fits perfectly	192
9.2.4	Questions arising from other goals	192
9.2.5	How many questions does AQUA ask?	192
9.3	Conclusion: "Goal-based programs"	193

Bibliography		194
---------------------	--	------------

List of Figures

1.1	Question-driven understanding	11
3.1	The structure of volitional explanations.	32
3.2	The religious fanatic explanation pattern.	35
3.3	The depressed teenager explanation pattern.	37
3.4	The religious fanatic explanation pattern, revisited.	40
3.5	Modelling different points of view.	44
4.1	Anomaly detection questions	53
5.1	Theory of inference control.	80
5.2	Inferences supporting the religious fanatic hypothesis.	81
5.3	The structure of a hypothesis tree.	86
6.1	Learning situation indices for XPs.	102
6.2	Learning stereotype indices for XPs.	104
6.3	Associating new questions with XPs.	109
6.4	Elaborating an XP through incremental learning.	112
8.1	Control structure: The understanding cycle	145
8.2	Representing frames in memory	147
8.3	Equivalence between frame language representations and graph structures in memory.	149
8.4	Packaging hierarchy of event structures.	150
8.5	Concepts in AQUA's memory	151
8.6	Relations in AQUA's memory	153
8.7	Role relations in AQUA's memory	155

LIST OF FIGURES

8.8 Primitive causal relations in AQUA's memory.	158
8.9 Questions in, questions out.	188

Chapter 1

The nature of questions

We're flooding people with information. We need to feed it through a processor. A human must turn information into intelligence or knowledge. We've tended to forget that no computer will ever ask a new question.

— Adm. Grace Hopper, Feb 5, 1987.

In October 1983, the U.S. was shocked by reports of a suicide bombing mission that destroyed the U.S. marine barracks in Beirut. Newspapers ran detailed stories describing the mission. Editorials such as “Who are these smiling killers?” (New York Times, Dec 18, 1983) tried to explain the “readiness to die for a political belief” in terms of the Japanese Kamikaze pilots of World War II, religious mysticism, and the psychology of the people who apparently perceived such suicidal actions as being life-enhancing.

Many such incidents were reported in the following year, including the infamous car bombing attack on the U.S. embassy at Beirut in September 1984. But then these detailed reports started changing in character. They became more sketchy and less inclined to offer explanations, and were often relegated to the less important sections of the paper. For example, consider the following story (New York Times, Nov 27, 1985, page A9), in which the emphasis is on the journalistic details of the event rather than any understanding of the motivations of the people involved:

S-1: Suicide bomber strikes Israeli post in Lebanon.

SIDON, Lebanon, November 26 — A teenage girl exploded a car bomb at a joint post of Israeli troops and pro-Israeli militiamen in southern Lebanon today, killing herself and causing a number of casualties, Lebanese security sources said. ...

A statement by the pro-Syrian Arab Baath Part named the bomber as Hamida Mustafa al-Taher, born in Syria in 1968.

The statement said she had detonated a car rigged with 660 pounds of explosives in a military base for 50 South Lebanon Army men and Israeli intelligence and their vehicles.

Of course, the reporters were catering to what they thought were the changing interests of their readers. They hadn't really understood these fanatics any better, nor had they begun to sympathize with them. But they had learned about the stereotypical “religious fanatic commits suicide bombing”

situation. Stories seldom differed from this situation in significant ways, and so they ceased to raise any new issues as far as the readers were concerned.

Occasionally a novel story was published, one that did raise new questions or provided the reader with fresh insight into an old question. For example, consider the following excerpt from a rather unusual story which appeared on the front page (New York Times, April 14, 1985):

S-2: Boy Says Lebanese Recruited Him as Car Bomber.

JERUSALEM, April 13 — A 16-year-old Lebanese was captured by Israeli troops hours before he was supposed to get into an explosive-laden car and go on a suicide bombing mission to blow up the Israeli Army headquarters in Lebanon. ...

What seems most striking about [Mohammed] Burro's account is that although he is a Shiite Moslem, he comes from a secular family background. He spent his free time not in prayer, he said, but riding his motorcycle and playing pinball. According to his account, he was not a fanatic who wanted to kill himself in the cause of Islam or anti-Zionism, but was recruited for the suicide mission through another means: blackmail.

Here was a suicide bomber who was *not* a fanatic, who did not fit the stereotype. This story provides the reader with a novel answer to the motivational question of why Lebanese kids were going on these suicidal missions, but raises a whole host of new questions: Who blackmailed him? How could you blackmail anyone into killing himself? What could he possibly want more than his own life? After reading the story, I found myself wondering about several issues that I had never thought about before.

This dissertation addresses a central issue in memory-based story understanding that has been largely ignored by researchers so far: *What happens to an understanding system as a result of reading a story? How do the issues that the understander is interested in affect the reading process, and how do these interests change during the process of reading?*

We present a story understanding program that is intended as a computer model of the *understander's memory* in addition to being a model of the *understanding process* itself. The program learns and refines its world model by reading newspaper stories in the terrorism domain.

1.1 Question-driven understanding

The intuition underlying the theory of question-driven understanding is that *asking questions is central to understanding*. To illustrate what this means, consider the blackmail story S-2 above. If one wants to learn more about the motivations of the terrorists in the Middle East, this story is interesting because it is anomalous. The usual stereotype of the Shiite religious fanatic does not hold here. Instead, this story raises many new questions. Here are some of the questions that were voiced by a class of graduate students when this story was read to them:

1. Why would someone commit suicide if he was not depressed?
2. Did the kid think he was going to die?
3. Are car bombers motivated like the Kamikaze?
4. Does pinball lead to terrorism?

5. Who blackmailed him?
6. What fate worse than death did they threaten him with?
7. Why are kids chosen for these missions?
8. Why do we hear about Lebanese car bombers and not about Israeli car bombers?
9. Why are they all named Mohammed?
10. How did the Israelis know where to make the raids?
11. How do Lebanese teenagers compare with American teenagers?

Some of the questions seem pretty reasonable, (e.g., "Did the kid think he was going to die?"), but some are rather silly in retrospect (e.g., "Does pinball lead to terrorism?"). Some, though perfectly reasonable questions, aren't central to the story itself, but instead relate to other things that the person concerned was reminded of, things that he was wondering about or interested in (e.g., "Why do we hear about Lebanese car bombers and not about Israeli car bombers?").

The claim is that an understander has questions already extant in memory before it begins to read a story. These questions are left over from its previous experiences. As the understander reads the story, it remembers these questions and thinks about them again in a new light. This raises further questions that the understander begins to think about. Certainly after reading this story, one expects to have several questions representing issues one was wondering about which weren't resolved by the story. For example, in this story it turns out that the boy was blackmailed into going on the bombing mission by threatening his parents. This makes one think about the question "What are family relations like in Lebanon?" This question remains in memory after one has finished reading the story. To the extent that one is interested in this question, one will read stories about the social life in Lebanon, and one will relate other stories to this one. To cite another example, one of the students in the class repeatedly related the story to the IRA because he was interested in similar issues about Ireland.

Thus, in addition to starting out with questions, the understander ends up with more questions after it finishes reading the story. These include new questions that were raised while reading the story, as well as old questions that the understander was reminded of but were not resolved by the story. *Furthermore, the new questions that are raised depend on the questions that the understander started with.* For example, consider story S-1 again:

S-1: Suicide bomber strikes Israeli post in Lebanon.

SIDON, Lebanon, November 26 — A teenage girl exploded a car bomb at a joint post of Israeli troops and pro-Israeli militiamen in southern Lebanon today, killing herself and causing a number of casualties, Lebanese security sources said. ...

A statement by the pro-Syrian Arab Baath Part named the bomber as Hamida Mustafa al-Taher, born in Syria in 1968.

The statement said she had detonated a car rigged with 660 pounds of explosives in a military base for 50 South Lebanon Army men and Israeli intelligence and their vehicles.

This is a fairly stereotypical suicide bombing story. It offers no explanation of the girl's motivations, concentrating instead on the journalistic details of the event. If the understander believes that suicide bombers are religious fanatics, the explanation is obvious; it can use the stereotypical religious fanatic explanation to fill in the missing pieces of the story. The questions that this story

raises focus on the religious beliefs of the girl. These questions essentially predict what the understander already expects to see (e.g., "Is she a Shiite Moslem?"). These questions attempt to confirm the understander's belief in the stereotypical explanation it already knows about.

However, if the understander has just read the blackmail story S-2, the explanation may not be quite so obvious. Was the girl really a religious fanatic? Was she was also coerced in some manner? The questions that were raised during the blackmail story, and the hypotheses that were created, affect the questions that are raised by this story. They affect the issues that the understander thinks about while reading the story, and the interpretations that it draws.

Thus understanding is a process of relating what one reads to the questions that one already has. These questions represent the *knowledge goals* of the understander, i.e., the things that the understander wants to learn about [Ram, 1987; Schank and Ram, 1988; Schank, 1986; Hunter, 1989; Dehn, 1989]. The point of reading is to find answers to these questions so as to attain a more complete understanding of the issues the understander is interested in. However, while doing this, many new questions get raised, which in turn guide the understanding of future stories and affect the interpretations that are drawn.

The intuition, then, is that asking questions is central to understanding. This raises two issues:

- **Content:** What kinds of questions are there? How does the understander know which questions to ask?
- **Process:** What difference do questions make? What effect do they have on the understanding process? How are questions managed?

Let us discuss these questions from a cognitive perspective as well as a computational one.

1.1.1 Cognitive motivations: Goal-based understanding

When we compare the way people read newspaper stories with how computer programs typically read them, we notice the following differences:

1. People are biased. They interpret stories in a manner that suits them. They jump to conclusions. Computer programs, on the other hand, are usually designed to read stories in an objective manner, and to extract the "correct" or "true" interpretation of a story to the extent that they can.
2. People don't read everything in great detail. They concentrate on details that they find relevant or interesting, and skim over the rest. In contrast, computer programs are designed to attend to every aspect of a story that is within the scope of their knowledge structures. Consequently, they either process the entire story in great depth, or else they skim everything in the story. They can not decide which aspects to process in detail and which ones to ignore.
3. People change as they read. They never read the same story twice in the same way. They notice different things the second time around, or they simply get bored. After reading a story, they interpret other similar stories differently. Computer programs, in contrast, are not adaptive; they always read a given story the same way.

What makes people different from computer programs? What is the missing element that our theories don't yet account for? The answer is simple: *People read newspaper stories for a reason: to learn more about what they are interested in.* Computers, on the other hand, don't. In fact, computers don't even have interests; there is nothing in particular that they are trying to find out when they read.

Parsing has conventionally been viewed as the extraction of syntactic information from sentences (the "syntactic school"), or as the translation of sentences into conceptual representations of meaning (the "conceptual school"). However, while understanding language does involve these processes, both views usually overlook the basic question of goals: *Why do we read in the first place?* How do the goals of the understander affect the understanding process?

In other words, if a natural language system is to read and understand a language the way people do, it can't be just a "translator." People differ from translators in one critical respect — they have *goals*. One reads because one *wants* to read. One wants to find things out. One has questions one wants answers to, interests one wants to pursue, issues one wants to learn about. If a computer program is to be a model of language understanding, it should also read for a "purpose."

Of course, people have several goals that do not make sense to attribute to computers. One might read a restaurant guide in order to satisfy hunger or entertainment goals, or to find a good place to go for a business lunch. Computers do not get hungry, and computers do not have business lunches.

However, these *physiological* and *social goals* give rise to several *intellectual* or *cognitive goals*. A goal to satisfy hunger gives rise to goals to find information: the name of a restaurant which serves the desired type of food, how expensive the restaurant is, the location of the restaurant, etc. These are goals to acquire information or knowledge, and are called *knowledge goals*. These goals can be held by computers too; a computer might "want" to find out the location of a restaurant, and read a guide in order to do so in the same way as a person might. While such a goal would not arise out of hunger in the case of the computer, it might well arise out of the "goal" to learn more about restaurants.

In other words, knowledge goals also arise from the desire to learn, to pursue one's intellectual interests, to improve one's model of the world. These goals can be viewed as *questions* about the domain of interest. To be interested in terrorism, for example, is to have a lot of questions about various aspects of terrorism, and to think about these questions in the context of input data about terrorism, such as newspaper stories about terrorist incidents. The point of reading these stories is to answer one's questions, as well as to reveal flaws or gaps in one's model of terrorism in order to try to improve this model. These gaps give rise to new questions which in turn stimulate further interest in terrorism.

Both computers and people can be "interested" in terrorism in this sense. In either case, it would follow that when one reads, one would constantly relate what one is reading to one's questions, and to what one knows about the domain. This, of course, is why one gets reminded of other things as one reads, why one get side-tracked into other chains of thoughts and finds oneself wondering about other questions that are brought up.

In contrast with people, therefore, a computer has only one underlying goal: to learn and improve its world model.¹ However, this (and, in the case of people, other physical and social goals) gives rise to knowledge goals that then drive the understanding process. Parsers that do not include a theory of how one relates what one reads to one's knowledge goals and memory are at best just translators; they will never constitute a theory of natural language *understanding*.

1.1.2 Computational motivations: What are the knowledge goals of an understanding program?

Understanding, then, consists of asking questions and trying to answer them. Although questions are the crux of the theory of question-driven understanding, it would defeat the purpose to build a "question-asking" program per se. Instead, questions should arise naturally as *knowledge goals* of the program during various stages of the understanding process. This means that the program should ask a question only when it has a goal to acquire that piece of knowledge, i.e., when it needs to know the answer for the purposes of understanding the story. In other words, questions should be *functionally useful* to the overall goals of the system.

The theory of questions presented in this dissertation depends on a theory of *understanding tasks*, the basic tasks of an understander. These tasks include the integration of facts with what the understander already knows, the detection of anomalies in the text which identify flaws or gaps in the understander's model of the domain, the formulation of explanations to resolve those anomalies, the confirmation and refutation of potential explanations, the learning of new explanations for use in understanding future situations. These are the basic tasks that an understander needs to be able to perform.

In order to carry out these tasks, the understander needs to integrate the text, which is often ambiguous, elliptic and vague, with its world knowledge, which is often incomplete. In formulating an explanation, for example, the understander may need to know more about the situation than is explicitly stated before it can decide which is the best explanation. However, it is impossible to anticipate when a particular piece of knowledge will be available to the understander, since the real world (in the case of a story understanding program, the story) will not always provide exactly that piece of knowledge at exactly the time that the understander requires it. *Thus the understander must be able to suspend questions in memory, and reactivate them at the right time when the information it needs becomes available.* In other words, the understander must be able to remember what it needs to know, and why.

Understanding tasks, therefore, generate information subgoals or *questions*, representing what the understander needs to know in order to carry out the current task, be it explanation, learning, or any other cognitive task. These questions constitute the specific knowledge goals of the understander. Reading is a process of seeking answers to these questions in the input, which in turn raises new questions while answering old ones.

We have argued that modelling goal-driven behavior is a reasonable task from the point of view of cognitive modelling, and also that the ability to suspend and reactivate questions is essential to

¹Since computers will eventually be expected to interact with the physical world (e.g., robots) and the social world (e.g., employees), they will also be expected to have some of the physical or social goals that we currently attribute only to people.

deal flexibly with real world input. In addition, we should also provide functional arguments for why programs (or humans) have goals. What difference do goals make? Why should we design our programs to be able to represent and reason about their own goals or questions? This dissertation will attempt to demonstrate that:

- Questions can be used to focus attention on interesting facts, and limit the inferences drawn from them.
- The explicit representation of program goals (questions) facilitates the discovery of unexpected opportunities to satisfy the goals (answer the questions).
- The understander learns by answering its questions and raising new ones.
- Viewing reading as a goal-driven process provides a natural way to integrate other cognitive processes that could also be used as plans to pursue knowledge goals, such as case-based reasoning, analogy or tweaking.

In addition, if questions are used to focus attention, it is important to ask the “right” questions to make sure that the understander focuses on the relevant issues, so that it does not miss the point of the story. *The depth of understanding that the understander achieves depends on the questions that it asks.* Thus, in addition to discussing the processing issues mentioned above, we will also discuss the types of questions that an understander might ask. Of particular interest to the problem of understanding goal-based stories is the underlying model of motivational explanations, which will be discussed along with the questions that it gives rise to.

Let me now describe this view of understanding, and the assumptions that underlie it, in more detail.

1.1.3 Knowledge structures aren't perfect

Conventional script, frame or schema-based theories assume that understanding means finding an appropriate script, frame or schema² in memory and fitting it to the story. Schemas in memory are assumed to be “correct;” if an applicable schema is found, the story is understood.

However, this model is inadequate since an understander's memory is always incomplete. Knowledge structures often have gaps in them, especially in poorly understood domains. These gaps correspond to what the understander has not yet understood about the domain. Even if a schema appears to be correct, novel experiences or stories may reveal flaws in the schema or a mismatch with the real world. Furthermore, the schema may not be indexed correctly in memory.

1.1.3.1 Mis-indexed knowledge structures

If a ready-made schema cannot be found that fits the story perfectly because the schema isn't indexed correctly in memory, a conventional schema-based understander would simply fail. However, in a

²I will use the neutral term *schema* for knowledge structures that encapsulate “canned” or stereotypical information about some situation, when for the purposes of the discussion at hand the distinctions between various kinds of schemas (scripts, frames, MOPs, explanation patterns, etc.) are irrelevant.

realistically sized memory, knowledge structures would not always be indexed along with every conceivable scenario that they could be relevant in, nor could one expect the understander to try all possible structures in each situation it encountered. For example, one couldn't expect the blackmail schema to be associated with suicide bombing *a priori*, nor would one want to pull up blackmail as a possible hypothesis each time a suicide bombing is encountered. This is a central issue that understanders must be designed to deal with.

1.1.3.2 Gaps in knowledge structures

Even if a relevant schema is found, it is often the case that the understander's knowledge of the scenario in question will be incomplete. Clearly it would be impossible to stuff in all the knowledge about every situation into our machines. Instead, we would expect many schemas to be incomplete, or to have questions attached to them representing what the understander didn't understand about those situations in the past, or simply to be very sketchy.

In the blackmail story S-2, for example, even if one assumes that blackmail was known to be a possible explanation for suicide bombing, it could hardly be a fully-fleshed out causal chain. Instead, there would be many gaps in that chain, many questions attached to that explanation: *How does one blackmail a child? What could anyone want more than their own life?* In such a situation, there is more to understanding a story than fitting it into a schema; the understander has the opportunity to answer some of these pending questions if the story happens to provide these answers, thus improving its understanding of the scenario represented by the schema.

1.1.3.3 Incremental refinement of knowledge structures

Even if the schema looks perfect to begin with, new stories that differ in small but interesting ways would raise new questions about the scenario, questions that the understander could not have anticipated in advance. The religious fanatic explanation for suicide bombings in the Middle East, for example, is one that most people have and apply automatically, but there are many stories about fanatics that are just a little bit different from the standard scenario. For example, consider applying the religious fanatic explanation to the following story (Los Angeles Times, April 10, 1985):

S-3: 4 Die in Suicide Attack Against Israelis.

JERUSALEM — A young guerrilla driving a car filled with explosives blew it up in a suicide attack Tuesday against a group of Israeli guards in Lebanon, killing three and wounding two others, an Israeli military source confirmed.

Besides the driver, identified by the guerrilla group as a 16-year-old girl, the blast killed two Israeli soldiers and a Druze member of the Israeli-backed South Lebanon Army. The wounded were Israeli soldiers, the source said. ...

A spokesman for the group identified the driver as Sana Mohaydaleh. In an interview taped before the incident and shown Tuesday after the attack on Beirut television, a girl identified as Mohaydaleh said: "I hope that my soul will unite with the souls of all the martyrs before me. I decided on martyrdom to free our land because I saw the misery of my countrymen under the occupation. I hope I will be successful and able to kill the highest number possible of our enemies."

Assuming that the understander starts out knowing about religious fanatics, this story fits pretty well into the stereotypical scenario. Conventional understanders would be able to apply the schema to the story, but they would ignore the novel aspects of the story which are often the most interesting aspects. Instead, an understander should be able to focus on these aspects of the story and learn from them by modifying its schemas to accommodate these variations. This kind of *incremental learning* is an essential part of the understanding process.

While reading story S-3 above, an understander might think about why these suicide bombers are all teenagers, or perhaps about the strange fact that a television interview with this girl was videotaped before the mission. Is there something general to be learned by reading this story? Certainly, these issues are more interesting than other more stereotypical facts in the story, such as what kind of car was used or how many people were killed. These are the issues that the understander should focus on in this story.

1.1.4 What is the output of the understanding process?

This has always been a controversial issue since the answer obviously depends on why one is parsing in the first place. For a front-end to a database, the output might be a sequence of database commands. For the syntactic processor, the output might be a syntax tree. Traditionally, parsers have always output some sort of representation of the input. In addition, story understanding programs are often designed to answer questions to demonstrate that they have adequately understood a story (e.g., [Lehnert, 1978]).

In contrast, we claim that *asking* questions is itself central to understanding. The point of reading is to answer one's own questions, to fill in the gaps in one's memory structures. The point of reading is to *learn*. An understander builds a representation of a story only to the extent that it needs to relate the story to its questions. In doing this, of course, several new questions might be raised.

For example, suppose the understander has read many "religious fanatic does suicide bombing" stories and has a well-understood schema to explain this type of event in memory. Now suppose it encounters the blackmail story S-2. What would the output of the understander be? Of course, the understander would build a partial representation of the story. But along with that, it would ask several new questions, arising from the conflict between its stereotypical schemas and the novelty of this story. It would learn a new explanation for suicide bombing, "blackmailed into suicide bombing." It would form a new index to associate "blackmail" with "suicide bombing." *The changes that would occur in the understander as a result of having read the story are the real outputs of the understanding process.*

How do these changes affect the understanding process during the rest of this story, and during future stories? In story S-2, it turns out that the boy's father had been involved in a very expensive car accident, and the boy was blackmailed by using this as a lever to threaten his family's future:

S-2: (continued) Abu Hassan [the Amal security chief] sent a messenger to [Mr. Burro], telling him to come to his office for a talk. The meeting, Mr. Burro recalled, was a carefully balanced combination of inducements and threats. ...

"You know your father has had this accident," Abu Hassan said, "and your financial conditions are not good. ... We have the power to take care of everything. ... The future of your family is in your hands." ...

The youth was told that if he did not agree to the suicide mission, Abu Hassan would cause problems for his father. Abu Hassan also apparently hinted that he was ready to cut off Mr. Burro's only source of income.

Mr. Burro made it clear that he had been terrified of Abu Hassan and of what he might do to his father.

Even if the understander already has a "blackmailed into suicide bombing" schema, it is unreasonable to assume that this schema will be complete in every detail. Instead, there would be questions attached to this schema, representing what the understander was left wondering about after learning that schema: *What could the person being blackmailed value more than his own life?* When the understander attempts to apply its "blackmailed into suicide bombing" schema to this story, *the result isn't simply an understanding of the new story; it is the answering of these questions.*

Thus the result of understanding consists of answers to old questions about the domain, plus of course a set of new questions. In contrast to the traditional view of understanding as a "story in, representations out" process, therefore, understanding is better described as a "questions + story in, answered questions + new questions out" process, as depicted in figure 1.1.

Since the point of reading is to try to answer one's questions, one could think of questions as *knowledge goals* or *cognitive goals*, representing "things that the understander wants to find out." Thus we call the process described in this dissertation *goal-driven* or *question-driven understanding* [Schunk and Ram, 1988].

1.1.5 Understanding is subjective

Consider the following story (New York Times, Aug 10, 1984):

S-4: Garbage can bomb kills 3 in Beirut.

BEIRUT, Lebanon, August 9 — A bomb concealed in a garbage can exploded today in Moslem West Beirut, killing 3 people and wounding 15. Within minutes, armed [Shiite] militiamen rushed into the streets to assert control. ... The Shiites stopped cars to check documents and they took 9 or 10 people into custody. ...

The radio station of the Christian militia, the Voice of Free Lebanon, noted that all the detained were Christians. It said they had been beaten and added that the treatment proved that "any stick was good enough for beating Christians." The bombing, it said, "was obviously not a Christian affair."

What would it mean to have understood this story? Obviously, different people would understand it differently. Someone not interested in the Middle East problem might not read it at all. A reader interested in the mechanics of terrorist incidents might concentrate on the unusual instrument (the garbage can) that was used. Another reader might read this story as a "yet another crazy fanatic in Lebanon" story, and not remember any of the details at all. Yet another reader might think about the fact that it is now the Christians that are using terrorist tactics against the Moslems, whereas it had always been the other way.

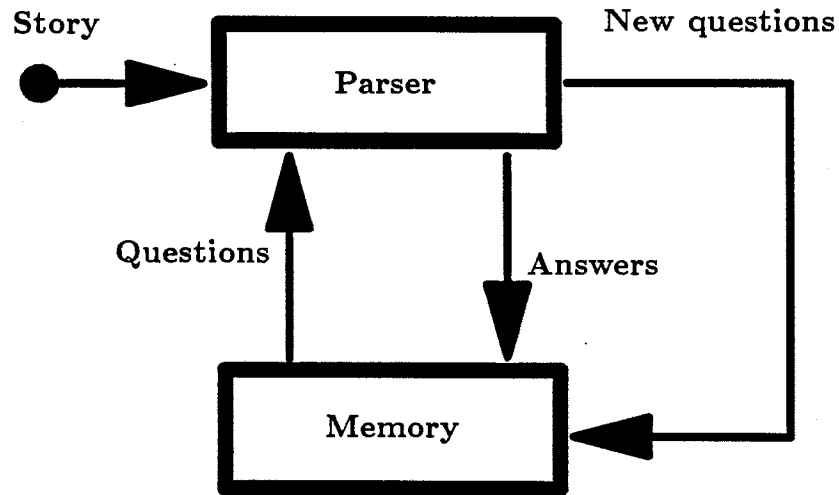


Figure 1.1: Question-driven understanding. In contrast to the traditional view of understanding as a “story in, representations out” process, we view understanding as a “questions + story in, answered questions + new questions out” process. A theory of understanding, therefore, must include a theory of memory and learning in addition to a theory of parsing.

Understanding is a subjective process [Carbonell, 1979]; the understander’s beliefs and hypotheses determine how the story is interpreted. This is a natural consequence of the theory of question-driven understanding since it is the questions in memory that determine which inferences are drawn. This will be discussed in more detail below.

1.1.6 Understanding is dynamic

The idea of subjective understanding does not only apply across readers; even the same person would read a story, or another like it, differently the second time around. Depending on the questions that were raised by a story, one would read future stories differently. For example, consider reading the following story (New York Times, February 27, 1986) before and after one has read the blackmail story S-2:

S-5: Lebanon car bomb kills driver, hurts 7 at Palestinian site.

BEIRUT, Lebanon, February 26 — A car bomb exploded today at the entrance of the largest Palestinian refugee district in southern Lebanon, killing the driver and wounding seven people.

The police said the explosion occurred outside the Ain Khilwe camp, near the port of Sidon A guard at the entrance of the camp ... said he saw the driver trying to get out of the car. "He struggled with the door, then the whole car exploded with him inside," the guard said.

On the surface, this story looks like a stereotypical suicide bombing story. The only quirk in this story is that the driver appears to have changed his mind. But this is pretty understandable; perhaps he was frightened and changed his mind at the last minute. Or perhaps it was part of the plan all along that the driver would jump out of the moving vehicle.

However, if one has just read the blackmail story, one could view this story as confirmation of the hypothesis that suicide bombers do not volunteer for these missions, they are forced into them by extortion (or perhaps exhortation). This story does not prove that the hypothesis is "true," of course. The point here isn't whether one can prove one has the "right answer." The point is that *the questions and hypotheses currently in memory can affect one's interpretation of a story*, or at least the thoughts that cross one's mind as one reads. Since these questions and hypotheses change as the understander reads and learns more about the domain, the understanding process is dynamic. A story would be understood differently depending on what the understander had read previously. In particular, an understander would process a story differently if it were to read the same story again, since its goals in reading the story, the questions it is interested in, would be different.

1.1.7 Controlling inferences

The combinatorial explosion of inferences has always been one of the classic problems in AI. Resources are limited, and inferences potentially infinite; the understander needs to be able to determine which inferences are useful to draw from a given piece of text. But unless one considers the goals of the understander, it is very difficult to give a principled definition of what it means for an inference to be "useful." Since questions represent the understander's goals, the understander should draw those inferences from the text that will help it answer its questions.

Of course, it isn't possible to draw every possible inference from the text hoping to answer a question somewhere in memory. Instead, questions that are currently active should be used to focus the understander and guide inferencing. The understander processes text only as far as it appears to be useful to its goals. This is a heuristic process only since there is no way to guarantee that only useful inferences will be drawn. Thus the understander would expect to miss some connections, as well as find others that it was not expecting.

We call this process *graded parsing* or *variable-depth parsing*. Every new fact read by the understander is processed to some extent; this bottom-up processing draws the basic inferences without which it couldn't do anything at all. For example, when it reads a sentence, the understander always does the work necessary to figure out the main action in the sentence, as well as its actor. It then tries to see if the new fact could address any of the questions that it has. If the fact looks potentially relevant to some of its questions, the understander then processes that fact further, using those questions to drive further inferencing. To understand this process, consider the following sentence:

S-6: John dislikes chocolate.

This sentence would be processed into a representation of John's goal of avoiding chocolate, or perhaps of John's dislike for chocolate³. But now there are many inferences the understander could draw from this, some of them plausible but unlikely, some implausible but not impossible. Instead of drawing all these inferences, the understander tries to retrieve questions from its memory that would be answered by this new fact. If a question is answered, it goes back to why the question was asked in the first place, and restarts the chain of thought that posed the question. For example, if John was a suspect in a murder case, and there were chocolate particles found on the rug near the victim, the understander would draw the inferences necessary to verify the hypothesis that John was the murderer. This is accomplished by continuing the reasoning task (in this case, the hypothesis verification task) that is awaiting the answer, and allowing it to draw the inferences that it needs from the answer. In other words, *the understanding tasks that gave rise to the questions determine the inferences that need to be drawn once the questions are answered.*

1.1.8 The problem of interestingness

This view of understanding gives us a handle on the problem of interestingness [Hidi and Baird, 1986; Schank, 1979]. Questions represent what the understander is *interested* in finding out at any time for the purpose of understanding the story. They are used to determine which are the interesting things to pursue or connect up. Thus the interestingness of a fact in a story depends on how likely the fact is to answer a question one is wondering about, or to raise new questions that would be worth thinking about. Since the main point of reading is to learn about the domain, interestingness can be defined as *a guess at what one thinks one might learn from paying attention to a fact to a question, were one to process it in detail.* For example, since anomalies arise from contradictions between standard explanations known to the understander and unusual situations described in the input, anomalies are interesting because the understander may be able to modify its explanations or even learn a new explanation by reading the story.

As defined above, interestingness is a heuristic to focus attention rather than a precise measure of what one might learn by processing something in detail. This heuristic might miss some connections that would be found by the more detailed inferences that an in-depth parser might make, but people don't always parse stories in great depth either. It is important to guess at the interestingness of a fact without processing it in detail, because otherwise the purpose of focussing attention on interesting facts in order to control inferences would be defeated.

Furthermore, this notion of interestingness is dynamic; the interests of the understander change as it reads, which is of course true for people too. A system must be able to determine the interestingness of a fact dynamically with respect to its goals (questions in the case of an understanding system).

1.1.9 Opportunism and the memory model

It would, of course, be impractical to maintain a list of "pending questions," and check every question in that list every time a new fact was encountered. Thus questions, along with suspended understanding tasks that gave rise to them, must be indexed in memory. The memory model is

³The particular representation used is not relevant to this discussion.

based on the theory of dynamic memory [Schank, 1982; Kolodner, 1980; Lebowitz, 1980]. Rather than build an independent representation, which then has to be integrated into memory and related to previous episodes as a separate stage in the understanding process, an understander parses text directly into memory and tries to relate it to its questions. In turn, this memory, and in particular the questions currently in memory, are actively used to guide the parser as described above.

Since questions are indexed in memory, it is quite likely that an understander will find the answers to questions other than the ones it is currently thinking about as it reads. In other words, knowledge goals can be satisfied *opportunistically* during the course of understanding. These questions could be ones raised earlier in the same story, or even during some other story that it read in the past.

For example, one of the questions that is raised during many suicide bombing stories is why they choose kids for these missions. One doesn't actively think about this question all the time, of course. However, one often finds answers to questions like these even when the question is only "at the back of one's mind." Consider the following story (New Haven Register, August 6, 1988):

S-7: 200,000 children in world's armed forces, study finds.

GENEVA — An estimated 200,000 children are under arms worldwide, most of them forcibly recruited but some urged to enlist by their parents ...

According to the survey, examples of the problem reportedly included:

- Illegal street roundups in Afghanistan to recruit youths under 15.
- Abduction of boys under the legal draft age of 18 by army recruiters in El Salvador.
- Lowering of Iran's conscription age to 13, with voluntary enlistment by parental consent for younger children.
- Introduction by South Africa of compulsory military training at age 16 ...
- Use of volunteers under age 15 in Honduras and Morocco.

This story provides an answer, in fact, several possible answers, to the question of why kids are recruited for suicide bombing missions in Lebanon. If the understander is interested in this question and is reminded of it when it reads this story, it can use this opportunity to answer the question, or propose new hypotheses. Thus questions can be answered *opportunistically*, either within the same story in which they were raised, or during future stories.

1.2 The AQUA program

AQUA⁴ is a question-driven story understanding program that reads newspaper stories about terrorism [Ram, 1987; Schank and Ram, 1988]. AQUA builds causal and motivational explanations for the events in the story in order to understand why the characters acted as they did, or why certain events occurred or did not occur. AQUA has been designed as a question generation program. The program asks questions as it processes a story, and then uses these questions to drive the understanding process. As a consequence, it is interested in those facts that are relevant to the questions that it currently has. These questions represent *knowledge goals*, which are cognitive subgoals of the basic task of processing and understanding the text.

⁴ *Asking Questions and Understanding Answers.*

AQUA implements a theory of explanation-based understanding cast in a question-driven framework. In addition to the "factual" stories that many understanding programs deal with, AQUA reads "human interest" stories in the terrorism domain. It can achieve a better level of understanding of these stories than conventional script-based understanders, since its questions need not only be generated from slots in stereotypical scripts or MOPs.⁵ In fact, many interesting questions about a story arise from the detection of anomalies in the story, and the construction of explanations for those anomalies. These questions are used to guide understanding in the same way as questions from other sources.

AQUA's basic goal in reading is to answer its questions and to improve its understanding of the domain (terrorism). Thus AQUA's output consists of answers to old questions about the domain, plus of course a set of new questions. Rather than having them be pre-programmed, AQUA generates those questions based on its understanding goals, i.e., AQUA's questions are *functional* in explanation and understanding. Questions are indexed in an opportunistic memory, along with suspended understanding tasks that are awaiting the answers. Thus AQUA knows not only what it wants to find out, but why it needs that piece of information.

Since questions represent the knowledge goals of the understander, they should also provide the focus for learning. In addition to asking questions, therefore, AQUA can learn from answers to these questions. As implemented, AQUA improves its explanatory knowledge of its domain by incremental refinement of this knowledge using answers to questions that arise from the explanation process.

1.3 Issues addressed in this thesis

We have described understanding as a process that is driven by one's questions or knowledge goals. In fact, parsing can be viewed as a method or a plan that a system might use to try to answer these questions; other plans a system might employ include various inferencing techniques, such as analogy, case-based reasoning, default reasoning, and so on. Our theory of question-driven understanding addresses the following issues:

1. **Questions:** What kinds of questions are there? In other words, what are the knowledge goals of an understanding program?
2. **Inference control:** How do questions guide processing? How can they be used to focus attention on relevant input? Which inferences should be drawn from a piece of text? How much processing should be performed on a new fact, given that resources are limited? Are there a set of bottom-up inferences that it is always useful to make?
3. **Subjective interpretation:** How do questions affect interpretation? What is the output of the parsing process?
4. **Opportunistic memory model:** How do we get reminded about questions raised in the past, other than the ones that we are currently thinking about? What happens to memory, and the questions in memory, as we read?

⁵A MOP, or *memory organization packet*, is a memory structure which contains information about how other memory structures are linked together in frequently occurring combinations. MOPs are composed of *scenes* which describe how and where a particular set of actions take place, which in turn can point to *scripts* that embody specific aspects of these scenes [Schank, 1982].

5. **Integrated processing:** Where do higher-level processes, such as explanation, case-based reasoning, default inference, analogy, etc., fit in? Do these come after parsing, or are they integrated together? Is there a basic mechanism that could be used to integrate higher-level processes with parsing actions in a uniform way?
6. **Learning:** What does one learn from novel answers to one's questions? How does memory change during understanding? How does this affect the processing of future stories?
7. **Interestingness:** Are some kinds of questions more interesting than others? How do we determine what is interesting and what is not? Should we spend our limited resources on processing a new fact?

One could view questions as being similar to the "predictions" or "expectations" of previous schema-based understanders, which arise from slots in schemas that are activated during the understanding process. The differences lie in the types of questions that AQUA asks, the knowledge structures that give rise to the questions, and in their use by the understander. In a sense, this dissertation is an attempt to raise the standard of the questions that our programs are capable of asking, to the point that questions begin to look less like "predictions" about the story and more like knowledge goals or interests of the program. Chapter 2 discusses these issues in greater detail.

Chapter 3 presents a general theory of motivational explanations that can be used to understand goal-based stories. Chapter 4 discusses the theory of questions based on a taxonomy of understanding tasks, including explanation questions which are based on the theory of motivational explanations of chapter 3. The model of opportunistic memory is discussed in chapter 5, along with processing issues related to question-driven understanding and their impact on the design of the understander. Learning and interestingness are discussed in chapters 6 and 7, respectively. Chapter 8 discusses the implementation of AQUA, and presents a detailed program transcript to illustrate how the different aspects of the question-driven understanding process are integrated together. Chapter 9 discusses the range of stories that this approach can handle, and concludes with a discussion of goal-based or interest-based programs.

Chapter 2

Previous work in story understanding

2.1 Integrated processing

One of the central problems in natural language understanding is the integration of higher level reasoning processes, such as explanation, case-based reasoning, default inference, analogy, and so on, with parser level processes in a uniform and psychologically plausible way. Do higher-level processes come after parsing, or are they integrated? Are they implemented by different mechanisms, or does the same basic mechanism underlie both kinds of processes?

We believe that understanding is an integrated process. The understander brings all the relevant knowledge to bear as early as possible, be it syntactic, conceptual, or explanatory, in order to make the best use of it. This is called the *integrated processing hypothesis* [Birnbbaum, 1986; Schank *et al.*, 1978; DeJong, 1979; Lebowitz, 1980; Schank and Birnbbaum, 1984].¹ However, there is still the issue of the *mechanism* by which the understander brings different kinds of knowledge to bear. Traditionally, parser-level questions or *requests* [Riesbeck, 1975], such as those determining actor-action relationships or noun group meanings, have been generated and used by parser-specific mechanisms, whereas higher-level knowledge, such as that used for explanation, is stored in the memory of the system and applied at a later stage.

Early understanders usually had two separate modules, a parser and an inferencer (e.g., SAM [Cullingford, 1978], ELI [Riesbeck, 1975], Wilks's parser [Wilks, 1973], LUNAR [Woods *et al.*, 1972]). Even in systems that were integrated, semantic and pragmatic processing was motivated by syntactic analysis (e.g., SHRDLU [Winograd, 1972], PARSIFAL [Marcus, 1980]). Later understanding systems moved towards more integrated models (e.g., SOPHIE [Brown and Burton, 1975], POLITICS

¹This hypothesis is in contrast with modular theories of language understanding, which assume the existence of an autonomous subsystem for syntactic analysis (e.g., [Fodor *et al.*, 1974; Marcus, 1980; Fodor, 1983]). This subsystem is assumed to be either completely separate from the inferential subsystem, or able to interact with it in a limited manner. The issue of integrated vs. modular processing is discussed in [Birnbbaum, 1986; Schank and Birnbbaum, 1984; Fodor, 1985].

[Carbonell, 1979], IPP [Lebowitz, 1980], FRUMP [DeJong, 1979]). These were usually prediction-based or "top-down" (e.g., SAM [Cullingford, 1978], Ms. MALAPROP [Charniak, 1977b], PAM [Wilensky, 1978], POLITICS, IPP, FRUMP, DMAP [Riesbeck and Martin, 1985]). AQUA continues the integrated processing paradigm.

AQUA is based on the premise that asking questions is central to understanding. This is due to the interaction between the requirements of the integrated processing hypothesis, and the nature of the real world. The understander needs to be able to bring different kinds of knowledge to bear on a problem, and use it for reasoning as appropriate. But it is impossible to anticipate when a particular piece of knowledge will be available to the understander, since the real world (in AQUA's case, the story) will not always provide exactly the piece of knowledge at exactly the time that the understander requires it. Thus the understander must be able to suspend questions in memory, and reactivate them at the right time when the information it needs becomes available. In other words, the understander must be able to remember what it needs to know, and why. It must be able to ask the right questions, answer them by reading the text, and continue the reasoning processes that the answers were needed for.

The text level model used in AQUA is similar to the text processing models underlying FRUMP, IPP and MOPTRANS. Since AQUA is not so much an extension of previous *parsers* as an extension of previous story *understanders*, text-level questions will receive the least attention here.

In addition to story understanding, AQUA is also a model of memory and learning. IPP was one of the first programs to shift its focus from parsing to memory. Even though it was a story understanding program, the real emphasis was on the interaction of memory with the parsing process, and the evolution of memory as the program read more stories. DMAP [Riesbeck and Martin, 1985; Martin, in preparation] is a good example of a recent program in which memory is integrated with the parsing process. AQUA takes this view further by integrating the explanation process into the theory of memory-based parsing. The emphasis is still on changing memory. However, unlike IPP whose memory changed through feature correlation, AQUA's memory changes through the process of incremental learning of explanations and their indices.

2.2 The nature of stories

In contrast with typical understanding programs that deal mainly with "factual" stories (e.g., FRUMP or IPP), AQUA is designed to deal with human interest stories. These stories focus on goals, motivations and emotions of particular individuals, and offer explanations for their actions.² The stories are interesting because these goals and motivations are often those that the understander would itself possess. These stories are similar in nature to the stories that PAM read, except for the fact that they are newspaper stories.

Since human interest stories are often about anomalous goal configurations, novel motivations, and so on, they are by nature not amenable to script-based processing which is designed for stereo-

²Story understanding also involves issues at the level of discourse structure, such as the text structure of the story, the style of presentation or argumentation, and the intentions of the author in telling the story. These issues are outside the scope of this dissertation.

typical situations. Novel stories provide an opportunity to learn; thus understanding, learning and memory are all related issues to be addressed.

2.3 The nature of predictions

If questions are used to drive processing, one could view questions as predictions or expectations in the traditional sense. The question then arises, what is the difference between traditional expectation-based parsers and the goal-based parser that we are proposing?

The real issue here is the nature of expectations. In ELI [Riesbeck, 1975] and CA [Birnbaum and Selfridge, 1979], predictions are generated by complex, procedurally-oriented word definitions instead of high-level memory structures. In more recent "integrated" parsers, such as POLITICS [Carbonell, 1979], FRUMP [DeJong, 1979], IPP [Lebowitz, 1980], BORIS [Dyer, 1982] and MOPTRANS [Lytinen, 1984], parsing is integrated with the rest of understanding by using high-level memory structures as a source of expectations about the input. In these parsers, background knowledge is used for both parsing and inferencing, and text analysis proceeds using pragmatic expectations in addition to syntactic and semantic knowledge. However, the expectations in these systems are still static, giving a "canned" flavor to these programs. Furthermore, these programs are incapable of understanding novel situations for which they do not have scripts.

For example, let us consider a program such as FRUMP [DeJong, 1979] as a question generation program. FRUMP had a database of "sketchy scripts" (also called frames or schemas) for different situations, such as terrorism and earthquakes. Each script contained a set of slots to be filled in when understanding a story about that kind of situation. For example, the earthquake script had slots for the Richter scale reading, the number of people killed, and so on. The slots, therefore, represented the questions that the system asked every time it read about an earthquake. They also represented the limit of what the system could understand about earthquake stories. FRUMP would miss the point of a story about an earthquake in Pisa in which the Leaning Tower was destroyed, because it simply didn't have a slot for "famous monuments destroyed" in its earthquake script. In other words, it would never think of asking the question, "Which famous monuments were destroyed?"

To extend the range of situations the system can handle, we could, of course, add the missing slot to the other slots in the script, along with all the other slots we might need. But then the system would ask the question "Which famous monuments were destroyed?" every time it read an earthquake story. Since all slots in all scripts are not equally interesting in every situation, it would still have the problem of deciding which slots are interesting in a given situation even if all the required slots were available.

Most story understanders avoid this issue and pursue all of them with equal enthusiasm. For example, the MOPTRANS program was an in-depth parser. Since it did not have explicitly represent its goals or questions in memory, it couldn't skim the text since there was no principled way to determine which facts were relevant to its current goals. An understander should be able to decide which facts deserve extra attention, and process those in more detail while skipping over the irrelevant facts.

Furthermore, it is unreasonable to assume that all the required slots, even if represented somewhere in the system, would be available in all situations that required them. For example, the slots or predictions provided by the blackmail schema would be available in a suicide bombing situation only if the programmer had anticipated the connection between blackmail and suicide bombing. But clearly it would be impossible to stuff all the required knowledge for all possible situations into a machine. We might compromise and stuff in a "lot" of knowledge as a start. But a machine that relied only on previously built-in knowledge would be able to understand just the situations that it was designed for. In order to be considered intelligent, it would need to be able to deal with novel situations that it didn't already have the knowledge to deal with. In the blackmail example, we would like the system to be able to learn to use the predictions from the blackmail schema in situations involving suicide bombing after reading a story that provided the connections for the first time.

2.4 The source of predictions

There is also the issue of where these expectations or questions come from. ELI made conceptual predictions at the textual level, where as IPP made episodic predictions at the story level. In other words, IPP asked what we call "memory-level questions," questions derived from dynamic memory operations. For example, it would ask journalism questions (who, what, where, when) in an effort to fill out these details so that it could build generalizations based on these slots. However, questions need not only be generated from slots in scripts. In fact, many interesting questions about a story arise from the detection of anomalies in the story, and the construction of explanations for those anomalies. These questions are used to guide parsing in the same way as questions from other sources.

Thus the top-down paradigm of matching input strings to pre-existing frameworks remains essentially the same; the real issue is what the framework is, and how it affects the understanding process. In early programs, this framework consisted of conceptual dependency representations [Schank, 1972]. In SAM and FRUMP, the framework consisted of static knowledge structures (scripts), which in IPP and CYRUS was superseded by dynamic knowledge structures (MOPs, scenes) as the focus shifted from parsing to memory [Schank, 1982; Lebowitz, 1980; Kolodner, 1980]. Many other memory structures have been proposed to provide top-down guidance to the understander, e.g., [Minsky, 1975; Bobrow *et al.*, 1977; Charniak, 1977a; Goldstein and Roberts, 1977; Wilensky, 1978].

In AQUA, the "framework" to which input is "matched" consists of questions derived from explanatory structures (in addition to those derived from the dynamic memory structures of previous programs). The "issue skeletons" of FRUMP, in a sense, were proposed in anticipation of the need for larger explanatory structures representing causal knowledge; however, these too were static in the same sense as sketchy scripts were. AQUA's "explanation patterns" or XPs³ are dynamic in the way that its MOPs are, which of course raises the additional issues of how these structures are learned, represented, indexed, retrieved and used.

³XPs are knowledge structures representing stereotypical patterns of causality, and will be described in chapter 3.

2.5 Inference control

The problem of controlling inferences in a principled manner is central to story understanding. Integrated theories of understanding are goal-directed and relatively top-down, although they should be bottom-up to the extent needed to ensure that they are not totally insensitive to unusual input. How much bottom-up processing should be done? In AQUA, an input is processed bottom-up only to the extent required to determine its interestingness to the program.

Most story understanding programs attend to every aspect of the story in equal depth. They either always skim the text, or they process everything in depth. In contrast, AQUA processes text only as far as it appears to be useful to its questions. It tries to draw only those inferences that are likely to be useful. Which inferences are drawn, therefore, depend both on the understander's goals, as well as its current hypotheses about the situation. If an input does turn out to answer a question, the understanding task that is awaiting the answer is restarted, which then determines which further inferences the program draws from that input.

POLITICS used "focus of attention" criteria to determine the subjective interest of input to the understander, which in turn guided the processing of that input. These criteria judged the relevance of the input to the goals and beliefs specified by the ideology of the program. AQUA uses similar heuristics to control inferences by focussing its attention on input that is relevant to what it is trying to find out by reading the story.

In addition, AQUA also attends to input that is unusual or anomalous since such input signal problems with the program's model of the world. In other words, AQUA attends not only to input that is already anticipated (in the sense that it is relevant to questions, goals or beliefs that it already has), but also to input that, though unexpected, provides it with an opportunity to learn more about its domain.

2.6 Interestingness

POLITICS judged the interestingness of an input using focus of attention heuristics based on its ideology, as described above. Since its ideology was static, one disadvantage of this theory was that the interests of the program, though they could be "switched" between its pre-programmed ideologies, never really changed. In contrast, AQUA's interests are based on its questions which are always changing.

Since FRUMP skimmed the text, it too could be said to have a theory of interestingness since it processed only the "most important" points in the story. However, this notion of interestingness was also canned since it had no real theory of interestingness; it was the responsibility of the programmer to put in just those slots that he thought were interesting in a given situation. But it is not possible to pre-determine interestingness in this static way. Instead, the interestingness of various aspects of the input should be determined dynamically based on their relevance to the program's goals.

In other words, interestingness is not a pre-determined property of particular types of situations. It is neither inherent in the situation nor in the system, but rather arises from the relationship between the stimulus and the goals of the system. A system with no goals would have no reason

to find any input more interesting than any other, nor would any particular piece of information be universally interesting for all systems unless they shared the same goals.

The implicit goals of an understanding program are to understand the story, build explanations for anomalies, and develop, verify and refine its world model. These goals give rise to questions, which represent the specific knowledge goals of the program. This dissertation is about how questions arise from the basic goals of an understanding program, and how these in turn drive understanding towards the eventual satisfaction of these goals.

Thus, the real issues are, what questions are extant in memory, and how do these questions relate to the depth of understanding that is required? In AQUA, interest in a concept is triggered by its likely relevance to questions or understanding goals, and continuing interest is determined by its continuing significance to these goals. Thus interestingness is defined functionally as a criterion for inference control towards satisfaction of the program's goals.

Furthermore, questions and interests change as the program reads. AQUA would never read the same story twice in the same way, unless the story was so uninteresting that it failed to learn anything from it. Thus, like IPP, AQUA is based on a dynamic memory, but unlike IPP, memory changes through a process of anomaly detection and explanation rather than feature correlation.

2.7 Subjective interpretation

Early computer models of subjective belief include PARRY [Colby, 1973], BELIEVER [Schmidt *et al.*, 1978], the Goldwater Machine [Abelson and Carroll, 1965; Abelson, 1973] and POLITICS [Carbonell, 1979]. Of these, AQUA is closest in spirit to the POLITICS program. POLITICS demonstrated that an understander's interests and beliefs influence the interpretation of a story. Different understanders may attribute different goals to an actor, which in turn influences how the actor's actions are interpreted relative to these goals.

In addition, an understander focuses its attention on those aspects of a story that affect it personally. This determines which consequences are analyzed and remembered in memory, and which ones are ignored.

The open question here is what it means for something to affect an understander personally. In other words, what are the goals of an understander? In POLITICS, particular ideologies that were already programmed into the system could be tagged as being "interesting" or "personally relevant." In contrast, the approach used in AQUA is to allow the program to evolve its own set of interests that are functional to the purpose of the program (learning about terrorism), by letting the questions that arise from this purpose *be* its goals. These interests can be used to focus attention on those aspects of the story that would help it achieve its purpose. One could say that AQUA has "adopted" the goals that it has questions about, in the sense that it is interested in aspects of stories about such goals as if they were its own.

Birnbaum argues that the understander should be able to prefer interpretations that promote coherence, i.e., those that can be "explained" in the causal context of the story [Birnbaum, 1986]. In addition, if more than one causally coherent interpretation is found, AQUA prefers interpretations

that match its questions and hypotheses if it can find one. In other words, one of the factors in evaluating an interpretation is the degree to which the interpretation bears on current hypotheses.

In BORIS, for example, inferential processing is done after the word has already been disambiguated using selectional restrictions and scriptal lexicons. In contrast, AQUA builds multiple hypotheses for ambiguous words and noun groups, which can be discriminated later if the need arises. For example, word sense disambiguation and noun group connection are done by building text-level hypotheses that can later be specified using a mechanism similar to the specialization mechanism of MOPTRANS. Text-level hypotheses are managed using the same general mechanisms that are used for managing hypotheses built for, say, explaining anomalies.

For example, consider a program such as IPP which disambiguates words by matching scriptal expectations. IPP does not address the issue of which expectations are more important or more likely to be satisfied, which in turn would determine which of the possible interpretations is more plausible or more salient in the context of current expectations. Since AQUA's expectations are derived from explanations and explanatory hypotheses in addition to scriptal lexicons, it can use its general explanation evaluation mechanisms, which are influenced by the goals and hypotheses of the understander, to evaluate its expectations and therefore decide which interpretation to prefer.

2.8 Questions and opportunistic memory

AQUA's memory is built on top of an opportunistic memory architecture [Hammond, 1988; Hayes-Roth and Hayes-Roth, 1979; Birnbaum and Collins, 1984; Birnbaum, 1986; Dehn, 1989], which provides a uniform way to integrate question-based processing for different types of understanding tasks. The underlying memory model is based on the theory of dynamic memory [Schank, 1982], such as that used by IPP [Lebowitz, 1980] or CYRUS [Kolodner, 1980]. In addition, AQUA's memory contains questions indexed along with the concepts in memory, each being represented along with its own reason for being asked. In this respect, AQUA's memory is similar to the memory of the IVY program [Hunter, 1989]. When questions in memory are answered, perhaps opportunistically, AQUA can restart the suspended computation that was awaiting the answer to the question.

The integrated processing hypothesis states that the understander brings different kinds of knowledge to bear during all stages of the understanding process. In order to achieve this, both parser-level and higher-level questions should be used in the same way to guide processing. All these questions, therefore, must be indexed in memory in a uniform manner, rather than being kept on arbitrary lists in different parts of the system. Along with each question, the understander stores a representation of what gave rise to the question, so that when the question gets answered it can continue with that line of reasoning, whether this be a parser-level task, an explanation task, or any other understanding task.

Other predictive story understanders also use predictions to guide understanding, but these predictions or "questions" are not indexed in memory, but rather kept on a list in the working memory of the parser. This has two disadvantages. One, if the list of predictions grows large (as it would in a story of realistic length), the understander must either laboriously match incoming facts to all the predictions on the list, or else it must delete predictions on some arbitrary basis (e.g.,

at the end of every sentence). Two, predictions can only be satisfied if they are explicitly being considered by the program, whereas if they were indexed in memory they could be satisfied in an opportunistic manner.

A further advantage of an explicit representation of questions as memory structures is that a program could reason about its questions. The difference between questions and merely having unfilled slots in a script is that, although both could be thought of as "expectations," in the latter case the program does not know *why* it has those expectations. This is also the difference between having a question and calling a "subroutine." Calling a *factorial* subroutine does not constitute the question *What is the factorial of n?* since neither the subroutine nor the Lisp interpreter can reason about *why* the answer is desired, nor can it decide to "hang" until later information comes in and notify the process that needed the answer. Of course, suspended subroutine calls could be used to *implement* questions; these would qualify as questions if a program could reason about the decision to suspend and restart these calls, and if it could use their presence in memory to guide its processing.

Another way to put this is that questions are grounded in particular subprocesses of the explanation process, and play a central functional role in those processes *which the system can reason about*. There is an *explanation* behind these questions, which the slot-based predictions of programs such as IPP and FRUMP lacked.

2.9 Learning and explanation

Another problem with some story understanders is that, unlike people, they do not learn as a result of their experiences. An exception was the IPP program, which was based on a dynamic memory. The program generalized stories that it read, thus modifying its own memory. However, these generalizations were based on feature correlations between instances; the program had no way of determining which generalizations were meaningful since it could not distinguish between similarities that were coincidental or superficial and those that were causal or explanatory. AQUA's XPs represent causal generalizations and therefore serve as explanatory categories instead of correlational ones.

Although IPP formed categories for expectation failures, it didn't explain those failures, or use them for focussing the understanding process. Furthermore, IPP's expectations, like FRUMP's, were derived solely from slots in its scripts or MOPs. While script-based processing is useful in stereotypical situations, an understander should be able to analyze and explain anomalies, and use its explanations and hypotheses to provide expectations in novel situations. Ideally, therefore, we would like to integrate both processes into a uniform understanding algorithm, using both types of knowledge to generate questions about the input text.

Explanation-based learning programs such as GENESIS [Mooney and DeJong, 1985] can learn new XPs from scratch through the generalization of causal features from a novel story. In addition, however, an understander should also be able to modify or elaborate old XPs incrementally (e.g., [Schank, 1986; Kass and Owens, 1988]). It should also be able to re-use old XPs in new contexts, and learn new indices for these XPs as a consequence.

Since questions or knowledge goals represent what the program needs to know, they provide the focus for learning. IVY is a good example of a model of goal-driven learning [Hunter, 1989]. The program's knowledge acquisition goals represent what it needs to learn. IVY invokes various knowledge planning strategies to satisfy these knowledge goals. This is similar to the goal-driven behavior of AQUA, except that AQUA learns by reading stories instead of through knowledge acquisition planning. In fact, reading can be viewed as a special type of plan for satisfying knowledge goals. Furthermore, IVY's goals or questions were directed towards forming categories in memory for the purpose of diagnosis, whereas AQUA's questions are directed towards detecting anomalies and forming explanations for the purpose of achieving a causal understanding of the domain.

2.10 Conclusion

To summarize, then, while integrated memory-based parsing is the right idea, we would like to build a story understander that:

- asked questions like FRUMP, but not just static questions based on scripty knowledge, so that it could understand novel and non-scripty stories.
- integrated the input into a dynamic memory like IPP, but used a theory of explanation to generate expectations, and detect and resolve anomalies, so that it learned explanatory categories rather than correlational ones.
- explicitly represented its questions or goals in memory like IVY, so that it had a sense of what it was trying to learn.
- used its questions or goals to focus attention on interesting facts like POLITICS, but determined interestingness dynamically using its questions, so that its interests were functional with respect to its understanding goals, as well as dynamic.

In this dissertation, we will describe a natural language understanding system that is designed as a question generation program. The system asks questions as it processes a story, and then uses these questions to drive the understanding process. As a consequence, the system is interested in those facts that are relevant to the questions that it currently has. Thus the Richter scale reading of an earthquake would be interesting only if it was actually relevant to something it wanted to find out, and not simply because it was a slot in the earthquake script.

Furthermore, AQUA can achieve a better level of understanding since questions need not only be generated from slots in scripts. In fact, many interesting questions about a story arise from the detection of anomalies in the story, and the construction of explanations for those anomalies. These questions are used to guide parsing in the same way as questions from other sources.

Chapter 3

Theory of explanation

What would it mean for a machine to be intelligent? The basic requirement is that the machine should be capable of understanding the world around it. In order to do this, the machine must have a lot of knowledge about the world. But clearly there is more to intelligence than containing a lot of world knowledge, for then encyclopedias and data base programs could be said to be intelligent. People use their knowledge to help them function effectively in the complicated world they live in. Furthermore, people know what they know; they are *introspective* about their own knowledge as well as their reasoning processes. This does not mean that their introspection is infallible, of course; people introspect naïvely as opposed to “correctly.” Finally, people can *learn* from their experiences and update their models of the world as a result. If machines are to become intelligent, they must be able to do the same.

The central issue is that of *change*. Intelligent beings change as a result of their experiences. They accomplish this change through the process of attempting to *explain* what they do not understand. When a novel or poorly understood situation is processed, it is interpreted in terms of knowledge structures already in memory. As long as these structures provide expectations that allow the understander to function effectively in the new situation, there is no problem. However, if these expectations fail, the understander is faced with an *anomaly*. The world is different from its expectations. What does the understander need to understand about the failed structures to allow it to learn from the anomaly?

Clearly, the understander needs to know *why* it made those predictions. It also needs to explain *why* the failure occurred, i.e., to identify the knowledge structures that gave rise to the faulty expectations, and to understand why its domain model was violated in this situation. Finally, it must store the new experience in memory for future use. It must understand the causal structure underlying the situation so that the new knowledge can be indexed in memory using features that are likely to be good *predictors* of when this knowledge might be useful.

Thus *explanation* is a central problem in understanding and learning. Explanations need not be based on detailed causal models of the world. Regardless of the level of expertise of the understander in a particular domain, it must try to build causal models based on what it knows before it can

learn more about that domain.¹

3.1 What is an explanation?

The need for an explanation arises when some observed fact doesn't quite fit into the understander's world model, i.e., the understander detects an *anomaly*. An explanation is a knowledge structure that makes the anomaly go away. To illustrate the nature of such a structure, let us consider some candidate explanations for the anomaly underlying the following popular joke:

S-8: Why do firemen wear red suspenders?

- (1) Because it is always raining in New Haven.
- (2) To keep their pants up.
- (3) Because red, the symbol of warning, is the color of the fire brigade's uniform.
- (4) Because red suspenders look funny if they aren't part of a uniform.

Consider (1). This does not seem like an explanation for S-8. The reason isn't that (1) is false, but rather that there seems to be no causal connection between (1) and S-8. Thus it is not sufficient for a proposed explanation to be true; *an explanation must be causally connected to the anomaly*. It must contain a set of premises and a causal chain linking those premises to the anomalous proposition. If the understander believes the premises, the proposition ceases to be anomalous since the causal interactions underlying the situation can now be understood.

However, not all causal structures are explanations. For example, (2) is causally relevant to S-8, but it still doesn't feel like an explanation. To understand why, let us make the anomaly in S-8 explicit. The real question isn't "Why do firemen wear red suspenders?", but rather one of:

S-9: Why do firemen wear only red suspenders? If firemen are a representative sample of the general population, we would expect them to wear suspenders of various colors, and even belts.

S-10: Why doesn't everyone wear red suspenders? If red suspenders are indeed attractive or desirable, we would expect everyone to wear red suspenders, not just firemen.

The reason that the joke is funny is that (2) misses the point of the question. If the point is made explicit as in S-9, (3) is a possible explanation for the anomaly. Alternatively, if the real question is intended to be S-10, (4) is a possible explanation. The point is that, in order to qualify as an explanation, a causal description must address the underlying anomaly.

To state this another way, *an explanation must address the failure of the understander to model the situation correctly*. In addition to resolving the incorrect predictions, it must also point to the erroneous aspect of the chain of reasoning that led to the incorrect predictions. An explanation is *useful* if it allows the understander to learn; the claim here is that *an explanation must be both causal and relevant in order to be useful*.

¹Learning can also be based on models of other domains. Although the problem of analogy is outside the scope of this dissertation, most analogical learning techniques also rely on causal models (e.g., [Falkenheimer-87, 1987; Adelson, 1988; Burstein, 1988; Ranney and Thagard, 1988]).

An explanation, therefore, must address two questions:

1. Why did things occur as they did in the world?
2. Why did I fail to predict this correctly?

The answer to the first question is called a *domain explanation* since it is a statement about the causality of the domain. The second is called an *introspective* or *meta-explanation* since it is a statement about the reasoning processes of the understander. Let us consider the second one first.

3.1.1 Introspective or meta-explanations

One of the questions an explanation must address is why the understander failed to make the correct prediction. This could happen in one of the following ways:

1. **Novel situation:** The understander did not have the knowledge structures to deal with the situation.²
2. **Incorrect world model:** The knowledge structures that the understander applied to the situation were incomplete or incorrect.
3. **Mis-indexed domain knowledge:** The understander did have the knowledge structures to deal with the situation, but it was unable to retrieve them since they were not indexed under the cues that the situation provided.

When an explanation is built, the understander needs to be able to identify the kind of processing error that occurred and invoke the appropriate learning strategy. The theory of question-driven understanding provides a natural way to do this. Since questions arise from understanding tasks, a question in memory represents *why the question was asked* in addition to *what the question is asking about*. The question underlying an anomaly, therefore, represents the chain of reasoning that led up to the anomaly. When the question is answered, the understander uses this chain of reasoning to place the answer appropriately in memory.

For example, if the processing error arose because of a gap in the knowledge structure being applied to the situation, the question generated by that gap represents both the knowledge that is being sought to fill the gap, as well as the fact that this knowledge, when it comes in, should be used to fill in the gap in the original knowledge structure. Similarly, if a question arose due to a mis-indexed knowledge structure, the answer to the question, when available, should be used to re-index the knowledge structure appropriately. The issue of learning will be treated in detail in chapter 6.

3.1.2 Domain explanations

The domain explanation is a causal chain that demonstrates why the anomalous proposition might have occurred by introducing a set of premises that causally lead up to that proposition. For example, consider the question:

²A *knowledge structure* is an encoding of a piece of world knowledge that can be used to understand situations encountered in the world by "fitting" the situation into an instance of the structure. The knowledge structure *applies* to the situation if it matches the situation. An applicable knowledge structure provides an "understanding" of the situation by allowing the understander to package up the relevant details of the situation in a coherent manner.

S-11: Why did John go to the grocery store?

A domain explanation for this question might be the following:

- (1) John wanted to buy some milk.
- (2) John decided to shop at a grocery store so that he could buy the milk.
- (3) John went to the grocery store so that he could shop there.

The premise of this explanation is that John wanted to buy milk. If the understander believes or can verify the premises of an explanation, the conclusion is said to be explained. Explanations are often verbalized using their premises. Thus in normal conversation this explanation would be stated succinctly as "Because he wanted to buy some milk." However, the real explanation includes the premises, the causal chain, and any intermediate assertions (such as (2) above) that are part of the causal chain.

How might an understander construct such an explanation? PAM [Wilensky, 1978] used a set of planning rules connecting typical goals and plans of people, and chained them together to form explanations such as the above. However, this is too inefficient in complicated situations, where the causal chains could be several steps long. To get around this problem, AQUA uses pre-stored explanations for stereotypical situations. These explanations represent standard patterns that are observed in these situations, and hence are called *explanation patterns* [Schank, 1986].

An explanation pattern (XP) is a stock explanation for a stereotypical situation. For example, "religious fanatic does terrorist act" is a standard XP many people have about the Middle East terrorism problem. One might think of them as the "scripts" of the explanation domain. When AQUA sees a situation for which it has a canned XP, it tries to apply the XP to avoid detailed analysis of the situation from scratch.³

Domain explanations can be divided into two broad categories, physical and volitional.

3.1.2.1 Physical explanations

Physical explanations link events with the states that result from them, and further events that they enable, using causal chains similar to those of [Rieger, 1975] and [Schank and Abelson, 1977]. For example, consider the following story:

S-12: Car bomber blows up Israeli headquarters in Lebanon, killing himself.

If the understander has never read a story about a suicide bombing before, it might ask the question "How can a car be used to blow up a building?" The answer to this question is a physical explanation:

³Unlike scripts, however, XPs are flexible since they contain a description of the *causality* underlying a situation in addition to a description of the situation itself. This allows XPs to be useful in novel situations, while retaining the advantages of pre-stored structures in stereotypical situations. Chapter 6 discusses the incremental elaboration of XPs in novel situations.

- (1) A car is a physical object.
- (2) A car can contain explosives.
- (3) A car can be propelled by driving it.
- (4) Explosives can be blown up by the sudden impact of a car colliding with a building.
- (5) A building can be blown up by blowing up explosives in its immediate vicinity.

Thus the explanation is that the bomber drove an explosive-laden car into the building, the impact caused the explosives to detonate, which caused the building to blow up.

3.1.2.2 Volitional explanations

Volitional explanations link actions that people perform to their goals and beliefs, yielding an understanding of the *motivations* of the characters. Volitional explanations thus correspond to the filling out of the "belief-goal-plan-action" chain [Schank and Abelson, 1977; Wilks, 1977; Wilensky, 1978; Schank, 1986], although, as we will see below, we need to expand the vocabulary of this chain in order to model such explanations adequately.

For example, the understander might ask a different question on reading story S-12, such as "Why would someone commit suicide if they are not depressed?" An explanation for this question, such as the religious fanatic explanation, must provide a motivational analysis of the reasons for committing suicide. For this reason, volitional explanations are also called motivational explanations.

Volitional explanations fall into two broad categories:

1. **Abstract explanation patterns** for why people do things. These are standard high-level explanations for actions, such as "Actor does action because the outcome of action satisfies a goal of the actor."
2. **Stereotypical explanation patterns.** These are specific explanations for particular situation, such as "Shiite Moslem religious fanatic goes on suicide bombing mission."

For example, an explanation of type 1 for a suicide bombing story could be "Because he wanted to destroy the Israeli headquarters more than he wanted to stay alive" (an instance of *xp-sacrifice*). An explanation of type 2 would be simply "Because he was a religious fanatic." The internal causal structure of this explanation could then be elaborated to provide a detailed motivational analysis in terms of explanations of the first type if necessary.

Both kinds of explanations are represented as XPs. Stereotypical XPs are categorized according to the abstract XP category they fall into. AQUA resorts to abstract XPs when it is unable to find a stereotypical XP specific to a given situation.

3.2 Decision models: Representing volitional explanations

Since the domain of AQUA focuses on human interest stories, the program deals mainly with volitional explanations. To construct such explanations, AQUA relates the actions in which the characters in the story are involved to the *outcomes* that those actions had for them, the *goals*, *beliefs*,

emotional states and *social states* of the characters as well as priorities or *orderings* among the goals, and the *decision process* that the characters go through in *considering* their goals, goal-orderings and likely outcomes of the actions before deciding whether to do those actions. A detailed volitional explanation involving the planning decisions of a character is called a *decision model*, and is illustrated in figure 3.1.

3.2.1 The outcome of an action

Every action results in some set of states that may or may not be beneficial to the people involved in that action, depending on their goals at that time. The outcome of an action, therefore, must be modelled *from the point of view of a particular volitional agent* involved in that action. The most common volitional participants are actor⁴ and planner, but any role involving a volitional agent must potentially be explained. For example, in the blackmail story S-2, the boy is the object of the recruiting action, the actor being a group of terrorists. The outcome of this action should be modelled both from the point of view of the boy and from the point of view of the terrorists.

In figure 3.1, the volitional role of an agent is represented by a *volitional-role-relation*, which is a relation between the action or mop and the *volitional-agent*. The mop results in a collection of states, which collectively comprise the outcome of the mop. The *self-outcome* relation shown represents the outcome of the mop as viewed from the point of view of a particular *volitional-agent*. Thus each *volitional-role-relation* has a corresponding *self-outcome*, which could be either *pos* (positive), *neg* (negative) or *mixed*.

3.2.2 The decision process

Every agent involved in an action makes a *decision* about whether to participate in⁵ that particular volitional role in the action. Such decisions represent the *planning* process that the agent underwent prior to the action. A complete model of this process requires a sophisticated vocabulary of goals, goal interactions, and plans, such as that of [Wilensky, 1983] or [Hammond, 1986]. In order to represent finer distinctions in reasoning about motivations, AQUA uses an extended planning decision vocabulary.

There are three basic kinds of decisions:

1. **Choice:** The agent *chooses* to participate or not to participate in a given volitional role in some action. AQUA must be able to model why he made this choice.
2. **Agency:** The agent is *induced* to participate or not to participate in a given volitional role in an action. This is similar to the previous case in that the agent "enters" the action of his own volition. The difference is that here the agent is acting under the agency of another agent, so that AQUA needs to be able to model inter-agent interactions [Schank and Abelson, 1977; Wilensky, 1983; Ram, 1984].

⁴Typewriter font represents actual vocabulary items used by AQUA.

⁵By "participate in," I mean "play a volitional role in," whether this is the role of actor, planner or object.

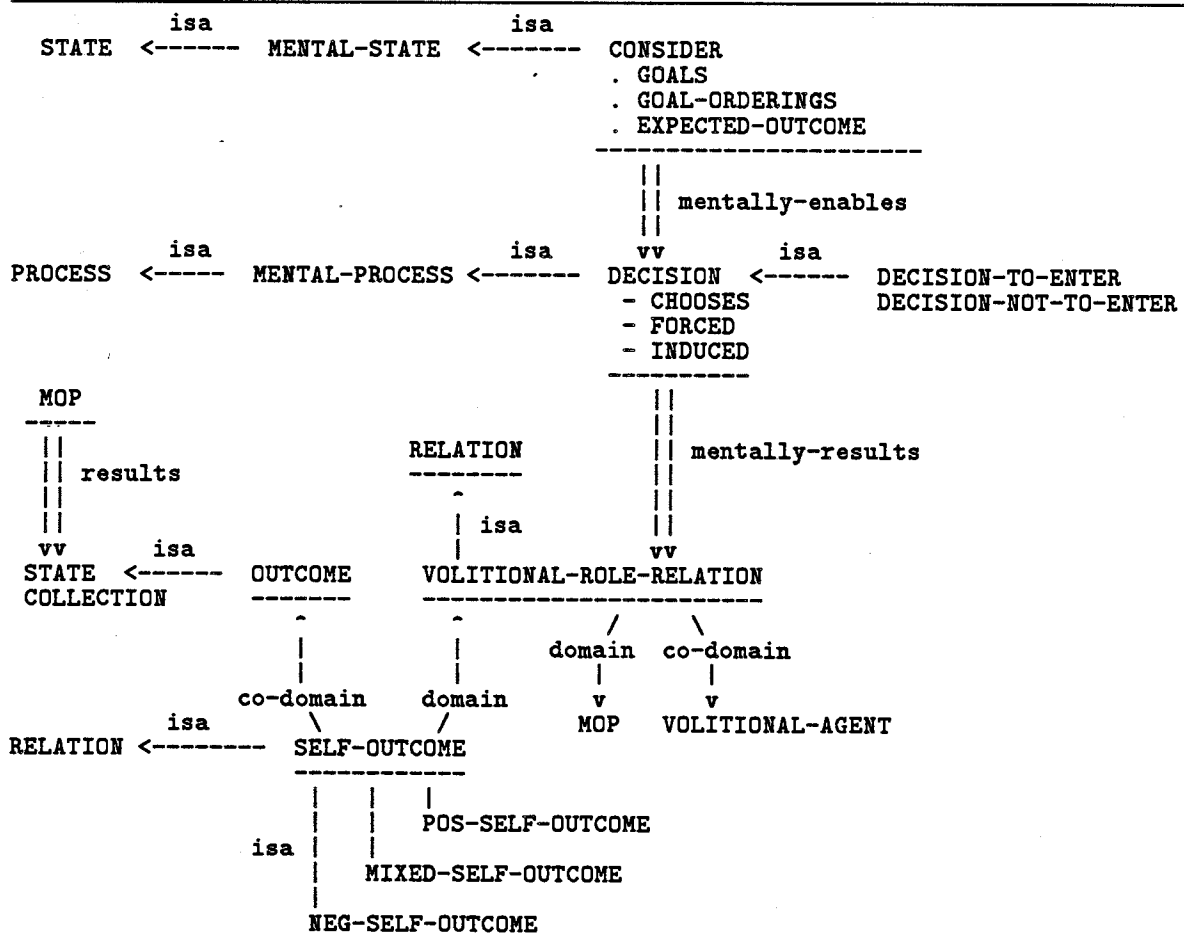


Figure 3.1: The structure of volitional explanations. A volitional-agent participates in some volitional-role in a mop, which then results in an outcome (a collection of states). Prior to this, the volitional-agent undergoes a decision process in which he considers his goals, goal-orderings and expected-outcome, which then mentally-results in the volitional-role-relation being considered being true (in) or false (out) depending on the outcome of the decision.

3. **Coercion:** The agent is *forced* to participate or not to participate in a given volitional role in an action. This case arises when an agent is physically coerced into participation or non-participation.

The decision to participate in an action is represented using a **decision structure** (see figure 3.1), which is a kind of **mental-process**. In our ontology, a **process results** in certain states of the world; thus a **decision mentally-results** in the **volitional-role-relation** becoming true or false (depending on the outcome of the decision), i.e., in the participation or non-participation of a **volitional-agent** in a particular **volitional-role** in a **mop**.

3.2.3 Considerations in decisions

The understander also needs to reason about what an agent was *considering* as he made a particular decision. Considerations model the goals and beliefs of an agent, along with orderings among these goals and expected outcome of the action being considered. Considerations are composed of the following constituents:

1. **Goals:** goals considered by the agent while deciding whether or not to participate in an action.
2. **Goal-orderings:** the agent's prioritization of these goals.
3. **Expected-outcome:** the agent's beliefs about what the outcome of the action is likely to be.

This is represented by the **consider** node in figure 3.1. As shown, a **consider** has the three components mentioned above. In our ontology, a **consider** is a kind of **mental-state** which **mentally-enables** the decision process, which in turn **mentally-results** in the **volitional-role-relation** that the agent was considering.

Each of these constituents may itself need to be explained further. For example, the understander might question the social or mental (e.g., emotional) states that **initiated** a particular goal or goal-ordering in an agent, or how a particular belief about the outcome of an action came about. Explanations, therefore, may need to be *elaborated* according to the demands of the story and the interests of the understander. The decision model representation of planning decisions allows AQUA to perform this kind of reasoning.

Decision models provide a theory of motivational coherence for stories involving volitional agents. When a decision model is applied to the actions of a given character in a story, it may give rise to questions based on faulty assumptions or inconsistencies identified in the application of the decision model to the story. These inconsistencies signal anomalies, which must be explained by determining whether different parts of the decision model (e.g., the goals of the agent, his beliefs about the outcome, or his volition in deciding to perform the action) are actually present as assumed. This is done via a question-based process that will be discussed in chapters 4 and 5.

3.2.4 Components of explanation patterns

Explanation patterns have four main components:

1. **PRE-XP-NODES:** Nodes that represent what is known before the XP is applied. One of these nodes, the EXPLAINS node, represents the particular action being explained.
2. **XP-ASSERTED-NODES:** Nodes asserted by the XP as the explanation for the EXPLAINS node. These comprise the premises of the explanation.
3. **INTERNAL-XP-NODES:** Internal nodes asserted by the XP in order to link the XP-ASSERTED-NODES to the EXPLAINS node.
4. **LINKS:** Causal links asserted by the XP. These taken together with the INTERNAL-XP-NODES are also called the internals of the XP.

An explanation pattern states that the XP-ASSERTED-NODES lead to the EXPLAINS node (which is part of a particular configuration of PRE-XP-NODES) via a set of INTERNAL-XP-NODES, the nodes being causally linked together via the LINKS. In other words, an XP is a causal chain composed of a set of nodes connected together using a set of LINKS (causal rules or XPs). The "antecedent" of this causal chain is the set of XP-ASSERTED-NODES, the "internal nodes" of the causal chain are the INTERNAL-XP-NODES of the XP, and the "consequent" is the EXPLAINS node. The difference between XP-ASSERTED-NODES and INTERNAL-XP-NODES is that the former are merely asserted by the XP without further explanation, whereas the latter have causal antecedents within the XP itself.

Let us illustrate these with some examples.

3.2.5 Example: The religious fanatic explanation

One explanation that AQUA has for the motivations of car bombers in the Middle East is what we might call the "religious fanatic" XP. This is a stereotypical motivational explanation pattern for the reason why religious fanatics perform terrorist acts against opponent religious groups.

In terms of the vocabulary introduced above, the religious fanatic XP can be represented as shown in figure 3.2. The explanation has the following parts:⁶

1. **Explains:** Why volitional-agent A did a terrorist-act M, with results =
 - (1) death-state of A
 - (2) destroyed-state of target, a physical-object whose owner is an opponent religious group.
2. **Premises:**
 - (1) A is a shiite-moslem
 - (2) A is a religious-fanatic, i.e., A has high-religious-zeal
3. **Internals:**

⁶ A detailed computer representation of the religious fanatic XP, showing the PRE-XP-NODES, XP-ASSERTED-NODES, INTERNAL-XP-NODES and LINKS, is presented in section 5.3.1.

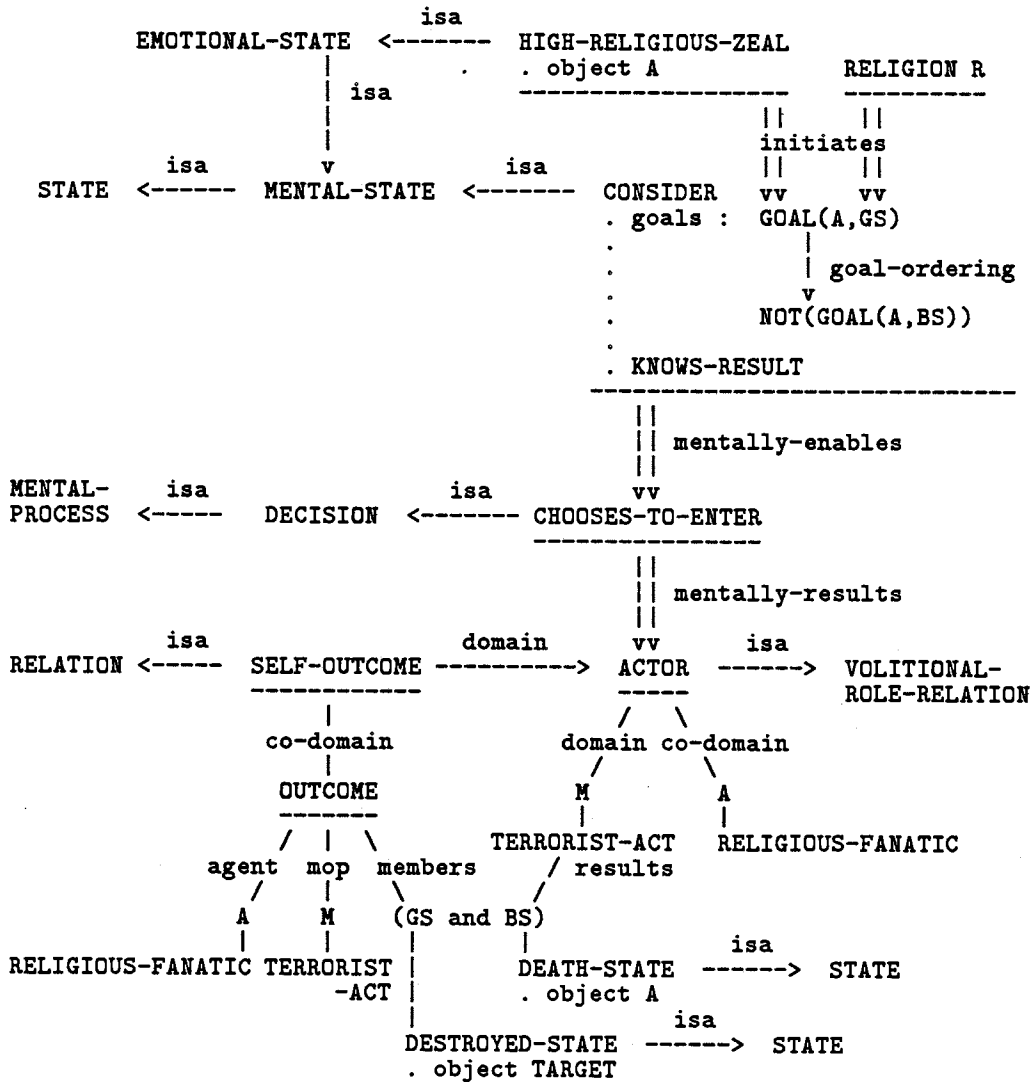


Figure 3.2: The religious fanatic explanation pattern. A is the agent, R his religion, M the action he chooses to do, and GS and BS the good and bad outcomes for A as a result of doing that action.

- (1) A is religious and believes in the Shiite Moslem religion R (an **emotional-state**, perhaps caused by a **social-state**, such as upbringing).
- (2) A is strongly zealous about R (an **emotional-state**).
- (3) A wants to spread his religion R (a **goal**, initiated by (1) and (2)).
- (4) A places a high priority on his goal in (3), and is willing to sacrifice other goals which we would normally place above the religion goal (a **goal-ordering**, initiated by (1) and (2)).
- (5) A believes that performing terrorist acts against opponent religious groups will help him achieve his goal in (3) (a **belief or expected-outcome**).
- (6) A knows that the performance of a terrorist act may result in a negative outcome for him (an **expected-outcome**).
- (7) A weighs his goals (3), goal-orderings (4), and likely outcomes (5) and (6) (a **consideration**).
- (8) A decides to do the terrorist act M (a **decision**, based on the considerations in (7)).
- (9) A does the terrorist act M (an **action or mop**, whose actor is A).
- (10) The terrorist act has some outcome for A, which is either positive or negative as viewed from the point of view of A's goals and goal-orderings (a **self-outcome**).

As shown in figure 3.2, the complete explanation pattern represented as a *graph structure* relating nodes in memory to other nodes. AQUA has several such explanation patterns indexed in memory, which are instantiated to form hypotheses about the motivations of the characters in the story.

3.2.6 Example: The depressed teenager explanation

Let us consider another example of the decision model representation of XPs. This is the "depressed teenager" explanation, a standard explanation for why teenagers commit suicide. The XP is shown in figure 3.3 and is read as follows:

1. **Explains:** Why volitional-agent A did an action M, with results = death-state of A.
2. **Premises:** A is a depressed-teenager.
3. **Internals:**
 - (1) A is a depressed-teenager (a **stereotype**).
 - (2) A is depressed (an **emotional-state**, asserted by the stereotype in (1)).
 - (3) A wants to achieve his death-state (a **goal**, initiated by (2)).
 - (4) A believes that performing action M will result in his own death (an **expected-outcome**).
 - (5) A weighs his goals (3) and likely outcomes (4) (a **consideration**).
 - (6) A decides to do the action M (a **decision**, based on the considerations in (5)).
 - (7) A does the action M (a **mop**, whose actor is A).
 - (8) M results in the death-state of A.

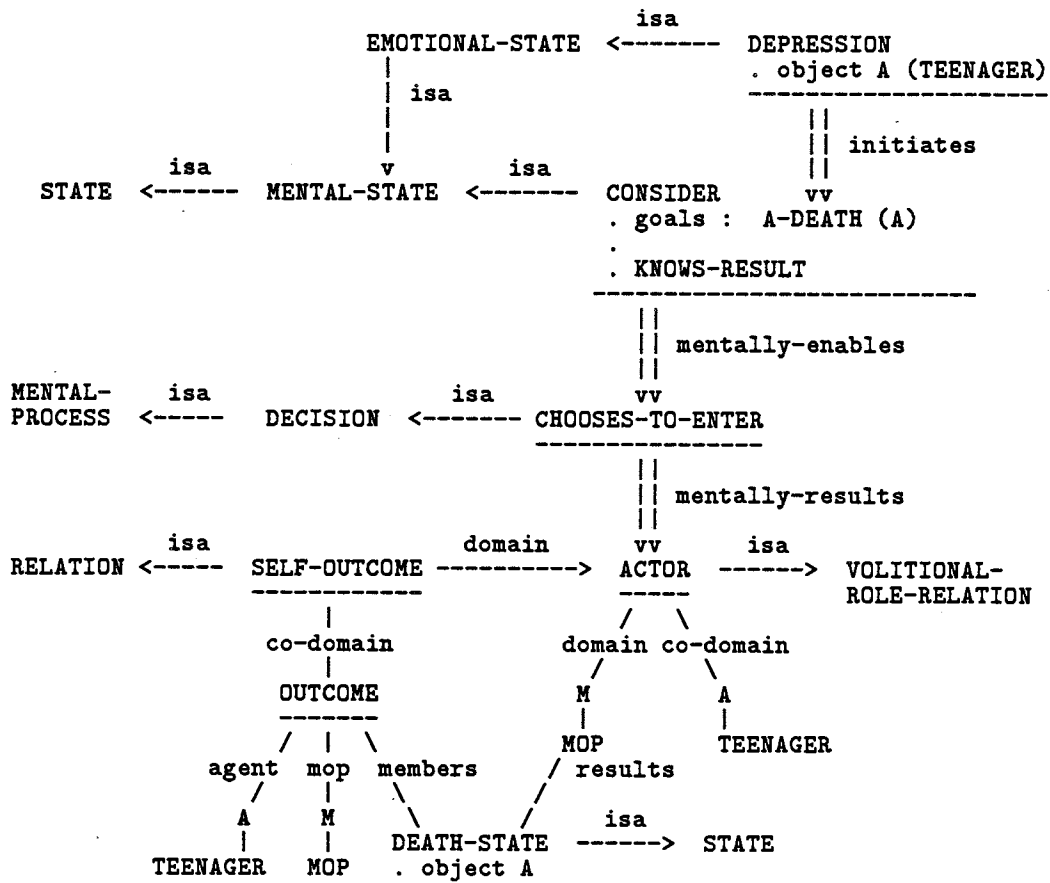


Figure 3.3: The depressed teenager explanation: Depression leads teenagers A to want to die, which leads them to perform actions M that they know will lead to their own death.

3.3 Elaborating explanations

It would be tedious to reason at this level of detail for every action in every story. The purpose of XPs is to package stereotypical causal chains into larger units that can then be used for reasoning without having to consider all the details contained within. The larger units can then be elaborated on demand when the understander needs to get at the details, or they in turn can be composed together to form yet larger explanations. Thus explanations need to be *additive* and *compositional*.

3.3.1 Additivity of explanations

Explanations are additive in the sense that the understander can put explanations together to construct larger explanations. For example, the depressed teenager XP shown in figure 3.3 can be combined with another explanation for how teenagers get depressed in the first place, say, because they have a bad family life. This involves merging the XP-ASSERTED-NODES of one explanation with the PRE-XP-NODES of the other. These will then become INTERNAL-XP-NODES of the combined explanation.

In this example, the combined explanation is that teenagers commit suicide because they have a bad family life, one of the internal nodes of which is the state of depression.

3.3.2 Compositionality of explanations

To be useful, explanations must be expressed at a suitable level of abstractions. For example, suppose the understander is trying to explain the action:

S-13: John drove to the grocery store.

Here are two possible explanations for this action:

- (1) Because he wanted to buy some milk.
- (2) He knew that driving to the store would enable him to buy some milk, and he knew that driving to the store would take up some time, and he preferred to give up that time in order to buy the milk, so he decided to drive his car, which resulted in him being at the grocery store, which was a positive outcome for him since he wanted to buy some milk.

If John is already known to have a goal of getting milk, the explanation that the understander builds should be (1) rather than (2). Although (2) is not incorrect, it is probably unnecessary to reason at this level of detail. Instead, the understander should construct explanation (1), *selectively elaborating* portions of this explanation as it needs to, depending on the task at hand. In order to allow the understander to recover the detailed explanation, causal links in explanation patterns must be complete explanations in themselves. A primitive link, such as the *initiates* link or inference rule [Schank and Abelson, 1977], can be viewed as a one-step causal explanation for a goal in terms of a belief or a mental state. An XP can use either primitive links or other XPs to link together nodes in the explanation.

For example, the religious fanatic explanation shown in figure 3.2 is actually represented in AQUA as shown in figure 3.4. As shown in these two figures, this XP involves both primitive links (*initiates*) as well other XPs (*xp-sacrifice*). The abstract XP *xp-sacrifice* packages up the causality underlying the standard situation of trading off one goal for another. All that needs to be added to *xp-sacrifice* to use it in a larger XP is an explanation of why the actor had the goals and goal-orderings that he did (in this case, because of his religion).

3.3.3 Levels of explanation

One might argue that all volitional explanations can theoretically be derived from a general theory of motivations. Even if this were possible, however, it would certainly be horribly inefficient. XPs are an attempt to associate specific stereotypical causal rules with specific stereotypical situations and people. An understander should try to apply specific XPs whenever they are applicable, resorting to general motivational inference only when necessary. This "specific preempts the general" principle is similar to the "elsewhere principle" in linguistics.

AQUA uses abstract XPs as organizing classes for specific XPs. For example, the religious fanatic explanation is a type of *xp-sacrifice* explanation. When trying to explain an action, AQUA first tries to retrieve a specific XP that applies to that situation. If an XP is found, it attempts to apply the XP to the situation. If there is no specific XP for that situation, or no applicable XP is found, AQUA moves up a level and tries to apply its abstract motivational XPs. Thus for a suicide bombing story, AQUA would first try the explanation "Because he was a religious fanatic" before building an abstract explanation such as "Because the actor preferred to blow up the target even at the expense of his own life."

3.4 Indexing explanations in memory

Ideally, an XP should be indexed in memory such that it is retrieved only in those situations in which it is applicable. But this is impossible in practice. For example, consider the applicability conditions for "blackmail." In general, blackmail is a possibility whenever "someone does something he doesn't want to do because not doing it results in something worse for him."⁷ But trying to show this in general is very hard. Thus, in addition to general applicability conditions, an understander must learn specific, sometimes superficial, features that suggest possibly relevant XPs even though they may not completely determine the applicability of the XP to the situation. For example, a classic blackmail situation is one where a rich businessman who is cheating on his wife is blackmailed for money using the threat of exposure. If one read about a rich businessman who suddenly began to withdraw large sums of money from his bank account, one would expect to think of the possibility of blackmail. However, one does not think of blackmail when one reads a story about suicide bombing, although theoretically it is a possible explanation.

⁷Once blackmail has been retrieved as a possible explanation, it must be verified by confirming that the action was indeed performed under threat from another agent. This is the central to the blackmail explanation, and is represented by the XP-ASSERTED-NODES of *xp-blackmail*.

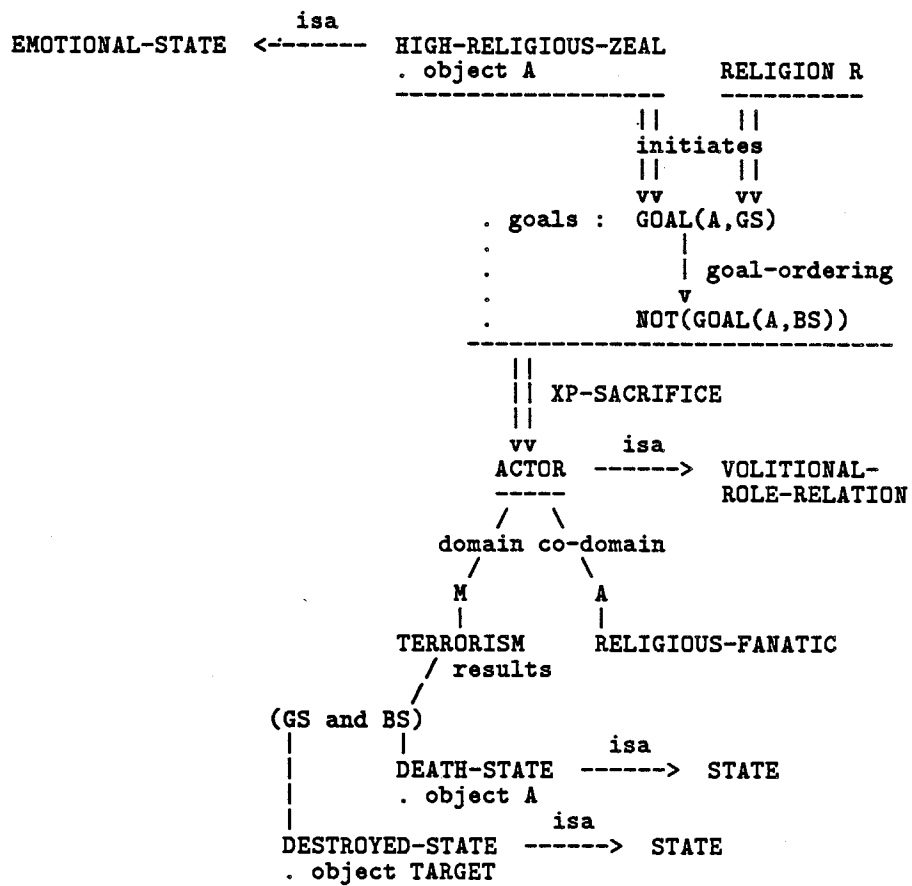


Figure 3.4: The religious fanatic explanation pattern, revisited. The XP invokes a abstract XP, xp-sacrifice, as a causal link to package up the causality of trading off one goal for another.

The point is that XPs are associated with stereotypical situations and people in memory. An understander needs to learn the stereotypical categories that serve as useful indices for motivational explanations. This is a type of inductive category formation [Diettrich and Michalski, 1981]; however, the generalization process is constrained so that the features selected for generalization are those that are causally relevant to the explanations being indexed [DeJong, 1983; Sussman, 1975; Schank *et al.*, 1986; Hammond, 1986; Mitchell *et al.*, 1986]. AQUA uses an explanation-based learning technique for identifying those features that play a causal role in the explanation, which are then used for indexing XPs in memory.

Every XP has three types of indices, the category index, the situation index, and the stereotype index.

3.4.1 Category index

This is the abstract "type" of the XP. The particular anomaly being explained often provides a clue as to which category of XP is needed. For example, if the anomaly was one where an actor violated one of his own goals, AQUA would look for an XP of the type *xp-sacrifice*.

3.4.2 Situation index

AQUA's event memory is based on a MOP hierarchy such as that used by CYRUS [Kolodner, 1980] or IPP [Lebowitz, 1980]. XPs in memory are indexed off the MOP representing the type of situation to which they apply. For example, XPs for suicide are indexed off the *suicide* MOP.

3.4.3 Stereotype index

When one reads a story about suicide, the explanations one thinks of depend on one's perception of the person who committed suicide. If the person is a high school teenager, explanations such as "depressed teenager" or "drug-related suicide" come to mind. If a terrorist commits suicide during a terrorist attack in the Middle East, the "religious fanatic" explanation comes to mind. If the person was a Japanese pilot during World War II, one might think of the "Kamikaze" explanation. Similarly, when one reads about someone infiltrating a computer system with a computer virus, the explanation that comes to mind depends on whether the person is a "high school teenager," an experienced "computer hacker," or a "foreign national."

Thus the understander's perception of the actor is an important index for retrieving XPs. A *character stereotype*, or simply *stereotype*, is a memory structure consisting of a set of goals, goal-orderings, plans and beliefs that describe the motivations of a particular type of person, along with other features a person of that type would be expected to have. Stereotypes serve as *motivational categories* of characters, i.e., they categorize people according to their motivations in performing particular actions.

For example, here is a stereotype of a particular kind of American teenager. The stereotype is composed of the volitional elements that make up decision models in volitional explanations, along with other typical features.

Goals: Entertainment, sex, school, ...

Goal-orderings: Entertainment over school, ...

Beliefs: Not religious, ...

Plans, activities: Dating, studying, riding motorcycles, ...

Features:

- Age: between 13 and 19 (definitional)
- Nationality: American (definitional)
- Religion: none/atheism
- ...

All the features listed in the stereotype are only typical, unless explicitly marked as definitional. Stereotypes model typical collections of volitionality (goals, goal-orderings, plans and beliefs) that make up motivational or volitional explanations.⁸ The features of a stereotype are used for recognition. For example, a 14-year-old American boy who is an atheist is a good candidate for the stereotype shown above. In order to explain a particular action performed by such a person, AQUA would retrieve XPs indexed under that action and this stereotype.

The main constraint on the knowledge represented in stereotype structures is based on the function of these structures in the understanding process. Since stereotypes are used for retrieving explanations, they must provide the kind of discrimination that are needed for indexing XPs in memory. Thus a stereotype must contain a typical collection of goals, goal-orderings, plans and beliefs that together play a causal role in the XP that is indexed under that stereotype, as well as predictive features for these elements. For example, the stereotype shown above would be useful only if there was an XP indexed under it which relied on the goals, goal-orderings, beliefs and plans that are part of this stereotype.

These constraints also apply to stereotypes that are built by the program itself. New situation and stereotype indices for XPs are learned when AQUA reads stories in which XPs are used in novel contexts. This is discussed in chapter 6.

3.5 The explanation cycle

An explanation-based understander must be able to detect anomalies in the input, and resolve them by building motivational and causal explanations for the events in the story in order to understand why the characters acted as they did, or why certain events occurred or did not occur.

The process model for the task of explanation consists of the following steps:

Anomaly detection

- Ask anomaly detection questions

⁸In this respect, stereotypes are similar to the *ideologies* of POLITICS [Carbonell, 1979] and the *personal-themes* of ABBY [Domeshek, 1988].

XP retrieval

- Ask XP retrieval questions
- Retrieve specific XPs indexed under action description and actor stereotype
- Apply specific XP, or abstract XP if no specific XPs

XP application

- Ask XP applicability questions, suspend if necessary
- (Tweak)
- Instantiate nodes
- Instantiate links

Hypothesis verification

- Ask hypothesis verification questions
- Suspend if necessary
- Confirm/refute later when HVQs are answered

The reasoning performed in each step is characterized by the kinds of questions asked during that step, which will be discussed in chapter 4. Here we discuss the processes underlying these tasks.

3.5.1 Anomaly detection

Anomaly detection refers to the process of identifying an unusual fact that needs explanation. The anomalous fact may be unusual in the sense that it violates or contradicts some piece of information in memory. Alternatively, the fact may be unusual because, while there is no explicit contradiction, the understander fails to integrate the fact satisfactorily in its memory. Anomalies fall into two categories:

1. **Physical anomalies:** Reasoning about the physical causality behind the observed events. E.g., if this is the first suicide bombing story one has read, one might think about the question "How can a car be used as a bomb?"
2. **Volitional anomalies:** Reasoning about why the characters in the story acted as they did. E.g., "Why would someone commit suicide if they were not depressed?"

Anomaly detection is done via a set of *anomaly detection questions* that are always asked of each incoming fact. These questions focus the understander on a particular aspect of the situation that might be anomalous. Since the domain of AQUA focuses on human interest stories, the program deals mainly with volitional anomalies. These anomalies focus on the decision that the actor took when he decided to do the action, which is represented using a decision model. Inconsistencies between the story and the decision model signal anomalies; AQUA checks this by questioning different parts of the decision model (e.g., the goals of the agent, or the agent's beliefs about the outcome) with respect to the story.

Story event	Tracking the motivations of:	
	THE BOY	THE TERRORISTS
recruit		why recruit kids?
suicide bombing	religious fanatic?	
blackmail	what's worse than death?	what's worse than death?
threaten parents	family relations?	
indoctrinate		why bother indoctrinate?

Figure 3.5: Modelling different points of view. Different sets of questions arise from the story depending on which decision model one is interested in, requiring different sets of explanations.

In addition to the reasoning about the actor of an action, AQUA also considers the planner's reasons for planning that action, if the actor and the planner are different. This involves building a similar decision model from the planner's point of view. For example, the blackmail story S-2 could be viewed from two different points of view, as shown in figure 3.5.

AQUA builds separate explanations for the actor and planner of every event if they are different. If the actor and planner are the same person or group, the default explanation for the planner is simply that the planner planned the action because he wanted to to carry it out successfully.

3.5.2 XP retrieval

When faced with an anomalous situation, AQUA tries to retrieve one or more XPs that would explain the situation. In order to do this, AQUA asks a series of *XP retrieval questions* about the situation. These questions are also called anomaly resolution questions, since their intent is to find explanations that help resolve anomalies.

As described earlier, AQUA uses three types of indices to retrieve XPs, the category index, the situation index, and the stereotype index. If the anomaly detection process results in an anomaly being detected, the type of the anomaly determines the abstract category of XP that is needed. This is done via a series of questions which are discussed in the next chapter.

Starting from the action being explained, AQUA walks up the MOP hierarchy looking for XPs of the appropriate category that are indexed under the appropriate stereotype. In order to do this, AQUA matches the actor of the action with the stereotype(s) that index explanations for that action. As discussed earlier, this is a heuristic to find XPs that might be relevant without having to do the inferencing required to check if they are indeed relevant, which requires determining whether the causality underlying the situation matches that underlying the XP. If no stereotypical XP is found, the abstract XP is applied.

The planning/decision model underlying AQUA's theory of volitional explanations is powerful enough to represent volitional anomalies, as well as the indices to XPs that will resolve these anomalies. The fact that these anomalies can be derived from this model, as discussed in chapter 4, supports the claim that the model provides a general theory of volitional explanations.

3.5.3 XP application

Once a set of potentially applicable XPs is retrieved, the understander tries to use them to resolve the anomaly. This involves instantiating the XP, filling in the details through elaboration and specification, and checking the validity of the final explanation.

When an XP is retrieved, it is instantiated to form a hypothesis. This involves unifying the EXPLAINS node of the XP with the description of the situation being explained, and instantiating the INTERNAL-XP-NODES and LINKS. If all the PRE-XP-NODES and INTERNAL-XP-NODES of the XP fit the situation, the hypothesis is applicable. If the unification fails, the hypothesis is rejected.⁹

3.5.4 Hypothesis verification

The final step in the explanation process is the confirmation or refutation of possible explanations, or, if there is more than one hypothesis, discrimination between the alternatives. A hypothesis is a causal graph that connects the premises of the explanation to the conclusions via a set of intermediate assertions. Verifying a hypothesis involves walking through this graph and checking to see if its assertions are known or inferrable. For example, although there is no real difficulty in applying the religious fanatic explanation in the blackmail story S-2, the explanation rests on certain assumptions. To verify the hypothesis, AQUA asks what the religion of the boy was, and whether he believed fanatically in that religion. When it reads that "the boy was a Shiite Moslem but not a religious fanatic," it answers these questions and refutes the hypothesis.

To verify a hypothesis, AQUA checks each of the XP-ASSERTED-NODES of the hypothesis which represent its premises. A hypothesis is accepted if all the XP-ASSERTED-NODES are true (in), and rejected if any of these nodes is false (out). This process is repeated for all the possible hypotheses.

When this is done, AQUA is left with one or more alternative hypotheses. Partially confirmed hypotheses are maintained in a data dependency network called a *hypothesis tree* (see section 5.3.2), along with questions representing what is required to verify these hypotheses. These questions are called *hypothesis verification questions* and will be discussed in the next chapter.

3.6 Evaluating explanations

AQUA tries to maintain alternative hypotheses in the order of how good the corresponding explanation is for the given anomaly. There are five criteria for evaluating the goodness of an explanation:

1. **Believability:** Do I believe the XP from which the hypothesis was derived? This is not an issue when all XPs in memory are believed, but for a program that learns new XPs, some of which may be incomplete, the believability of the XP is an important criterion in deciding whether to believe the resulting hypothesis.

⁹There is also the possibility of modifying the hypothesis to fit the situation [Schank, 1986; Kass *et al.*, 1986]. AQUA does not deal with this issue.

2. **Applicability:** How well does the XP apply to this situation? Did it fit the situation without any modifications? Since AQUA does not tweak its XPs [Kass *et al.*, 1986], it does not deal with this issue.
3. **Relevance:** Does the XP address the underlying anomaly? This is ensured by indexing XPs by their abstract types, and using the anomaly characterization to determine the type of XP to retrieve.
4. **Verification:** How definitely was the explanation confirmed or refuted? This can be determined by looking at the HVQs of the hypothesis. AQUA uses a simple measure, the number of unverified HVQs, to determine the degree to which an explanation has been confirmed.
5. **Specificity:** How specific is the XP? Is it abstract and very general (e.g., a proverb), or is it detailed and specific? AQUA prefers specific explanations to abstract ones.

Although evaluating explanations can be quite difficult, the heuristics given above are sufficient for the purposes of AQUA. A general theory of evaluating explanations would need to specify how these and other criteria can be computed using better and possibly domain-specific heuristics (see, for example, [Heider, 1958; Kelley, 1967; McDermott, 1974; Wilensky, 1983; Pazzani, 1988; Leake, in preparation]).

3.7 Conclusion

Explanation is the central component of the understanding process. This chapter described the theory of explanation used in AQUA. The theory is couched in a question-based framework. The following chapters will describe the theory of questions as well as the other components of a question-driven understander.

Chapter 4

Theory of questions

Although questions are the crux of the theory of question-driven understanding, it would defeat the purpose to build a “question-asking” program per se. Instead, questions should arise naturally as *knowledge goals* of the program during various stages of the understanding process. This means that the program should ask a question only when it has a goal to acquire that piece of knowledge, i.e., when it needs to know the answer for the purposes of understanding the story. In other words, questions should be *functionally useful* to the overall goals of the system. Thus before we talk about a taxonomy of questions, it is necessary to develop a taxonomy of *understanding tasks*, the basic tasks of an understander. Having done this, we can use this taxonomy to categorize the different types of questions that the program would ask.¹

In addition to parser-level tasks such as noun group connection, pronoun reference, etc., an understander must integrate facts from the text with what it already knows in order to understand the input. It must be able to detect anomalies, formulate explanations to resolve those anomalies, confirm or refute potential explanations, and learn new explanations. It must be able to figure out what the text relates to in memory, i.e., which knowledge structures in memory should be used to interpret the input, and to what extent it is worth processing.

These are the basic tasks of an understander. These tasks give rise to specific questions or understander goals, which represent the information that the understander needs to perform the tasks. Understander goals fall in four main categories, corresponding to the tasks they are useful for.

1. **Text goals** – Knowledge goals of a text analysis program that arise from text-level tasks. These are the questions that arise from basic syntactic and semantic analysis that needs to be done on the input text, such as noun group attachment or pronoun reference. These tasks are not unique to AQUA, of course; all parsers must perform this sort of processing. In AQUA, however, a question or hypothesis generated from a text goal is indexed in memory and used

¹Note that this taxonomy will not cover all types of questions. For example, AQUA would have no reason to utter indirect speech acts [Austin, 1962; Searle, 1969], examination questions, or questions asked for the purpose of specific perlocutionary effects, such as asking a child a question to get the child to think about something rather than to find out the answer, since these would not arise from its goals as a program that learns by reading newspaper stories.

in the same way as questions and hypotheses arising from higher-level tasks. The details of this scheme, as well as its advantages, will be discussed in the next chapter.

2. **Memory goals** – Knowledge goals of a dynamic memory program that arise from memory-level tasks. A dynamic memory must be able to notice similarities, match incoming concepts to stereotypes in memory, form generalizations, and so on. Questions arising from these tasks are the memory goals of the program.
3. **Explanation goals** – Goals of an explainer that arise from explanation-level tasks, including the detection and resolution of anomalies, and the building of motivational and causal explanations for the events in the story in order to understand why the characters acted as they did, or why certain events occurred or did not occur.
4. **Relevance goals** – Goals of any intelligent system in the real world, concerning the identification of aspects of the current situation that are “interesting” or relevant to its own goals.

Let us discuss the questions that arise from these tasks.

4.1 A taxonomy of questions

A functional theory of questions, as I argued above, must be based on a taxonomy of the understander goals that the questions are in service of. Asking these questions is necessary for the task from which the question arises. In turn, these questions drive the understander to read the story in an effort to find answers. Thus questions represent the *knowledge goals* of the understander, in the sense that they represent what the understander wants to learn.

Understanding, then, is the relating of new information to questions in memory in an attempt to learn general answers to those questions. Before discussing these issues further, it is useful to develop a taxonomy of questions and to discuss some examples.

AQUA’s theory of explanation was presented in chapter 3. The explanation process gives rise to explanation questions. In addition to explanation, AQUA also performs text-level and memory-level tasks, which in turn give rise to text-level and memory-level questions.

Each question focuses on a different aspect of the story. For example, explanation questions focus on different types of anomalies, and on explanations for these anomalies. Asking an anomaly detection question is essential to detecting the corresponding anomaly. For example, asking the question *Does the actor want the outcome of this action?* is essential to the detection of a goal violation anomaly, in the sense that the program could not notice the anomaly if it did not focus on the goals of the agent, i.e., if it did not think of asking the question.

To put this another way, the questions asked by the understander affect the final understanding that the understander comes to. Thus it is important for the understander to ask the “right” questions in order to achieve a detailed understanding of the situation. For the purpose of understanding stories involving motivations of people, we present a taxonomy of motivational questions that focus on those motivational aspects of stories that are needed to build volitional explanations based on the planning/decision model presented in the previous chapter.

The taxonomy of questions is based on the understanding tasks that AQUA needs to perform when it reads a story. The categories will be illustrated using program transcripts showing examples of questions asked by AQUA as it reads a car bombing story.²

4.1.1 Explanation questions

The goals of any explanation-based understander include the detection and resolution of anomalies, and the building of motivational and causal explanations for the events in the story in order to understand why the characters acted as they did, or why certain events occurred or did not occur. The process model for the task of explanation consists of the following steps, repeated here for convenience:

Anomaly detection

- Ask anomaly detection questions

XP retrieval

- Ask XP retrieval questions
- Retrieve specific XPs indexed under action description and actor stereotype
- Apply specific XP, or abstract XP if no specific XPs

XP application

- Ask XP applicability questions, suspend if necessary
- (Tweak)
- Instantiate nodes
- Instantiate links

Hypothesis confirmation

- Ask hypothesis verification questions
- Suspend if necessary
- Confirm/refute later when HVQs are answered

Questions that arise from these tasks are called *explanation questions*. The reasoning performed in each step is characterized by the kinds of questions asked during that step. Let us consider each of these steps from the point of view of these questions.

²Detailed program transcripts are presented in chapter 8.

4.1.1.1 Anomaly detection questions

Anomaly detection refers to the process of identifying an unusual fact that needs explanation. Anomalies fall into two categories:

1. **Physical anomalies:** Reasoning about the physical causality behind the observed events. E.g., if this is the first suicide bombing story one has read, one might think about the question "How can a car be used as a bomb?"
2. **Volitional anomalies:** Reasoning about why the characters in the story acted as they did. E.g., "Why would someone commit suicide if they were not depressed?"

Questions that help the understander in this task are called *anomaly detection questions*. These questions focus the understander on a particular aspect of the situation that might be anomalous. Once an anomaly is detected, the understander uses the anomaly characterization to retrieve potential explanations to resolve the anomaly. Thus the questions involved in the rest of the explanation process can be thought of as *anomaly resolution questions*.

Since the domain of AQUA focuses on human interest stories, the program deals mainly with volitional anomalies. Volitional anomalies can be categorized into two broad types:

1. **Stereotype violation:** Explicit violation of stereotypical information in memory. E.g., the anomaly "Why do they choose kids for bombing missions?" arises if one expects that the terrorists would recruit well-trained military men for these difficult missions.³

S-14: Terrorists recruit boy as car bomber.

Trying to explain WHY DID THE TERRORIST GROUP RECRUIT THE BOY TO DO THE CAR BOMBING?

Anomaly! THE BOY is not a typical MILITARY AGENT.

2. **Missing explanation:** Lack of explanation in memory. These anomalies arise, not out of explicit contradictions, but out of the lack of some information that was expected to be present in memory. In other words, these anomalies arise out of the noticing of gaps in memory. E.g., the anomaly "Why would a person who was not a fanatic go on a suicide bombing mission?" arises when the understander realizes that its standard religious fanatic explanation is inapplicable, and it has no further explanations that apply to this situation. Similarly, in the following example, AQUA detects an anomaly for a novel action for which it has no explanations in memory:

S-15: The terrorist group surrendered to the Israeli police.

³AQUA uses a simple template-based natural language generator to describe concepts in memory. In the following transcripts, the generator output has been cleaned up to some extent for the sake of readability. Since AQUA actually uses the representation of questions in memory, not their printed output, the actual form of the output is not relevant to the operation of the program.

Trying to explain WHY DID THE TERRORIST GROUP SURRENDER TO THE ISRAELI POLICE?⁴

Anomaly! THE SURRENDER violates THE GOAL OF THE TERRORIST GROUP TO PRESERVE THE INDEPENDENCE OF THE TERRORIST GROUP.

Characterized outcome as a BAD outcome for the ACTOR

Searching for stereotypical XPs

Anomaly! No XPs for why THE TERRORIST GROUP SURRENDERED TO THE ISRAELI POLICE.

The first anomaly in the above transcript arises from a contradiction, whereas the second one arises when AQUA encounters a gap in its memory. If AQUA did have an applicable XP to start with, or is able to fill this gap by learning a new XP that explains why terrorists surrender, the XP would be retrieved and applied to the story. This action would then be "explained" (and therefore not anomalous) by virtue of the fact that it has been fitted into an XP that the program has.

Volitional anomalies detected by asking a series of anomaly detection questions about the input. These questions arise by questioning different parts of the planning/decision models discussed in the previous chapter. For example, the understander could question the goals of the agent, or his beliefs about the expected outcome of the action, or his volition in choosing to perform the action.

Each of these questions uncovers a different type of anomaly, and proposes a different type of explanation for the anomaly. For example, questioning the goals of the agent allows the understander to detect goal violation anomalies, in which the agent performs an action that violates his own goals.

In order to notice anomalies and build explanations based on the planning/decision model of volition, AQUA's questions must be generated from this model. This is done by walking over the representation of abstract XPs and questioning the applicability conditions of these XPs (see chapter 3). For example, the abstract XP "Actor chooses to perform an action with a negative outcome because he did not know that the action would result in this outcome" relies on the beliefs of the actor about the outcome of the action. This is represented as one of the PRE-XP-NODES of the XP. This node gives rise to the question: *Did the actor know that the action would have this negative outcome?*

Before asking this question, however, AQUA must determine whether the outcomes of the action are indeed negative from the point of view of the actor. Thus the XP "Actor chooses to perform an action because he wants the outcome of the action" must be tried first. This XP gives rise to the question: *Does the actor want the outcome of the action?* This question must be asked before the question about the actor's beliefs.

In principle, the understander could order its XPs at run-time by checking their applicability conditions to see which ones presuppose the others. As implemented, abstract XPs are organized

⁴For the purposes of this discussion, we assume that AQUA tries to explain every action it sees. The issue of what is uninteresting and what is worth explaining is discussed in chapter 7.

into a hierarchy that determine the order in which they will be checked.⁵ AQUA traverses down this hierarchy, generating questions based on the PRE-XP-NODES of each XP. These questions, therefore, can be viewed as comprising a discrimination net of volitional questions, in which each question raises further questions if its answer seems anomalous (see figure 4.1). For example, the question *Does the actor want the outcome of the action?*, if answered negatively, would signal an anomaly and raise further questions such as *Did the actor know the outcome of the action?* and *Is there another result of this action, perhaps currently unknown, which the actor desired even at the expense of the outcome that he didn't want.* These questions are indexed in memory and used to determine the interesting aspects of the story. The above questions represent the fact that AQUA is interested in the beliefs of the boy, as well as further results of the suicide bombing mission. When these questions are answered later in the story, the corresponding explanation is re-activated. This will be discussed later in greater detail; here we will discuss a taxonomy of anomaly detection questions without concern for their implementation.

Anomaly detection questions can be categorized as follows:

Decision questions: These questions focus on the decision that the actor took when he decided to do the action. Therefore, this is also a taxonomy of the planning decisions one would consider when deciding to do an action.

Personal goals:

- Does the actor want the outcome of this action?
- Does the actor want to avoid a negative outcome of not doing this action?
- Does the actor want a positive outcome of this action more than he wants to avoid a negative outcome of doing the same action?
- Does the actor enjoy doing that action?
- Does the actor habitually do this action?

Instrumentality:

- Is this action instrumental to another action that the actor wants to carry out?
- Is this action part of a larger plan that the actor is carrying out?

Interpersonal goals:

- Does the actor want a positive outcome of this action for someone he likes?
- Does the actor want to avoid a negative outcome of this action for someone he likes?
- Does the actor want a negative outcome of this action for someone he dislikes?
- Does the actor want a positive outcome of this action for a group that he belongs to?
- Does the actor want to avoid a negative outcome of this action for a group that he belongs to?
- Does the actor feel gratification in doing good for others?

⁵Since AQUA does not learn new abstract XPs, this hierarchy does not need to be dynamic. AQUA does learn new stereotypical XPs, which are specific versions of abstract XPs for particular situations. This involves learning new situation and stereotype indices for the new XP (see chapter 6). However, this does not require modification of the hierarchy of abstract XPs since new XPs are indexed using existing abstract XPs as the category index, corresponding to the leaves of the discrimination net of figure 4.1.

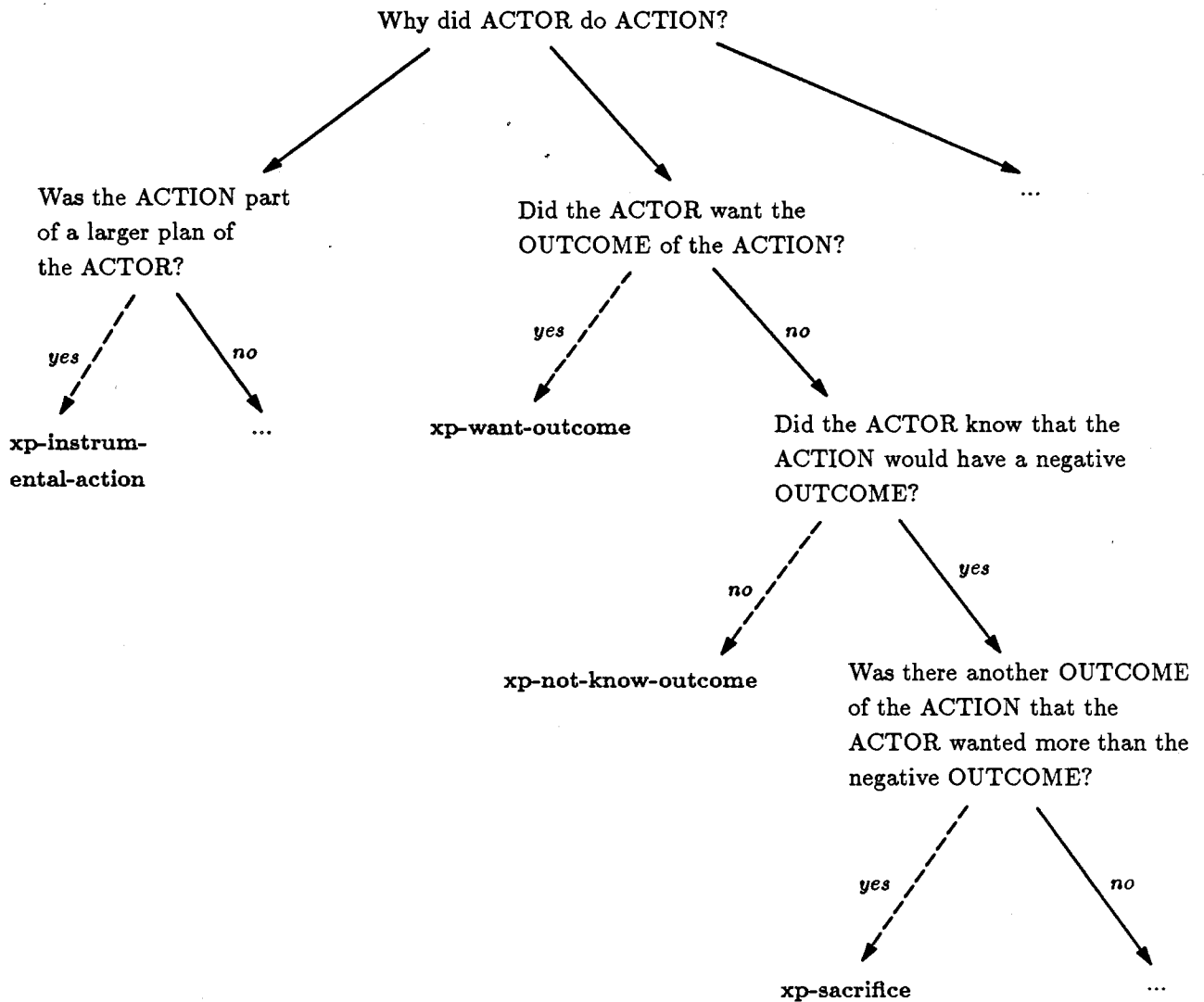


Figure 4.1: Anomaly detection questions arise from questioning different parts of the planning/decision model used to represent volitional explanations. These questions can be viewed as comprising a discrimination net of volitional questions, as shown here in a simplified form. This net corresponds to the hierarchy of abstract XPs represented in the program; the questions shown here are generated using the PRE-XP-NODES of abstract XPs in this hierarchy. If an anomaly is detected, the discrimination net also determines the *category index* of the XP required.

Social control:

- Did someone with social control over the actor ask him to perform the action?
- Did someone with social control over the actor force him to perform that action?

Knowledge and beliefs:

- Did the actor know the probable outcomes of the action?
- Did the actor believe that the action would have a positive outcome for him?
- Did the actor know about the possible negative outcome of the action?

Interference questions: These questions focus on possible interference from external sources.

- Did someone want to block the actor's goal?
- Did someone want to prevent this state of the world? Would this state of the world violate this person's goals?
- Did someone want the actor to be involved in this action?
- Did the actor accidentally get involved in this action?

For example, consider the above story S-15 again. These are the questions that lead to the anomaly begin detected in this story:

S-15: The terrorist group surrendered to the Israeli police.

Trying to explain WHY DID THE TERRORIST GROUP SURRENDER TO THE ISRAELI POLICE?

Did THE TERRORIST GROUP want the outcome of THE SURRENDER?

Characterizing the outcome CAPTURED-STATE
of THE SURRENDER
from the point of view of THE TERRORIST GROUP (the ACTOR)

DOES THE TERRORIST GROUP WANT TO ACHIEVE THE CAPTURED STATE OF THE TERRORIST GROUP?

Anomaly! THE SURRENDER violates THE GOAL OF THE TERRORIST GROUP TO PRESERVE THE INDEPENDENCE OF THE TERRORIST GROUP.

Characterized outcome as a BAD outcome for the ACTOR

In the above output, only the questions that led to the anomaly are shown. To take a more complete example, consider the following story:

S-16: Terrorists recruit boy as car bomber.

A 16-year-old Lebanese got into an explosive-laden car and went on a suicide bombing mission to blow up the Israeli army headquarters in Lebanon. ...

The teenager was a Shiite Moslem but not a religious fanatic. He was recruited for the mission through another means: blackmail.

When AQUA reads this story, it asks the following anomaly detection questions which focus on the personal goals in the actor's decision model:

Trying to explain WHY DID THE TEENAGE LEBANESE BOY DO THE SUICIDE BOMBING?

Was THE SUICIDE BOMBING instrumental to another action?

Does THE TEENAGE LEBANESE BOY typically do SUICIDE BOMBINGS?

Did THE TEENAGE LEBANESE BOY want the outcome of THE SUICIDE BOMBING?

Characterizing the outcomes DEATH-STATE and DESTROYED-STATE
of THE SUICIDE BOMBING
from the point of view of THE TEENAGE LEBANESE BOY (the ACTOR)

DOES THE TEENAGE LEBANESE BOY WANT TO ACHIEVE THE DEATH OF THE TEENAGE LEBANESE BOY?

Anomaly! The SUICIDE BOMBING violates THE GOAL OF THE BOY TO PRESERVE THE LIFE STATE OF THE BOY.

DOES THE TEENAGE LEBANESE BOY WANT TO ACHIEVE THE DESTRUCTION OF THE ISRAELI ARMY HEADQUARTERS IN LEBANON?

No relevant GOALS found

Did THE TEENAGE LEBANESE BOY want to avoid a negative outcome of not doing THE SUICIDE BOMBING?

No relevant OUTCOMES found

Did THE TEENAGE LEBANESE BOY enjoy doing the SUICIDE BOMBING?

No relevant GOALS found

Did THE TEENAGE LEBANESE BOY habitually do SUICIDE BOMBINGS?

No relevant ACTIVITIES found

Characterized outcome as a BAD outcome for the ACTOR

In addition to the above questions, which focus on the actor's reasons for performing an action, AQUA also considers the planner's reasons for planning that action, if the actor and the planner are different. Many of the questions are similar to the above questions. For example, AQUA would ask whether the planner wanted an outcome of the action, whether he knew the outcome of the action, etc. Thus for the above story, AQUA also asks the following questions:

Trying to explain WHY DID THE TERRORIST GROUP PLAN THE SUICIDE BOMBING?

Did THE TERRORIST GROUP want the outcome of THE CAR BOMBING?

Characterizing the outcomes: DEATH-STATE and DAMAGED-STATE
of MOP: THE SUICIDE BOMBING
from the point of view of: THE TERRORIST GROUP (the PLANNER)

DOES THE TERRORIST GROUP WANT TO ACHIEVE THE DEATH OF THE TEENAGE
LEBANESE BOY?

No relevant GOALS found

DOES THE TERRORIST GROUP WANT TO ACHIEVE THE DESTRUCTION OF THE
ISRAELI ARMY HEADQUARTERS IN LEBANON?

Matches typical GOALS

Did THE TERRORIST GROUP typically do SUICIDE BOMBINGS?

Matches typical ACTIVITIES

No anomaly detected

In addition, AQUA also asks the following questions which focus on the interaction between the planner and the actor:

Planner questions: Given an action that was planned and executed by different people or groups:

- Did the action result in a positive outcome for both the planner and the actor?
- Did the planner select the actor knowing that the action would result in a negative outcome for the actor?

For example, if one wonders why the planner gave the necessary resources to the actor and then realizes that one of the planner's goals was achieved by the action, one can hypothesize that a contract or exchange exists between the two.

When answered, anomaly detection questions either result in an anomaly being detected, in which case the anomaly needs to be resolved through explanation, or in no anomaly being detected, in which case an explanation has implicitly been found. For example, if the answer to "Did the action result in a positive outcome for the actor?" is yes, there is no anomaly; the explanation implicit in this question is the abstract one of "Actor does action to satisfy a goal." An explanation at this level is sufficient in many situations; a deeper explanation is required only when there is an anomaly at this level.

4.1.1.2 XP retrieval questions

These questions are also called anomaly resolution questions, since their intent is to find explanations that help resolve anomalies. XP retrieval questions fall into two broad categories, corresponding to the taxonomy of motivational explanations:

1. Abstract XP retrieval questions.
2. Stereotypical XP retrieval questions.

When faced with an anomalous situation, AQUA tries to retrieve one or more XPs that would explain the situation. In order to do this, AQUA asks a series of *XP retrieval questions* about the situation.

XP retrieval questions focus the understander's attention on a particular aspect of a situation, or allow it to view a situation in a particular way, with the intention of finding an explanation that might underlie it. These questions fall into two broad categories, corresponding to the taxonomy of motivational explanations:

1. **Abstract XP retrieval questions:** These questions focus the understander's attention on the goals, plans and beliefs of the character. For example, the question "Did the boy want to kill himself?" focuses on the boy's goals with respect to this particular outcome, the death of the boy. If the answer to this question is yes, an explanation might be that the boy actually did want to kill himself (a suicidal teenager perhaps), and suicide bombing was just a bizarre way of doing this. Another question of this type is "Did the boy know he was going to die?", which focuses on the boy's beliefs.
2. **Stereotypical XP retrieval questions:** These questions attempt to view the character as belonging to a particular stereotype. Their intent is to enable the understander to retrieve stereotypical XPs specific to the particular situation. For example, the question "Was the boy a zealous Shiite Moslem?" focuses on the religious beliefs of the boy. An affirmative answer to this question would cause AQUA to retrieve the "religious fanatic" XP, since this XP is indexed under a religious fanatic stereotype.

As argued earlier, it is too inefficient to reason from scratch, starting from the general theory of motivations. Stereotypical XPs provide a way to associate specific stereotypical causal rules with specific stereotypical situations and people. The features characterizing these stereotypes result in XP retrieval questions.

Thus an XP retrieval question is a heuristic to find XPs that might be relevant without having to do the inferencing required to check if they are indeed relevant, which requires determining whether the causality underlying the situation matches that underlying the XP. If the question results in an XP being retrieved, the understander does the rest of the work to determine if the XP is indeed relevant.

XP retrieval questions at the abstract level of abstract XPs focus on the goals, goal priorities, beliefs, and decisions of the people involved, whereas those at the specific level of stereotypical XPs focus on particular stereotypes of people that are known to be associated with those kinds of actions. Thus XP theory replaces general matching techniques for goals and plans with specific applicability conditions for stereotypical situations. The former are applicable to a greater range of situations but harder to determine; the latter are specific to particular situations but easier to match, infer, apply and verify. This is the basic principle underlying script, frame and schema-based theories.

Here is a taxonomy of XP retrieval questions. The taxonomy at the level of abstract XPs mirrors the taxonomy of anomaly detection questions. Since the particular XP retrieval questions at the level of stereotypical XPs depend on the stereotypes currently in memory, this category will be illustrated using examples rather than a taxonomy.

Abstract XP retrieval questions:

Decision anomalies:

- **Anomaly: goal-violation**

Situation: Actor does action that results in a negative outcome.

Questions:

- Did the actor actually want this outcome (i.e., did we misperceive his goals?)
- Did the action result in another outcome that the actor wanted even at the expense of the negative outcome?
- Was the actor forced into doing this action?
- Did the actor know that the action would have this negative outcome?
 - * Did the actor have enough information about the environment?
 - * Did the actor project the effects of the action correctly?

For example, since there is a goal-violation anomaly in the above story, AQUA asks the following abstract XP retrieval questions:

Anomaly! The SUICIDE BOMBING violates THE GOAL OF THE BOY TO PRESERVE THE LIFE STATE OF THE BOY.

Searching for abstract XPs

DID THE SUICIDE BOMBING RESULT IN A STATE?
and WAS THE GOAL OF THE BOY TO ACHIEVE THE STATE MORE IMPORTANT THAN THE GOAL OF THE BOY TO PRESERVE THE LIFE STATE OF THE BOY?

No other RESULTS of THE SUICIDE BOMBING known
Suspending explanation

DID THE BOY BELIEVE THAT THE SUICIDE BOMBING WOULD RESULT IN THE DEATH STATE OF THE BOY?

No relevant BELIEFS found
Suspending explanation

These questions are indexed in memory, and are used to determine the interesting aspects of the story. The above questions represent the fact that AQUA is interested in the GOALS and BELIEFS of the boy, as well as further results of the suicide bombing mission. If a question is answered later in the story, the corresponding explanation is re-activated. For example, suppose that the story were to be continued as follows:

S-16: The boy was told that the car bombing would not cause him any harm.

Answering question: DID THE BOY BELIEVE THAT THE SUICIDE BOMBING WOULD RESULT IN THE DEATH STATE OF THE BOY?

with: THE BOY DID NOT BELIEVE THAT THE SUICIDE BOMBING WOULD RESULT IN THE DEATH STATE OF THE BOY.

Restarting suspended explanation

THE BOY DECIDED TO DO THE SUICIDE BOMBING DESPITE THE VIOLATION OF THE GOAL OF THE BOY DO PRESERVE THE LIFE STATE OF THE BOY

because THE BOY DID NOT KNOW THAT THE SUICIDE BOMBING WOULD RESULT IN THE DEATH STATE OF THE BOY.

Of course, this still does not explain why the boy did the suicide bombing, but the **goal-violation** anomaly is resolved. Once this explanation is confirmed, the other hypotheses are retracted.

Continuing with the taxonomy of abstract XP retrieval questions:

- **Anomaly: goal-violation or unusual-goal-ordering**

Situation: Actor does action that results in a positive outcome and a negative outcome.

Questions:

- Does the actor prefer to achieve the positive outcome even at the expense of the negative outcome?
- Does the actor actually want to avoid the negative outcome?
- Can the goal violated by the negative outcome be pursued later?

XPs:

- Goal priority elevation in particular contexts.
 - * Goal violated by negative outcome can be pursued later.
 - * Goal of positive outcome is temporarily urgent.
 - * Short term goals preferred to longer term goals.
 - * Personal goals preferred to group goals.
 - * Difficult goals postponed.
 - * New goals from wanting what others have.
- Actor's goal priorities were misperceived.
 - * Personal differences (individual, parental).
 - * Group differences.
 - * Cultural differences.

- **Anomaly: bad-plan-choice**

Situation: Actor does action to achieve a goal even though another action looks better.

Questions:

- Did the second action have a negative side effect for the actor?
- Did the actor know about the second action?
- Was the actor capable of performing the second action?
- Is the first action better in the long run?
 - * Is the cumulative effect of the first action better?
 - * Does the first action keep more options open?
- Does the actor enjoy doing the first action?

- **Anomaly: failed-opportunity**

Situation: Actor doesn't do action that would have resulted in a positive outcome for actor.

Questions:

- Did the actor actually want this outcome (i.e., did we misperceive his goals)?

- Did the actor know that the action would result in the positive outcome?
- Did the action also result in a negative outcome for the actor?
- Was the actor capable of performing that action?
- **Anomaly: unmotivated-action**
 Situation: Actor does an action that doesn't satisfy any of his goals. Questions:
 - Did the actor think that the action would satisfy one of his goals?
 - Does the action actually satisfy a goal (i.e., did we misperceive the situation)?

Planner anomalies:

- **Anomaly: malicious-intent**
 Situation: Planner knowingly recruits actor for action that results in negative outcome for actor.
 Questions:
 - Was the planner's real intention to achieve a negative outcome for the actor?
 - Did the planner want some other outcome of the action, but also wanted to avoid the negative outcome from happening to himself?
 - Was the planner willing to sacrifice the actor's goal to achieve a goal of his own?
 - Did the planner want both the outcomes, i.e., was he killing two birds with one stone?
 - Was the planner in turn forced to make this decision?

In the above story, the actor-planner interaction gives rise to the following questions:

The PLANNER, THE TERRORIST GROUP, is not the same as the ACTOR,
THE BOY

Anomaly! The PLANNER, THE TERRORIST GROUP, planned an action with
a BAD outcome for the ACTOR, THE BOY.

Searching for abstract XPs

DID THE TERRORIST GROUP WANT TO ACHIEVE THE DEATH STATE OF THE
BOY?

DID THE TERRORIST GROUP WANT TO DESTROY THE ISRAELI ARMY
HEADQUARTERS?

and DID THE TERRORIST GROUP WANT TO AVOID THE DEATH STATE OF THE
TERRORIST GROUP?

WAS THE GOAL OF THE TERRORIST GROUP TO DESTROY THE ISRAELI ARMY
HEADQUARTERS MORE IMPORTANT THAN THE GOAL OF THE TERRORIST
GROUP TO PRESERVE THE LIFE STATE OF THE BOY?

If any of these questions are answered, the corresponding XP is retrieved and applied to the story. If the explanation can be confirmed, the anomaly is resolved.

Stereotypical XP retrieval questions:

- XP: Religious fanatic does suicide bombing.
Questions:
 - Was the actor religious?
 - (If in the Middle East) Was the actor a Shiite Moslem?
- XP: Depressed teenager commits suicide.
Questions:
 - Was the actor a stereotypical teenager?
 - Was the actor depressed?

In the current example S-16, AQUA finds the boy's suicide bombing action anomalous. It tries to retrieve XPs to explain this action and finds the religious fanatic XP indexed under the **Lebanese-person** stereotype and the **suicide-bombing** MOP:

Searching for stereotypical XPs

Asking EQ: IS THE BOY A TYPICAL TEENAGER?

Asking EQ: WHY WOULD A TEENAGER DO A SUICIDE BOMBING?

Situation index = SUICIDE-BOMBING

Stereotype index = TEENAGER

No XPs found

Asking EQ: IS THE BOY A TYPICAL LEBANESE PERSON?

Asking EQ: WHY WOULD A LEBANESE PERSON DO A SUICIDE BOMBING?

Situation index = SUICIDE-BOMBING

Stereotype index = LEBANESE-PERSON

Retrieved stereotypical XPs:

XP-RELIGIOUS-FANATIC (category index = XP-SACRIFICE)

AQUA also tries to retrieve abstract XPs to explain the anomaly. For example, since the anomaly is one in which the actor performs an action with a negative outcome for himself, AQUA asks the abstract XP retrieval questions that were described earlier. Abstract XPs are used only if no specific stereotypical XPs can be found. For example, consider the question:

IS THE GOAL OF THE TEENAGE LEBANESE BOY TO ACHIEVE THE DESTRUCTION OF
THE ISRAELI ARMY HEADQUARTERS IN LEBANON MORE IMPORTANT THAN THE
GOAL OF THE TEENAGE LEBANESE BOY TO PRESERVE THE LIFE OF THE
TEENAGE LEBANESE BOY?

Not known

This question, when answered, would lead to **xp-sacrifice**, but since a stereotypical XP of this type has already been found (**xp-religious-fanatic**), the abstract XP is not used to explain this anomaly.

4.1.1.3 XP application questions

Once a set of potentially applicable XPs is retrieved, the understander tries to use them to resolve the anomaly. This involves instantiating the XP, filling in the details through elaboration and specification, and checking the validity of the final explanation. Questions that help the understander elaborate explanations, or collect more information about the input, to help it construct a coherent understanding of the input are called *XP elaboration questions* and *data collection questions*, respectively.

When an XP is retrieved, it is instantiated to form a hypothesis. This involves the following steps:

- Unify the EXPLAINS node of the XP with the description of the situation being explained. Mismatches between the EXPLAINS node and the situation become XP application questions.
- Instantiate the INTERNAL-XP-NODES and LINKS.
- Check each of the INTERNAL-XP-NODES to see if it is true. If all the INTERNAL-XP-NODES are known, continue.
- Otherwise generate XP application questions from those INTERNAL-XP-NODES that are not already known to be true and cannot be inferred.
- Check if there are old questions attached to the XP, representing what the understander does not fully understand about that XP. Instantiate these questions with respect to the current situation.

If this process raises any XP application questions, the explanation is suspended until answers to these questions become known. To illustrate this, consider the application of *xp-religious-fanatic*, retrieved in the previous step, to the current story:⁶

Applying XP-RELIGIOUS-FANATIC to WHY DID THE BOY DO THE SUICIDE
BOMBING.

THE BOY DID THE SUICIDE BOMBING
because THE BOY WAS A RELIGIOUS FANATIC.

Unifying EXPLAINS node

Installing NODES

Installing LINKS

THE BOY IS A RELIGIOUS FANATIC
because THE BOY IS A SHIITE MOSLEM

THE BOY WANTS TO ACHIEVE THE DESTRUCTION OF THE ISRAELI ARMY
HEADQUARTERS
because THE BOY IS A RELIGIOUS FANATIC

⁶These questions correspond to the actual definition of *xp-religious-fanatic* shown in section 5.3.1. As noted earlier, the generator output has been cleaned up in this trace for readability.

THE GOAL OF THE BOY TO ACHIEVE THE DESTRUCTION OF THE ISRAELI ARMY
 HEADQUARTERS IS MORE IMPORTANT THAN THE GOAL OF THE BOY TO
 PRESERVE THE LIFE STATE OF THE BOY
 because THE BOY IS A RELIGIOUS FANATIC

THE BOY DECIDED TO DO THE SUICIDE BOMBING
 because XP-SACRIFICE

In this case, there are no XP application questions since the religious fanatic XP is applicable to the situation. However, it still needs to be verified.

4.1.1.4 Hypothesis verification questions

The final step in the explanation process is the confirmation or refutation of possible explanations, or, if there is more than one hypothesis, discrimination between the alternatives. Hypothesis verification questions (HVQs) are questions that arise from this task. For example, although there is no real difficulty in applying the religious fanatic explanation in story S-16, the explanation rests on certain assumptions. To verify the hypothesis, AQUA asks what the religion of the boy was, and whether he believed fanatically in that religion. When it reads that "the boy was a Shiite Moslem but not a religious fanatic," it answers these questions and refutes the hypothesis.

To generate the HVQs for a hypothesis, AQUA checks each of the XP-ASSERTED-NODES of the hypothesis:

- If all the XP-ASSERTED-NODES are true, accept the hypothesis.
- If any of the XP-ASSERTED-NODES is false, reject the hypothesis.
- Otherwise, generate HVQs for the hypothesis from those XP-ASSERTED-NODES that are not already known to be true and cannot be inferred.

In the *xp-religious-fanatic* example, this raises the following hypothesis verification questions:

Installing HVQs to verify XP:

WHAT IS THE RELIGION OF THE TEENAGE LEBANESE BOY?
 WHAT IS THE RELIGIOUS ZEAL OF THE TEENAGE LEBANESE BOY?

This process is repeated for all the possible hypotheses. When this is done, AQUA is left with one or more alternative hypotheses, each with its own set of HVQs. This is represented using a *hypothesis tree* as described in section 5.3.2.

When new facts come in, AQUA checks to see if these facts would answer any questions in memory (see chapter 5). If an HVQ is answered, AQUA re-examines the hypothesis to see whether it has been confirmed or refuted:

- If the HVQ is answered negatively, refute the hypothesis.

- If the HVQ is answered positively and this is the last HVQ for the hypothesis, confirm the hypothesis and refute its competitors.
- In each case, re-evaluate belief in corresponding hypothesis.

Thus when an HVQ is answered, AQUA knows what to do with the answer since the hypothesis structure represents the suspended explanation task that is waiting for the answer.

In the current story, the *xp-religious-fanatic* hypothesis is eventually refuted when AQUA reads the sentence:

S-16: The teenager was a Shiite Moslem but not a religious fanatic.

Answering question: WHAT IS THE RELIGION OF THE BOY?
with: THE BOY IS A SHIITE MOSLEM.

Answering question: WHAT IS THE RELIGIOUS ZEAL OF THE BOY?
with: THE BOY IS NOT VERY ZEALOUS ABOUT THE SHIITE
MOSLEM RELIGION.

Refuting hypothesis:
THE BOY DID THE SUICIDE BOMBING
because THE BOY WAS A RELIGIOUS FANATIC.

4.1.2 Text-level questions

Text-level questions arise from the basic syntactic and semantic analysis that needs to be done in the input text. Although the particular questions that AQUA generates depend on the situation, all these questions fall into one of the following categories:

4.1.2.1 Actor-action identification

The basic question here is, "Who was the actor of the action?"

For example, in story S-16, when AQUA reads "...and went on a suicide bombing mission", it asks "Who was the actor of the mission?" This question prompts it to retrieve the syntactic subject of the sentence, "the 16-year-old Lebanese", and fill the ACTOR slot of the newly created frame for "suicide bombing mission" with the frame representing the Lebanese teenager.

Processing word: MISSION

Instantiating concept: MISSION

Trying to attach BOMBING to THE MISSION
Trying to attach SUICIDE to THE BOMBING MISSION

Asking basic questions for THE SUICIDE BOMBING MISSION

WHO DID THE SUICIDE BOMBING MISSION?

Searching for the ACTOR of THE SUICIDE BOMBING MISSION
Using sentence subject: THE TEENAGE LEBANESE BOY

Filling ACTOR slot of THE SUICIDE BOMBING MISSION

The actor-action identification question can also be answered by a memory-level task. For example, when AQUA reads "was recruited ... through ... blackmail", AQUA asks "Who blackmailed the boy?" and searches memory to see if it can infer the actor. In this example, AQUA has to read further to find the answer.

The reason that this question is a good question to ask is that after the basic representation is built, AQUA tries to determine whether the action is anomalous. An action can be anomalous in different ways, of course, but detecting anomalies with respect to the actor's goals is a central explanation task. For this task, AQUA needs to determine the actor of the action.

4.1.2.2 Attachment questions

1. Noun group connection: Is this a noun phrase? How do I connect the two nouns together?

For example, does the phrase "car bombing" refer to the bombing of a car, or a suicide bombing using a car? Two nouns, such as "car" and "bombing", can connect together in different ways, depending on the language used to describe the connection. If it becomes important to determine exactly what is meant, AQUA must use its syntactic knowledge of noun phrases, as well as the semantics of the nouns comprising the phrase, to figure out the meaning.

In this story, the car can be either the target or the instrument of the bombing. AQUA represents this by building a vague representation that could be refined to either of the possibilities. This represents the question "Was the car the target of the bombing or the instrument of the [suicide] bombing"?

Trying to attach THE CAR to THE BOMBING

Initial list of candidates: (TARGET INSTRUMENT)

Building two hypotheses for CAR BOMBING:
THE CAR IS THE TARGET OF THE BOMBING.
THE CAR IS THE INSTRUMENT OF THE BOMBING.

When it reads further, AQUA disambiguates "car bombing" to the latter meaning.

Specializing THE BOMBING to SUICIDE BOMBING

Confirming hypothesis:

THE CAR IS THE INSTRUMENT OF THE BOMBING.

Attaching THE CAR to the INSTRUMENT slot of THE BOMBING

At this point, an understander might learn the frozen expression "car bombing." As currently implemented, AQUA does not perform any text-level learning, but it would be straightforward to implement a simple mechanism to learn phrases such as "car bombing." In any particular story, however, AQUA would still need to build the two hypotheses shown above since both meanings of "car bombing" are commonly used.

2. **Adjectives, prepositional attachment:** These are other aspects of the noun phrase attachment problem and are handled in a similar manner. Prepositions result in syntactic predictions being made, which are also represented as questions. For example, when AQUA reads "recruit boy as ...", it predicts a description of the task that the boy was recruited for. This is represented as the question "For what purpose did the terrorists recruit the boy?" When "suicide bomber" is read, it matches this question and is placed in the appropriate slot of the recruit frame.

Processing word: AS

Word type: FW
Function: PREPOSITION

Predicting filler for the REQUESTED-MOP slot of THE RECRUIT

WHAT WAS THE PURPOSE OF THE RECRUIT?

Transforming to: WHAT TERRORIST ACT WAS PLANNED BY THE
TERRORIST GROUP?

Processing word: CAR
Processing word: BOMBER

Trying to attach THE CAR to THE BOMBING

Answering question: WHAT WAS THE PURPOSE OF THE RECRUIT?
with: THE CAR BOMBING

Satisfying prediction from AS
Filling REQUESTED-MOP slot of THE RECRUIT

4.1.2.3 Reference questions

The basic question here is, "Who or what does this description refer to?" A description could be a pronoun (e.g., "he") or a definite reference (e.g., "the boy" or "the teenage bomber"). When AQUA reads a description, it builds a representation that specifies what is being described. It then searches memory for a referent matching the description. The search process is the same as the one used, for example, when it needs to search for a "teenage bomber" for a different task such as explanation.

Searching for referent for THE LEBANESE TEENAGER

Merging concepts:

THE YOUNG BOY

THE LEBANESE TEENAGER

The merging process unifies the two frames and replaces them with a single frame that represents the concept [Cullingford, 1978; Charniak, 1988]. If more than one possible referent is found, AQUA builds two or more hypotheses and processes them using the same mechanisms as for noun phrase or explanation hypotheses.

4.1.2.4 Disambiguation and specialization questions

AQUA's memory is constantly trying to specialize or refine vague descriptions. For example, when AQUA reads "get into a car", it builds a representation of a "get into" action whose *to* slot is a *car*. Since getting into a *container* is the action of entering it, AQUA automatically specializes the representation to *enter*:

Filling *TO* slot of GET INTO with THE CAR

Looking for possible specializations

ENTER is a GET INTO with *TO* = CONTAINER

Specializing GET INTO with *TO* = CAR to ENTER

Thus disambiguation is carried out by building vague descriptions (such as "get into") that are specialized to more specific descriptions (such as "enter") as their slots get filled with specific fillers. This is similar to the process used by the MOPTRANS parser [Lytinen, 1984].

4.1.2.5 Questions or requests?

Text-level questions are similar to *requests* [Riesbeck, 1975]. There are two main advantages of the question-based formulation of text-level tasks. Traditionally, parsing requests have been generated and used by parser-specific mechanisms, whereas higher-level knowledge, such as that used for explanation, is stored in the memory of the system and applied at a later stage. The uniform representation of questions generated by different processes allows us to design an integrated system in an easy and natural manner, as discussed in chapters 5 and 8.

Secondly, from a pedagogical point of view, the explicit generation of text-level questions forces us to pin down exactly which bottom-up parsing tasks need to be done before anything can be done at all. These tasks correspond to the text-level questions that the program always generates.

For example, AQUA always asks the question "Who was the actor?" whenever it sees an action. This means that AQUA always activates what would be the "look for actor" requests for every sentence describing an action. This is a good strategy because it is difficult to formulate motivational explanations for actions whose actors are unknown.

4.1.3 Memory-level questions

AQUA's memory is based on the theory of dynamic memory [Schank, 1982; Kolodner, 1980; Lebowitz, 1980]. An understander relies on its memory to notice similarities (reminding), relate incoming concepts to stereotypes in memory, and provide expectations for further processing. AQUA is a memory-based understander since it parses directly into memory; the representations are built in memory, and memory in turn provides expectations to the parser [Riesbeck and Martin, 1985].

The memory-level tasks of the understander give rise to the following kinds of questions:

4.1.3.1 Journalism questions

These questions seek the who, what, when and where of the story. Their purpose is to try to fill in the details that are typically mentioned in newspaper stories. These questions are relatively uninteresting in themselves, unless in filling in these details the understander detects an anomaly.

4.1.3.2 Reference questions

Reference is the process of determining whether two concepts refer to the same thing. At the text level, reference questions are triggered by pronouns and definite noun phrases, but referential cues may be non-textual too. Some examples of reference questions are:

- Is this bombing the same bombing that was predicted?
- Is the 16-year-old teenager the same as the boy that was recruited?
- Is the explosive-laden car the same car that was implicitly part of the suicide bombing mentioned in the headline?

4.1.3.3 Attachment questions

Attachment questions try to determine the connection between two concepts. At the text level, these questions are triggered by noun groups and prepositional phrases, as described above. However, these questions can also be generated by memory- and explanation-level tasks.

4.1.3.4 Stereotype activation questions

As discussed in section 3.4.3, much of the program's knowledge consists of stereotypes. These could be stereotypical features of concepts, stereotypical activities and goals of people, or stereotypical motivational explanations for actions. One of the central tasks of AQUA's memory is the activation of stereotypes. This gives rise to the following types of questions:

- Does this instance fit this stereotype?
- Is this a typical role filler for this stereotype?
- Does this person have these stereotypical goals?

- Can I view this instance as one that matches this stereotype?
- Does this group of instances follow a pattern?

For example, consider the stereotype that was described in section 3.4.3:

Goals: Entertainment, sex, school, ...

Goal-orderings: Entertainment over school, ...

Beliefs: Not religious, ...

Plans, activities: Dating, studying, riding motorcycles, ...

Features:

- Age: between 13 and 19 (definitional)
- Nationality: American (definitional)
- Religion: none/atheism
- ...

This stereotype describes a particular kind of American high school teenager, and is used to index stereotypical XPs for different actions such as do-drugs and suicide. Consider the following story:

S-17: Illinois, March 6 - A young girl killed herself with a gun ... Diana, 15, was studying in the 10th grade at Rockville High School.

When AQUA reads this story, it builds a representation of the suicide. Since suicide is an action that violates the actor's p-life goal, AQUA tries to explain the action. Indexed under **suicide** and **high-school-teenager** are two stereotypical XPs, the "drug-related suicide" XP and the "depressed teenager" XP. Before it can retrieve these XPs, AQUA must determine whether the actor, the young girl, matches the **high-school-teenager** stereotype.

Trying to explain WHY DID THE YOUNG GIRL COMMIT SUICIDE?

Did THE YOUNG GIRL want the outcome of THE SUICIDE?

Characterizing the outcome DEATH-STATE
of THE SUICIDE

from the point of view of THE YOUNG GIRL (the ACTOR)

DOES THE YOUNG GIRL WANT TO ACHIEVE THE DEATH OF THE YOUNG GIRL?

Anomaly! The SUICIDE violates THE GOAL OF THE YOUNG GIRL TO PRESERVE
THE LIFE STATE OF THE YOUNG GIRL.

Searching for stereotypical XPs

Asking EQ: IS THE YOUNG GIRL A TYPICAL HIGH SCHOOL TEENAGER?

Does THE YOUNG GIRL have stereotypical GOALS?
Matched THE GOAL TO ATTEND SCHOOL

Does THE YOUNG GIRL have stereotypical GOAL-ORDERINGS?
No GOAL-ORDERINGS known

Does THE YOUNG GIRL have stereotypical BELIEFS?
No BELIEFS known

Does THE YOUNG GIRL have stereotypical ACTIVITIES?
Matched ATTENDING SCHOOL

Does THE YOUNG GIRL have stereotypical FEATURES?
Matched AGE = TEENAGE AGE (definitional)
Matched NATIONALITY = AMERICAN (definitional)
RELIGION is unknown (typical)

THE YOUNG GIRL could be A TYPICAL HIGH SCHOOL TEENAGER

Asking EQ: WHY WOULD A TYPICAL HIGH SCHOOL TEENAGER COMMIT SUICIDE?

Retrieved stereotypical XPs:
XP-DRUG-RELATED-SUICIDE
XP-DEPRESSED-TEENAGER

4.1.3.5 MOP and scene inference questions

The representation of events in a dynamic memory using MOPs and scenes has been discussed in [Schank, 1982; Kolodner, 1980; Lebowitz, 1980]. Since this dissertation focuses on *explanations* rather than event structures, the representation of events in AQUA simply follows that of standard dynamic memory theory. AQUA makes MOP and scene inferences in order to build a representation of the events in the story. For example, upon reading about the recruiting of a boy for a suicide bombing mission, AQUA asks the questions:

- How did they locate the boy?
- How did they persuade the boy to do the suicide bombing?

These questions serve as predictions of specific scenes that the story might be expected to mention. However, AQUA makes no effort to elaborate the scenes of every action unless the question is also essential for an explanation task. MOP and scene inferences fall into the following categories, which are illustrated by examples from the output of AQUA as it reads the suicide bombing story S-16.

1. Predicting the scenes of a MOP. For example, recruit predicts locate and persuade:

Predicting SCENES for THE RECRUIT
HOW DID THE TERRORIST GROUP LOCATE THE BOY?
HOW DID THE TERRORIST GROUP PERSUADE THE BOY?

2. Finding the correct context for a scene. For example, blow-up-bomb is part of bombing.

Answering question WHAT IS THE GOAL SCENE OF THE CAR BOMBING?
with THE BLOWING UP OF THE BOMB.

Satisfying prediction from TO:
Filling GOAL-SCENE slot of THE CAR BOMBING

3. Activating definitional inferences. For example, suicide by definition results in the death-state of the actor.

Activating definitional inferences from SUICIDE:
THE SUICIDE RESULTED IN THE DEATH STATE OF THE BOY.

4. Specializing or refining vague descriptions [Lytinen, 1984]. For example, a person with age = 16 is a teenager; a ptrans with instrument = car is a drive.

Filling AGE slot of THE PERSON
Specializing PERSON with AGE = SIXTEEN to TEENAGER

Filling GENDER slot of THE TEENAGER
Specializing PERSON with GENDER = FEMALE to TEENAGE GIRL

Filling TO slot of GET INTO
Specializing GET INTO with TO = CAR to ENTER

4.1.3.6 Reminding and memory search questions

A dynamic memory must be able to recognize unique combinations, notice similarities (both superficial and structural), and retrieve memory structures given a specification. Some examples of these questions are:

- Why are they all named Mohammed?
- Why do we hear about Lebanese car bombers and not about Israeli car bombers?
- Are there any other instances of teenage suicide bombers in memory?

AQUA's memory is designed as a discrimination net in the style of [Kolodner, 1980]. A question-driven understander must be able to pose the following types of questions:

- Can I guess who the actor of this action might be?
- Can I guess where this might have happened?
- Can I guess how this might have been performed?

These issues are outside the central focus of this dissertation. The current implementation of AQUA uses simple memory search techniques to answer such questions.

4.1.3.7 Analogy and generalization questions

Noticing similarities and forming generalizations are an integral part of a dynamic memory. Reminders give rise to questions that seek generalizations, such as "Are car bombers motivated like the Kamikaze?" These generalizations could be based either on superficial similarities or on deeper explanatory or causal similarities. IPP, for example, used an integrated model of understanding and generalization based on surface similarities [Lebowitz, 1980]. This suffers from the familiar problem of feature selection [Schank *et al.*, 1986; DeJong, 1983]. AQUA uses its explanations to guide generalization. This will be pursued in chapter 6.

4.1.4 Relevance questions

All intelligent systems in the real world are, in some sense, constantly asking the questions "Why does this matter to me? How does this relate to my goals?" People have questions already in memory before they begin to read a piece of text. These questions are left over from previous experiences. As they read, they remember these questions and think about them again in a new light. Certainly after reading the blackmail story S-2, we expect to have several questions representing issues we were wondering about which weren't resolved by the story. In story S-2 it turns out that the boy was blackmailed into going on the bombing mission by threatening his parents. This makes us think about the question "What are family relations like in Lebanon?" which remains in memory after we have finished reading the story. To the extent that we are interested in this question, we will read stories about the social life in Lebanon, and we will relate other stories to this one. To cite another example, a person we read this story to repeatedly related the story to the IRA because he was interested in similar issues about Ireland.

Similar arguments can be made for systems performing other cognitive tasks, such as planning. Any system with goals will ultimately be concerned about identifying aspects of the current situation that are relevant to its own goals. Hence questions arising from this task are called *relevance questions*. These questions determine the "interestingness" of the input, in the sense that the input is interesting only if it addresses one's goals or helps to answer one's questions. Thus the interestingness of a fact in a story can be thought of as a measure of how likely is the fact to answer a question one is wondering about, or, alternatively, a measure of what one might expect to learn from that fact, were one to process it in detail. This will be discussed further in chapter 7.

4.2 Conclusion

Questions represent the knowledge goals of the understander, and arise from various understanding tasks. Thus questions are a useful way of characterizing these tasks. This view of understanding also allows us to design an integrated system in an easy and natural manner, as discussed in next chapter.

Chapter 5

Theory of understanding

We now turn to processing issues related to question-driven understanding, and their impact on the design of the understander. The basic goal of reading, as we argued, is to notice gaps in memory and try to fill in these gaps. These gaps give rise to questions or knowledge goals, representing what the understander wants to learn about. Thus a question-driven understander can be viewed as a memory program whose task is to grow more complete by answering the questions that it has.

A program can use different methods to answer its questions. It might try to infer the answers by using default reasoning, or by reasoning by analogy to past cases. Alternatively, it might try to get the information from the outside world, say, by asking someone, or by reading stories or, as in the case of AQUA, news articles. Since AQUA is trying to create mini-theories with respect to the questions that it needs to answer, a superficial reading of these stories will not suffice; AQUA must read at the “theory creation” end of Schank’s “understanding spectrum” [Schank, 1986]. However, people seldom read stories “in depth;” AQUA must be able to control the inferences that it makes according to the mini-theories it is currently pursuing. Let us consider the effect of these issues on the design of the understander.

5.1 Processing issues

5.1.1 Integration of higher-level reasoning processes

One of the central problems in natural language understanding is the integration of higher level reasoning processes, such as explanation, case-based reasoning, default inference, analogy, and so on, with parser level processes in a uniform and psychologically plausible way. Do these processes come after parsing, or are they integrated? Are they implemented by different mechanisms, or does the same basic mechanism underlie both kinds of processes?

We believe that understanding is an integrated process. The understander brings all the relevant knowledge to bear as early as possible, be it syntactic, conceptual, or explanatory, in order to make the best use of it. This is called the *integrated processing hypothesis* [Birnbaum, 1986; Schank *et al.*,

1978; DeJong, 1979; Lebowitz, 1980; Schank and Birnbaum, 1984]. However, there is still the issue of the *mechanism* by which the understander brings different kinds of knowledge to bear. Traditionally, parser-level questions or *requests* [Riesbeck, 1975], such as those determining actor-action relationships or noun group meanings, have been generated and used by parser-specific mechanisms, whereas higher-level knowledge, such as that used for explanation, is stored in the memory of the system and applied at a later stage.

However, in order to make effective use of different kinds of knowledge at the level of the parser, both parser-level and higher-level questions should be used in the same way to guide processing. All these questions, therefore, must be indexed in memory in a uniform manner, rather than being kept on arbitrary lists in different parts of the system. Along with each question, the understander stores a representation of what gave rise to the question, so that when the question gets answered it can continue with that line of reasoning, whether it be a parser-level task or an explanation task.

For example, consider the following story:

S-18: John asked his mother if he could borrow the car for the dance.

Suppose AQUA needs to explain why "he" would want to borrow the car (an explanation-level task). In order to do this, AQUA needs to figure out who "he" refers to. The explanation-level task thus gives rise to a specific text-level question: *Who does "he" refer to?*

To answer this question, AQUA could use a text-level strategy, *Look backwards for a male person*, which in this case yield "John" as the referent. However, consider the following similar story:

S-19: John asked Bill if he could borrow the car for the dance.

In this case, the strategy *Look backwards for a male person* doesn't work because there are two possible referents, John and Bill. If AQUA were to look for the most recent referent, it would make the wrong connection since "he" refers to John, not to Bill. However, here is another similar story:

S-20: John told Bill that he could borrow the car for the dance.

In this story, "he" probably refers to John, although it could refer to Bill in certain contexts. To finish the example, consider:

S-21: John told Mary that he could borrow the car for the dance.

The pronoun "he" now certainly refers to John. The point is that it takes a combination of text-level and higher-level knowledge to answer this question, which in turn is derived from an explanation-level task. When this question is answered, regardless of which kind of knowledge was used in the particular story, AQUA continues with the task of explaining why "he" would want to borrow the car. Furthermore, if the same question had arisen from a different task, say, a text-level task, it would not make any difference in how AQUA tried to answer it.

AQUA asks questions at all levels, and indexes them uniformly in memory. When these questions are answered, the tasks that depend on the answer are restarted. The advantage of integrated processing is that different sources of knowledge can be brought to bear on pursuing these answers in a uniform manner [Birnbaum, 1986].

5.1.2 Subjective understanding

People understand stories differently. This is known as *subjective understanding* [Carbonell, 1979]. The understander's interests and beliefs determine how a story is interpreted. Furthermore, since understanding is driven by the understander's questions or knowledge goals, the questions in memory determine which inferences are drawn from a story. If the understander is trying to confirm a particular hypothesis, it will focus on those inferences that are relevant to the hypothesis. In addition, if there are multiple possible interpretations, it will tend to prefer those interpretations that confirm its hypotheses rather than those that seem anomalous.

In other words, the understanding process is *interest-driven* or *goal-driven* in the sense that it is sensitive to the current questions that the understander is seeking answers to. This requires that AQUA be able to determine the potential relevance of a given concept to its questions. Thus the interestingness of a new concept depends on its relevance to questions in memory, and the interestingness of a hypothesis in memory depends on whether there are unresolved questions attached to the hypothesis.

AQUA's memory permits three kinds of interestingness queries:

1. AQUA can determine whether a new (possibly only partially activated) concept is interesting by indexing to possibly relevant questions and checking if the new concept could directly answer a question.
2. AQUA can determine whether a concept already in memory is interesting (or a particular aspect of that concept is interesting) by checking if it has any unanswered questions attached to it.
3. AQUA can determine how it would like a new concept to be further specified (thus favoring disambiguation towards satisfaction of its goals), again by checking if a particular specification would address an extant question. This is called *interest-based specialization*.

These queries result in a *subjective bias* in decisions made while reading a story. For example, consider the following story again:

S-5: Lebanon car bomb kills driver, hurts 7 at Palestinian site.

BEIRUT, Lebanon, February 26 — A car bomb exploded today at the entrance of the largest Palestinian refugee district in southern Lebanon, killing the driver and wounding seven people.

The police said the explosion occurred outside the Ain Khilwe camp, near the port of Sidon A guard at the entrance of the camp ... said he saw the driver trying to get out of the car. "He struggled with the door, then the whole car exploded with him inside," the guard said.

On the surface, this looks like a stereotypical suicide bombing story. The only anomaly is that the driver tried to do something (get out of the car) which would disenable his own plan (suicide bombing):

Trying to explain WHY DID THE DRIVER TRY TO OPEN THE DOOR?

Retrieved instrumentality XP:
XP-INSTRUMENTAL-ACTION

Applying XP-INSTRUMENTAL-ACTION to THE DRIVER TRIED TO OPEN THE DOOR.
 THE DRIVER TRIED TO OPEN THE DOOR TO ENABLE THE EXIT.

Trying to explain WHY DID THE DRIVER TRY TO EXIT THE CAR?

Anomaly! THE EXIT (attempted) disenables THE DRIVE
 which is instrumental to THE CAR BOMBING.

This anomaly results in the following questions:

Anomaly: disenable-own-plan

Situation: Actor does action that disenables his own plan.

Questions:

- (1) Did the actor change his mind about carrying out the plan?
- (2) Was this actually part of the plan (i.e., did we misperceive his plan?) Perhaps the driver was not a suicide bomber; he was carrying out a different plan in which he intended to jump out at the last minute.
- (3) Was the actor forced into doing carrying out this plan?

When AQUA reads this story, the explanation that it builds depends on the hypotheses that are currently in its memory. If AQUA knows about car bombings in which the actor jumps out at the last minute, it confirms hypothesis (2). If, on the other hand, AQUA has just read the blackmail story S-2, it has the hypothesis that suicide bombers do not volunteer for these missions, but rather are forced into them using blackmail. In this situation, AQUA builds a different explanation for the anomaly by confirming hypothesis (3):

Retrieved abstract XPs:
 XP-ACTOR-CHANGED-MIND
 XP-FORCED-TO-DO-ACTION

Applying XP-ACTOR-CHANGED-MIND to THE DRIVER TRIED TO EXIT THE CAR.

THE DRIVER TRIED TO EXIT THE CAR
 because THE DRIVER CHANGED HIS MIND ABOUT DOING THE CAR BOMBING.

Installing HVQs to verify XP:
 DID THE DRIVER WANT TO PREVENT THE DRIVER DOING THE CAR BOMBING?

Applying XP-FORCED-TO-DO-ACTION to THE DRIVER TRIED TO EXIT THE CAR.

THE DRIVER DID THE CAR BOMBING
 because THE DRIVER WAS FORCED TO DO THE CAR BOMBING.

Installing HVQs to verify XP:
 DID THE DRIVER NOT VOLUNTEER TO DO THE CAR BOMBING?

Matches hypothesis in memory!

Confirming hypothesis:
 THE DRIVER DID NOT VOLUNTEER TO DO THE CAR BOMBING.

In this story, AQUA cannot prove that its hypothesis is correct. The point here isn't whether one can prove one has the "right answer," but rather that understanding is biased by questions and hypotheses currently in memory. This is why different people often interpret the same story in different ways.

5.2 Controlling inferences

Ultimately, the point of reading is to learn more about the world. The understander's world model grows more complete as it reads a story and relates it to gaps and questions in memory. These questions should be used to focus the understanding process, since answering the questions is the goal of reading in the first place. In particular, the questions currently in memory should be used to control the inferences that are drawn from the text.

How do questions guide inferencing? Ideally, only those inferences should be drawn that lead to conclusions that the program needs to know. But this is not always possible in practice. Let us consider some of the issues involved in the problem of inference control.

5.2.1 Bottom-up vs. top-down processing

Given that the basic task of a question-based understanding program is to try to answer questions in its memory by reading stories, there is an obvious choice to be made in the design of the program:

Text-driven: A text-driven program would read the text, build representations for it, and then process it to see if it addressed any questions in memory.

Question-driven: A program that was totally question-driven would pick the most interesting or urgent question in memory and try to answer it via inference, reading text, or indeed any other method it had available.

Each of these approaches has its disadvantages. The text- or data-driven method is completely bottom-up. It tries to process in detail everything in the hope that it might turn out to be relevant. Instead, we would like the program to concentrate on those aspects that were of interest to it. In other words, we would like the process to be driven by the interests or goals of the understander.

The goal- or question-driven method, when taken to the extreme, is too top-down. A program built using this method would only see what it was looking for already. Furthermore, rather than expending resources in trying to answer a question immediately, it might be advantageous to index the question in memory, to be answered later when the opportunity arises. Finally, so as not to overlook obvious information and be sufficiently sensitive to the exigencies of the input, the process should be data-driven as well. The interaction between these requirements is non-trivial.

AQUA is designed as a compromise between these two approaches. The basic understanding cycle that it uses is as follows. The parser reads the story word by word, trying to build a basic conceptual structure to represent the input. As quickly as possible, this structure is related to the questions in memory. If the new structure answers a question in memory, the inferences that were

awaiting that answer are restarted. Thus the program only draws those inferences that are required to match the new structure to the question, and those that are demanded by the task that generated the question in the first place.

This method is called "graded parsing" or "variable-depth parsing". The process is *data-driven* to the extent that pieces of the input for which there are no explicit expectations but which are likely to be relevant are processed to the extent necessary to determine their relevance. At the parsing level, this amounts to making a judicious *skip/don't skip* decision, based on estimates of the relevance of the input using *top-down* guidance from the questions in memory.

In practice, this means that there is a set of bottom-up questions that the program always asks of incoming text. Further processing of the input is done only if these questions raise other questions that need answers, or if the input turns out to answer some question already in memory. These correspond to the two diamonds in figure 8.1 on page 145. These diamonds attempt to determine which facts the understander should focus on. This decision is made using a set of heuristics that will be described in chapter 7.

The top-down component of this process involves the use of questions to determine which inferences to draw from the fact being processed. These inferences fall into two categories: inferences required for matching the fact to questions, and inferences drawn after matching a potential answer to a question.

5.2.1.1 Inferences required for matching

The process of matching a question to a potential answer often spawns further questions, which in turn could require further inferencing to answer. When a new instance is being processed, therefore, the questions that are retrieved as being possibly relevant determine the inferences that need to be made about the new instance. For example, suppose the understander reads the following sentence:

S-22: The boy was recruited for the mission by the Islamic Jihad.

Suppose it had previously asked the question, "Was the boy blackmailed by a group affiliated with the police?" Does S-22 constitute an answer to this question? The answer is yes only if the Islamic Jihad is known to be affiliated with the police. If this is not already known to the understander, a new question is generated, and the matching process is suspended until an answer to the new question is found. This could be found by further inferencing, or by reading the story further.

Thus the inferences that the understander needs to make from S-22 are determined by the questions that S-22 is relevant to. There are many other inferences that could be made from this sentence, e.g., that the boy had approached the Islamic Jihad, that the Islamic Jihad is a terrorist group, and so on. But there is no point in wasting resources in making these inferences and checking to see if they are plausible if this is not something that the understander needs to know about.

AQUA's matcher allows three kinds of matches:

1. **Intersection match:** Either of the two concepts being matched could have extraneous information. Only the information that overlaps is required to match. This is useful for tasks such as reference. For example, "the suicide bombing" and "the car bombing" could refer to the same bombing, but "the 16-year-old boy" and "the terrorist group" are incompatible.

2. **Subset match:** One of the concepts must be completely matched, but the other could have extraneous information. This is useful for matching questions to potential answers, as in example S-22 above, since the answer is required to satisfy all the constraints of the question. For example, the question "Was the boy a teenager?" is not answered by "The boy was young," but it is answered by "The boy was 16 years old."
3. **Equality match:** Both concepts must match completely; neither can specify any extraneous information that doesn't correspond to some piece of the other concept. This is useful for internal bookkeeping tasks such as variable binding.

The matcher is designed so that incomplete subset and equality matches automatically generate questions that need to be answered before the match goes through. These questions are indexed in memory, and the match is suspended, to be restarted later when the questions are answered.

5.2.1.2 Inferencing after a question is answered

When a new piece of information comes in, the understander tries to determine whether it could answer a question that is pending in its memory. If a question is answered, the source of the question determines the further processing that needs to be done on the new piece of information. In other words, the task that gave rise to the question now receives the new information, which is then processed by that task.

For example, consider the following sentence:

S-23: The boy was 16 years old.

What inferences should be drawn from this sentence? Suppose that this is used to answer the question, "Was the boy a teenager?" This now tells the understander that it was the fact that he was a teenager, as opposed to male or 16 years of age, that is important. Furthermore, the understander also knows *why* this is important, since it knows why the question was asked in the first place. For example, if the question arose from a memory-level similarity-based generalization task, the understander might now try to construct a new generalization in memory, say, that suicide bombers in Lebanon are often teenagers. Alternatively, if the understander was trying to confirm the hypothesis that the boy was recruited because he was a gullible teenager, the understander would now try to confirm that explanation. This would require that it determine whether the boy was gullible if it didn't know this already.

The point is that the gullibility of a teenage boy is not something one would normally try to infer from sentence S-23. Rather, it is the reason one wants to know the age of the boy that determines which inferences should be drawn once the age is known. This is accomplished by continuing the reasoning task that is awaiting the answer, and allowing it to draw the inferences that it needs from the answer.

5.2.2 Theory of inference control

Although the question-asking heuristics used by AQUA are not represented formally within the program, it is useful to formalize a theory of inference control. Suppose T is an understanding task,

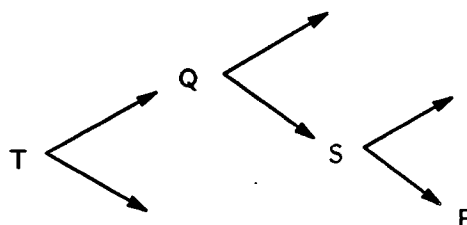


Figure 5.1: Theory of inference control. T is an understanding task, Q is a question generated by that task, S is a subquestion, and F is an input fact that would answer the subquestion.

and Q is a question generated by that task. Let S be a subquestion of Q, and F be an input fact that would answer the subquestion S.

This can be represented as shown in figure 5.1. The lines represent chains of inferences between the nodes. For example, in the blackmail story S-2, T is the goal to explain the motivations of the teenage suicide bomber, Q is the question "Why did the boy do the suicide bombing?", S is the hypothesis verification question "Is the boy a religious fanatic?", and F is the input fact "The boy is a Moslem but not a fanatic." This can be represented as shown in figure 5.2.

The central question of inference control is, how do we connect the input fact F to the question Q, so that the original task T can run? Even assuming that there are inference rules in memory that can complete the chain $Q \rightarrow F$, it is nevertheless too expensive to try all the combinations of these rules since there are far too many of them, and they can combine in too many different ways. This is the familiar problem of combinatorial explosion.

There are two possible strategies, corresponding to the bottom-up and top-down approaches for understanding:

Top-down: Backward¹ chain from the question Q until the new fact F is derived.

Bottom-up: Forward chain from the input fact F until the question Q is derived.

As before, AQUA's "graded parsing" method combines the two into a process that may be described as follows:

- Backward chain a little from Q and generate subquestions S in memory.
- Wait for input facts F to come in.
- Forward chain from the input to answer the subquestions S.

¹The "forward" and "backward" directions are arbitrary since they depend on how the rules are written. This is not relevant to the argument.

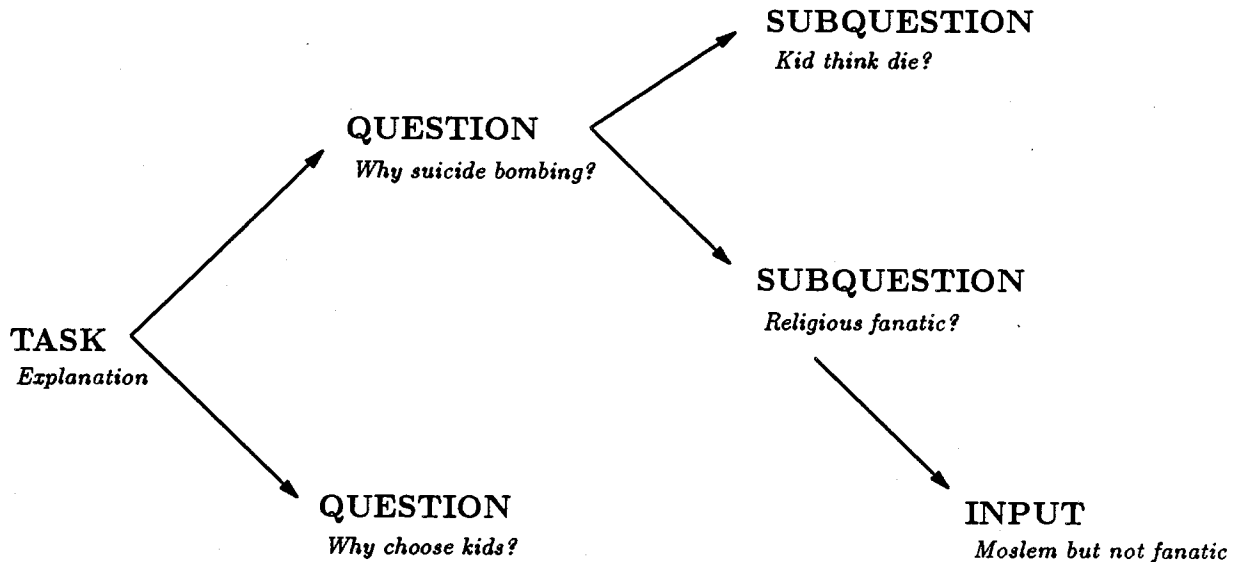


Figure 5.2: Inferences supporting the religious fanatic hypothesis, corresponding to the inference structure of figure 5.1.

- Answer the main question Q and continue with task T.

The real issue here, of course, is how much backward chaining should be done at the time the subquestions are answered, and how much forward chaining should be done when the input comes in. In other words, what kinds of subquestions S, and how many of them, should one generate?

The answer depends on five factors:²

1. **Certainty of inference:** The probability of the inference rules used to transform Q to S, or the likelihood that the conclusion will be true. If the subquestion strongly supports the original question, it is worth making the inference. In a logic system where an inference rule represents a deduction, this probability is 1.
2. **Cost of inference:** The cost of making the inferences needed to transform Q to S, which includes the cost of retrieving the inference rule as well the cost of checking its applicability. In some systems, this might also include the cost of asserting the question S, if, say, this involved side-effecting the world or performing experiments. The cheaper the inference, the more it is

²A sixth factor, the likelihood of the question being answered, is discussed below.

worth the system's while to make it. In AQUA, this is the cost of retrieving and applying an explanation pattern.

3. **Usefulness of question:** The usefulness of the conclusion, i.e., the usefulness of the original question Q . Since Q is in service of the task T , this is the same as the importance of performing that task. If Q were in the service of more than one task, one would need to combine the importance of all these tasks. If the task is very important, it is worth making the inference even if it is very expensive to do so.

AQUA tries to answer all its questions. However, some questions may be more important than others. Chapter 7 will discuss heuristics for determining the relevance and interestingness of a question.

4. **Likelihood of question being useful:** The likelihood that the conclusion will be useful, i.e., the likelihood that the question Q will actually turn out to be useful in performing the task T . However, if the task is very important, it might be worth trying to pursue Q even if there is only a small chance that it will be useful. In AQUA, questions are only generated from tasks that require them, and so this likelihood is 1.
5. **Indexing cost:** The cost of keeping indexed questions in memory and matching to them. If there are too many questions in memory, it might be too expensive to find them or to match input to potentially relevant questions. This cost depends on the scheme used to maintain questions in memory. Since AQUA indexes its questions in memory, the number of questions in memory does not affect the cost of finding them, since questions are indexed using their concept specifications along with known facts. However, if there are several identical questions indexed in the same place in memory, matching the input to all these questions and running their associated tasks could be expensive.

In addition, there is the issue of how much forward chaining should be done from an input that does not match any questions exactly, in order to find questions in memory that the input might answer. For this purpose, bidirectional search is better than a purely bottom-up or purely top-down backtracking-style approach. Since it takes $b^{d/2}$ inferences to connect each question to a given fact, where b is the branching factor of the tree and d is the depth of the inference chain, there can be up to $b^{d/2}$ questions in memory before bidirectional search becomes as slow as unidirectional search with a single question, which takes b^d inferences [Birnbau, 1986]. A better approach would be to use a marker-passing or spreading activation scheme to find possible questions in memory (e.g., [Charniak, 1983; Charniak, 1986; Granger *et al.*, 1986; Norvig, 1987].)

However, these are "weak" theories in the sense that the forward chainer does not rely on knowledge of the types of questions that the backward chainer is likely to generate, and the backward chainer in turn does not rely on knowledge of the inferences that the forward chainer is likely to make. To do better than this, we would like the question generator to ask the types of questions whose answers are likely to be inferred by the bottom-up processing that is always performed on incoming facts. For example, since the understander needs to know results of actions in order to built motivational explanations based on the goals of the actor, it is reasonable to assume that the forward chainer would always try to determine the results of an action, if possible, either by reading the story or via inference. Thus if a question is transformed into a question about the results of an action, it is reasonable to stop backward chaining and suspend the question at this point, since

the results question is likely to be answered during the forward chaining process. The last criteria for inference control, therefore, is **the likelihood of a question being answered**, which depends on knowledge about the inferences that are likely to be made by the understander's own inference processes.

AQUA does not represent this theory of inference control formally. In other words, there are no explicit functions to compute each of these metrics and to make a decision based on them. However, the heuristics to determine the interestingness of questions and to index questions in memory have been designed keeping these metrics in mind so that the process is efficient.

The main concern in AQUA has been the asking of questions, i.e., the formulation of the questions or hypotheses that the forward chaining procedure is expecting to find. Even with a "weak" approach, asking the right questions is an important problem; with a more knowledge-based approach, this becomes even more critical. The bottom-up processing performed by AQUA on incoming facts corresponds to the running of focus of attention or interestingness heuristics, and will be described in chapter 7. Incoming facts, as well as bottom-up inferences drawn from these facts, are used to retrieve questions using the strategies described in section 5.4.5.

The mechanisms implemented in AQUA include:

- **Question retrieval:** finding suspended goals that F might satisfy.
- **Question indexing:** indexing S so that it is found almost only when it is indeed needed.
- **Process scheduling:** restarting suspended goals that depend on questions when the questions are answered.
- **Hypothesis management:** deleting alternative questions and hypotheses when a question is answered, because their likelihood of being useful decreases since an alternative has been found.

In addition, one might consider heuristics to:

- fade out old useless goals as time goes by, because indexing cost increases and usefulness decreases.
- expend extra resources on inferring an answer to a question that is asked repeatedly, because usefulness and likelihood of being useful increase, and indexing cost would increase if this were not done.
- decide not to bother indexing a question if it was very unlikely to be useful or too expensive to keep around, or if it could easily be derived again later.

Ideally, AQUA should incorporate a *knowledge planner* based on this theory of inference control which uses the interestingness and utility of questions (knowledge goals) to explicitly guide its decisions. Regardless of implementation, however, this is a useful way of thinking about programs that reason about their knowledge goals. For AQUA, these metrics have served as useful constraints when making design decisions.


```

;; An assertion that the AGENT preferred destroying the target to preserving his life.

(goal-ordering      [goal-ordering (goal-actor =agent)
                    (lesser-goal =violated-goal)
                    (greater-goal =satisfied-goal)])

;; XP-ASSERTED-NODES - nodes asserted by the XP as the explanation for the EXPLAINS node.
;; INTERNAL-XP-NODES - internal nodes asserted by the XP in order to link up XP-ASSERTED-NODES to
;; the EXPLAINS node.

;; The boy was a Shiite Moslem and believed fanatically in the religion.

(religion-category [shiiite-moslem.0])
(belief-state      [religious-belief-state
                   (domain =agent)
                   (believed-item [religious-belief (domain =agent)
                                   (co-domain =religion-category)])
                   (belief-strength [fanatic.0])])

;; Explanation links:
;;
;; AGENT is a RELIGIOUS-FANATIC
;; because AGENT is a SHIITE-MOSLEM
;;
;; AGENT wants A-GOAL (DESTROYED-STATE (TARGET))
;; because AGENT is a SHIITE-MOSLEM and AGENT is a RELIGIOUS-FANATIC
;;
;; AGENT prefers A-GOAL (DESTROYED-STATE (TARGET)) over P-LIFE (AGENT)
;; because AGENT is a SHIITE-MOSLEM and AGENT is a RELIGIOUS-FANATIC
;;
;; AGENT did the BOMBING
;; because the BOMBING satisfied a greater goal even though it violated a lesser goal
;; (PLAN-SELECTION, represented as an abstract XP involving a DECISION structure).

(links          (set: {[emotionally-results (domain =religion)
                                             (co-domain =belief-state)]

                    [emotionally-initiates (domain =belief-state)
                                             (co-domain =satisfied-goal)]

                    [emotionally-initiates (domain =belief-state)
                                             (co-domain =goal-ordering)]

                    [xp-sacrifice (agent =agent)
                                   (action =explains)
                                   (role =role)
                                   (good-state =good-state)
                                   (main-result =main-result)
                                   (bad-state =bad-state)
                                   (side-effect =side-effect)
                                   (satisfied-goal =satisfied-goal)
                                   (violated-goal =violated-goal)
                                   (goal-ordering =goal-ordering)]})))

```

This is the definition of the religious fanatic XP as actually seen by the program. The XP is described in section 3.2.5. When the definition is read into memory, it is converted into a graph representation similar to that shown in figure 3.2.

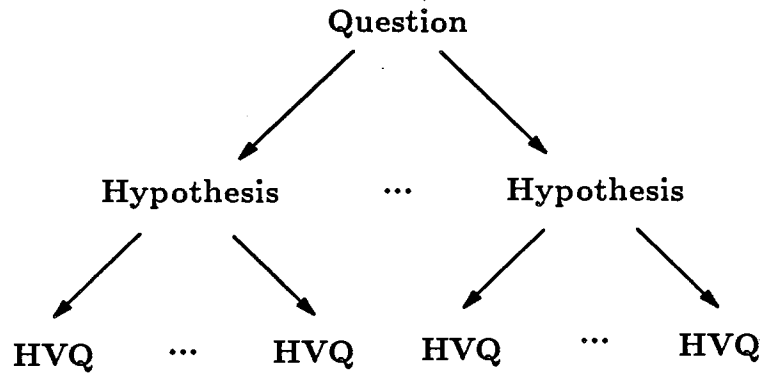


Figure 5.3: The structure of a hypothesis tree.

5.3.2 Representing hypotheses

When there are multiple possible answers for a question, AQUA maintains a *hypothesis tree* to represent the alternatives. For example, when there is more than one possible explanation for an anomaly, the hypothesis tree represents alternative explanations, as well as verification questions for these explanations. These questions in turn can have hypotheses for their answers, which in turn can have further verification questions.

The structure of a hypothesis tree is similar to an OR-AND tree, as shown in figure 5.3. Each ANSWER node is a hypothesis for the QUESTION, which will ultimately be answered by one of these nodes. Each HVQ node is a verification question for ANSWER. Every HVQ must be answered before the hypothesis represented by that ANSWER node is confirmed. HVQ nodes are just QUESTION nodes which in turn could have alternative ANSWER nodes under them.

If all the HVQ nodes for a particular hypothesis are confirmed, that hypothesis is confirmed as the answer for the question, and its alternatives are refuted. If any HVQ for a hypothesis is refuted, i.e., an answer is found that does not match the concept specification of the HVQ, the corresponding hypothesis is refuted and the ANSWER node deleted from the tree (see section 5.4.6).

The nodes in the tree are not special or separate. The hypothesis tree is a structure that links regular nodes representing frames in AQUA's memory, and is similar to a data dependency network such as that of [Doyle, 1979]. Each node in memory is tagged with TRUTH and STATUS values to allow AQUA to represent the ontological status of that node.

5.3.3 The TRUTH slot

The TRUTH slot holds a label that represents AQUA's belief in the concept represented by a node in memory. The label **in** (**out**) mark concepts that are known to be true (false) [Doyle, 1979]. The label **hypothesized** marks nodes whose truth is unknown. All hypotheses are marked as being **hypothesized**. Concepts that are hypothesized to be true (false) are marked as being **hypothesized-in** (**hypothesized-out**).

Thus there are five possible values for the TRUTH slot:

in	known to be true
hypothesized-in	hypothesized to be true
unknown	truth unknown
hypothesized-out	hypothesized to be false
out	known to be false

5.3.4 The STATUS slot

Since all frame representations that AQUA deals with are nodes in memory, the STATUS slot is used to tag nodes to indicate the type of information they represent. This tag can have one of the following values:

story-instance	real instance from current story
inferred-prototype	prototype inferred as slot filler
predefined-instance	predefined instance already in memory
var	dummy variable frame for variable binding
question	hypothesized instance representing an instance

In addition, AQUA distinguishes between types and instances or tokens of those types.

5.4 Opportunistic memory architecture

AQUA's memory is built on top of an opportunistic memory architecture [Hammond, 1988; Hayes-Roth and Hayes-Roth, 1979; Birnbaum and Collins, 1984; Birnbaum, 1986; Dehn, 1989], which provides a uniform way to integrate question-based processing for different types of understanding tasks. The underlying memory model is based on the theory of dynamic memory [Schank, 1982], such as that used by IPP [Lebowitz, 1980] or CYRUS [Kolodner, 1980]. In addition, AQUA's memory contains questions indexed along with the concepts in memory, each with its own reason for being asked. When these questions are answered, perhaps opportunistically, AQUA can restart the suspended computation.

The mechanisms required for this process include:

- A memory architecture for opportunistic question answering.
- A method of suspending reasoning processes when there is insufficient data for them to operate, and a method of restarting suspended processes when the data comes in, when the questions that they depend on get answered.

- A mechanism for generating questions for different types of reasoning, and associating them in memory with the corresponding suspended processes.
- A method of indexing questions in memory so that they can be found when possible answers come in. Obviously, all questions cannot always be active at all times, nor is it feasible to check every incoming fact against every pending question. Thus we would like questions to be found only when they are potentially relevant.

5.4.1 The understanding cycle

The processing cycle in AQUA implements the following algorithm. There are three interacting steps: READ, EXPLAIN and GENERALIZE.

5.4.1.1 The READ step

AQUA reads a piece of text, guided by the questions in memory. It tries to answer these questions using the new piece of information.

Read some text, focussing attention on interesting input as determined below. Build minimal representations in memory.

Retrieve extant questions indexed in memory that might be relevant, i.e., whose concept specifications are satisfied by the new input. Use these questions as an interestingness measure to focus the read above.

Answer the questions retrieved in the previous step. Unify the answer with each question, and restart the suspended process represented by the task specification. E.g., if the question is an HVQ:

Answer HVQ by either confirming or refuting it.

Propagate back to the hypothesis that the question originated from.

Confirm/refute hypotheses. If the HVQs of a hypothesis are confirmed, confirm the hypothesis and refute its competitors. If any HVQ of a hypothesis is refuted, refute the corresponding hypothesis.

Explain the new input if necessary, i.e., if interesting and not already explained.

5.4.1.2 The EXPLAIN step

The EXPLAIN step implements the explanation cycle that was discussed in section 3.5, and is summarized here:

Check input for anomalies

Formulate XP retrieval questions

Retrieve XPs

Apply XP to input:

If in applying the XP an anomaly is detected, **characterize** the anomaly and **explain** it recursively.

If the XP is applicable to the input:

Construct hypothesis by instantiating the explanation pattern.

Construct HVQs to help verify or refute the new hypothesis.

Index HVQs in memory to allow them to be found in the next step.

Answer questions by reading further, focussing attention on input concepts that trigger questions in memory.

Confirm/refute hypotheses when their HVQs are answered, as appropriate.

5.4.1.3 The GENERALIZE step

Since questions represent the knowledge goals of the understander, they should provide the focus for learning. AQUA can:

Generalize novel answers to its questions.

Index these answers in memory, so that the task that originally generated the question would now find the information instead of failing.

This is discussed in chapter 6.

5.4.2 Process model for running understanding tasks

Each understanding task in the above algorithm is implemented using the following process model:

Run understanding task

- Generate questions representing what that task needs to know. If answers are known, continue task, otherwise:
 - Index questions in memory.
 - Suspend understanding task.
 - Restart later when answers come in.
 - Learn if answers are novel.

The basic top-level task is as follows:

Process new fact

- Index into memory to see if new fact is relevant to any questions in memory. If a question is found, answer question and restart suspended tasks.
- Run bottom-up tasks on new fact (e.g., interestingness heuristics, anomaly detection).

The top-level process model is simple and uniform. Individual tasks are characterized by the questions that they generate. These tasks run as individual routines which interact through memory.

The opportunistic memory model provides a clean framework for implementing the theory of question-driven understanding, although other implementations are certainly possible.

5.4.3 Representation of questions

When an understander is trying to reason about something, e.g., it is trying to explain something that seems anomalous, and it needs to know something that isn't there in memory, it formulates a question that is indexed in memory at the point at which it expected to find the information. The question consists two parts:³

1. **Concept specification:** what the question is looking for.
2. **Task specification:** what to do with the answer once it comes in, which depends on why the question was asked.

5.4.3.1 Concept specification

The concept specification represents the information that the question is looking for. This is represented using a memory structure that specifies what would be minimally acceptable as an answer to the question. A new piece of knowledge is an answer to a question if it matches the specification completely. The answer could specify more than the question required, of course.

The concept specification looks like any other memory structure, except that its TRUTH slot is marked with the label *hypothesized*, *hypothesized-in* or *hypothesized-out*, as appropriate. When the question is answered, the concept becomes in or out.

For example, consider the question "For what purpose did the terrorist group recruit the boy?" This is represented as an unknown action mop.42:

```
(define-instance RECRUIT.3
  (isa      (set: {persuade-method}))
  (actor    [terrorist-group.9])
  (requested-mop [mop.42])
  ...)

(define-instance MOP.42
  (isa      (set: {mop}))
  (planner  [terrorist-group.9])
  (actor    [boy.12])
  (status   [question])
  (truth    [hypothesized]))
```

³In general, a question might also specify *knowledge plans* to use in seeking answers, or good places to look for the answers [Hunter and Silbert, 1987].

The question frame is indexed in memory like a normal frame. For example, since the question, when answered, will be used to fill the `requested-mop` slot of `recruit.3`, the question frame is placed in that slot.

Similarly, the question "Was the boy a Shiite Moslem?" is represented as a hypothesized-in religion assertion `religion.68`:

```
(define-instance BOY.12
  (isa      (set: {boy}))
  (age     ...)
  (nationality ...)
  (religion [religion.68
              (isa      (set: {religion}))
              (domain   [boy.12])
              (co-domain [shiiite-moslem.0])
              (status   [question])
              (truth    [hypothesized-in])]))))
```

5.4.3.2 Task specification

The task specification represents what to do with the answer once it comes in, which depends on why the question was asked. This may be represented either as a procedure to be run, or as a declarative specification of the suspended task. When the question is answered, either because the program actively pursued it, or opportunistically while it was processing something else, the suspended process that depends on that information is restarted, as discussed in section 5.4.6.

In the actual implementation of AQUA, I have used both declarative and procedural representations at different times. For example, since the question "Is the boy a Shiite Moslem?" is an HVQ for the `xp-religious-fanatic` explanation, the `task-specification` slot of `religion.68` might contain either a closure of the form:

```
(define-frame RELIGION.68
  (isa      (set: {religion}))
  (task-specification (closure: (lambda ()
                                  (restart-explanation ...))))
  ...)
```

or a declaration of the form:

```
(define-frame RELIGION.68
  (isa      (set: {religion}))
  (task-specification (set: {(hvq-for xp-religious-fanatic.65)}))
  ...)
```

Both representations are equivalent for the purposes of restarting suspended understanding tasks. However, if the program needs to reason about the purpose of the question, a declarative representation is necessary because it allows the program to access the internals of the task that produced the question. For example, if the program is trying to decide which of two questions is more interesting or important, it might use a heuristic that preferred explanation questions to, say, text-level questions. In this case, a closure would not suffice as a task specification.

5.4.4 Indexing questions

Where should a question be placed in memory? Since a potential answer to a question may arrive at any time, particularly when the question may not even be "active," the question must be indexed in memory exactly where the answer would be placed when it does come in. This ensures that the question will be found without extensive searching through lists of questions. The issue of the amount of backward chaining that should be done at this point, before the question is indexed, is addressed in section 5.2.2.

Thus questions are indexed in memory on the basis of their concept specifications. Further, since memory gives rise to expectations that guide the parser, these questions are used to generate such expectations when the concepts to which they are attached are active.

5.4.5 Retrieving questions

When a new fact comes in, either because it is read in the story, or because it is inferred for some other reason, AQUA needs to retrieve questions in memory that the fact could be relevant to. The questions retrieved determine how useful that fact is. AQUA's question retrieval strategies take advantage of the fact that questions are indexed on the basis of their concept specifications in an inheritance hierarchy.

AQUA uses three question retrieval strategies:

5.4.5.1 Type retrieval

When a new concept is activated, questions indexed off the types of the concept are retrieved. The concept is matched against the question to ensure that it would indeed answer the question, as described in section 5.2.1.1. For example, if AQUA reads about a car, it retrieves questions off the car concept to see if the car it read about could answer any of these questions.

5.4.5.2 Relation retrieval

Slots and slot fillers specify relations between concepts. For example, the *results* slot specifies a causal relation of a particular kind between a *mop* and a *state*. Similarly, the *actor* slot specifies a participatory relation between a *mop* and a *volitional-agent*.

Relations are represented as frames in memory (e.g., see [Wilensky, 1986]). This enables AQUA to reason about the relations themselves. For example, the proposition "the boy knew that the bombing would result in his own death" is represented as a *knowledge-state* of the boy whose *mobject* is the *results* relation between the bombing MOP and the *death-state* whose object is the boy.

```

(define-instance KNOWLEDGE-STATE.23
  (isa      (set: {mental-state}))
  (domain   [boy.12])
  (co-domain [physically-results.20
    (isa      (set: {results}))
    (domain   [bombing.18
      (actor [boy.12])
      (planner [terrorist-group.9])
      (object ...)
      (results [death-state.19])])]
    (co-domain [death-state.19 (domain [boy.12])])]))

```

In order to represent this kind of proposition, the `results` relation must be represented explicitly in addition to being a slot in the `bombing` frame. Chapter 8 discusses the representation of relations in more detail.

Questions seeking relations between concepts are indexed in the appropriate slots in these concepts. When a new relation is activated, questions indexed in these slots are retrieved. The new relation is then matched against the questions to ensure that it would indeed answer the questions.

For example, if AQUA reads that a bombing resulted in the actor's death, it retrieves questions indexed in the `results` slot of the `bombing` frame, as well as the inverse slot of the `death-state` frame, to see if any questions were seeking this information. There are three possibilities:

- A question specifies both the action and the resulting state. In this case the question is trying to verify the relation.
- A question specifies only the action. In this case the question is trying to find out what the action resulted in, which is answered with `death-state`.
- A question specifies only the resulting state. In this case the question is trying to find out what caused the state, which is answered with `bombing`.

Other relations are handled in a similar manner. The three possibilities shown above are merely for the purposes of exposition; AQUA does not actually have to reason about the individual possibilities since the matcher automatically takes care of this, as described in section 5.2.1.1.

5.4.5.3 Specialization retrieval

To determine how a new concept should be specialized, AQUA retrieves questions indexed off the specializations of this concept to see if a particular specialization would address an extant question. As discussed in section 5.1.2, this allows the understanding process to be sensitive to the questions that the understander is currently seeking answers to. This strategy is used for determining the interestingness of a concept and will be discussed in chapter 7.

5.4.6 Answering questions

When a question is posed, there are many ways to try to answer it. An understander could use inference techniques, such as analogy or default reasoning, to try to figure out the answer, or it

could just read more of the story in the hope of finding the answer. As currently implemented, AQUA uses simple inference techniques to see if it can answer the question, failing which it resorts to reading further. The uniformity of the design makes it easy to add different inference techniques if required. For example, AQUA can use the TWEAKER program [Kass, in preparation] to infer answers to questions by tweaking past cases.

Inferences made to answer questions fall into two categories:

1. **Definitional inferences:** These inferences are made using knowledge that is part of the definition of concepts in memory. For example, it is definitional of the **suicide** concept that suicide results in the **death-state** of the actor. Similarly, **gender = male** is definitional of **male-person**, and therefore of all its specializations in memory, such as **boy**. These inferences can always be made with certainty.
2. **Stereotypical inferences:** For example, the **p-life** goal to preserve one's life is stereotypical of all people, but not definitional. Similarly, most Lebanese people are Moslems, but this not always true; thus **religion = moslem** is stereotypical of **lebanese-person** but not definitional. Inferences made using this knowledge can only be confirmed by reading the story.⁴

When questions are answered, either through inference or by reading the story, suspended tasks that depend on the answer are restarted. This is straightforward since the task specification is part of the representation of the question. For example, suppose AQUA has built two hypotheses to explain a suicide bombing:

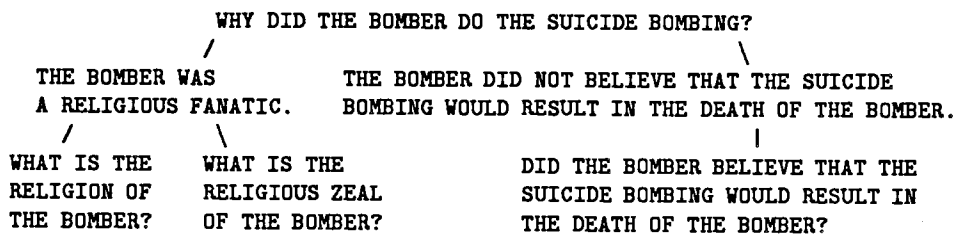
Applying XP-RELIGIOUS-FANATIC to THE BOMBER DID THE SUICIDE BOMBING.
 THE BOMBER DID THE SUICIDE BOMBING BECAUSE THE BOMBER WAS A
 RELIGIOUS FANATIC.

Installing HVQs to verify XP:
 WHAT IS THE RELIGION OF THE BOY?
 WHAT IS THE RELIGIOUS ZEAL OF THE BOY?

Applying XP-NOT-KNOW-OUTCOME to THE BOMBER DID THE SUICIDE BOMBING.
 THE BOMBER DID NOT BELIEVE THAT THE SUICIDE BOMBING WOULD RESULT
 IN THE DEATH OF THE BOMBER.

Installing HVQs to verify XP:
 DID THE BOMBER BELIEVE THAT THE SUICIDE BOMBING RESULTED IN THE
 DEATH OF THE BOMBER?

The hypothesis tree for this story is as follows:



⁴AQUA assumes that the stories it reads are always true.

If the HVQs of the religious fanatic hypothesis are answered, that hypothesis is confirmed and its competitors are refuted:

Answering question: WHAT IS THE RELIGION OF THE BOMBER?
with: THE BOMBER IS A SHIITE MOSLEM.

Answering question: WHAT IS THE RELIGIOUS ZEAL OF THE BOMBER?
with: THE BOMBER IS VERY ZEALOUS ABOUT THE SHIITE MOSLEM
RELIGION.

Confirming hypothesis:
THE BOMBER DID THE SUICIDE BOMBING
because THE BOMBER WAS A RELIGIOUS FANATIC.

Refuting alternative hypothesis:
THE BOMBER DID NOT BELIEVE THAT THE SUICIDE BOMBING WOULD RESULT
IN THE DEATH OF THE BOMBER.

Retracting HVQs:
DID THE BOMBER BELIEVE THAT THE SUICIDE BOMBING RESULTED IN THE
DEATH OF THE BOMBER?

Thus AQUA uses the hypothesis tree structure to manage alternative hypotheses and questions. The concept specifications of its questions represent what AQUA needs to know, and the task specifications represent why it needs to know this information. These are then used to restart suspended tasks when the information comes in and the questions are answered.

5.4.7 Advantages of the opportunistic memory model

The opportunistic memory model provides a natural implementation of the theory of question-driven understanding. The advantages of this model have been discussed throughout this section and are summarized here:

1. The top-level process model is simple and uniform.
2. Various understanding tasks are implemented uniformly and can interact via memory. Many interesting questions come from the interaction between explanation and memory, and this model provides a useful way of studying the process.
3. Other reasoning processes, such as tweaking, can be integrated easily.
4. The pursuit of answers to questions can be viewed as a planning task. This enables one to write heuristics for determining the best strategy for achieving knowledge goals. Since these goals are represented explicitly, the system can decide to pursue or abandon them, or just leave them in memory until they are achieved.
5. Questions can be answered opportunistically. Again, this is possible only because questions are represented explicitly in memory.
6. Bottom-up (text-driven) and top-down (goal-driven) processing are combined, allowing the system to use its questions to determine the interestingness of input without being totally insensitive to unusual input.

Chapter 6

Theory of learning

Novel stories often involve people with goals, plans and beliefs that are different from those that the understander knows about. Such stories result in anomalies, which arise from contradictions between the observed situation and expectations provided by knowledge structures in memory.

When there is a processing failure, the understander needs to be able to identify the problem and invoke the appropriate learning strategy to correct the problem. In order to do this, the understander must be able to reason about its own reasoning process that led to the problem. The theory of question-driven understanding provides a natural way to do this. Since questions arise from understanding tasks, a question in memory represents *why the question was asked* in addition to *what the question is asking about*. The question underlying an anomaly, therefore, represents the chain of reasoning that led up to the anomaly. When the question is answered, the understander can use this chain of reasoning to place the answer appropriately in memory.

Thus, questions, the knowledge goals of the understander, provide the focus for learning. Although this is true for all questions arising from understanding tasks, we will concentrate mainly on explanation questions. As implemented, AQUA improves its explanatory knowledge of its domain by learning from answers to explanation questions.

6.1 Incremental theory formation: Where do XPs come from?

Case-based reasoning and learning programs deal with the issue of using past cases to understand, plan for, or learn from novel situations [Kolodner *et al.*, 1985; Simpson, 1985; Alterman, 1986; Bain, 1986; Hammond, 1986; Schank, 1986; Turner, 1986; Sycara, 1987; Kolodner, 1988]. This happens according to the following process:

- Use problem description to get reminded of old case.
- Retrieve the results (lessons, XPs, plans) of processing the old case and give them to the understander, planner or problem-solver.

- Adapt the results from the old case to the specifics of the new situation.
- Apply the adapted results to the new situation.

The intent behind case-based reasoning is to avoid the effort involved in re-deriving these lessons, XPs or plans by simply reusing the results from previous cases. However, this process assumes that past cases are well understood and provide good "lessons" to be used for future situations. This assumption is usually false when one is learning about a novel domain, since cases encountered previously in this domain might not have been understood completely. Instead, it would be reasonable to assume that the understander would have questions associated with such cases, representing what it hadn't yet understood about the case.

Even if past cases are not well understood, they can still be used to guide processing in new situations. However, in addition to using the past case to understand the *new* situation, the understander can also learn more about the *old* case itself. In other words, the understander can perform a kind of *opportunistic learning* during case-based reasoning.

For example, suppose the understander retrieves past explanations from situations already in memory, and uses them to build explanations to understand new situations. If the old explanation is not well understood, there is the possibility of learning more about that explanation in the process of applying it to a new situation. Since the understander has already made the effort to retrieve the old explanation, it should use this opportunity to update its memory and build a better explanation for the future. It may even re-understand the old case in a new light and come to a better understanding of it. If the old case was inappropriately retrieved, the understander may also learn better retrieval cues or indices for the old explanation so that it is retrieved appropriately in the future.

Now consider the theory of explanation based on XPs that we have presented in this dissertation. How are stereotypical XPs, such as those used by AQUA, formed in memory? Explanation-based learning provides a way to generalize novel explanations encountered in particular situations into XPs that are applicable to similar situations in the future [DeJong, 1983; DeJong and Mooney, 1986; Mitchell *et al.*, 1986]. However, it is difficult to determine the correct level of generalization.

Furthermore, many stories do not provide enough information to prove that the explanation is correct. The understander must often content itself with two or more competing hypotheses, or otherwise jump to a conclusion. This means that the understander's memory of past cases does not always contain "correct" cases or "correct" explanations, but rather one or more hypotheses about what the correct explanation might have been.¹ These may be hypotheses that were entertained but never verified, or hypotheses that the understander believed in that were grudgingly discredited due to weakly believed data, or even confirmed hypotheses that are still partly suspect. These hypotheses often have questions attached to them, representing what is still not understood or verified about those hypotheses.

As the understander reads new stories, it is reminded of past cases, and of old explanations that it has tried. In attempting to apply these explanations to the new situation, its understanding of

¹ Actually, a single story or episode can provide more than one "case," each case being a particular interpretation or dealing with a particular aspect of the story. For an explanation program, each anomaly in a story, along with the corresponding hypothesis tree, can be used as a case.

the old case gradually gets refined. Old hypotheses get elaborated or generalized, and bad ones get rejected.

In some cases, the understander eliminates all but one hypothesis. This then becomes the "standard interpretation" of the old case and serves as a summary of it. Whenever the old case is recalled, it is always viewed in that way, and the "standard lesson" attached to it is used to understand new stories in that same light. Details irrelevant to this lesson fade away. To use Schank's term, the old case becomes *ossified*; it begins to look more and more like an XP.

Thus XP learning is an incremental process of theory formation [Ram, 1989]. A detailed and situation-specific explanation gradually becomes generalized and can be applied to a larger variety of situations. New indices are learned as the understander learns more about the range of applicability of the XP. The case-turned-XP is re-indexed in memory and is more likely to be recalled only in relevant situations.

6.2 Incremental learning in AQUA

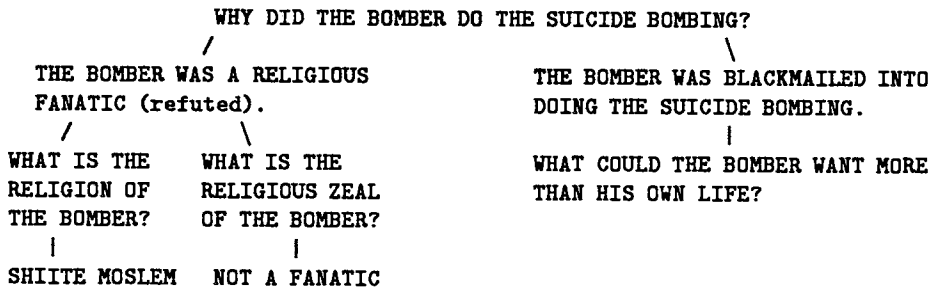
Although AQUA does not start out with detailed episodic cases, its learning process has been designed keeping the above view of XP learning in mind. AQUA's "case memory" consists of XPs that have been used to explain past situations. AQUA improves its explanatory knowledge of the domain through a process of re-indexing and incremental modification that is similar to the ossification process described above.

For example, suppose AQUA has just read the blackmail story:

S-16: A 16-year-old Lebanese got into an explosive-laden car and went on a suicide bombing mission to blow up the Israeli army headquarters in Lebanon. ...

The teenager was a Shiite Moslem but not a religious fanatic. He was recruited for the mission through another means: blackmail.

After reading this story, AQUA has the following hypothesis tree in memory:



This is what the "case" S-16 looks like in memory. It is a hypothesis tree representing an anomaly raised during this story, one or more hypotheses constructed by applying known XPs to the anomalous situation, verification questions for these hypotheses, and possibly answers to these questions.

In other words, AQUA's understanding of this case consists of the explanation questions that were raised while reading the story, as well as answers to these questions, if any. What does AQUA learn from this story? Furthermore, what does AQUA learn by applying this case to another suicide bombing story?

Many learning programs have dealt with the problem of learning new XPs that the program does not already have (e.g., [Mooney and DeJong, 1985]). However, even if an XP is already available for the current situation, a system could learn by elaborating the XP further, or by filling out details that were not well understood before, or by re-indexing the XP in memory.

I call this kind of learning *incremental learning*, since the system improves its explanatory knowledge of the domain in an incremental fashion rather than by learning new XPs as a whole. In the above example, even though AQUA already has the blackmail XP in memory, it learns the following:

- A new situation index, *suicide-bombing*, for *xp-blackmail*.
- A new stereotype, *lebanese-teenager*, to use as a stereotype index for *xp-blackmail*.
- A new question for future applications of *xp-blackmail* to *suicide-bombing*: *What could the bomber want more than his own life?*
- An answer to this question when another suicide bombing story involving blackmail is read: *The bomber wanted to protect his family even at the expense of his own life*. This results in the blackmail explanation being elaborated.

In the former two cases, AQUA is learning new indices for an existing XP, *xp-blackmail*; in the latter two, it is modifying the XP itself. Let us discuss how AQUA performs these types of learning.

6.3 Learning indices for explanation patterns

Regardless of whether a new XP is learned from scratch or by applying an existing XP to a new situation, the XP needs to be indexed in memory appropriately so that it can be used in future situations in which it is likely to be useful. In other words, the understander should be able to learn new indices for the XP [Leake and Owens, 1986].

For example, consider the blackmail story S-2. This story is interesting because it presents a novel explanation for suicide bombing. It presents a suicide bomber who was *not* a fanatic, i.e., who did not fit the "religious fanatic" stereotype. The story spends some effort in trying to convince the reader that the boy fits the typical "teenager" stereotype. It then presents a novel motivational explanation for suicide bombing: blackmail.

This explanation is novel, not because one has never heard of blackmail, but because it is used in a novel context. Blackmail is not the first explanation that comes to mind when one reads a suicide bombing story. In fact, the story raises a host of questions: Who blackmailed him? How could you blackmail anyone into killing himself? What could he possibly want more than his own life? These questions need to be answered before the explanation can be considered adequate for this context.

In addition to learning a new explanation for suicide bombing, however, an understander also needs to learn when to apply this explanation. It should associate the XP with particular contexts

in which it is likely to be applicable (the *situation index*). It might also learn not to apply an XP in some situations. Finally, an understander should also learn new character stereotypes to serve as indices to the XP (the *stereotype index*).²

In this story, AQUA learns a new context for blackmail (suicide bombing), as well as a new character stereotype representing the type of person who one might expect to see involved in a "blackmailed into suicide bombing" explanation. This stereotype is built through generalization under causal constraints from the hypotheses that were considered, including the ones that were ultimately refuted. This is a type of explanation-based learning [DeJong, 1983; Sussman, 1975; DeJong and Mooney, 1986; Hammond, 1986; Mitchell *et al.*, 1986], since the causality of the situation (as represented by *xp-blackmail*) provides constraints on which aspects of the situation are retained in the learned structure.

Answering question: WHY DID THE BOY DO THE SUICIDE BOMBING?
with: THE BOY WAS BLACKMAILED INTO DOING THE SUICIDE
BOMBING.

Novel explanation for A SUICIDE BOMBING!

Indexing XP-BLACKMAIL
Situation index = SUICIDE BOMBING
Stereotype index = STEREOTYPE.79 (the newly learned stereotype)

Applying XP-BLACKMAIL to THE BOY DID THE SUICIDE BOMBING.

...

Let us discuss how AQUA learns these indices.

6.3.1 Learning situation indices

AQUA learns new contexts (e.g., "suicide bombing") for stereotypical XPs (e.g., "blackmail") which are then used as situation indices for these XPs in the future. The main issue here is how far the context should be generalized before it is used as an index. In the above example, should the new situation index for *xp-blackmail* be *suicide-bombing*, *suicide*, *bombing*, *destroy*, or indeed any *mop* with a negative side effect for the actor? The issue here isn't one of correctness but of utility. As discussed earlier, *xp-blackmail* is a possibility whenever the actor does something he would ordinarily not do because of a negative side effect. However, XP theory tries to replace generalized reasoning of this form with specific reasoning about stereotypical situations. The latter is more efficient even though it is less general.

After reading the above story, for example, one would expect to think of blackmail when one reads another story about a suicide bombing attack. However, one would probably not think of blackmail on reading any story about suicide, say, a teenager killing himself after failing his high school examinations, even though theoretically it is a possible explanation.

Thus in the above example, AQUA uses *suicide-bombing* as the new situation index for *xp-blackmail*. The newly learned index is shown in figure 6.1. After reading several stories about

²The *category index* remains the same since the type of the XP hasn't changed.

blackmail, AQUA would know about different stereotypical situations in which to use the blackmail explanation, rather than a generalized logical description of every situation in which blackmail is a possible explanation. In other words, AQUA would have indexed a copy of *xp-blackmail* under all the MOPs for which it has seen *xp-blackmail* used as an explanation. Whenever these MOPs are encountered, AQUA would retrieve the blackmail XP (if the category and stereotype indices are also present). The reason that a copy of the original XP is used is that the XP, once copied, may need to be modified for that particular situation as discussed below.

AQUA can still understand other blackmail situations that it has not learned about as yet, since it can use its abstract XPs to form hypotheses. In addition, AQUA can be extended to use a case-based reasoning module to adapt specific blackmail situations it knows about to situations it doesn't know about, as described later. Thus *not having a situation index for an XP does not necessarily mean that the XP cannot be applied to the situation, but rather that this XP is not one that would ordinarily come to mind in that situation.*

6.3.2 Learning stereotype indices

The main constraint on a theory of stereotype learning is that the kinds of stereotypes learned must be useful in retrieving explanations. In other words, they must provide the kinds of discrimination that are needed for indexing XPs in memory. Since volitional explanations are concerned with goals, goal orderings, plans and beliefs of characters, the learning algorithm must produce typical collections of goals, goal-orderings, plans and beliefs, along with predictive features for these elements.

AQUA learns new character stereotypes to use as stereotype indices for XPs that it already has. When AQUA encounters an anomaly, it retrieves XPs and uses them to build hypotheses to explain the anomaly. It then reads the story further, trying to verify these hypotheses. If a novel explanation is found, AQUA generalizes the features of the character involved in the explanation to form a new stereotype (*stereotype.79* in the following example).

Answering question: WHY DID THE BOY DO THE SUICIDE BOMBING?
with: THE BOY WAS BLACKMAILED INTO DOING THE SUICIDE BOMBING.

Novel explanation for A SUICIDE BOMBING!

Building new stereotype STEREOTYPE.79:

Typical goals: P-LIFE (in)
A-DESTROY (ISRAELI OBJECT) (out)
AVOIDANCE-GOAL (SOMETHING) (question)

Typical goal-orderings:
AVOIDANCE-GOAL (SOMETHING) over P-LIFE (question)

Typical beliefs: RELIGIOUS-ZEAL = NOT A FANATIC (in)

Typical features: AGE = TEENAGE AGE (hypothesized)
RELIGION = SHIITE MOSLEM (hypothesized)
GENDER = MALE (hypothesized)
NATIONALITY = LEBANESE (hypothesized)

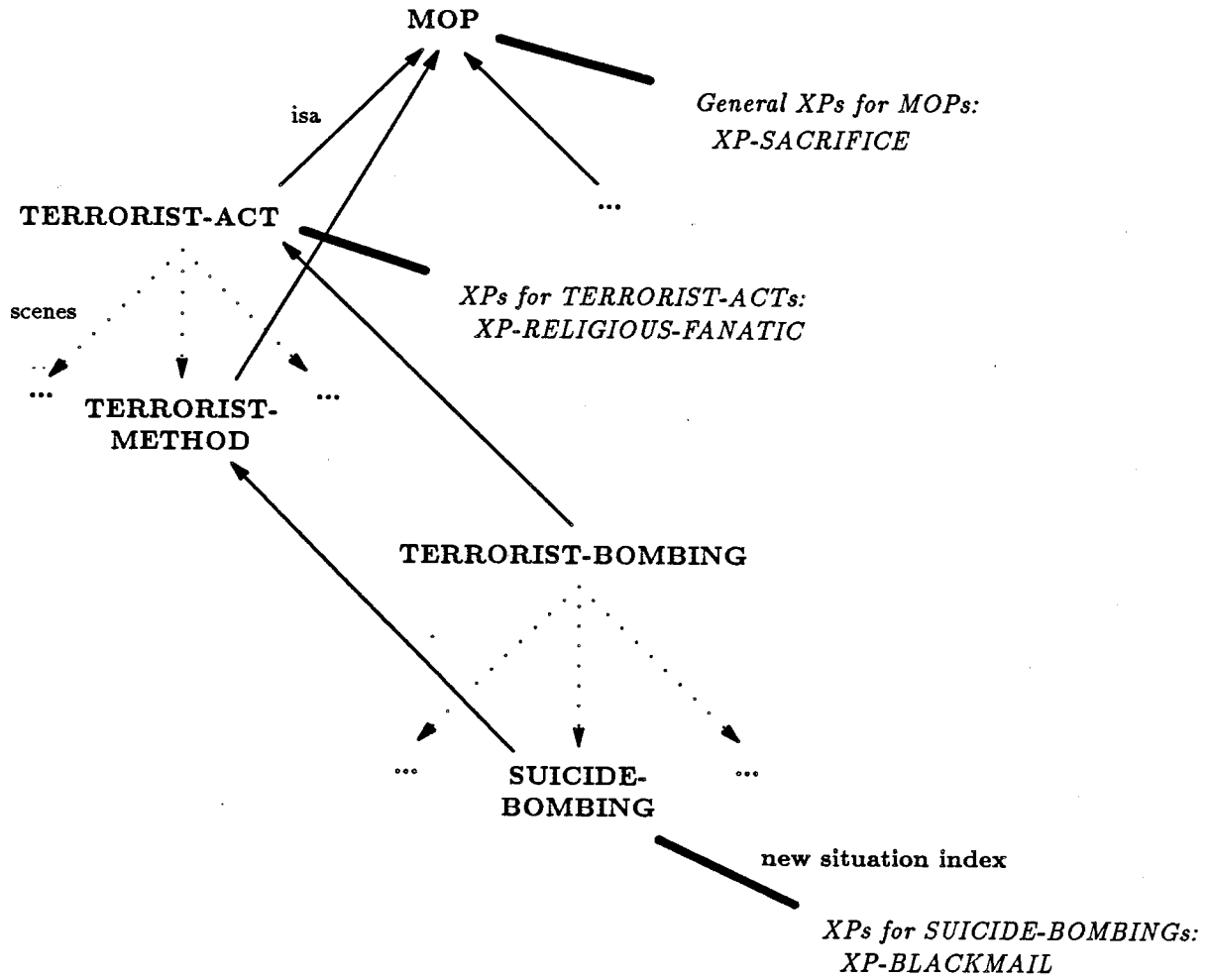


Figure 6.1: Learning situation indices for XPs. Upward lines represent isa links, and downward dotted lines represent scenes of MOPs. Heavy lines represent situation indices, which point from MOPs to XPs. Here, AQUA has just built a situation index from suicide-bombing to a copy of xp-blackmail.

Indexing XP-BLACKMAIL

Stereotype index = STEREOTYPE.79

Applying XP-BLACKMAIL to THE BOY DID THE SUICIDE BOMBING.

...

The label in (out) marks features that are known to be true (false) of this stereotype (see section 5.3.3). These features are definitional of the stereotype. The label question marks features that are in but incomplete. In this case, SOMETHING refers to an unknown goal that needs to be filled in when the information comes in. This is represented as a goal with an unknown goal-object.

Finally, the label hypothesized marks features that were true in this story but were not causally relevant to the explanation. These features are retained for the purposes of recognition and learning. Since AQUA does not assume that its explanations are complete, there is the possibility of learning more about this explanation in the future that would help to determine whether these features have explanatory significance. This has not yet been implemented in AQUA.

The stereotype is used to index the new explanation in memory (figure 6.2). After reading this story, AQUA uses the new stereotype to retrieve the blackmail explanation when it reads other stories about Lebanese teenagers going on suicide bombing missions. For example, if AQUA reads story S-24 after reading the blackmail story, it would ask the following questions:

S-24: 4 Die in Suicide Attack Against Israelis.

JERUSALEM — A young guerrilla drove an explosive-laden car into a group of Israeli guards in Lebanon. The suicide attack killed three guards and wounded two others. ...

The driver was identified by the guerrilla group as a 16-year-old girl, Sana Mohaydaleh. Before the attack, she said that she was prepared to die in order to kill her enemies because she loved her country.

Trying to explain WHY DID THE GIRL DO THE SUICIDE BOMBING?

Searching for stereotypical XPs

Retrieved stereotypical XPs:

XP-BLACKMAIL-SUICIDE-BOMBING

Applying XP-BLACKMAIL-SUICIDE-BOMBING to THE GIRL DID THE SUICIDE BOMBING.

Installing HVQs to verify XP:

DID THE TERRORIST GROUP PERSUADE THE GIRL BY THREATENING TO DO A MOP THAT WOULD RESULT IN A STATE?

DID THE GIRL WANT TO PREVENT THIS STATE?

DID THE GIRL NOT VOLUNTEER FOR THE SUICIDE BOMBING?

This example illustrates the effect of learning on AQUA's understanding of future stories. Note that these questions are very different from the ones that AQUA asked when it read the same

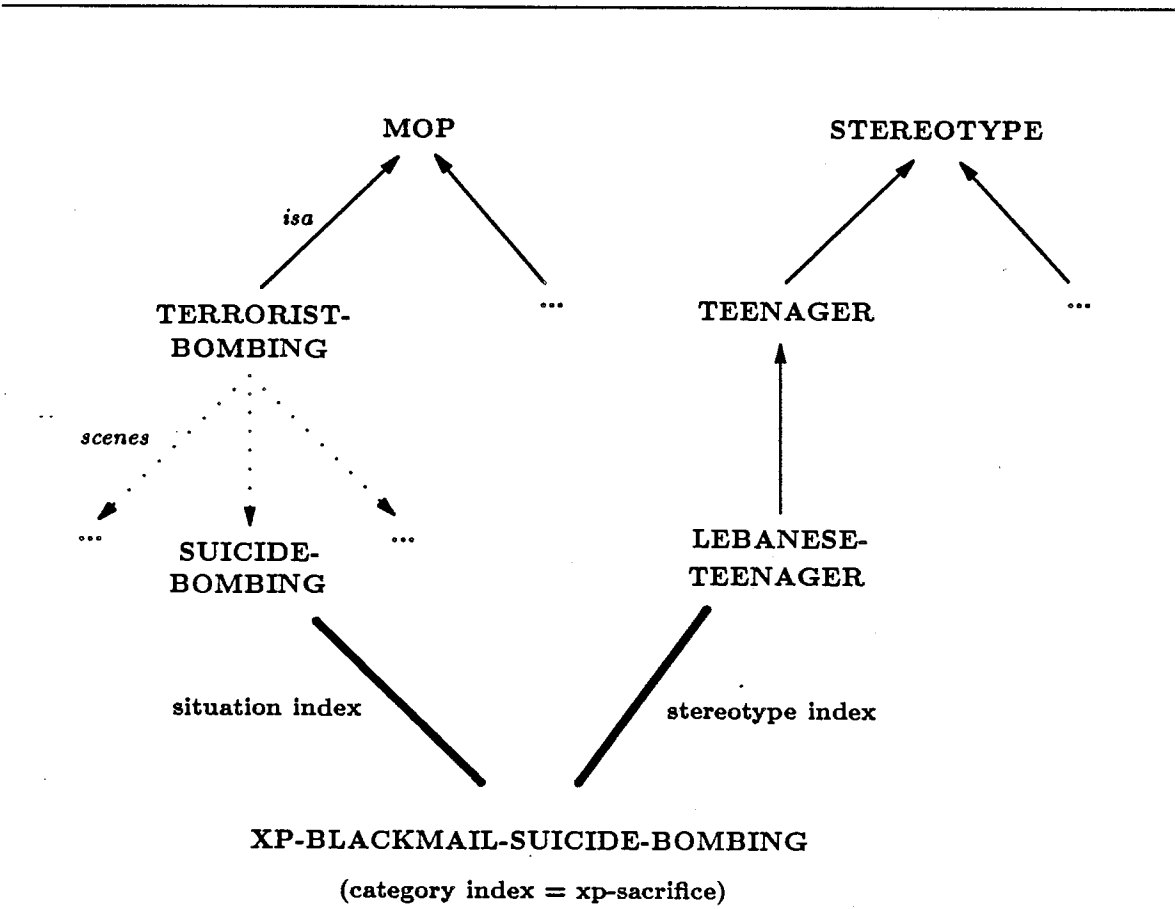


Figure 6.2: Learning stereotype indices for XPs. Upward lines represent *isa* links, and downward dotted lines represent *scenes* of MOPs. Heavy lines represent indices to XPs. Here, AQUA has just built a stereotype index from *stereotype.79*, representing a *lebanese-teenager*, to *xp-blackmail-suicide-bombing*.

story without previously having read the blackmail story, as demonstrated earlier. Thus this example also illustrates the effect of current hypotheses on the understanding of the story (subjective understanding).

6.3.2.1 Learning from successful explanations

How does AQUA build the new stereotype? Clearly, much of the stereotype comes from the motivational aspects of the blackmail explanation. AQUA retains those goals, goal orderings and beliefs of the character in the story that are causally implicated in the blackmail explanation. Since blackmail relies on a goal ordering between two goals, one of which is sacrificed for the other, the stereotype must specify that the character has a goal that he or she values above *p-life* [Schank and Abelson, 1977]. The stereotype also specifies that the character would normally not have the goal of destroying objects belonging to the Israeli, since this is part of the blackmail explanation.

In other words, *xp-blackmail* specifies that:

- (1) The blackmailee has a goal G1.
- (2) The blackmailee does not have the goal G2 (since otherwise he or she would satisfy the goal without needing to be threatened).
- (3) The blackmailee had a third goal G3, which he or she valued above goal G1.
- (4) The blackmailer threatened to violate G3 unless the blackmailee performed an action that satisfies G2, even though the action would have a negative effect of violating G1.

The semantics of the word "blackmail," as described by prototype theory, specify a prototypical blackmail as represented by the stereotypical XP shown above.³ For example, in a classic blackmail situation where, say, a rich businessman is threatened with photographs of his affair:

- (1) The businessman has the goal to keep his money.
- (2) The businessman does not have the goal to increase the blackmailer's wealth.
- (3) The businessman has the goal to keep his affair secret from his wife, which is more important than his goal to hold on to his money.
- (4) The blackmailer threatens to violate the businessman's goal to keep his affair secret unless the businessman gives the blackmailer some money, which would satisfy the blackmailer's goal to get more money even though it would violate the businessman's goal to keep his money.

In the suicide bombing situation, AQUA infers the following goals and goal-orderings for the actor (corresponding to (1), (2) and (3) above):

Building new stereotype (STEREOTYPE.79):

Inferring GOALS from XP-BLACKMAIL (successful):

THE ACTOR WANTED TO PRESERVE HIS OWN LIFE.

³The threat involved in blackmail is usually, but not always, a threat of exposure; other kinds of threats may be closer to "extortion." However, the underlying volitionality in both cases is the same, and represented by the same decision model.

THE ACTOR DID NOT WANT TO DESTROY A PHYSICAL OBJECT
BELONGING TO THE ISRAELI ARMY.⁴

THE ACTOR WANTED TO AVOID SOMETHING.

Inferring GOAL-ORDERINGS from XP-BLACKMAIL (successful):

THE GOAL OF THE ACTOR TO AVOID SOMETHING WAS MORE IMPORTANT
THAN THE GOAL OF THE ACTOR TO PRESERVE HIS OWN LIFE.

These goals and goal-orderings are added to the stereotype being built. At this point, *stereotype.79* has the following features:

Typical goals: P-LIFE (in)
A-DESTROY (ISRAELI OBJECT) (out)
AVOIDANCE-GOAL (SOMETHING) (question)
Typical goal-orderings:
AVOIDANCE-GOAL (SOMETHING) over P-LIFE (question)

6.3.2.2 Learning from failed explanations

Many explanation-based learning programs learn only from positive examples (e.g., [Mooney and DeJong, 1985; Segre, 1987]). However, it is also possible to apply this technique to learn from negative examples (e.g., [Mostow and Bhatnagar, 1983]). AQUA uses refuted hypotheses to infer features that should *not* be present in the newly built stereotype. These are features which, if present, would have led to the hypothesis being confirmed.

For example, in the blackmail story, AQUA knows that the person being blackmailed is not a religious fanatic, since the religious fanatic explanation, which depended on this fact, has been refuted. The kind of person likely to be blackmailed into suicide bombing is, therefore, not a religious fanatic.⁵ This feature is recorded in the newly built stereotype.

Building new stereotype (STEREOTYPE.79):

XP-RELIGIOUS-FANATIC failed because:

THE BOY DID NOT BELIEVE FANATICALLY IN THE SHIITE MOSLEM
RELIGION.

Inferring BELIEFS from XP-RELIGIOUS-FANATIC (failed):

THE ACTOR DID NOT BELIEVE FANATICALLY IN A RELIGION.

⁴Clearly, this assertion should be generalized to state that the actor did not want to destroy a physical object belonging to the *enemy*, not specifically to the *Israeli army*. Although this has not yet been implemented in AQUA, it would be straightforward to implement an explanation-based generalizer that could build this generalization under causal constraints from *xp-blackmail*. Currently, AQUA generalizes only by walking up the inheritance hierarchy.

⁵As before, this is a stereotypical inference and not a logically correct one. A religious fanatic could indeed be blackmailed into suicide bombing; however, on reading a story about a religious fanatic going on a suicide bombing mission, *xp-blackmail* would not be the first XP to come to mind.

This results in a new belief being added to `stereotype.79`:

Typical beliefs: RELIGIOUS-ZEAL = NOT A FANATIC (in)

The reason that learning from the failed explanation works in this example is that the blackmail explanation specifies that the person being blackmailed would normally not have the goal to perform that action. There would be no point in blackmailing a religious fanatic into doing a suicide bombing if he would be willing to do it anyway. In order to make this inference, the understander needs to reason about the planning behavior of the blackmailer.

The features of the failed explanation to include depend on the nature of the failed explanation itself. For example, even though depressed teenagers often commit suicide, they can also die in other ways. Reading about a teenager who was hit by a truck does not automatically rule out the fact that he was depressed, even though the "depressed teenager commits suicide" XP is inapplicable. However, it does mean that `xp-depressed-teenager` should not be indexed under "hit by a truck," at least on the basis of this example.

Thus learning from failed explanations depends on the nature of the failed explanation, as well as its relationship to the successful explanation. AQUA uses a simple algorithm to learn from failed explanations, but more work needs to be done in this area.

6.4 Modifying existing explanation patterns

AQUA learns new XPs primarily through the incremental modification of existing XPs, and the learning of new indices for these XPs. Whenever an explanation question is answered, AQUA can learn from the new answer by elaborating the XP or re-indexing it in memory, depending on the task that spawned the explanation question in the first place. If an explanation question is raised but not answered, such as an unverified HVQ or an unconfirmed adaptation of an old hypothesis, the question is attached to the XP and raised when that XP is applied in future situations. Whenever the question is answered, the XP is modified to reflect the answer. These processes result in the XP being modified in an incremental manner.

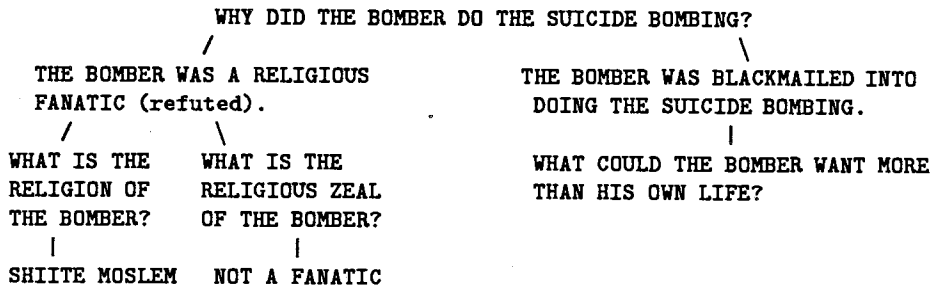
6.4.1 Associating new questions with XPs

We have described AQUA's understanding algorithm as a "story + questions in, answers + new questions out" process. One of the results of applying an XP to an anomalous situation is a hypothesis that resolves the anomaly. However, another result is often the raising of new questions that the understander hadn't thought of before. These questions are associated with the XP, and may be answered when the XP is applied in the future. When they are answered, the understander can elaborate and modify the XP, thus achieving a better understanding of the causality represented by the XP.

Asking new questions and indexing them in memory along with its XPs is an essential part of the learning process. These questions represent what the understander does not quite understand about

the XPs that it already has, i.e., what the understander needs to learn about. These questions give rise to XP application questions (see section 4.1.1.3) during future stories.

For example, consider the following hypothesis tree built after reading the blackmail story:



Suppose AQUA reads the blackmail story with only the religious fanatic XP for suicide bombing in memory. When reading this story, AQUA is handed an explanation for the suicide bombing: the story explicitly mentions that the bomber was blackmailed. In a sense, then, the story has been understood since an explanation for the bombing has been found. However, one could not really say that AQUA had understood the story if it didn't ask the question *What could the boy want more than his own life?* Unless this question is raised while reading the story, one would have to say that AQUA had missed the point of the story.

Trying to explain WHY DID THE BOY DO THE SUICIDE BOMBING?

Answering question: WHY DID THE BOY DO THE SUICIDE BOMBING?
with: THE BOY WAS BLACKMAILED INTO DOING THE SUICIDE BOMBING.

Novel explanation for A SUICIDE BOMBING!

AQUA now uses the blackmail XP to explain why the boy did the suicide bombing, as described in chapter 5. This results in a hypothesis tree, whose verification questions correspond to unconfirmed nodes of the instantiated XP. In the current example, AQUA is left with the following question (see figure 6.3):

Unanswered questions:

DID THE BOY WANT TO PREVENT A STATE MORE THAN THE BOY WANTED TO PRESERVE THE LIFE STATE OF THE BOY?

New questions for XP-BLACKMAIL-SUICIDE-BOMBING:

DID THE ACTOR WANT TO PREVENT A STATE MORE THAN THE ACTOR WANTED TO PRESERVE THE LIFE STATE OF THE ACTOR?

After reading the story, AQUA has learned a new XP for suicide bombings: blackmail. It has not fully understood about the motivations of suicide bombers, of course, since this story does not provide answers to the questions that are pending. However, it has learned the questions to ask about such a situation should it be encountered again, as shown in figure 6.3.

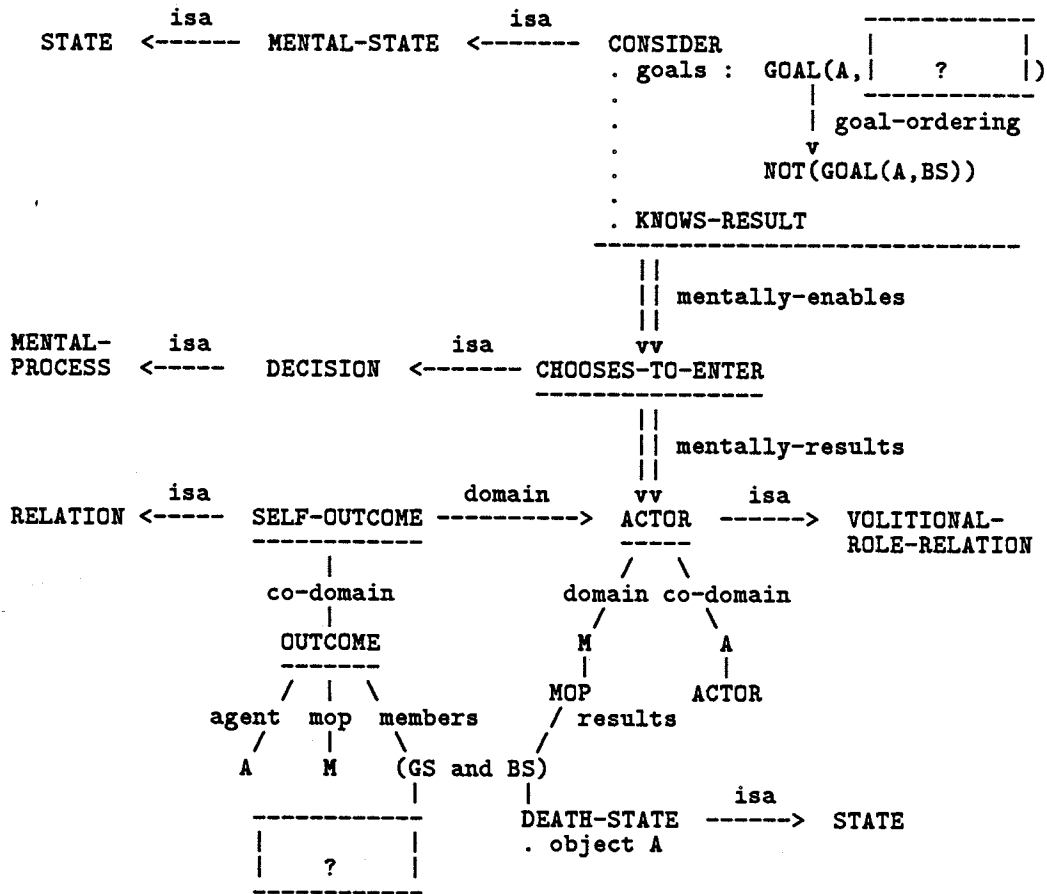


Figure 6.3: Associating new questions with XPs. The unknown goal represents the new question, *What could the actor want more than his own life?* This is depicted as an empty box, representing a gap in the program's knowledge. The XP is elaborated by filling in this gap when this question is answered.

6.4.2 Incremental refinement of XPs by answering questions

In addition to raising new questions, of course, an understander must answer the questions that is already has in order to improve its knowledge of the domain. AQUA performs incremental learning by answering questions associated with XPs.

For example, consider the following story:

S-25: JERUSALEM — A young girl drove an explosive-laden car into a group of Israeli guards in Lebanon. The suicide attack killed three guards and wounded two others. ...

The driver was identified as a 16-year-old Lebanese girl. ... Before the attack, she said that a terrorist organization had threatened to harm her family unless she carried out the bombing mission for them. She said that she was prepared to die in order to protect her family.

When this story is read, AQUA retrieves the new `xp-blackmail-suicide-bombing` and applies it to the story. The question that is pending along with this explanation is also instantiated:

Trying to explain WHY DID THE GIRL DO THE SUICIDE BOMBING?

Searching for stereotypical XPs

Retrieved stereotypical XPs:

`XP-BLACKMAIL-SUICIDE-BOMBING`

Applying `XP-BLACKMAIL-SUICIDE-BOMBING` to THE GIRL DID THE SUICIDE BOMBING.

Instantiating pending questions:

DID THE GIRL WANT TO PREVENT A STATE MORE THAN THE GIRL WANTED TO PRESERVE THE LIFE STATE OF THE GIRL?

This question is answered when AQUA reads this story further:

Answering question: DID THE GIRL WANT TO PREVENT A STATE MORE THAN THE GIRL WANTED TO PRESERVE THE LIFE STATE OF THE GIRL?

with: THE GIRL WANTED TO PROTECT THE FAMILY OF THE GIRL MORE THAN THE GIRL WANTED TO PRESERVE THE LIFE STATE OF THE GIRL.

Elaborating `XP-BLACKMAIL-SUICIDE-BOMBING`:

Answering question: DID THE ACTOR WANT TO PREVENT A STATE MORE THAN THE ACTOR WANTED TO PRESERVE THE LIFE STATE OF THE ACTOR?

with: THE ACTOR WANTED TO PROTECT THE FAMILY OF THE ACTOR MORE THEN THE ACTOR WANTED TO PRESERVE THE LIFE STATE OF THE ACTOR.

New questions for XP-BLACKMAIL-SUICIDE-BOMBING:

WHY DID THE ACTOR WANT TO PROTECT THE FAMILY OF THE ACTOR
MORE THAN THE ACTOR WANTED TO PRESERVE THE LIFE STATE OF
THE ACTOR?

AQUA builds a new node to explain the node that was previously under question. The question is replaced by the new node representing the *protect-family* goal. This node becomes one of the XP-ASSERTED-NODES of *xp-blackmail-suicide-bombing*. Since no explanations are known for the newly added node, this in turn becomes a new question about the elaborated XP. The question is seeking a reason for the unusual *goal-ordering* of the actor, in which *protect-family* is given a higher priority than *p-life*.

When the elaborated XP is applied to a new suicide bombing story, the new node will now be one of the premises of the hypothesis, causing AQUA to ask whether the actor was trying to protect his family. This reflects a deeper understanding of this particular scenario and is shown in figure 6.4. The new question (not shown in the figure) will also be instantiated, causing AQUA to look for an explanation for the unusual *goal-ordering*. Should new questions be raised and then answered during future stories, AQUA will again be able to elaborate this XP in a similar manner. Thus AQUA evolves a better understanding of the "blackmailed into suicide bombing" XP or "case" through a process of question asking and answering.

6.5 Conclusion: Learning from explanation questions

We have shown how AQUA learns new indices for XPs that it already has, and how it elaborates these XPs to achieve a better understanding of the situations they represent. This type of learning corresponds to the incremental modification of AQUA's explanatory knowledge by the answering of explanation questions. In general, an understander would also use explanation-based generalization techniques to learn new XPs through generalization of causal features from a novel story [DeJong, 1983; DeJong and Mooney, 1986; Mitchell *et al.*, 1986], but this has not been implemented in AQUA.

In order to learn a particular type of knowledge such as an XP or an index, this knowledge must be explicitly represented in the program. A program that represents inference rules or schemas in memory, and builds explanations via backchaining through these rules or schemas, has an opportunity to learn new rules or schemas whenever this backchaining results in a novel combination of structures. However, it cannot learn applicability conditions for these rules unless this kind of knowledge is explicitly represented and available for reasoning. AQUA builds explanations, not by backchaining through rules, but by retrieving larger structure (XPs) from memory and applying them to the current situation. Since the indices for these structures are represented in the program, AQUA can learn new indices for XPs.

To summarize, AQUA can:

- Learn new contexts for stereotypical XPs that it already has. For example, AQUA learned a new *situation index*, suicide bombing, for the blackmail XP.

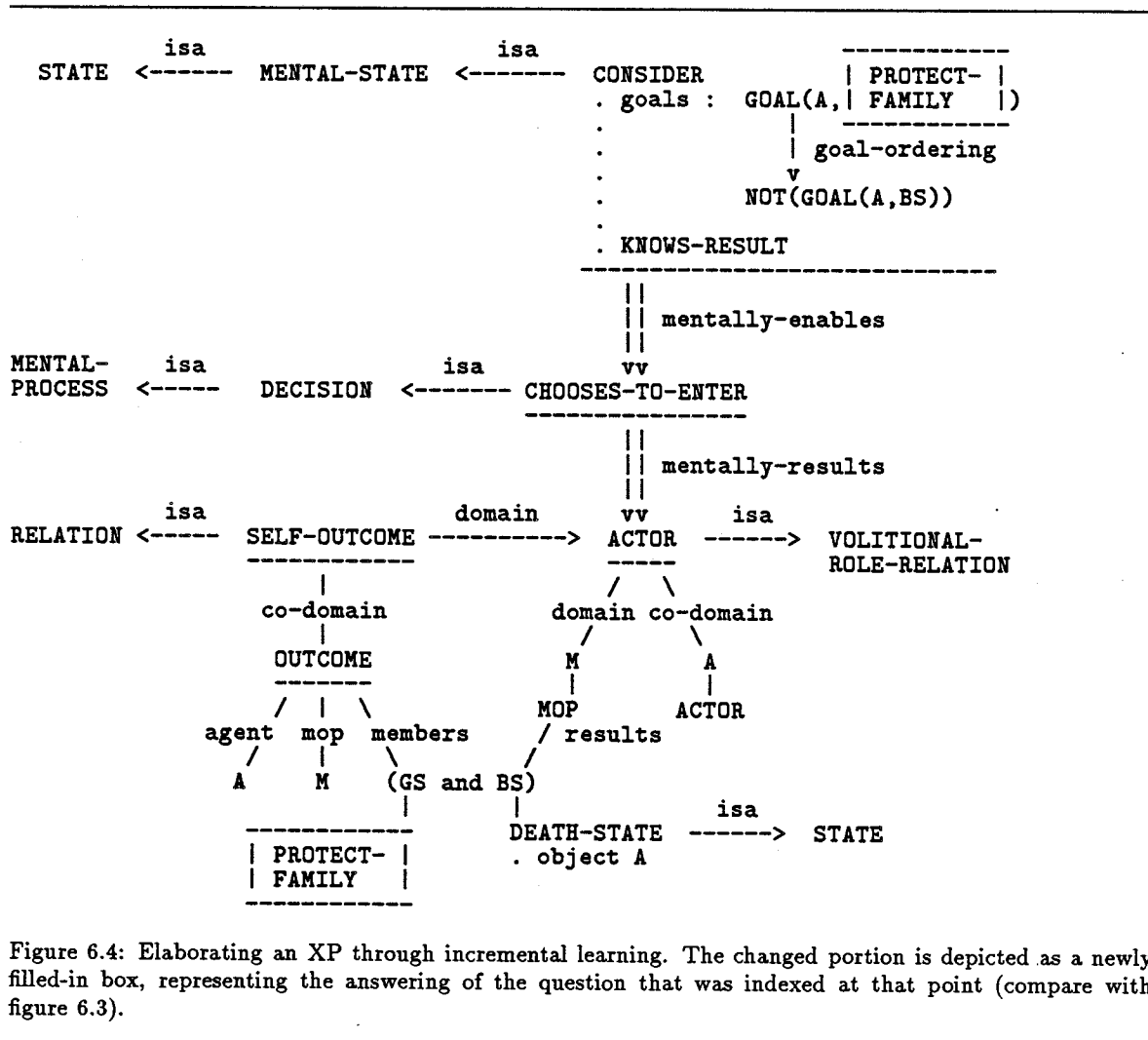


Figure 6.4: Elaborating an XP through incremental learning. The changed portion is depicted as a newly filled-in box, representing the answering of the question that was indexed at that point (compare with figure 6.3).

- Learn new character stereotypes to use as *stereotype indices* for stereotypical XPs that it already has. For example, AQUA learned a new stereotype representing a typical Lebanese teenager who might be blackmailed into suicide bombing, which was used to index the blackmail XP.
- Associate new questions to XPs. Asking the right questions is as important to understanding as is answering them. Learning the questions to ask when the XP is next applied is a central issue since this allows AQUA to reason about what it does not yet know but wants to find out. For example, though AQUA learns a new XP, blackmail, for suicide bombing, the XP is not perfectly understood in this context. As discussed above, there are questions attached to the XP, such as *What could the bomber want more than his own life?* AQUA focuses on this question while reading suicide bombing stories in the future in order to learn about this aspect of suicide bombing.
- Incrementally refine existing XPs by answering questions associated with them. Having learned the questions to ask in future stories, AQUA fills in the gaps in its XPs that these questions represent and improve its understanding of these XPs.

6.6 Extensions

In this section, we will briefly discuss how AQUA might be interfaced to other programs to perform the following kinds of learning:

- (In conjunction with an explanation-learning program) Learning new XPs.
- (In conjunction with a case-adaptation program) Case-based learning.

6.6.1 Extension: Learning new XPs

Although AQUA focuses on incremental learning, it would be relatively straightforward to interface AQUA to an explanation-based generalization program that learned new XPs (e.g., GENESIS [Mooney and DeJong, 1985]). Learning a new stereotypical XP (such as `xp-religious-fanatic`) of a given abstract type (such as `xp-sacrifice`) involves the following steps:

- Build a new XP structure using explanation-based generalization.
- Make the new XP be `isa` the abstract type of the XP.
- Index the new XP in memory under the following indices:

Category index: the abstract type of the XP.

Situation index: the type of action or mop being explained.

Stereotype index: a new stereotype built by generalizing the features of the volitional-agent involved in the action or mop.

- Check each of the INTERNAL-XP-NODES and LINKS of the XP to see it was confirmed. For each that is not confirmed, generate a question and attach it to the new XP. These questions represent what the understander needs to learn about this XP when it is used in future situations.

Once a new XP is learned and indexed in memory, it gives rise to XP retrieval questions as well as XP application questions during future stories. If the XP is found to be applicable, a hypothesis is built and hypothesis verification questions are generated as usual.

As mentioned earlier, AQUA does not perform the first step of this process, in which a new XP is built through explanation-based generalization. However, it can take a new XP that might be created by an explanation-based generalization program and incorporate it into memory by performing the remaining steps in the above process. These steps are the same as those used to learn new indices for XPs that it already has, and to associate new questions to such XPs, which were discussed earlier.

6.6.2 Extension: Learning by case adaptation

AQUA's strategies for answering questions include simple memory search, inference, and reading stories. However, the modular design of AQUA makes it easy to interface it with other reasoning programs. For example, AQUA could use a case-based reasoning module to infer answers to questions by adapting answers from similar past situations that it knows about.

The TWEAKER program tries to construct novel explanations by adapting hypotheses from previous explanations [Kass, in preparation]. AQUA could take advantage of this program by learning new XPs from cases that may not immediately be applicable to the current situation. This allows it to perform case-based learning through the adaptation of past cases.

AQUA interacts with TWEAKER in the following ways.⁶

6.6.2.1 XP retrieval

During the XP retrieval step, AQUA tries to retrieve stereotypical XPs to explain an anomaly that it has detected. If no such XPs are found, AQUA tries to apply the abstract XPs that it has. However, similar XPs might exist in memory for similar anomalies in different domains. AQUA uses TWEAKER as an "analogy module" or "case-based reasoning module" to adapt these XPs into an XP that would apply in the current situation.

For example, suppose AQUA has not read about religious fanatics before, and it does not have a "religious fanatic" XP in memory. However, suppose it does know about Kamikazes, and has an XP, *xp-kamikaze*, in memory which states that the Kamikazes performed suicide bombing missions to protect their country since their love for their country (*patriotic-zeal*) was more important to them than their *p-life* goal. (A detailed representation of *xp-kamikaze* is shown in chapter 8.)

⁶The transcripts illustrate how a case adaptation program can be used as a module in AQUA. How the adaptation actually gets done is discussed in [Kass, in preparation] and is beyond the scope of this dissertation.

Given its knowledge of *xp-kamikaze*, AQUA uses TWEAKER to adapt *xp-kamikaze* to the terrorist bombing situation by replacing "patriotic zeal" by "religious zeal" and use this to explain the otherwise similar suicide bombing.⁷

Anomaly! The SUICIDE BOMBING violates THE GOAL OF THE BOY TO PRESERVE
THE LIFE STATE OF THE BOY.

Searching for stereotypical XPs
No XPs found

Calling TWEAKER

Input question: WHY DID THE TEENAGE LEBANESE BOY DO THE SUICIDE
BOMBING MISSION?

In memory somewhere: WHY DID THE KAMIKAZE PILOT DO THE SUICIDE
BOMBING MISSION?

Answer: An instance of XP-KAMIKAZE

Adapting XP-KAMIKAZE to THE SUICIDE BOMBING
Replacing PATRIOTIC ZEAL with RELIGIOUS ZEAL

Output from TWEAKER: A new XP XP-RELIGIOUS-FANATIC

Applying XP-RELIGIOUS-FANATIC to THE BOY DID THE SHOOTING.

...

The question that the old explanation answered is analogous to the question that the new explanation is going to answer, since both are of the form "Why did X do an action that resulted in the death of X?" These questions can be used to construct the mapping between the two domains.

6.6.2.2 XP application

After an XP has been found, AQUA matches the PRE-XP-NODES and INTERNAL-XP-NODES of the XP to the situation to see if the XP is applicable. If the match returns any out nodes, the XP is ordinarily discarded. However, AQUA can use TWEAKER as an "adaptation module" to try to make the XP applicable to the situation by adapting the out nodes.

For example, suppose that AQUA has an overly specific "religious fanatic" XP in memory that is applicable only to suicide bombings. This XP would ordinarily not be applicable to the following story:

S-26: A high-ranking general in the Israeli army was injured today as a teenage Lebanese boy attempted to kill him. The boy was shot by the general's guards.

However, AQUA can understand this story with the help of the adaptation module:

⁷In the following transcripts, *slanted type* represents the output of the case-based reasoning and adaptation module. *Typewriter font* represents AQUA's output, as before.

Applying XP-RELIGIOUS-FANATIC to THE BOY SHOT THE GENERAL.

Unifying EXPLAINS node

Was there a BOMBING?
No

Is the ACTOR known?
Yes (THE BOY)

DID THE ACTION RESULT IN THE DEATH STATE OF THE ACTOR?
Yes (THE DEATH STATE OF THE BOY)

DID THE ACTION RESULT IN THE DESTROYED STATE OF AN ISRAELI
PHYSICAL OBJECT?
Yes (THE GENERAL)

XP-RELIGIOUS-FANATIC is not applicable to THE BOY SHOOTING THE GENERAL.

Calling TWEAKER

Adapting XP-RELIGIOUS-FANATIC to THE SHOOTING
Replacing SUICIDE BOMBING with any DESTROY action

Applying XP-RELIGIOUS-FANATIC to THE BOY SHOT THE GENERAL.

...

6.6.2.3 Inferencing for HVQ verification

When an applicable XP is found, it is instantiated to form a hypothesis. The XP-ASSERTED-NODES, which are the premises of the hypothesis, are turned into hypothesis verification questions. This hypothesis must then be verified by finding answers to its HVQs. AQUA uses TWEAKER as an inferencer to help determine the truth of the HVQ nodes by adapting them into something else that is already known or perhaps easier to verify.

To continue the Kamikaze example, the new *xp-religious-fanatic* that is built by adapting *xp-kamikaze* does not specify the religion that the actor was zealous about, since it is not possible to determine this by examining the Kamikaze case alone. Suppose AQUA tries to apply this new XP to the shooting story S-26. The HVQs that are generated are:

Applying XP-RELIGIOUS-FANATIC to THE BOY SHOT THE GENERAL.

Installing HVQs to verify XP:

WHAT IS THE RELIGION OF THE TEENAGE LEBANESE BOY?
Not known

WHAT IS THE RELIGIOUS ZEAL OF THE TEENAGE LEBANESE BOY?
Not known

In order to infer whether the boy was fanatic about his religion, AQUA needs to transform the question into something more specific. This is what TWEAKER is designed to do.

Calling TWEAKER

*Input questions: WHAT IS THE RELIGION OF THE TEENAGE LEBANESE BOY?
and WHAT IS THE RELIGIOUS ZEAL OF THE TEENAGE
LEBANESE BOY?*

*Inferring RELIGION from NATIONALITY
Inferring RELIGIOUS-ZEAL from RELIGION*

*Output: THE BOY IS A SHIITE MOSLEM
and THE BOY IS VERY ZEALOUS ABOUT THE SHIITE MOSLEM
RELIGION.*

Confirming hypothesis:

THE BOY SHOT THE GENERAL BECAUSE THE BOY WAS A RELIGIOUS FANATIC.

...

6.6.2.4 Subjective interpretation

Subjective interpretation, as described earlier, means that the interpretation of a story is influenced by the hypotheses already in the understander's memory. When building multiple hypotheses to explain an anomaly, AQUA jumps to conclusions if one of these hypotheses matches an earlier hypothesis in memory (see section 5.1.2).

However, a new hypothesis might not directly match an old hypothesis, but yet might be similar enough to warrant making a conclusion. For example, after reading the blackmail story, AQUA learns that suicide bombers are often forced into performing these missions rather than volunteering for them. This is a new hypothesis in memory. As illustrated in section 5.1.2, this leads AQUA to prefer an explanation based on coercion if it can find one.

However, in a story where coercion is possible but not explicitly stated, AQUA could use TWEAKER to adapt the new hypothesis so that it confirms the old hypothesis that it already believes. This is similar to the "interest-based specialization" process discussed in section 5.1.2, which uses matching and specialization instead of adaptation to achieve this result.

6.7 Conclusion: Memory and understanding

In section 5.4.1.3, we introduced the *generalize* step of AQUA:

Generalize novel answers to its questions.

Index these answers in memory, so that the task that originally generated the question would now find the information instead of failing.

All novel stories provide an opportunity to learn. Noticing similarities and forming generalizations are an integral part of a dynamic memory. As discussed in section 4.1.3, reminders give rise to questions that seek generalizations. Such questions are called analogy or generalization questions.

These generalizations could be based either on superficial similarities or on deeper explanatory or causal similarities. For example, the IPP program used an integrated model of understanding and generalization based on surface similarities [Lebowitz, 1980]. IPP changed by learning new generalizations in memory. To restate this in terms of the taxonomy of questions presented in chapter 4, IPP learned by answering its memory-level questions.

However, the understander might learn from other types of questions as well. For example, various syntactic learning techniques could be used to learn from text-level questions. AQUA does not perform any text-level learning.

An understander could also learn from answers to explanation questions. This is the issue that AQUA focuses on. The advantage of using explanations to guide generalization is that it avoids the familiar problem of feature selection [Schank *et al.*, 1986; DeJong, 1983].

IPP was one of the first programs to shift its focus from parsing to memory. Even though it was a story understanding program, the real emphasis was on the interaction of memory with the parsing process, and the evolution of memory as the program read more stories. AQUA takes this view further by integrating the explanation process into the theory of memory-based parsing. The emphasis is still on changing memory. However, unlike IPP whose memory changed through feature correlation, AQUA's memory changes through the process of incremental learning of explanations and their indices.

Chapter 7

Theory of interestingness

Combinatorial explosion of inferences has always been one of the classic problems in AI. Resources are limited, and inferences potentially infinite; an understander needs to be able to determine which inferences are useful to draw from a given piece of text. But unless one considers the goals of the understander, it is very difficult to give a principled definition of what it means for an inference to be "useful." Since questions represent the understander's knowledge goals, the understander should draw those inferences from the text that will help it answer its questions.

A program that uses questions to guide understanding is an improvement over one that processes everything in equal detail, i.e., one that is completely text-driven. An understander that is completely text-driven would process everything in detail in the hope that it might turn out to be relevant. To avoid this, the understander should draw only those inferences which would help it find out what it needs to know.

In other words, questions can be thought of as representing what the understander is *interested* in finding out at any point for the purpose of understanding a story. The understander uses its questions to focus its attention on the "interesting" aspects of the story, where "interesting" can be defined as "relating to something the understander wants to find out about."

In this dissertation, we have shown how an understander can use its questions to focus inferences for understanding tasks. To improve on this even further, the understander needs a way of determining which questions are worth pursuing and which ones are not. Not all questions are equally important, nor are all answers equally valuable. The understander needs to be able to determine the priorities of its questions, based on how useful the questions are to achieving its overall goals.

Why would an understander need to find something out in the first place? Ultimately, the point of reading is to learn more about the world. Questions arise when reading a story reveals gaps or inconsistencies in the world model. It is useful to focus attention on such questions because they arise from a "need to learn." For example, questions arising from anomalous facts are more useful than those arising from routine stereotypical facts, since in the former case the understander may learn something new about the world.

Thus, in addition to using questions to control inferences, the understander also needs to be able to determine the priorities of the questions themselves, depending on how likely the understander is

to learn something by thinking about these questions. Facts that raise questions of greater priority should be processed in greater detail.

This gives us two basic ways in which a fact can turn out to be worth processing:

1. **Top-down:** A fact that answers a question is worth focussing on since it helps to achieve a knowledge goal of the understander, which in turn allows the understander to continue the reasoning task that was awaiting the answer.
2. **Bottom-up:** A fact that raises new questions is worth focussing if the questions arise from a gap or inconsistency in the understander's knowledge base, since the understander may be able to improve its knowledge base by learning something new about the world.

These correspond to the two diamonds in figure 8.1 on page 145. These diamonds attempt to determine which facts the understander should focus on. This decision is made using a set of heuristics that will be described below.

The decision to focus attention corresponds closely with the notion of "interestingness." When an understander focuses on a particular fact and processes it in greater detail, it can be said to be "interested" in that fact. AQUA's focus of attention heuristics cause it to focus on those facts that correspond to aspects of the story that people would consider interesting. For example, while reading the blackmail story, AQUA focuses its attention on the act of suicide bombing mission as a whole (rather than, say, the act of driving the car), the religion of the boy (rather than the color of his hair), and the goal of the boy to preserve his own life (rather than other goals that he might have). These are precisely the interesting aspects of this story from a human interest point of view.¹

For this reason, AQUA's focus of attention heuristics can also be thought of as *interestingness heuristics*. These heuristics provide a functional definition of "interestingness" as a criterion for focussing attention: *Interestingness is a guess at what one thinks one might learn from paying attention to a fact or a question.* The guess must be made without processing the fact or question in detail, because otherwise the purpose of focussing attention to control inferences would be defeated. Thus the interestingness heuristics described below are indeed heuristics rather than precise measures of the value of thinking about a fact or a question.

7.1 Interestingness heuristics

In order to use questions to control inferences, an understander must be able to determine the interestingness of questions based on their content, as well as the task that they arose from. It must also be able to identify facts in the story that are interesting by virtue of being relevant to questions that the understander is interested in.

There are two types of heuristics for determining interestingness:²

¹Since interestingness depends on one's goals, the theory of interestingness presented here does not cover interests that arise from goals that lie outside the scope of the basic understanding and learning tasks that AQUA performs. For example, a parent would be interested in the report card of his child. Since AQUA's goals do not include caring for children, it would not have any reason to be interested in a report card, unless the report card was anomalous with respect to AQUA's beliefs.

²This is orthogonal to the top-down/bottom-up distinction made earlier.

1. **Content-based:** The interestingness of some input depends on its content or domain (more specifically, on the relationship between its content and the system's goals). In other words, some things are more interesting than others, depending on their relationship to the system's goals. For example, if one is intending to fly KLM in the near future, a story about a KLM flight being hijacked would be very interesting even if it were a stereotypical hijacking story. The understander would try to draw those inferences that were relevant to its goal of flying KLM. Similarly, stories about people one personally knows are more interesting than stories about strangers.

These heuristics use the content of the fact or question to determine its interestingness. The issue here is, which particular facts should the understander focus on? Which particular facts does the understander need to learn about?

2. **Structure- or Configuration-based:** Some kinds of situations are more interesting than others. For example, expectation failures are interesting, regardless of the content of the particular expectation that failed. These heuristics use the structure of the knowledge to determine interestingness. The issue here is, can the structure of the situation be used to determine which aspects of the situation are worth focussing on? Which configurations of knowledge structures signal gaps that the understander needs to learn about?

Of course, in any particular situation, the two conditions need to be combined in order to determine the overall interestingness of the input. For example, an expectation failure that relates to a goal of the system would have a higher priority than one that does not, and so the system would be more interested in the former.

Both types of heuristics identify situations in which the understander might be able to learn something useful. In the first case, the understander might learn more about a person it knows, or it might learn a new way to achieve a goal that it has. In the second case, the understander might be able to update its world model by identifying gaps in its model.

An example of the first type of heuristic is the principle of goal identification used by the POLITICS program [Carbonell, 1979]:

If the understander of an event identifies with the goals of one of the actors, he will focus attention on inferences that lead to the fulfillment of these goals.

This is a content-based heuristic since it relies on the actual type of goal, not merely on the fact that there is a goal being pursued. A configuration-based heuristic, on the other hand, relies on particular relationships between concepts in memory, not specifically on what those concepts are. For example, POLITICS used the following configuration-based heuristic to focus its attention:

Objective/Means distinction: If there are two or more actions in an event, and some actions are instrumental to stated or implicit objectives of one of the actors, the understander should focus attention on the objectives and non-instrumental actions.

In other words, an instrumental action is less likely to be significant than a larger action that it is part of, regardless of what the particular actions are. The reason this is a good strategy for an understander, according to the definition of interestingness proposed earlier, is that the understander

is more likely to learn something by thinking about the larger action than it is by thinking about the instrumental action.

For example, consider a story about a car bombing mission. The act of driving the car, which is instrumental to the act of suicide bombing, is less interesting by the above criterion. Thus the understander should focus its attention on the suicide bombing rather than worry about details of the driving. The reason this is so is that the understander is more likely to learn something useful from thinking about the suicide bombing as a whole.

Note that this and any other heuristic is fallible, since it is easy to construct a situation in which it would be better to focus attention on, say, the driving because of some other criterion. Thus the recommendations of various interestingness heuristics would need to be combined before arriving at a final measure to be used to focus attention.

AQUA has several heuristics to judge interestingness. These heuristics can be categorized according to the type of understanding goals that they pertain to. Let us start with relevance goals.

7.1.1 Interestingness from relevance goals

Interestingness arising from personal relevance usually falls into the class of content-based interestingness, since particular goals, people, locations, etc. are identified as being interesting to the understander. Questions and facts involving these goals, people, locations, etc. are worth pursuing since the understander might learn something relevant to it by doing so.

7.1.1.1 What could be relevant to a program?

In order for something to be personally relevant to a program, the program must have a personality in the real world. There must be goals it wants to achieve, people it knows or has heard of, places it knows about or has grown up in, and so on. Stories relevant to this personality are interesting even if they do not involve anomalies or novel explanations.

For example, two of the focus of attention criteria used by POLITICS fall into this category: "goal identification" and "interest in VIP activities." Since POLITICS had a political ideology, it could be said to have a personality in the sense used here. Thus it could focus its attention on those aspects of a situation that were relevant to its goals.

Since AQUA does not have any real experiences outside of reading stories, its "personality" consists of its knowledge goals, i.e., the questions that it is interested in finding answers to. The people, institutions, objects and locations that it is interested in learning about are the people, institutions, objects and locations that are involved in these questions. The goals that are interesting to AQUA are goals of characters in these stories that it has questions about.

For example, consider the blackmail story:

S-2: Boy Says Lebanese Recruited Him as Car Bomber.

JERUSALEM, April 13 — A 16-year-old Lebanese was captured by Israeli troops hours before he was supposed to get into an explosive-laden car and go on a suicide bombing mission to blow up the Israeli Army headquarters in Lebanon. ...

The teenager, Mohammed Mahmoud Burro, ... was not a fanatic who wanted to kill himself in the cause of Islam or anti-Zionism, but was recruited [by the Islamic Jihad] through another means: blackmail.

Suppose AQUA has read several religious fanatic stories, but has not encountered any coercion stories so far. After reading the blackmail story, as described in earlier chapters, AQUA is left with several questions in memory, including:

- What did Mohammed value more than his own life?
- Why did the Islamic Jihad plan this mission?
- Why did the Islamic Jihad choose a teenager for the mission?

On the basis of these questions, AQUA will now be interested in stories involving the people or institutions it has questions about, such as:

- Another story about Mohammed Burro (relevance to known person)
- Another story about the Islamic Jihad (relevance to known institution)

For example, after reading the blackmail story, AQUA's representation for the Islamic Jihad has a question attached to it:

```
(define-instance ISLAMIC-JIHAD.0
  (isa      (set: {terrorist-group}))
  (nationality [lebanese.29])
  (questions (set: {xp.16 xp.21}))
  (activities (set: {[actor.15
                    (domain [recruit.15])
                    (explanations (set: {xp.16}))])
                [planner.271
                    (domain [car-bombing.20])
                    (explanations (set: {xp.21}))])
                ...}))
...)
```

When AQUA reads another story about the Islamic Jihad, it checks the representation of the `islamic-jihad.0` frame to see if it has any pending questions about this agent. It finds the two pending questions mentioned above: *Why did the Islamic Jihad plan this mission?* and *Why did the Islamic Jihad choose a teenager for the mission?* Thus the new story is interesting by the actor relevance heuristic (H-4, to be discussed below), since it is possible that the story will answer these questions.

AQUA will also be interested in stories involving the newly learned blackmail explanation, such as:

- Another story about someone being blackmailed into a suicide bombing mission (relevance to novel explanation)

AQUA will also be interested in a story involving goals or goal priorities that it has questions about, such as:

- Another story about someone valuing something over their own life (relevance to goal)

For example, after reading the blackmail story, AQUA is interested in the following story:

S-25: JERUSALEM — A young girl drove an explosive-laden car into a group of Israeli guards in Lebanon. The suicide attack killed three guards and wounded two others. ...

The driver was identified as a 16-year-old Lebanese girl. ... Before the attack, she said that a terrorist organization had threatened to harm her family unless she carried out the bombing mission for them. She said that she was prepared to die in order to protect her family.

Again, since S-25 addresses pending questions in memory, it is worth thinking about this story since it might provide an answer to these questions. In this case, as discussed in chapter 6, the story allows AQUA to learn more about the novel blackmail explanation, by answering the questions about the goals and goal priorities of the actor that were pending in memory.

One could say that AQUA has “adopted” the goals that it has questions about, in the sense that it is interested in stories³ about such goals as if they were its own. An alternative approach to the problem of where a program might get its goals is to “give” goals to the program, for example, by programming particular ideologies into the program as in POLITICS. In AQUA, this would be analogous to tagging particular goals or people as being “personally relevant” or “interesting” to the program.

In contrast, the approach used in AQUA is to allow the program to evolve its own set of interests that are functional to the purpose of the program (learning about terrorism), by letting the questions that arise from this purpose *be* its goals. These interests can be used to focus attention on those aspects of the story that would help it achieve its purpose.

Let us now discuss interestingness heuristics based on this notion of personal relevance.

7.1.1.2 Goal relevance

This is similar to the “goal identification” criterion used by POLITICS.

H-1: Goal relevance

A fact that could be instrumental to or could hinder a goal of the understander is more interesting than one that has no relevance to the understander’s goals. A fact that directly matches or conflicts with a goal of the understander is very interesting.

For example, if one is planning a trip on Lufthansa to visit Germany, both the following stories would be interesting:

S-27: Lufthansa announced cheap rates to Frankfurt today.

S-28: A Lufthansa flight was hijacked by terrorists last night. The plane was diverted to Athens.

³Or aspects of stories

As discussed earlier, AQUA's interest in particular goals arises from goals that it encounters in stories that it still has questions about. For example, suppose AQUA is presented with the following stories after it has read the blackmail story:

S-29: Lebanon car bomb kills driver, hurts 7 at Palestinian site.

S-30: Lebanese civilians protest Israeli occupation.

AQUA can determine that S-29 is more interesting than S-30 since S-29 is about a goal that AQUA is interested in, i.e., has questions about, after reading the blackmail story. This assumes that S-30 isn't interesting by some other criterion, say, by virtue of being anomalous. To demonstrate this, anomaly detection and other interestingness heuristics are turned off in the following trace.

Processing: LEBANESE CIVILIANS PROTEST ISRAELI OCCUPATION.

...

Trying to explain WHY DID THE LEBANESE CIVILIANS PROTEST ISRAELI
OCCUPATION?

Not doing anomaly detection

No pending questions triggered; not processing further

Now consider story S-29:

Processing: LEBANON CAR BOMB KILLS DRIVER, HURTS SEVEN AT
PALESTINIAN SITE.

...

Trying to explain WHY DID THE DRIVER DO THE SUICIDE BOMBING?

Not doing anomaly detection

Matches question in memory!

WHY DID THE TEENAGE LEBANESE BOY DO THE SUICIDE BOMBING?

Searching for XPs

...

As shown, AQUA continues to process S-29 by trying to answer the question that it found interesting, which involves building an explanation for the suicide bombing. In contrast, AQUA determines that S-30 is not interesting using the only heuristic it has available in this case, and does not process it any further. If anomaly detection was not disabled, AQUA would process S-30 to the extent required to determine that the story was not anomalous, say, until a stereotypical explanation was found.

7.1.1.3 Vicarious goals

Vicarious interests arise from goal configurations that are similar to one's own, or to those one is likely to have at some point. Stories about crisis goals or sudden goal changes, for example, are usually interesting from a vicarious point of view since the understander is likely to experience similar crisis goals or goal changes.

These heuristics are configuration-based, as opposed to content-based, since all goal changes are inherently interesting regardless of the particular goals involved. However, goal changes involving goals of personal relevance to the understander would be more interesting than those involving other goals.

H-2: Vicarious goal change

An action that drastically changes the goals of the planner or actor of the action is interesting.

H-3: Vicarious crisis goal

An action that initiates a crisis goal of the planner or actor of the action is interesting.

An example of a vicarious crisis goal is the self-preservation goal arising from acts of violence. Schank calls violence an "absolute interest" [Schank, 1979] since people universally seem to be interested in violence. This follows from the application of heuristic H-3 to the thematic goal of self-preservation which is universal among people.

7.1.1.4 Actor relevance

There are four degrees to which a particular person or institution can be relevant to an understander, each of which produces more interest than the previous one:

H-4: Actor relevance

Stories involving people that are completely unknown are the least interesting. More interesting than these are stories involving people who the understander has heard of. Still more interesting are stories involving people who are famous, such as celebrities. Stories involving people who the understander personally knows are the most interesting.

This heuristic is similar to the "actor relevance" and "interest in VIP activities" criteria of POLITICS. For example, compare the following stories:

S-31: Drew McDermott started up a new company.

S-32: Paul Newman started up a new company.

S-33: James Watson started up a new company.

S-34: John Doe started up a new company.

By the above heuristic, S-31 would be most interesting from my point of view, followed by S-32, S-33 and S-34 in that order.⁴

Since AQUA's knowledge of agents (people or institutions) is restricted to those agents that it has read about, AQUA can only distinguish between agents that it knows about and agents that it doesn't. Thus, as actually implemented the actor relevance heuristic looks more like this:

⁴Since interestingness is subjective, not everyone will agree with these measures. This and other examples should be read from the point of view of a hypothetical understander with the particular goals and interests that are assumed in the example.

H-4: Actor relevance

Stories involving people that are unknown are less interesting than stories involving people that the understander has read about earlier and still has questions about.

Thus questions involving people that are unknown to the understander would receive a lower priority, unless these questions happened to be interesting by some other criteria.

7.1.1.5 Object relevance

As in the case of actor relevance, there are four degrees to which an object can be relevant to an understander, each of which produces less interest than the previous one:

H-5: Object relevance

Stories involving objects with which the understander has personal experience are most interesting. Less interesting are stories involving objects that are famous. Still less interesting are stories involving objects that the understander has only heard of. Stories involving objects that are unknown to the understander are least interesting.

For example, compare the following earthquake stories:

S-35: My childhood apartment collapsed in an earthquake.

S-36: The Leaning Tower of Pisa collapsed in an earthquake.

S-37: The Jet Propulsion Lab collapsed in an earthquake.

S-38: One building was destroyed and two others damaged in an earthquake.

By the above heuristic, S-35 would be most interesting from my point of view, followed by S-36, S-37 and S-38 in that order.

Again, since AQUA's knowledge of objects is restricted to those objects that it has read about, it can only distinguish between objects that it knows about and objects that it doesn't. Thus the object relevance heuristic is actually implemented as follows:

H-5: Object relevance

Stories involving objects that are unknown are less interesting than stories involving objects that the understander has read about earlier and still has questions about.

AQUA mainly deals with volitional explanations, and so it seldom has pending questions about objects unless they play a causal role in a volitional explanation.

7.1.1.6 Location relevance

Similarly, interestingness based on the personal relevance of a place or location can be determined using the following heuristic:

H-6: Location relevance

Stories involving locations with which the understander has personal experience are most interesting. Less interesting are stories involving locations that are famous. Still less interesting are stories involving locations that the understander has only heard of. Stories involving locations that are unknown to the understander are least interesting.

For example, compare the following stories:

S-39: A suicide car bombing in Manhattan.

S-40: A suicide car bombing in Chile.

By the location relevance heuristic, S-39 is more interesting than S-40, unless the understander happens to be from Chile, or is interested in Chile for some other reason. As before, this heuristic is implemented in AQUA in the following form:

H-6: Location relevance

Stories involving locations that are unknown are less interesting than stories involving locations that the understander has read about earlier and still has questions about.

7.1.2 Interestingness from explanation goals

So far, we have seen interestingness heuristics that tried to identify situations involving goals, people, objects or places that are personally relevant to the understander. Such situations provide an opportunity to learn something of personal relevance, e.g., to learn something more about a person the understander is interested in, or to learn a new plan to achieve a goal that the understander has.

There is another class of heuristics which identify potential gaps in the understander's knowledge, in order to determine what the understander might learn from processing the given situation. Situations that are interesting according to these heuristics are those that allow the understander to improve its world model. These heuristics are based on explanation and memory goals of the understander.⁵ Let us start with explanation goals.

7.1.2.1 Anomaly detection: What needs to be explained

Anomalies arise when incoming facts do not fit in with what the understander expected to see. Anomalies are interesting because there is a possibility that the world model that underlies the failed expectations is incorrect. All the anomaly detection questions presented in section 4.1.1, if they lead to an anomaly, signal interesting aspects of the story.

H-7: Anomaly detection

All anomalies are interesting, and therefore all anomaly detection questions that lead to anomalies are interesting.⁶ These questions are always pursued in an attempt to form explanations to resolve the anomalies.

⁵Since AQUA does not perform any text-level learning, text-level questions are not interesting to AQUA according to the learning criterion for interestingness.

⁶As mentioned earlier, AQUA deals mainly with volitional anomalies.

Questions seeking XPs to explain anomalies are called XP retrieval questions. These questions are less interesting if XPs are easily available. If no XP is found, there is a gap in memory corresponding to an unexplainable anomaly. An XP retrieval question that seeks to fill in this gap is interesting for an understander that is trying to learn about this domain.

H-8: XP availability

An XP retrieval question that fails to find an XP in memory is more interesting than one where an XP can easily be found.

For example, consider the following stories:

S-41: A teenager went on a suicide bombing mission. Although he was a Shiite Moslem, he was not a fanatic.

S-42: A millionaire was found dead in his house this morning. His wife, who will inherit his fortune, has been detained for questioning.

From the point of view of XP availability, story S-41 is more interesting than story S-42. In the first story, the question "Why did the (non-fanatic) boy go on a suicide bombing mission?" does not have an answer. However, in the second story, the question "Why was the millionaire murdered?" has a standard answer that is readily available, namely, that the millionaire was murdered for the inheritance money.

Of course, this and any other heuristic might be overruled if other interestingness heuristics came into play. Heuristics like these encode rules that represent guesses at the interestingness of a question, *other things being equal*. Counter-examples can easily be found by playing up interestingness factors represented by other heuristics. The final interestingness measure is a combination of all these factors, not any single factor taken by itself. For example, a stereotypical explanation for the motivations of a person who one knows personally might be more interesting than an unusual explanation for the motivations of a person one doesn't know. This also depends on the goals of the understander, since an understander might be particularly interested in the motivations of a particular person for some other reason.

The interestingness of an explanation question also depends on the kind of explanation that the question is seeking. Questions seeking explanations for the motivations of individuals arise from *human interest stories*, and are usually more interesting, from the human interest point of view, than questions that seek explanations for the motivations of institutions.

H-9: Human interest

Questions involving motivations of individuals are more interesting than those involving motivations of institutions.

For example, consider the following story (Los Angeles Times, April 10, 1985):

S-3: 4 Die in Suicide Attack Against Israelis.

JERUSALEM — A young guerrilla driving a car filled with explosives blew it up in a suicide attack Tuesday against a group of Israeli guards in Lebanon, killing three and wounding two others, an Israeli military source confirmed.

Besides the driver, identified by the guerrilla group as a 16-year-old girl, the blast killed two Israeli soldiers and a Druze member of the Israeli-backed South Lebanon Army. The wounded were Israeli soldiers, the source said. ...

A spokesman for the group identified the driver as Sana Mohaydaleh. In an interview taped before the incident and shown Tuesday after the attack on Beirut television, a girl identified as Mohaydaleh said: "I hope that my soul will unite with the souls of all the martyrs before me. I decided on martyrdom to free our land because I saw the misery of my countrymen under the occupation. I hope I will be successful and able to kill the highest number possible of our enemies."

This story is interesting because it focuses on the motivations of the girl. The question "Why did the girl do the suicide bombing?" is more interesting than "Why did the guerrilla group plan the suicide bombing?" In this particular story, the reporter has anticipated this and focussed the story accordingly.

Note that heuristic H-9 judges interestingness from a human interest point of view. Human interest stories focus on the goals, motivations and emotions of particular individuals. From the point of view of politics or counter-planning, however, the motivations of the institution might be more interesting than those of an individual, unless the individual was an important political figure in that institution.

When searching for explanations for actions, instrumental actions are less interesting than the actions that they are instrumental to. This is similar to the objective/means criterion of POLITICS that was described earlier.

H-10: Instrumentality

If an action is instrumental to or part of another action, the former action is less interesting than the latter.

H-11: Instrumentality

If an action has more than one action instrumental to it, or a MOP has more than one scene, the most interesting of the instrumental actions is that which is the goal scene of the MOP (or the "maincons" of a script).

When AQUA encounters instrumental actions, it uses these heuristics to focus its attention on the most interesting action. Other actions are explained away by building a simple explanation that states that they were performed in service of the larger action. For example, in story S-3 above, AQUA concentrates on explaining why the girl did the suicide bombing mission (the main action). The explanation for why the girl drove the car (the instrumental action) is simply "Because it enabled the bombing."

Trying to explain WHY DID THE YOUNG GIRL DRIVE THE CAR?

THE DRIVE is instrumental to THE SUICIDE BOMBING.

Assumed: THE YOUNG GIRL DROVE THE CAR TO ENABLE THE SUICIDE BOMBING.

Trying to explain WHY DID THE YOUNG GIRL DO THE SUICIDE BOMBING?

...

The question *Why did the young girl drive the car?* has a low priority because of heuristic H-10 and is therefore not pursued in detail. If this heuristic was removed, AQUA would spend a lot more time exploring this action:

Trying to explain WHY DID THE YOUNG GIRL DRIVE THE CAR?

Not using instrumentality heuristic

Did THE YOUNG GIRL want the outcome of THE DRIVE?

Characterizing the outcome AT-LOCATION
of THE DRIVE
from the point of view of THE YOUNG GIRL (the ACTOR)

DID THE YOUNG GIRL WANT TO ACHIEVE BEING AT THE LOCATION?

Did THE YOUNG GIRL enjoy doing DRIVES?

Did THE YOUNG GIRL habitually do DRIVES?

Searching for stereotypical XPs

Asking EQ: WHY WOULD A PERSON DO A DRIVE?

Retrieved stereotypical XPs:

XP-DRIVE

Applying XP-DRIVE to WHY DID THE YOUNG GIRL DRIVE THE CAR?

THE YOUNG GIRL DROVE THE CAR
because THE YOUNG GIRL WANTED TO GO SOMEWHERE.

...

As demonstrated here, AQUA spends a lot of time drawing inferences to explain why the girl drove the car. It tries to determine whether driving a car is anomalous, whether the standard XP for driving applies, and so on. Instead, if the instrumentality heuristic is available, these questions have a lower priority than questions about the suicide bombing itself. AQUA spends a lot less time on questions about the driving as shown earlier, focussing its attention on the suicide bombing instead. Without the instrumentality heuristic, AQUA will ultimately process the suicide bombing too, but it would have wasted a lot of time on less interesting aspects of the story.

Similarly, habitual or commonly performed actions are uninteresting by the following heuristic:

H-12: Thematic or stereotypical action

If the actor, or a group that the actor belongs to, is known to perform such actions or select such plans in service of a known thematic goal, the action is not interesting.

Thus common plans for goals and routine thematic or occupation-related actions are uninteresting.

AQUA does not explain stereotypical actions in detail. The explanation it builds is simply "Because the actor often performs such actions in service of his goals," unless it has been unable to explain such actions in the past in which case it would have a pending question which would make this action interesting.

Trying to explain WHY DID THE TERRORIST GROUP PLAN THE SUICIDE BOMBING?

THE SUICIDE BOMBING is a TERRORIST ACT.

TERRORIST ACTs are often planned by THE TERRORIST GROUP.

Assumed: THE TERRORIST GROUP PLANNED THE SUICIDE BOMBING BECAUSE THE
TERRORIST GROUP PLANS TERRORIST ACTS.

As before, this allows AQUA to spend its time processing the more interesting aspects of the story. If these heuristics are absent, AQUA will still process the interesting aspects of the story, of course, but it will spend a lot more time processing uninteresting details as well.

7.1.2.2 XP application

When an XP is found, it is applied to the anomalous situation to try to explain the anomaly. If a stereotypical XP applies easily and directly, the story conforms to the XP that the understander already has, and is therefore not very interesting. On the other hand, if existing XPs do not apply to the situation, the story is novel and therefore interesting.

H-13: XP applicability

If an available XP applies easily and directly to the story, the story is not very interesting. More interesting is the case when an XP applies but leaves gaps which need to be filled in before the hypothesis is verified. The most interesting story is one in which available XPs do not fit the situation and need to be modified.

This heuristic follows from the claim that interestingness is a measure of what the understander might learn from processing the situation. Stories that identify gaps in the understander's memory are more interesting than those that fit into stereotypical molds that the understander already knows about, since the understander is unlikely to learn anything from processing the latter kind of stories.

For example, consider the following stories which provide specific explanations for a racehorse death:

S-43: A racehorse died of a drug overdose.

S-44: A racehorse was killed for the insurance money.

S-45: A retired racehorse died of old age.

The first story is the most interesting because the "drug overdose" XP is not directly applicable to the situation since horses cannot shoot themselves with drugs. Before a hypothesis can be built, the XP must be modified to state, for example, that the owner was giving it drugs to enhance its performance [Kass, in preparation]. In the second story, the "killed for the insurance money" XP applies without modification, but it must still be verified. In contrast, the third story is the least interesting since a stereotypical explanation is available, applicable, and leaves no gaps that need to be verified.

In AQUA, the process of applying XPs gives rise to XP applicability questions if the XP does not apply directly to the situation, and to hypothesis verification questions if the XP asserts facts

that are not yet known to be true. Thus heuristic H-13 is easily implemented by checking to see if the XP application step (section 4.1.1) gives rise to any questions. Since these questions, if any, are indexed in memory, they automatically guide future processing along with all other questions without any special mechanisms being needed for this particular type of question. This is illustrated in the next section.

7.1.2.3 Hypothesis verification

AQUA verifies its hypotheses by reading the story to try to find answers to hypothesis verification questions. These questions, once answered, help to confirm or refute hypotheses in the corresponding hypothesis tree (section 5.3.2). Facts in the story that help to verify or refute hypotheses are more interesting than facts that have no bearing on hypotheses currently in memory.

H-14: Hypothesis verification

An input fact is interesting if it helps to verify or refute a hypothesis, i.e., if it matches a hypothesis verification question in memory.

For example, consider the following stories:

S-46: A teenage boy blew up the U.S. embassy in Beirut in a suicide attack, killing himself and wounding several people.

S-47: John had wanted to date Mary for a long time, but Mary preferred people with red hair.

In the first story, the religious zeal of the boy is more interesting than the color of his hair, since it helps the understander confirm the religious fanatic hypothesis. In the second story, on the other hand, the color of John's hair is more interesting than his religious beliefs.

Thus the interestingness of a fact depends on the hypotheses and questions currently in memory. As described in section 5.2, this affects the inferences that would be drawn after reading the fact. For example, suppose story S-46 continued in one of two ways:

S-48: The boy was a Shiite Moslem and a religious fanatic.

S-49: The boy had red hair.

In the first case, AQUA builds a basic representation for the boy's religious beliefs, and then draws further inferences trying to verify the religious fanatic hypothesis. In the second case, however, AQUA builds a basic representation for the boy's hair, and continues reading the story without spending any further time on this fact. This happens automatically since sentence S-49 does not trigger any questions in memory, whereas S-48 matches a hypothesis verification question which, when answered, restarts the chain of reasoning that the suspended explanation task was pursuing.

For example, here is a transcript of AQUA reading S-49 and then S-48 after it has built the religious fanatic hypothesis.

Processing THE BOY HAD RED HAIR.

Instantiating concept: HAIR

Filling HAIR slot of THE BOY

No pending questions triggered; not processing further

Processing THE BOY WAS A SHIITE MOSLEM AND A RELIGIOUS FANATIC.

Instantiating concept: SHIITE MOSLEM RELIGION

Filling RELIGION slot of THE BOY

Answering question: WHAT IS THE RELIGION OF THE BOY?

with: THE BOY IS A SHIITE MOSLEM.

Instantiating concept: HIGH RELIGIOUS ZEAL

Filling RELIGIOUS-ZEAL slot of THE BOY

Answering question: WHAT IS THE RELIGIOUS ZEAL OF THE BOY?

with: THE BOY IS VERY ZEALOUS ABOUT THE SHIITE MOSLEM
RELIGION.

Restarting suspended task

Confirming hypothesis:

THE BOY DID THE SUICIDE BOMBING

because THE BOY WAS A RELIGIOUS FANATIC.

...

Thus AQUA spends more time processing the religion fact (in this case, drawing hypothesis verification inferences) than the color of hair fact. This is because the religion fact answers questions in memory, whereas the color of hair fact does not answer any questions and is therefore judged to be uninteresting.

7.1.3 Interestingness from memory goals

For a program such as AQUA, explanation goals are more interesting than memory or text-level goals. However, memory-level tasks also give rise to heuristics for interestingness which, as before, are based on trying to identify gaps in memory which give the understander an opportunity to learn.

The basic learning mechanism in a program such as IPP [Lebowitz, 1980] is that of similarity-based generalization, a process that builds categories in memory by noticing similarities between instances or sub-categories and building generalizations based on these similarities. Based on this theory of learning, one can write down the following heuristic for interestingness:

H-15: Uniqueness

A category with a unique example or specialization is more interesting than one with several examples. If two categories have unique examples, the category higher in the type hierarchy is more interesting than the one that is lower down.

For example, suppose the understander has a category for terrorist acts in Lebanon and has read several stories of this type, but it has never read a story about a car bombing story. A story about yet another terrorist act in Lebanon would be, by this heuristic, less interesting than the first story it reads about a car bombing in Lebanon. The first car bombing story, in turn, would be more interesting than the first car bombing story in Lebanon.

Another memory-based heuristic which can be used to judge the interestingness of categories is the following:

H-16: Symmetry

A category which lacks a symmetric category is more interesting than one for symmetric categories are known.

The symmetry can be along any of the dimensions used for similarity-based generalization in memory. For example, if the understander builds categories of occupations based on the gender of the actor, occupations in which both male and female actors are seen would be less interesting than those in which only males or only females are seen. This assumes, of course, that these categories play some functional role in achieving the overall goal of the understander, otherwise there would be no principled reason for either building the categories or judging their interestingness.

Since AQUA learns via explanation-based generalization, not similarity-based generalization, it does not use the uniqueness and symmetry heuristics to judge the interestingness of categories in memory. Instead, it uses the heuristics mentioned in the previous section which look for unique examples of XPs rather than unique categories in memory.

7.2 Computing interestingness by combining heuristics

We have presented a set of content- and configuration-based heuristics for judging interestingness based on different types of understander goals. This set is not exhaustive, of course, but it illustrates the type of heuristics that an understander would use to determine interestingness and focus its attention.

The final measure of interestingness is derived by combining the recommendations of all the applicable heuristics. This is used to judge the interestingness of AQUA's questions, as well as the interestingness of facts that might be relevant to these questions.

7.2.1 Interestingness of questions

The interestingness of a question is determined as follows:

H-17: To determine the interestingness of a question, apply all the interestingness heuristics to the question and combine the interestingness recommendations of each of the heuristics.

There is a potential problem here since the heuristics given above don't recommend specific interestingness values. For example, is an anomaly involving an uninteresting goal more interesting

than a stereotypical way of achieving a highly interesting goal? This problem has not been addressed in this dissertation. AQUA pursues every question that is judged to be interesting by one or more heuristics.

For example, here are some of the unanswered questions raised by the blackmail story:

Pending interesting questions:

DID THE BOY BELIEVE THAT THE SUICIDE BOMBING [would result] IN THE DEATH OF THE BOY?

DID THE BOY WANT TO PREVENT A STATE MORE THAN THE BOY WANTED TO PRESERVE THE LIFE STATE OF THE BOY [i.e., what could he want more than his own life?]

WHY DID THE TERRORIST GROUP RECRUIT THE BOY FOR THE SUICIDE BOMBING?

Pending uninteresting questions:

WHY DID THE BOY DRIVE THE CAR?

Assumed: THE BOY DROVE THE CAR TO ENABLE THE SUICIDE BOMBING [by the instrumentality heuristic].

WHY DID THE TERRORIST GROUP PLAN THE SUICIDE BOMBING?

Assumed: THE TERRORIST GROUP PLANNED THE SUICIDE BOMBING BECAUSE THE TERRORIST GROUP PLANS TERRORIST ACTS [by the stereotypical action heuristic].

As shown on pages 130 and 132, AQUA determines that it is not worthwhile building a detailed explanations for why the boy drove the car and why the terrorist group planned the suicide bombing. If these heuristics are turned off, AQUA will build explanations for these actions. This does not mean that it will not understand the rest of the story, of course, but it will spend time pursuing inferences that are less relevant to the point of the story.

7.2.2 Interestingness of new concepts

The interestingness of a new concept depends on its relevance to questions in memory. AQUA can determine whether a new (possibly only partially activated) concept is interesting using the following heuristic:

H-18: To determine the interestingness of a new concept, use the concept to index to possibly relevant questions, check if the new concept could answer one or more of these questions, and combine the interestingness measures of those questions that are answerable.

In other words, new facts provided by a story are interesting to the extent that they help AQUA answer its questions. The interestingness of these questions determine the interestingness of the new fact. This in turn is used to control the inferences drawn from the new fact, as discussed in section 5.2.

7.2.3 Interestingness of concepts in memory

The interestingness of a hypothesis in memory depends on whether there are unresolved questions attached to the hypothesis. AQUA can determine whether a concept already in memory is interesting (or a particular aspect of that concept is interesting) by checking if it has any unanswered questions attached to it. This is done using the following heuristics:

H-19: To determine the interestingness of a concept in memory, retrieve all the questions indexed off that concept and combine the interestingness measures of these questions.

H-20: To determine the interestingness of a particular aspect or slot of a concept in memory, retrieve all the questions indexed off that slot of the concept and combine the interestingness measures of these questions.

For example, when AQUA reads the blackmail story, it builds a basic representation for the boy as follows:

```
(define-instance BOY.18
  (isa (set: {bomber lebanese-teenager}))
  (truth [in])
  (status [story-instance])
  (age [young-age.78])
  (gender [male.0])
  (nationality [lebanese.161])
  (school ...)
  (religion ...)
  (color-of-hair ...)
  (goals ...)
  (...))
```

The story mentions that the boy was young, male, Lebanese and a bomber. But there are several other facts that the understander could infer about the boy: the school he attends, his religion, the color of his hair, his goals, and so on. Instead of trying to fill all these slots, the understander should be able to identify which slots are worth filling. In the blackmail story, it is the boy's religion and goals that are relevant, and not his school or the color of his hair.

When AQUA reads this story, it first tries to explain the boy's actions using the religious fanatic XP.⁷ After the hypothesis verification questions for this XP are installed, the frame representing the boy has questions attached to it:

⁷Detailed program transcripts are shown in chapter 8.

```

(define-instance BOY.18
  (isa      (set: {bomber lebanese-teenager}))
  (truth    [in])
  (status   [story-instance])
  (age      [young-age.78])
  (gender   [male.0])
  (nationality [lebanese.161])
  (religion (questions
              (set: {religion.446})))
  (religious-zeal (questions
                   (set: {religious-zeal.459})))
  (goals    (questions
              (set: {a-destroy.456})))
  (school   ...)
  (religion ...)
  (color-of-hair ...))

```

Due to heuristic H-20, AQUA is now interested in the *religion*, *religious-zeal* and *goals* of the boy, since these slots have questions attached to them. Similarly, when AQUA finds out that the boy was actually blackmailed into going on the mission, the *religion* and *religious-zeal* slots will no longer be interesting; instead, there will be a new set of questions arising from the blackmail explanation. Contrast this with a typical script-based story understander such as FRUMP [DeJong, 1979], which was interested in every slot simply because it was there in the script. In a sense, every slot in every active schema was a prediction or a question about the story. In AQUA, only the slots that are actually relevant to AQUA's reasoning tasks are turned into questions. Furthermore, these questions, and therefore AQUA's interests, keep changing as the hypotheses being considered change, as opposed to being statically pre-determined.

In summary, AQUA uses its heuristics to focus its attention on questions and facts that are interesting, either by virtue of answering questions that AQUA is interested in, or by raising new questions. This allows the understanding process to be sensitive to the questions that the understander is currently seeking answers to. For example, AQUA uses the above heuristics to determine how it would like a new concept to be further specified, thus favoring disambiguation towards satisfaction of its goals (see section 5.1.2).

7.3 Examples

In conclusion, let us illustrate AQUA's interestingness heuristics by using them to determine the interestingness of some example stories. AQUA does not perform this task by applying all its heuristics to the story taken as a whole; instead, it uses its heuristics to focus its attention on interesting questions and facts as they are encountered, as demonstrated by the above transcripts. However, it is a useful exercise to collect all these judgements in one place in order to see the contribution of individual heuristics.

7.3.1 Example: A stereotypical story with personal relevance

Consider the following story:

S-50: Suicide bomber strikes U.S. embassy in Beirut.

A teenage girl exploded a car bomb at the U.S. embassy in Beirut today, killing herself and causing a number of casualties, security sources said. A statement by an unidentified terrorist group claimed responsibility for the attack, adding that the girl was a martyr for the cause of Islam.

This story raises the question of the girl's motivations in doing the suicide bombing. AQUA's interestingness heuristics can be used to judge how interesting this question is.

H-10 Instrumentality: The suicide bombing is not instrumental to a known larger plan.

H-3 Vicarious crisis goal: The girl's action affects her p-life goal.

H-8 XP availability: A stereotypical XP is available (xp-religious-fanatic).

H-13 XP applicability: The stereotypical XP is applicable.

H-14 Hypothesis verification: The stereotypical XP is verified.

H-9 Human interest: The stereotypical XP discusses personal motivation.

H-15 Uniqueness: Many examples of this XP have been seen before.

H-4 Actor relevance: The actor and planner are unknown to the understander.

H-5 Object relevance: The U.S. embassy is personally relevant to an American understander.

H-1 Goal relevance: No personal goals are achieved or violated.

H-6 Location relevance: Beirut is not personally relevant to the understander.

In summary, this story is relatively uninteresting. It is interesting only to the extent that it is about the U.S. embassy (personal relevance of object), and that it discusses the motivations behind the violation of someone's p-life goal (vicarious crisis goal). However, since these motivations have a standard explanation that has been seen many times before, this story is not interesting even from the point of view of explanation.

7.3.2 Example: A novel story with no personal relevance

Now consider applying the same heuristics to the blackmail story:

S-2: Boy Says Lebanese Recruited Him as Car Bomber.

JERUSALEM, April 13 — A 16-year-old Lebanese was captured by Israeli troops hours before he was supposed to get into an explosive-laden car and go on a suicide bombing mission to blow up the Israeli Army headquarters in Lebanon. ...

What seems most striking about [Mohammed] Burro's account is that although he is a Shiite Moslem, he comes from a secular family background. He spent his free time not in prayer, he said, but riding his motorcycle and playing pinball. According to his account, he was not a fanatic who wanted to kill himself in the cause of Islam or anti-Zionism, but was recruited for the suicide mission through another means: blackmail.

H-10 **Instrumentality:** The suicide bombing is not instrumental to a known larger plan.

H-3 **Vicarious crisis goal:** The boy's action affects his p-life goal.

H-8, H-13 **XP availability and applicability:** A stereotypical XP (xp-religious-fanatic) but inapplicable. Another stereotypical XP (xp-blackmail) is applied in a novel context.

H-14 **Hypothesis verification:** xp-religious-fanatic is refuted. xp-blackmail applies, but raises new questions such as "What could the boy want more than his own life?" which must be answered before the hypothesis is completely filled out.

H-9 **Human interest:** The explanation discusses personal motivation.

H-15 **Uniqueness:** xp-blackmail has never been applied to a suicide bombing story before.

H-4 **Actor relevance:** The actor and planner are unknown to the understander.

H-5 **Object relevance:** The objects involved are unknown to the understander.

H-1 **Goal relevance:** No personal goals are achieved or violated.

H-6 **Location relevance:** Lebanon is not personally relevant to the understander.

This story, although not personally relevant, is interesting from the point of view of human interest. The story discusses novel explanations for the motivations behind the actions involving the violation of a shared thematic goal, p-life.

7.3.3 Example: Another novel story

Let us consider a final example to illustrate the failure of the above heuristics to capture certain types of interestingness.

S-4: Garbage can bomb kills 3 in Beirut.

BEIRUT, Lebanon, August 9 — A bomb concealed in a garbage can exploded today in Moslem West Beirut, killing 3 people and wounding 15. Within minutes, armed [Shiite] militiamen rushed into the streets to assert control. ... The Shiites stopped cars to check documents and they took 9 or 10 people into custody. ...

The radio station of the Christian militia, the Voice of Free Lebanon, noted that all the detained were Christians. It said they had been beaten and added that the treatment proved that "any stick was good enough for beating Christians." The bombing, it said, "was obviously not a Christian affair."

When the above heuristics are applied to this story, they provide the expected recommendations:

H-8, H-13 **XP availability and applicability:** A stereotypical XP (religious fanatic) is easy to find and apply.

H-9 **Human interest:** The story is about the motivations of an organization, not of any particular individual.

H-15 **Uniqueness:** The bombing uses an unusual instrument (a garbage can).

H-16 **Symmetry:** Shiite Moslems are the target of the bombing instead of the planners, as is usually the case.

H-4, H-5, H-6 **Personal relevance:** The actor, planner and location are unknown to the understander.

This story, therefore, is more interesting than a stereotypical story such as S-50, since it involves an unusual instrument (a garbage can). Also, the Shiites are the target of the bombing instead of the perpetrators, as is more common in the stories that are reported in American newspapers. However, since the motivational aspects of the story are not very different from typical religious fanatic stories, this story is less interesting than a novel story such as S-2.

This example also illustrates the limitations of these heuristics, since they fail to capture the grim humor in the story, which is partly what makes this story interesting. The heuristics given here judge relevance to understanding goals, but they make no evaluation of humor. Similarly, AQUA's heuristics do not evaluate other kinds of interestingness arising from discourse structure, since such measures of interestingness would not be functionally useful to AQUA's understanding goals which do not include discourse goals.

7.4 Conclusion: Interest-producing conditions

In chapter 4 we introduced *relevance goals*:

Any system with goals will ultimately be concerned about identifying aspects of the current situation that are relevant to its own goals. Questions determine the "interestingness" of the input, in the sense that the input is interesting only if it addresses one's goals or helps to answer one's questions.

In this chapter, we defined interestingness as a criterion for inference control. Since the understander needs to focus its attention on those inferences likely to help it achieve its overall goals, it must prioritize its questions in order to be able to devote its resources in pursuing questions that are most likely to be useful towards achieving these goals. Questions with higher priority are said to be "interesting" from the point of view of these understanding goals.

Thus, interestingness is a heuristic measure of the relevance of the input to the understander's goals or questions. Since the point of answering questions is to improve one's understanding of the domain, interestingness can also be thought of as a measure of the likelihood of learning something from the story if one processes it in detail.

This is a functional approach to the problem of interestingness [Hidi and Baird, 1986; Schank, 1979] from the perspective of the theory of question-driven understanding. A similar approach can be used for systems performing other cognitive tasks, such as planning, since these systems would also need to focus their attention on inferences that were relevant to goals arising from their tasks.

How would a system determine that a piece of information is "interesting?" Is interestingness a property of the information? Or is the interestingness of various types of information a fact about the system?

Interestingness is neither inherent in the information nor in the system, but rather arises from the interaction between the two. It arises from the interaction between the stimulus and the goals of the system. A system with no goals would have no reason to find any input more interesting than any other, nor would any particular piece of information be universally interesting for all systems unless they shared the same goals.⁸

Interestingness also depends on the current state of the knowledge of the system. There is an optimum level of knowledge at which an input that relates to that knowledge is most interesting. This occurs when just the right amount of knowledge is missing. If the understander knows too much about the domain of the input, the input is not interesting because it does not tell the understander anything new. If the understander knows too little, the input is too deviant and cannot be related to existing knowledge structures.

In other words, interestingness is a non-monotonic function of knowledge and uncertainty, which peaks at just the right amount of knowledge and uncertainty. Small deviations from expectations are optimal in creating interest [Kintsch, 1980]. Thus interestingness depends on a notion of "normality" [Schank, 1979], and arises from gaps in background knowledge. These gaps correspond to questions or knowledge goals for an understanding program such as AQUA.

In AQUA, interest in a concept is triggered by its likely relevance to questions or understanding goals, and continuing interest is determined by its continuing significance to these goals. This is related to the "goal satisfaction principle" of [Hayes-Roth and Lesser, 1976], which states that more processing should be given to knowledge sources whose responses are most likely to satisfy processing goals, and to the "relevance principle" of [Sperber and Wilson, 1986], which states that humans pay attention only to information that seems relevant to them. These principles make sense because cognitive processes are geared to achieving a large cognitive effect for a small effort. To achieve this, the understander must focus its attention on what seems to it to be the most relevant information available [Sperber and Wilson, 1986].

In this dissertation, we have shown how the understander could use interestingness heuristics to focus understanding tasks. Interestingness also affects text comprehension and recall [Anderson, 1982; Hidi *et al.*, 1982; Luftig and Greeson, 1983; Baird and Hidi, 1985], judging the goodness of stories [Brewer and Lichtenstein, 1982; Wilensky, 1980; de Beaugrande, 1982; Brewer, 1983], and appears to be crucial for motivation and memorability. These issues are beyond the scope of this dissertation.

AQUA uses heuristics to compute the relevance of a concept to the system's questions, as well as the priorities of the questions themselves. These heuristics judge both the structure of the knowledge (e.g., missing explanation) as well as its content (e.g., personal relevance). The final measure of interestingness is derived by combining the recommendations of all the applicable heuristics.

Once the interestingness of a question or piece of input has been determined, AQUA uses it to guide processing by focussing its resources on the more interesting aspects of the story. Since the heuristics are geared towards learning, this ensures that AQUA spends its time on those aspects of the story that are most likely to result in something useful being learned. Without its interestingness

⁸The "absolute interests" (sex, violence) of [Schank, 1979] arise from thematic goals which are universal among people.

heuristics, AQUA would still learn the same things, but it would spend a lot more time drawing inferences that ultimately turn out to be irrelevant.

Chapter 8

Program design: Putting it all together

We have discussed several related issues in understanding, including the nature of questions, explanation, inference control, subjective interpretation, integrated processing, opportunistic memory, learning and interestingness. We have also discussed the AQUA program, which is a question-driven understanding program that addresses these issues from a common viewpoint, that of questions viewed as knowledge goals of the program. This chapter discusses further details of the design of the AQUA program. The program is implemented in T/Scheme [Rees *et al.*, 1984] and runs on an Apollo DN3000.

8.1 Control structure: The understanding cycle

The basic control structure of AQUA is shown in figure 8.1, and corresponds to the understanding cycle discussed in section 5.4.1. In a sense, the initial processing is similar to that of typical “script-based” understanders: words in the input text are used to instantiate memory structures, and open slots in these memory structures are used as predictions for the rest of the story. However, there are three main differences (expressed here in a “slot-filling” terminology):

1. Typically, all open slots in newly instantiated structures are used as “requests” or “predictions,” and cause the understander to look for fillers for those slots. AQUA, however, uses its interestingness heuristics to mark interesting slots to be used as predictions or questions. In addition, slots can be marked as being interesting by understanding tasks when their values are needed but not yet known.
2. Open slots arise not only from script-like knowledge structures, but also from causal or explanatory structures.
3. Typically, the ultimate task of the understander is to fill in as many of these open slots as possible. However, the action of filling in a slot does not do anything more than provide a

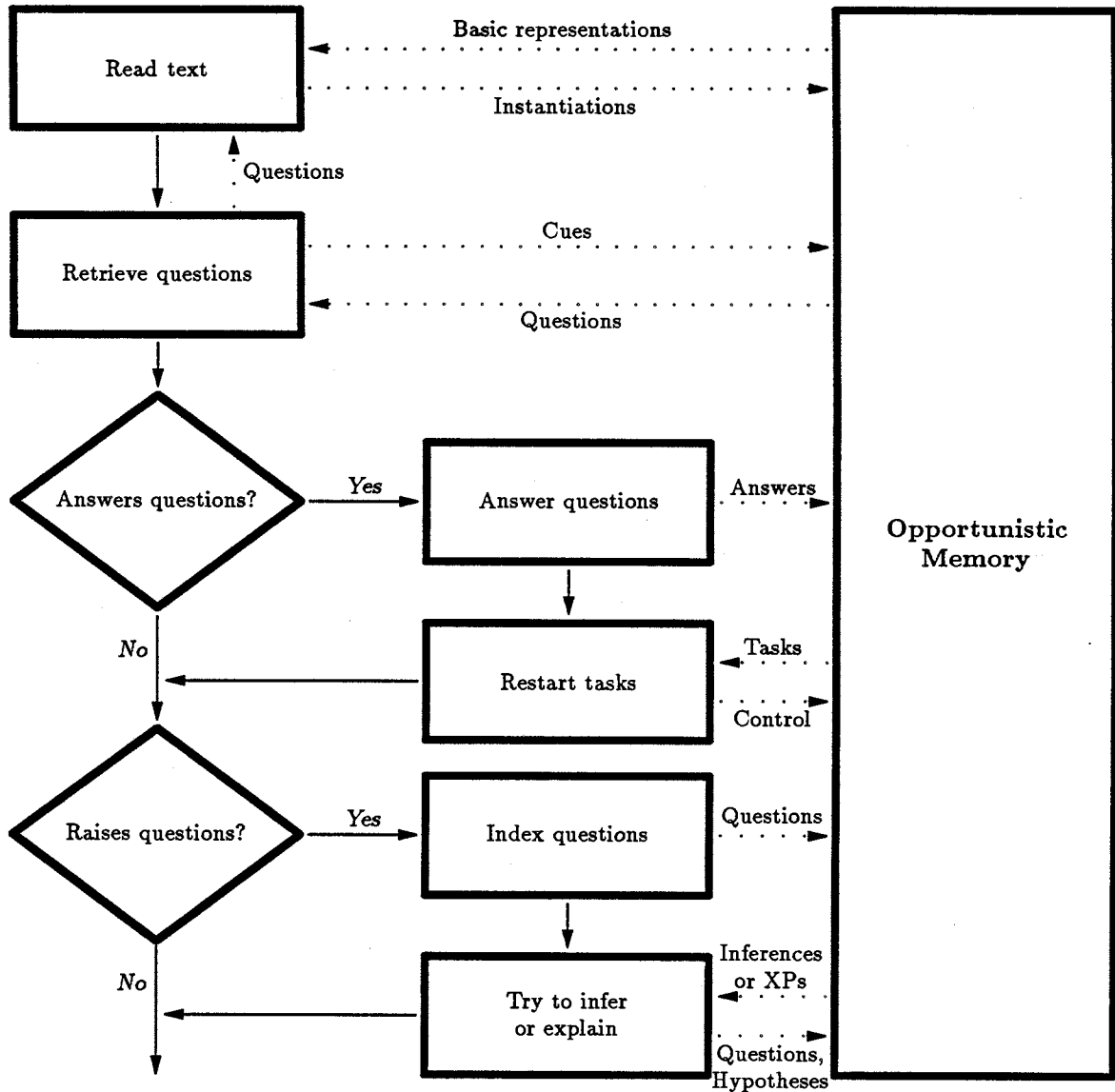


Figure 8.1: Control structure: The understanding cycle. A fact is interesting if it answers pending questions in memory, or if it raises new questions (e.g., via anomaly detection or other interestingness heuristics). Uninteresting facts pass vertically down with minimal processing; interesting facts cause suspended understanding tasks to be restarted, or new tasks to be created. New tasks can give rise to new questions, which are suspended along with the tasks if answers are not yet known and cannot be inferred.

value for that slot. In AQUA, however, slots aren't filled for their own sake, but rather for the sake of performing some kind of reasoning with that value (e.g., confirming a hypothesis).

Thus AQUA subscribes to the basic slot-filling idea, but extends this idea by (1) selecting which slots are worth filling, (2) using different kinds of knowledge structures to provide slots, and (3) remembering why particular slots need to be filled so that it can use the filled values when they become known.

8.2 Memory representations

8.2.1 The frame system

AQUA's memory is organized into a hierarchical semantic network of *frames* or schemas. We have developed a frame representation language to make it easier to enter such representations into the program.¹ This allows us to specify constraints on role fillers, variable bindings between role fillers, as well as to represent slots as relations between frames. Frames are organized into a multiple inheritance hierarchy, which is similar to a semantic net such as that used in KODIAK [Wilensky, 1986]. In addition to the information present in standard semantic networks, frames in memory are tagged with truth and status information as described in chapter 5.

8.2.2 Representing concepts

8.2.2.1 Frames, slots and facets

Concepts are represented as frames with slots. Each slot has a number of facets, representing different kinds of information about that slot:

value: The value of the filler for that slot. For example, the *actor* slot of an instance of a *mop* frame would have a pointer to an instance of a *volitional-agent* frame in its *value* facet.

default: A default or prototypical filler for that slot, e.g., a pointer to a *stereotype* frame.

constraints: Semantic constraints on fillers for that slot. These include type constraints as well as constraints between fillers of different slots, expressed as variable bindings between these fillers.

explanation: The explanation for that slot, e.g. a *volitional* explanation for why a *volitional-agent* chose to be the *actor* of the *mop*.

questions: A set of questions about this slot. Each question is represented as a frame consisting of a concept specification and a task specification, as described in chapter 5.

relation: A pointer to the relation between the frame and the filler of the *value* facet. The above facets are also accessible via the relation frame. Though this is redundant, it has allowed us to play with different representational ideas. Relation frames are described below.

¹Much of the frame representation language was developed by Eric Jones.

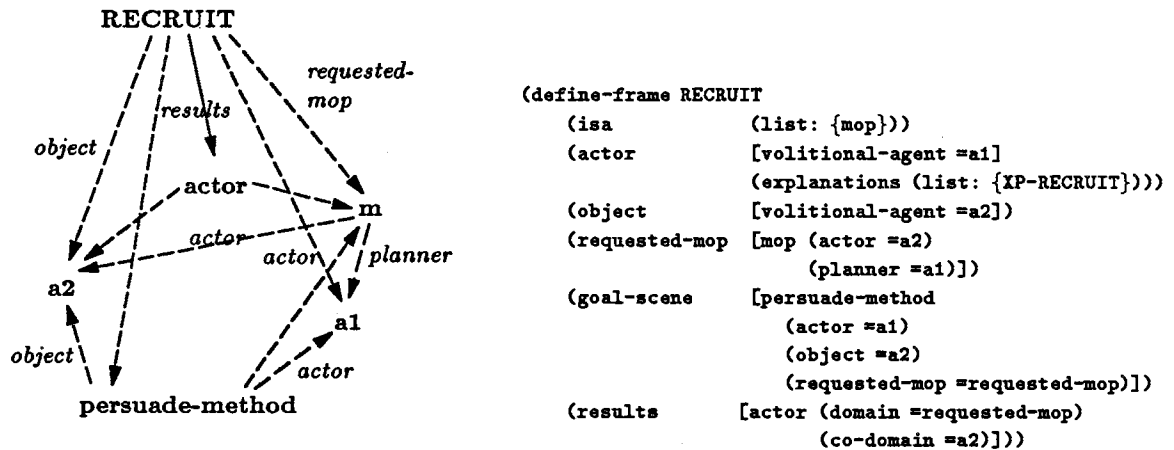


Figure 8.2: Frame language specifications of concepts represented as graph structures in memory. Variables used for role filler bindings are prefixed with = signs; here, =a1 and =a2 represent the two players in the recruit. Slot names can also be used as variables; thus, =requested-mop is a variable that is automatically bound to the requested-mop slot. Situation indices for explanations (here, xp-recruit) are stored in the explanations facet of the appropriate slot. The representation for recruit specifies that recruiting is an action in which an agent a1 requests another agent a2 to be the actor of the requested-mop m, of which a1 is the planner. The goal-scene of the recruit, in which the final result actually occurs, is an action which is of type persuade-method. The result of a successful recruit, is an action in which the actor relation between the requested-mop and a2 occurs, i.e., a2 performs the requested-mop.

If no facet is explicitly specified, it is assumed that the value facet is being referred to. A frame description in the frame representation language specifies a graph structure in memory that represents the concept, as shown in the example in figure 8.2.

8.2.2.2 Types of role fillers

Although slots in frames usually contain pointers to other frames, some slots can contain other kinds of values. For example, the results slot of a mop contains an unordered list of result frames.

Unordered lists are represented as mathematical sets with membership and equality predicates. For example, the questions facet is filled with a set of questions. Each member of this set is a question frame. If any of these frames ceases to be a question, it automatically ceases to belong to the set. If two frames in a set are found to refer to the same object, they are automatically combined into one set element.

Other types of fillers for slots include strings (e.g., the name slot of a person), numbers (e.g., the weight slot of a physical-object), and so on.

8.2.2.3 Relations

Slots and slot fillers specify relations between concepts. For example, the **results** slot specifies a causal relation of a particular kind between a mop and a state. Similarly, the **actor** slot specifies a participatory relation between a mop and a volitional-agent.

Relations are represented as full fledged frames in memory (e.g., see [Wilensky, 1986]). This enables AQUA to reason about the relations themselves. For example, the proposition "the boy knew that the bombing would result in his own death" is represented as a **knowledge-state** of the boy whose **mobject** is the **results** relation between the bombing MOP and the **death-state** whose object is the boy, as discussed in chapter 5. This could not be represented if the **results** relation was not represented explicitly, i.e., if the only representation of this fact was the presence of the **death-state** frame in the **results** slot of the bombing frame. The frame system representation of this proposition, as well as the equivalent graph representation in memory, is shown in figure 8.3.

The mapping from relation frames to slot names is done via the **slot** slot, which is a privileged slot in the relation frame. For example, the **slot** slot of the actor frame contains the symbol "actor." This tells AQUA that an actor relation is stored in the "actor" slot of its domain (a mop). This information is inherited by specializations of the actor frame, such as **driver** (the actor of a drive) or **bomber** (the actor of a bombing). For example, the **driver** relation is also stored in the "actor" slot of the drive frame. Thus slot names are arbitrary symbols which are meaningless in themselves; AQUA's knowledge about relations is represented in relation frames and not imputed to names of slots [McDermott, 1981].

8.2.2.4 The MOP hierarchy

MOP and scene connections are represented as causal relations between event structures. The **instrumental-scene** of a MOP is another MOP which results in a state that enables the first MOP. Similarly, the **goal-scene** of a MOP is that scene in which the main results of the overall MOP are actually achieved. Figure 8.4 shows the representation of event structures in a MOP-and-scene packaging hierarchy. Primitive actions are represented as mops with no scenes.

8.2.2.5 Ontology

There are two types of entities in memory:

- Entities that represent concepts in the domain. These include **volitional-agents**, **locations**, **mops**, **xps**, **goals**, **goal-orderings**, etc., and are organized in an inheritance and packaging hierarchy as shown in figure 8.5.
- Entities that represent internal reasoning concepts. These include **truth**, **status**, **suspended reasoning tasks**, etc.

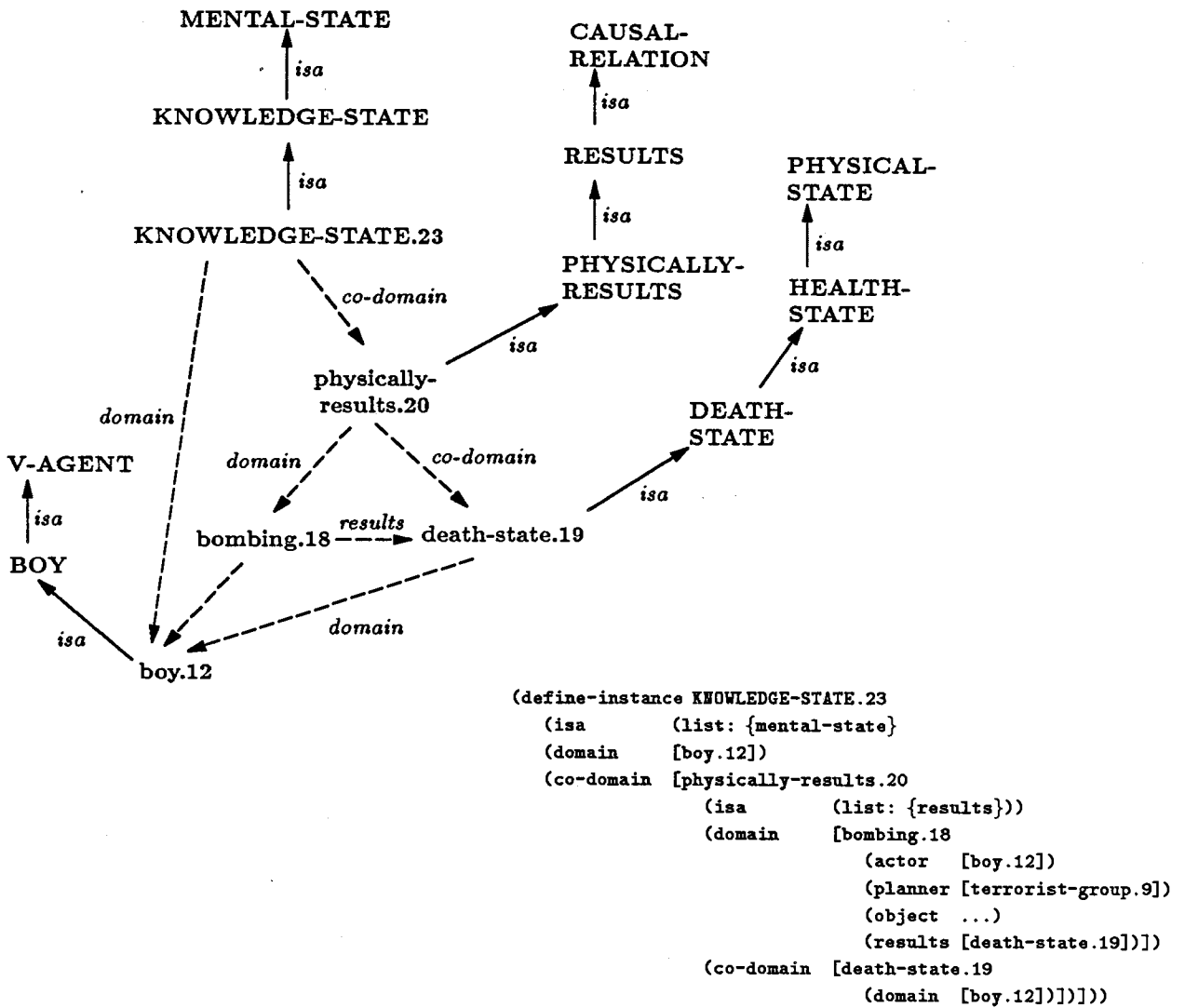


Figure 8.3: Equivalence between frame language representations and graph structures in memory. The proposition "The boy knew that the bombing would result in his own death" is represented as a knowledge-state of the boy, whose mobject (the co-domain of the knowledge-state relation) is the results relation between the bombing MOP and the death-state of the boy.

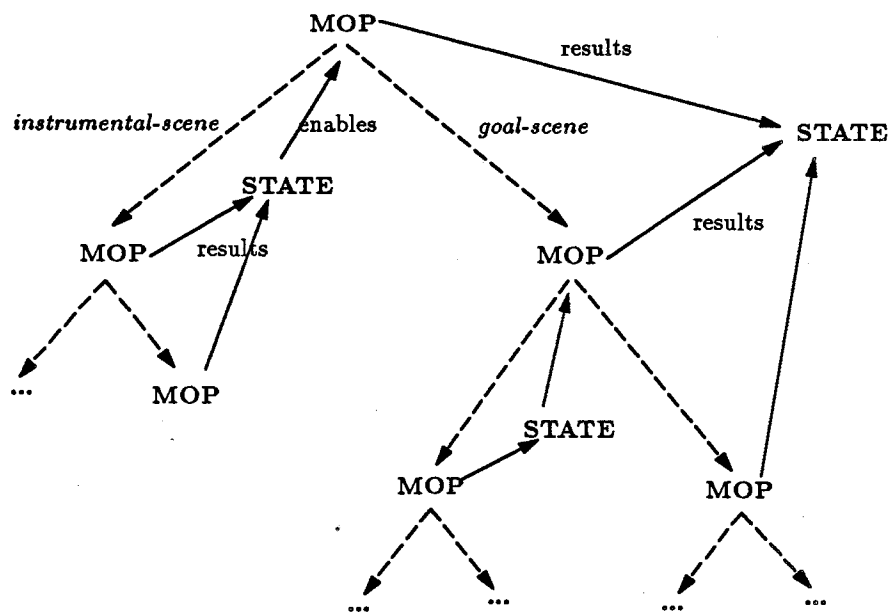


Figure 8.4: Packaging hierarchy of event structures. The *instrumental-scene* of a mop is another mop which results in a state which enables (or is a precondition of) the overall mop, and the *goal-scene* of a mop is a mop which achieves the main results of the overall mop. Dashed lines represent packaging links from mops to their scenes, and solid lines represent causal links between mops and states.

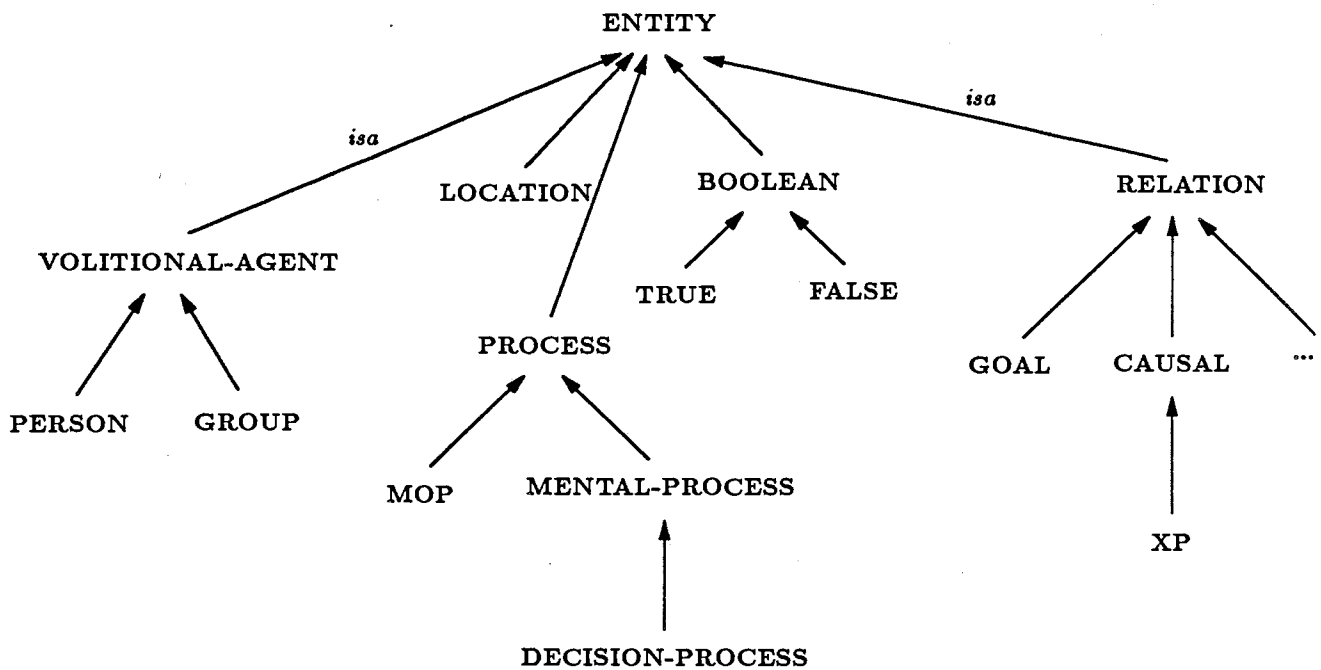


Figure 8.5: Some concepts in AQUA's memory. Concepts are represented as frames and are organized in an multiple inheritance and packaging hierarchy. The relation hierarchy is shown in more detail in figure 8.6. Upward solid lines represent *isa* links.

As described earlier, relations between frames are also represented as frames in an inheritance hierarchy, as shown in figure 8.6. A state is a special type of relation, e.g., a *knowledge-state* is a relation between a person and a fact that he knows; a *health-state* is a relation between an *animate-object* and a *health-state-value*.

Here are some examples of the types of relations used by AQUA:

Categorization relations: AQUA uses two basic categorization relations. All entities in memory are *isa* one or more entities higher up in the inheritance hierarchy. Furthermore, *volitional-agents* can be viewed as being typical character *stereotypes*. Both these relations have multiple values. For example, a *suicide-bombing* *isa* both *suicide* and *bombing* since it inherits aspects of each.

isa: entity → {entity}

stereotype-views: volitional-agent → {stereotype}

Attributes of inanimate objects: An attribute is a relation whose co-domain is an *attribute-value*, which could be one of an enumerate set of values, each of which is a frame of a given type (e.g., {male, female}, a set of nations) or an arbitrary value of a particular type (e.g., a string, a number, a *volitional-agent*). The co-domain can specify a binding for the domain. For example, the *purpose* of a *physical-object* is a *mop* of which the *physical-object* is an *instrument*. Here are some examples of attributes of physical objects:

<i>color:</i>	<i>physical-object</i>	→	<i>color-value</i> = {red, blue, ...}
<i>name:</i>	<i>physical-object</i>	→	<i>name-value</i> (a string)
<i>weight:</i>	<i>physical-object</i>	→	<i>weight-value</i> (a number)
<i>size:</i>	<i>physical-object</i>	→	<i>size-value</i>
<i>speed:</i>	<i>physical-object</i>	→	<i>speed-value</i>
<i>purpose:</i>	<i>physical-object</i>	→	[<i>mop</i> (<i>instrument</i> =domain)]
<i>creator:</i>	<i>physical-object</i>	→	<i>volitional-agent</i>
<i>owner:</i>	<i>physical-object</i>	→	<i>volitional-agent</i>

Attributes of volitional agents: Volitional agents (persons or groups of people) inherit attributes such as *weight* from *physical-object*. In addition, they also have attributes such as:

<i>gender:</i>	<i>person</i>	→	<i>gender-value</i> = {male, female}
<i>nationality:</i>	<i>volitional-agent</i>	→	<i>nation</i>
<i>religion:</i>	<i>volitional-agent</i>	→	<i>religion-type</i>
<i>language:</i>	<i>volitional-agent</i>	→	<i>language-type</i>
<i>age:</i>	<i>volitional-agent</i>	→	<i>age-value</i>
<i>occupation:</i>	<i>volitional-agent</i>	→	[<i>mop</i> (<i>actor</i> =domain)]

Physical states:

<i>physical-state:</i>	<i>physical-object</i>	→	{ <i>intact</i> , <i>destroyed</i> }
<i>health-state:</i>	<i>person</i>	→	{ <i>alive</i> , <i>dead</i> }

Here, *alive* in turn represents the set {*healthy*, *sick*}.

Spatial relations:

<i>contains:</i>	<i>physical-object</i>	→	<i>physical-object</i>
<i>at-location:</i>	<i>physical-object</i>	→	<i>location</i>

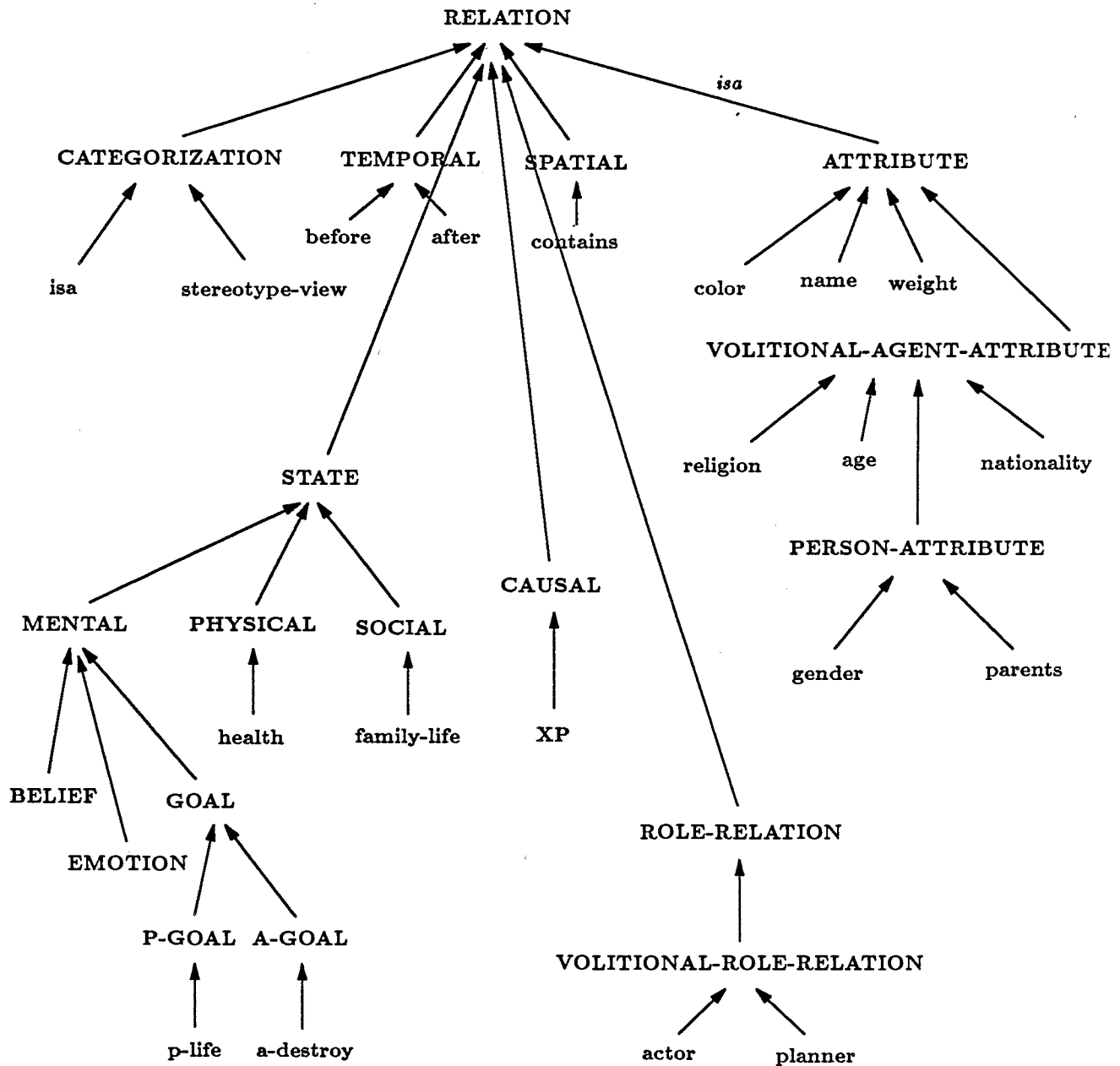


Figure 8.6: Some relations in AQUA's memory. Relations are represented as regular frames and are organized in a relation hierarchy. Role relations are shown in figure 8.7, and causal relations in figure 8.8. Upward solid lines represent *isa* links.

Temporal relations: AQUA uses a minimal representation of time since it does not do much temporal reasoning. Processes (including MOPs) and relations (including states) are tagged with starting and ending times. These are used to provide simple constraints on causal relations. For example, if a process results in a state, the start-time of the state is known to be after the start-time of the process (and, in some cases, the same as the end-time of the process). This provides simple causal constraints such as which processes a state could have resulted from.

start-time:	process	→	time-point
end-time:	process	→	time-point
before:	time-point	→	time-point
after:	time-point	→	time-point

Mental states: Mental states include goals, goal-orderings, belief-states and emotional-states.

goal: A goal is a mental-state, and represents a wanting relation between a volitional-agent (the goal-actor) and a desired state or relation (the goal-object). If the goal is out, it means that the agent does not want the goal-object of the goal.

Types of goals include preservation-goals, avoidance-goals, prevention-goals and achievement-goals.

goal-ordering: A goal-ordering is a relation between two goals, and represents the priorities of the goal-actor of these goals.

belief-state: A belief-state is a mental-state which describes the belief of an agent (the domain of the relation) in the truth of the assertion represented by the frame that is the co-domain. A knowledge-state, such as the one shown in figure 8.3, is a special case of a belief-state.

Social states: social-control, family-life.

Role relations: These include the object, instrument, from and to slots of a mop. Role relations are shown in figure 8.7.

The meanings of these slots are represented by the frames representing the corresponding role relations. For example, the object of a mop is defined to be a physical-object whose state is changed by the mop. This state change is the main result of the mop. Similarly, the instrument of a mop is a physical-object whose purpose is to bring about the state change in the object.

8.2.3 Representing explanation patterns

Chapter 3 discussed the representation of decision models, which are generalized volitional explanations. Instantiated explanations or hypotheses are stored in the **explanations** facet. A frame representation of the religious fanatic XP was presented in section 3.2.5. As another example of the representation of XPs, consider the representation of the Kamikaze explanation that was used to illustrate learning by case adaptation in chapter 6.

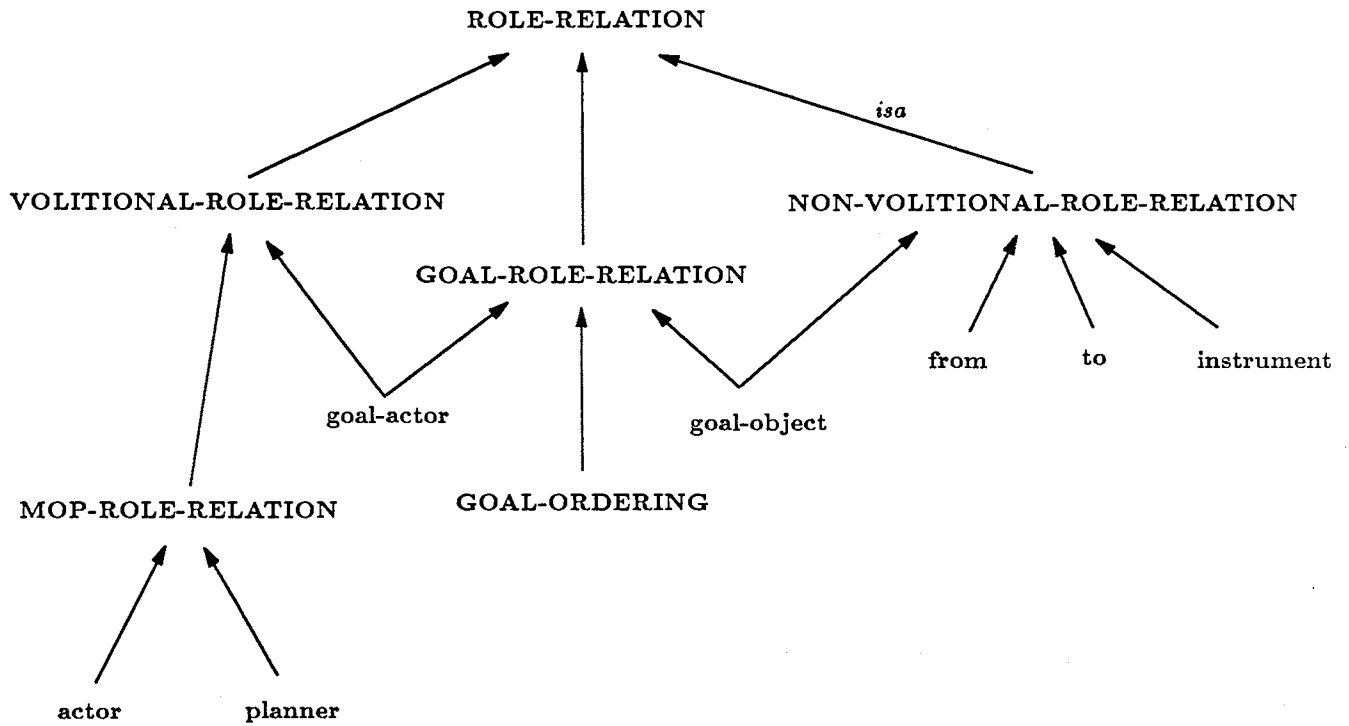


Figure 8.7: Some role relations in AQUA's memory. Role relations are part of the relation hierarchy shown in figure 8.6. Upward solid lines represent *isa* links.

```

(define-frame XP-KAMIKAZE

;; XP type - doing something with a bad outcome because you want the good outcome even more.

  (isa          (set: {volitional-xp}))
  (xp-type     (set: {xp-sacrifice}))

;; PRE-XP-NODES - nodes that need to be known before you can apply the XP.
;;
;; Explains:
;;   Why AGENT = KAMIKAZE-PILOT did a PLANE BOMBING
;;   with RESULTS (i) DEATH-STATE of AGENT
;;                 (ii) DESTROYED-STATE of TARGET, a PHYSICAL-OBJECT
;;                 whose OWNER is USA-MILITARY

  (explains    [bombing (actor    =agent)
                  (object    =target)
                  (instrument [zero-plane]
                    (main-result [destroyed-state =good-state (domain =target)])
                    (side-effect [death-state =bad-state (domain =agent)]
                      (opposite-of [alive-state =violated-state
                                    (domain =agent)])))]

  (target      [physical-object (owner =enemy [usa-military])])

;; The goals of the AGENT that were satisfied and violated.

  (satisfied-goal [a-destroy (goal-actor =agent)
                          (goal-object =good-state)])

  (violated-goal  [preservation-goal (goal-actor =agent)
                          (goal-object =violated-state)])

;; An assertion that the AGENT preferred destroying the target to preserving his life.

  (goal-ordering [goal-ordering (goal-actor =agent)
                          (lesser-goal =violated-goal)
                          (greater-goal =satisfied-goal)])

;; XP-ASSERTED-NODES - nodes asserted by the XP as the explanation for the EXPLAINS node.
;; INTERNAL-XP-NODES - internal nodes asserted by the XP in order to link up XP-ASSERTED-NODES
;; to the EXPLAINS node.

;; The pilot was Japanese and a zealous (fanatic) patriot.

  (nationality [japanese (domain =agent)
                    (co-domain =country)])

  (country     [japan.0])

  (belief-state [patriotic-belief-state
                  (domain =agent)
                  (believed-item [patriotic-belief (domain =agent)
                                    (co-domain =country)])
                  (belief-strength [fanatic.0])])

```

```

;; Explanation links:
;;
;; AGENT is ZEALOUSLY PATRIOTIC
;; because AGENT is JAPANESE
;;
;; AGENT wants A-GOAL (DESTROYED-STATE (TARGET))
;; because AGENT is JAPANESE and AGENT is ZEALOUSLY PATRIOTIC
;;
;; AGENT prefers A-GOAL (DESTROYED-STATE (TARGET)) over P-LIFE (AGENT)
;; because AGENT is JAPANESE and AGENT is ZEALOUSLY PATRIOTIC
;;
;; AGENT did the BOMBING
;; because the BOMBING satisfied a greater goal even though it violated a lesser goal
;; (PLAN-SELECTION, represented as an abstract XP involving a DECISION structure).

(links (set: {[emotionally-results (domain =nationality)
                                   (co-domain =belief-state)]

          [emotionally-initiates (domain =belief-state)
                                   (co-domain =satisfied-goal)]

          [emotionally-initiates (domain =belief-state)
                                   (co-domain =goal-ordering)]

          [xp-sacrifice (agent =agent)
                        (action =explains)
                        (role =role)
                        (good-state =good-state)
                        (main-result =main-result)
                        (bad-state =bad-state)
                        (side-effect =side-effect)
                        (satisfied-goal =satisfied-goal)
                        (violated-goal =violated-goal)
                        (goal-ordering =goal-ordering)]}))

```

This XP illustrates some of the features of the frame representation language, as well as the vocabulary used to represent volitional elements such as goals, goal-orderings and beliefs. The causal structure of the XP uses primitives causal links (such as *emotionally-results*), as well as abstract volitional XPs (*xp-sacrifice*) that are in turn represented using the vocabulary of planning/decision models, as described next.

8.2.3.1 Causal representations

XPs are causal structures composed of causal relations (e.g., *results*, *enables*) and elements of planning/decision models (e.g., *knowledge-of-outcome*). These were described in chapter 3, and are reviewed below.

Causal relations: A *causal-relation* is a primitive XP that can't be decomposed further. Its domain is a *pre-xp-node* that is causally connected to its *co-domain* (the *explains* node).

An XP can be viewed as a *causal-relation* that can either be treated as a single unit or elaborated if the details are required. Ultimately, all XPs consist of *causal-relations* (e.g., results of actions) as well as *decisions* that represent the basic plan selection process.

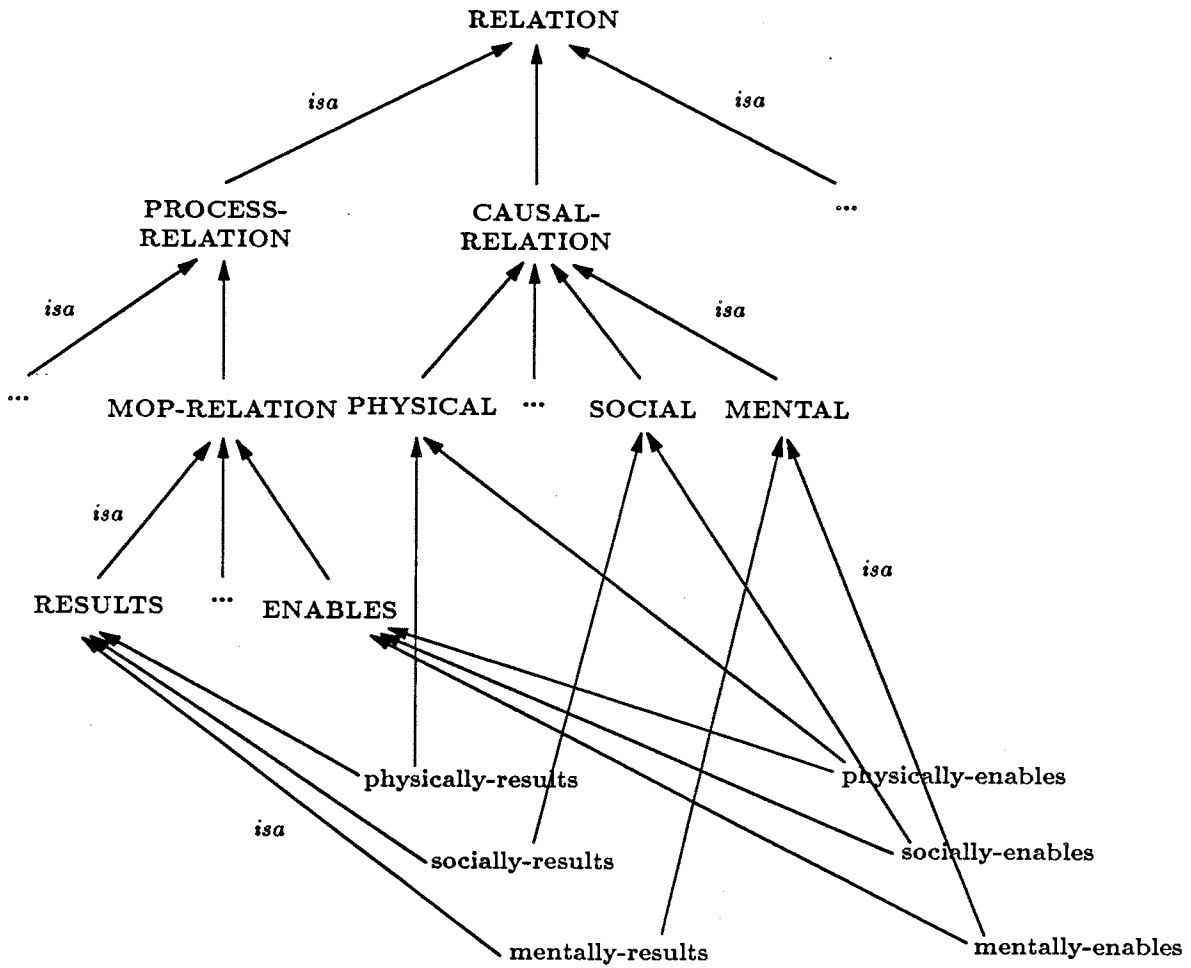


Figure 8.8: Some primitive causal relations in AQUA's memory. A causal relation is a primitive XP that can't be decomposed any further.

Outcomes: An outcome is a set of states, viewed as the (combined) result of a mop from the point of view of a volitional-agent participating in the mop (the actor, planner, etc.) or even from the point of view of an onlooker.

Considerations: A consideration represents the factors considered by a volitional-agent in deciding whether to participate in an action or mop, i.e., whether to play a volitional-role in that mop. These factors include the agent's goals, goal-orderings and beliefs about the expected outcome of the mop. Other factors that AQUA does not yet reason about include the emotional-state of the agent.

Decisions: A decision is a mental-process in which a volitional-agent considers participating in a particular volitional-role-relation of a mop. The outcome of the decision process is a state of having made a choice (either freely or via coercion or inducement) to participate or not to participate in the volitional-role-relation being considered, as discussed in chapter 3.

We have also developed similar representations for the decision to exit from a mop, but the current implementation of AQUA does not use these in a general manner.

8.3 Program example

This section presents a detailed program example to illustrate the process of question-driven understanding. In particular, we will demonstrate how the different aspects of this process that we have discussed are integrated together, using an annotated transcript of AQUA reading the blackmail story as an example, which will be followed by an example of incremental learning.

S-16: Terrorists recruit boy as car bomber.

A 16-year-old Lebanese got into an explosive-laden car and went on a suicide bombing mission to blow up the Israeli army headquarters in Lebanon. ...

The teenager was a Shiite Moslem but not a religious fanatic. He was recruited for the mission through another means: blackmail.

Processing story:

Processing word: TERRORISTS

Word type: TM
TM Definition: TERRORIST-GROUP

Instantiating concept: TERRORIST-GROUP

Processing word: RECRUIT

Word type: EB
EB Definition: RECRUIT

Instantiating concept: RECRUIT

The sentence is in ACTIVE voice
 Setting sentence SUBJECT PP to TERRORIST-GROUP.9

As discussed in section 4.1.2, basic syntactic and semantic processing is done via text-level questions, so that text-level processing is uniformly integrated with higher level processing. Similarly, memory-level processing, such as MOP and scene attachment, is also done via questions. For example, subject attachment, noun group connection and scene prediction are done as follows:

Asking basic questions for RECRUIT.15

WHO DID THE RECRUIT?

Using the sentence SUBJECT: THE TERRORIST GROUP
 to fill the ACTOR slot of THE RECRUIT

Answering question: WHO DID THE RECRUIT?
 with: THE TERRORIST GROUP.

Filling ACTOR slot of THE RECRUIT

Specializing THE RECRUIT with ACTOR = THE TERRORIST GROUP
 Specializing THE RECRUIT to TERRORIST RECRUIT

Predicting scenes for THE RECRUIT

WHAT IS THE INSTRUMENTAL SCENE OF THE RECRUIT?
 Transforming to: HOW DID THE TERRORIST GROUP LOCATE THE PERSON?

WHAT IS THE GOAL SCENE OF THE RECRUIT?
 Transforming to: HOW DID THE TERRORIST GROUP PERSUADE THE PERSON?

Predicting syntactic OBJECT of sentence

WHAT VOLITIONAL AGENT IS THE OBJECT OF THE RECRUIT?

Processing word: BOY

Word type: TM
 TM Definition: BOY

Instantiating concept: BOY

Asking basic questions for BOY.85

Using the sentence OBJECT: THE BOY
 to fill the OBJECT slot of THE RECRUIT

Answering question: WHAT VOLITIONAL AGENT IS THE OBJECT OF THE RECRUIT?
with: THE BOY

Processing word: AS

Word type: FW
Function: PREPOSITION

Resetting clause globals

Predicting fillers based on AS

Predicting filler for the REQUESTED-MOP slot of THE RECRUIT

WHAT WAS THE PURPOSE OF THE RECRUIT?
Transforming to: DID THE BOY DO THE TERRORIST METHOD PLANNED BY THE
TERRORIST GROUP?

Processing word: CAR

Word type: TR
TR Definition: CAR

Instantiating concept: CAR

Processing word: BOMBER

Word type: TM
TM Definition: BOMBING
ACTOR Definition: BOMBER.121

Instantiating concept: BOMBER
Instantiating concept: BOMBING

Asking basic questions for BOMBING.117

Trying to attach THE CAR to THE BOMBING

Initial list of candidates: (TARGET INSTRUMENT)

Building two hypotheses for CAR BOMBING:
THE CAR IS THE TARGET OF THE BOMBING.
THE CAR IS THE INSTRUMENT OF THE BOMBING.

Satisfying prediction from AS

Filling REQUESTED-MOP slot of THE RECRUIT

Answering question: WHAT WAS THE PURPOSE OF THE RECRUIT?
with: THE BOMBER WENT ON A CAR BOMBING MISSION.

Predicting scenes for THE CAR BOMBING

WHAT IS THE INSTRUMENTAL SCENE OF THE CAR BOMBING?

Transforming to: WHAT DID THE BOMBER DRIVE?

WHAT IS THE GOAL SCENE OF THE CAR BOMBING?

Transforming to: WHAT DID THE BOMBER BLOW UP?

Processing word: *PERIOD*

Word type: PUNCTUATION

Resetting phrase globals

Setting sentence SUBJECT PP to ()

Setting sentence SUBJECT MOP to ()

Resetting clause globals

Built representations for:

THE TERRORIST GROUP.

THE BOY.

THE CAR BOMBING.

THE TERRORIST GROUP RECRUITED THE BOY TO DO THE CAR BOMBING.

At this point, AQUA has finished reading the headline, and has built minimal representations for the concepts encountered (shown below). It now checks whether these concepts answer any pending questions or raise new ones. In this example, AQUA has not read any stories previously, and no questions are pending; however, new questions could still be raised which will then be used to focus the reading process during the rest of the story.

Processing THE TERRORIST GROUP

```
(define-instance TERRORIST-GROUP.9
  (isa (set: {terrorist-group}))
  (truth (value [in]))
  (status (value [story-instance])))
```

Running interestingness heuristics

No pending questions about THE TERRORIST GROUP.

No questions triggered by THE TERRORIST GROUP.

Processing THE BOY

```
(define-instance BOY^BOMBER.18
  (isa      (set: {boy bomber}))
  (truth    (value [in]))
  (status   (value [story-instance]))
  (age      (value [young-age.78]))
  (gender   (value [male.0])))
```

Running interestingness heuristics

```
No pending questions about THE BOY.
No questions triggered by THE BOY.
```

Processing THE RECRUIT

```
(define-instance RECRUIT.15
  (isa      (set: {terrorist-recruit}))
  (truth    (value [in]))
  (status   (value [story-instance]))
  (planner  (value [terrorist-group.9]))
  (actor    (value [terrorist-group.9])
            (relation [actor.41]))
  (object   (value [boy^bomber.18])
            (relation [object.83]))
  (instrument (value [inanimate-object.32]))
  (scenes   (set: {locate.16 persuade-method.21}))
  (instrumental-scene (value [locate.16])
                    (questions (set: {instrumental-scene.57})))
  (goal-scene (value [persuade-method.21])
             (questions (set: {goal-scene.73})))
  (requested-mop (value [car-bombing.20])
                (relation [requested-mop.108]))
  (results    (value (set: {actor.333}))))
```

```
THE TERRORIST GROUP RECRUITED THE BOY TO DO THE CAR BOMBING.
```

Running interestingness heuristics

```
No pending questions about THE RECRUIT.
No questions triggered by THE RECRUIT.
```

In addition to the above processing, actions must also be checked to see whether they need explanation. AQUA checks each of the **volitional-roles** in each action to see whether it needs to be explained. A **volitional-role** needs to be explained if it is not uninteresting (in which case a simple explanation is assumed) and is anomalous. Details of the explanation process will be deferred until later in the transcript when the suicide bombing mission is explained.

Running interestingness heuristics on ACTOR role:

THE TERRORIST GROUP RECRUITED THE BOY TO DO THE CAR BOMBING.

No questions triggered

No anomalies detected

Installing default explanation: XP-RECRUIT

Assumed:

THE TERRORIST GROUP DID THE RECRUIT.

because THE TERRORIST GROUP WANTED THE CAR BOMBING DONE BUT BY
SOMEONE ELSE.

Running interestingness heuristics on PLANNER role:

THE TERRORIST GROUP PLANNED THE RECRUIT.

No questions triggered

The PLANNER, THE TERRORIST GROUP, is also the ACTOR of THE RECRUIT
Not doing PLANNER explanation

Installing default explanation: XP-PLANNER

Assumed:

THE TERRORIST GROUP PLANNED THE RECRUIT.

because THE TERRORIST GROUP WANTED TO DO THE RECRUIT.

Running interestingness heuristics on OBJECT role:

THE OBJECT OF THE RECRUIT IS THE BOY.

No questions triggered

Asking anomaly detection questions for OBJECT.83

Anomaly! THE BOY is not a typical MILITARY AGENT.

Trying to explain why THE OBJECT OF THE RECRUIT IS THE BOY.

Searching for possible XPs for OBJECT.83

Anomaly! No explanations found.

From this point, details of lexical processing that have already been illustrated will be omitted in the interest of brevity. The transcript below illustrates automatic specialization of concepts, and the merging of co-referential concepts, in addition to scene prediction and the running of interestingness heuristics which are done as before.

 Processing word: A
 Processing word: 16-YEAR-OLD
 Processing word: LEBANESE

Trying to attach A TEENS AGE to THE LEBANESE PERSON
 Attaching A TEENS AGE to the AGE slot of THE LEBANESE PERSON

Specializing THE LEBANESE PERSON with AGE = A TEENS AGE
 Specializing THE LEBANESE PERSON to LEBANESE TEENAGER

Searching headline concepts for referent for THE LEBANESE TEENAGER

Merging concepts:
 THE BOY
 THE LEBANESE TEENAGER
 Resulting frame ISA (BOY BOMBER LEBANESE-TEENAGER)

 Processing word: GOT
 Processing word: INTO

The sentence is in ACTIVE voice

Using the sentence SUBJECT: THE BOY
 to fill the ACTOR slot of THE ENTER

Answering question: WHO DID THE ENTER?
 with: THE BOY

Predicting scenes for THE ENTER

Not asking WHAT IS THE INSTRUMENTAL SCENE OF THE ENTER?
 because no new information added over the default

Not asking WHAT IS THE GOAL SCENE OF THE ENTER?
 because no new information added over the default

Predicting syntactic OBJECT of sentence

WHAT CONTAINER IS THE OBJECT OF THE ENTER?

 Processing word: AN
 Processing word: EXPLOSIVE-LADEN
 Processing word: CAR

Trying to attach THE EXPLOSIVES to THE CAR
 Attaching THE EXPLOSIVES to the CONTAINS slot of THE CAR

Satisfying prediction from INTO
 Filling OBJECT slot of THE ENTER

Answering question: WHAT CONTAINER IS THE OBJECT OF THE ENTER?
with: THE EXPLOSIVE-LADEN CAR

Specializing THE ENTER with OBJECT = THE EXPLOSIVE-LADEN CAR
Specializing THE ENTER to ENTER VEHICLE

Processing word: AND

Word type: FW
Function: CONJUNCTION

Built representations for:

THE TEENAGE LEBANESE BOY.
THE EXPLOSIVE-LADEN CAR.
THE BOY ENTERED THE EXPLOSIVE-LADEN CAR.

Processing THE EXPLOSIVE-LADEN CAR

Running interestingness heuristics

No pending questions about THE EXPLOSIVE-LADEN CAR.
No questions triggered by THE EXPLOSIVE-LADEN CAR.

Processing THE ENTER

```
(define-instance ENTER-VEHICLE.710
  (isa      (set: {enter-vehicle}))
  (truth    (value [in]))
  (status   (set: {story-instance}))
  (main-result (value [at-location-volitional-agent.713]))
  (ptrans-object (value [boy^bomber^lebanese-teenager.18]))
  (actor     (value [boy^bomber^lebanese-teenager.18])
             (relation [actor.734]))
  (to       (value [car.110])))
THE BOY ENTERED THE EXPLOSIVE-LADEN CAR.
```

Running interestingness heuristics

No pending questions about THE ENTER.
No questions triggered by THE ENTER.

Running interestingness heuristics on ACTOR role:
THE BOY ENTERED THE EXPLOSIVE-LADEN CAR.

No pending questions triggered

Asking anomaly detection questions for ACTOR.333

Was THE ENTER instrumental to another action?
 Does THE BOY typically do ENTERs?
 Did THE BOY want the outcome of THE ENTER?

No anomalies detected

Processing word: WENT

Word type: EB
 Definition predicted by (CAR)
 EB Definition: DRIVE

Instantiating concept: DRIVE

The sentence is in ACTIVE voice
 Setting sentence SUBJECT PP to BOY^BOMBER^LEBANESE-TEENAGER.18

Initial list of candidates: (ACTOR INSTRUMENT)
 Using the sentence SUBJECT: THE BOY
 to fill the ACTOR slot of THE DRIVE

Answering question: WHO DID THE DRIVE?
 with: THE BOY

Answering question: WHAT IS THE INSTRUMENTAL SCENE OF THE BOMBING?
 with: THE BOY DROVE THE EXPLOSIVE-LADEN CAR.

Confirming hypothesis:
 THE CAR IS THE INSTRUMENT OF THE BOMBING.

Predicting scenes for THE DRIVE

WHAT IS THE INSTRUMENTAL SCENE OF THE DRIVE?
 Transforming to: WHAT DID THE BOY ENTER?

Not asking WHAT IS THE GOAL SCENE OF THE DRIVE?
 because no new information added over the default

Answering question: WHAT IS THE INSTRUMENTAL SCENE OF THE DRIVE?
 with: THE BOY ENTERED THE EXPLOSIVE-LADEN CAR.

Notice that AQUA has just answered a question pending in the instrumental-scene slot of the drive frame, which was asked earlier during the anomaly detection phase. This restarts the suspended explanation process, allowing AQUA to build a simple explanation for entering the car:

 WHAT IS THE INSTRUMENTAL SCENE OF THE DRIVE?
 was the last EQ for:
 THE BOY DID THE ENTER TO ENABLE THE DRIVE.

Restarting suspended explanation:
 THE BOY DID THE ENTER TO ENABLE THE DRIVE.

Applying XP-INSTRUMENTAL-SCENE.802 to ACTOR.734

Installing NODES for XP-INSTRUMENTAL-SCENE.802

THE BOY CHOSE THE ENTER AS A PLAN TO SATISFY A GOAL.

No HVQs for XP-INSTRUMENTAL-SCENE.802

Installing LINKS for XP-INSTRUMENTAL-SCENE.802

Inferred: THE SELECTION OF THE ENTER BY THE BOY TO SATISFY A GOAL
 MENTALLY RESULTED IN THE BOY DOING THE ENTER.

Finished installing XP-INSTRUMENTAL-SCENE.802

Assumed:

THE BOY ENTERED THE EXPLOSIVE-LADEN CAR.
 because THE BOY DID THE ENTER TO ENABLE THE DRIVE.

Predicting syntactic OBJECT of sentence

As described in chapter 7, the entering action is "explained away," being judged uninteresting by the instrumentality heuristic. The drive will be explained away in a similar manner, causing AQUA to focus its attention on the suicide bombing itself.

 Processing word: ON
 Processing word: A
 Processing word: SUICIDE
 Processing word: BOMBING
 Processing word: MISSION

Satisfying prediction from ON
 Filling INSTRUMENTAL-SCENE-OF slot of THE DRIVE

Trying to attach THE MISSION to THE BOMBING

Merging concepts:
 THE MISSION
 THE BOMBING
 Resulting frame ISA (BOMBING MISSION)

Trying to attach THE BOMBING to THE SUICIDE

Merging concepts:

THE SUICIDE

THE BOMBING

Resulting frame ISA (BOMBING MISSION SUICIDE)

Searching headline concepts for referent for THE SUICIDE BOMBING

Merging concepts:

THE CAR BOMBING

THE SUICIDE BOMBING

Processing word: TO
Processing word: BLOW
Processing word: UP
Processing word: THE
Processing word: ISRAELI
Processing word: ARMY
Processing word: HEADQUARTERS

Trying to attach THE ARMY to THE HEADQUARTERS

Attaching THE ARMY to the OWNER slot of THE HEADQUARTERS

Specializing THE HEADQUARTERS with OWNER = THE ARMY:

Specializing THE HEADQUARTERS to ARMY HEADQUARTERS

Trying to attach ISRAELI to THE ARMY

Attaching ISRAELI to the NATIONALITY slot of THE ARMY

Attaching THE ISRAELI ARMY HEADQUARTERS to the OBJECT slot of THE CAR
BOMBING

Processing word: IN
Processing word: LEBANON

Attaching LEBANON to the AT-LOCATION slot of THE ISRAELI ARMY
HEADQUARTERS

Processing word: *PERIOD*

Resetting phrase globals

Resetting clause globals

Built representations for:

LEBANON.
 THE ISRAELI ARMY HEADQUARTERS IN LEBANON.
 THE BOY DROVE THE EXPLOSIVE-LADEN CAR.
 THE BOY WENT ON A SUICIDE BOMBING MISSION.
 THE BOY BLEW UP THE EXPLOSIVE-LADEN CAR.

Processing LEBANON.

Running interestingness heuristics

No pending questions about LEBANON.
 No questions triggered by LEBANON.

Processing THE ISRAELI ARMY HEADQUARTERS IN LEBANON.

Running interestingness heuristics

No pending questions about THE ISRAELI ARMY HEADQUARTERS IN
 LEBANON.
 No questions triggered by THE ISRAELI ARMY HEADQUARTERS IN
 LEBANON.

Processing THE DRIVE.

THE BOY DROVE THE EXPLOSIVE-LADEN CAR.

Running interestingness heuristics

No pending questions about THE DRIVE.
 No questions triggered by THE DRIVE.

Running interestingness heuristics on ACTOR role:

THE BOY DROVE THE EXPLOSIVE-LADEN CAR.

THE DRIVE is instrumental to THE CAR BOMBING.

Installing instrumentality explanation: XP-INSTRUMENTAL-SCENE

Assumed:

THE BOY DROVE THE EXPLOSIVE-LADEN CAR.
 because THE BOY DID THE DRIVE TO ENABLE THE CAR BOMBING.

Now AQUA begins to process the car bombing act which is the central action in this story. As before, AQUA runs its interestingness heuristics, including anomaly detection, on each volition-

role in this mop. These will be shown in detail to illustrate the process of anomaly detection and explanation construction.

Processing THE CAR BOMBING

```
(define-instance CAR-BOMBING.20
  (isa (set: {car-bombing}))
  (truth (value [in]))
  (status (set: {story-instance}))
  (actor (value [boy^bomber^lebanese-teenager.18])
         (relation [actor.333]))
  (planner (value [terrorist-group.9])
          (relation [planner.271]))
  (instrument (value [car.110]))
  (instrumental-scene (value [drive.129])
                     (questions (set: {instrumental-scene.224})))
  (goal-scene (value [blow-up-bomb.120])
             (questions (set: {goal-scene.242}))))
```

THE BOY WENT ON A SUICIDE BOMBING MISSION.

Running interestingness heuristics

No pending questions about THE CAR BOMBING.

Running interestingness heuristics on PLANNER role:
THE TERRORIST GROUP PLANNED THE CAR BOMBING.

THE TERRORIST GROUP isn't the ACTOR of THE CAR BOMBING

Asking anomaly detection questions for PLANNER.271:

Does THE TERRORIST GROUP typically plan CAR BOMBINGS?

THE CAR BOMBING is a TERRORIST ACT.
TERRORIST ACTs are often planned by TERRORIST GROUPs.

Did THE TERRORIST GROUP want the outcome of THE CAR BOMBING?

Characterizing the outcome: ((DESTROYED-STATE) (DEATH-STATE))
of MOP: THE CAR BOMBING
from the point of view of: THE TERRORIST GROUP (the PLANNER)

DOES THE TERRORIST GROUP WANT TO ACHIEVE THE DAMAGE OR
DESTRUCTION OF THE ISRAELI ARMY HEADQUARTERS IN LEBANON?
Matches typical GOALS

DOES THE TERRORIST GROUP WANT TO ACHIEVE THE DEATH OF THE BOY?
 No relevant GOALS found

Installing explanation: XP-TYPICAL-ACTION

Assumed:

THE TERRORIST GROUP PLANNED THE CAR BOMBING.
 because TERRORIST GROUPS TYPICALLY PLAN CAR BOMBINGS.

Running interestingness heuristics on ACTOR role:
 THE BOY DID THE CAR BOMBING.

Asking anomaly detection questions for ACTOR.333

Was THE CAR BOMBING instrumental to another action?
 Does THE BOY typically do CAR BOMBINGS?
 Did THE BOY want the outcome of THE CAR BOMBING?

Characterizing the outcome: ((DESTROYED-STATE) (DEATH-STATE))
 of MOP: THE CAR BOMBING
 from the point of view of: THE BOY (the ACTOR)

DOES THE BOY WANT TO ACHIEVE THE DAMAGE OR DESTRUCTION OF THE
 ISRAELI ARMY HEADQUARTERS IN LEBANON?
 No relevant GOALS found

DOES THE BOY WANT TO ACHIEVE THE DEATH OF THE BOY?
 Anomaly! THE BOY WANTS TO ACHIEVE THE DEATH OF THE BOY.

Characterized outcome as a BAD outcome

Retrieved abstract XPs:
 XP-SACRIFICE
 XP-NOT-KNOW-OUTCOME

Searching for stereotypical XPs

Asking EQ: IS THE BOY A TYPICAL TEENAGER?
 Asking EQ: WHY WOULD A TEENAGER DO A SUICIDE BOMBING?

Situation index = SUICIDE-BOMBING
 Stereotype index = TEENAGER

No XPs found

Asking EQ: IS THE BOY A TYPICAL LEBANESE PERSON?
 Asking EQ: WHY WOULD A LEBANESE PERSON DO A SUICIDE BOMBING?

Situation index = SUICIDE-BOMBING
 Stereotype index = LEBANESE-PERSON

Retrieved stereotypical XPs:
 XP-RELIGIOUS-FANATIC (category index = XP-SACRIFICE)

AQUA's retrieves the xp-religious-fanatic XP shown in section 5.3.1. Since AQUA prefers specific XPs to abstract ones, it tries xp-religious-fanatic before xp-sacrifice. This will be shown in detail to illustrate the explanation process. First, AQUA walks through all the pre-xp-nodes to check whether the XP fits the story:

Trying to explain why THE BOY DID THE CAR BOMBING.
 using the XP: XP-RELIGIOUS-FANATIC

Checking if XP-RELIGIOUS-FANATIC.445 is applicable

WHO DID THE CAR BOMBING?
 Known (IN) -(node ACTOR.333)-

IS IT TRUE THAT THE BOY EXISTS?
 Known (IN) -(node BOY^BOMBER^LEBANESE-TEENAGER.18)-

DID THE BOY GO ON A SUICIDE BOMBING MISSION?
 Known (IN) -(node CAR-BOMBING.20)-

WHAT IS THE DAMAGED STATE OF THE ISRAELI ARMY HEADQUARTERS IN
 LEBANON?
 Known (IN) -(node DESTROYED-STATE.26)-

DID THE CAR BOMBING RESULT IN THE DAMAGE OR DESTRUCTION OF THE
 ISRAELI ARMY HEADQUARTERS IN LEBANON?
 Known (IN) -(node MAIN-RESULT.325)-

WHAT IS THE STATE OF HEALTH OF THE BOY?
 Known (IN) -(node DEATH-STATE.104)-

DID THE CAR BOMBING RESULT IN THE DEATH OF THE BOY?
 Known (IN) -(node SIDE-EFFECT.317)-

Note that AQUA has assumed that the target has been damaged or destroyed, using default inference for the results slot of blow-up-bomb.120. Since xp-religious-fanatic is applicable as far as AQUA can tell at this point, the XP is applied and a hypothesis built. This involves installing the causal chain represented by the internal-xp-nodes and links of the XP. Those nodes that are not yet known to be true become the hypothesis verification questions (HVQs) of the hypothesis.

 Applying XP-RELIGIOUS-FANATIC.445 to ACTOR.333

```
(f:define-instance ACTOR.333
  (isa      (set: {actor}))
  (truth   (value [in]))
  (status  (value [story-instance]))
  (co-domain (value [boy^bomber^lebanese-teenager.18]))
  (domain  (value [car-bombing.20])))
```

Installing NODES for XP-RELIGIOUS-FANATIC.445

THE BOY.
 THE ISRAELI ARMY HEADQUARTERS IN LEBANON.
 THE RELIGION.
 THE BOY WANTS TO ACHIEVE THE DAMAGE OR DESTRUCTION OF THE ISRAELI
 ARMY HEADQUARTERS IN LEBANON.
 THE BOY WANTS TO PRESERVE THE LIFE STATE OF THE BOY.
 THE GOAL OF THE BOY TO ACHIEVE THE DAMAGE OR DESTRUCTION OF THE
 ISRAELI ARMY HEADQUARTERS IN LEBANON IS MORE IMPORTANT THAN THE
 GOAL OF THE BOY TO PRESERVE THE LIFE STATE OF THE BOY.
 THE RELIGION OF THE BOY IS A RELIGION.
 THE BOY BELIEVES FANATICALLY IN THE RELIGION.

Installing HVQs for XP-RELIGIOUS-FANATIC.445

Adding HVQ: WHAT IS THE RELIGION OF THE BOY?
 Indexing question: WHAT IS THE RELIGION OF THE BOY?

Adding HVQ: WHAT IS THE RELIGIOUS ZEAL OF THE BOY?
 Indexing question: WHAT IS THE RELIGIOUS ZEAL OF THE BOY?

Transforming to: DOES THE BOY BELIEVE FANATICALLY IN THE
 RELIGION?

Adding HVQ: DOES THE BOY WANT TO ACHIEVE THE DAMAGE OR DESTRUCTION
 OF THE ISRAELI ARMY HEADQUARTERS IN LEBANON?
 Indexing question: DOES THE BOY WANT TO ACHIEVE THE DAMAGE OR
 DESTRUCTION OF THE ISRAELI ARMY
 HEADQUARTERS IN LEBANON?

Installing LINKS for XP-RELIGIOUS-FANATIC.445

Inferred: THE RELIGION OF THE BOY MENTALLY RESULTED IN THE FANATIC
 BELIEF IN THE RELIGION.
 (truth HYPOTHESIZED-IN) -(node EMOTIONALLY-RESULTS.462)-

Inferred: THE FANATIC BELIEF IN THE RELIGION MENTALLY INITIATED THE GOAL OF THE BOY TO ACHIEVE THE DAMAGE OR DESTRUCTION OF THE ISRAELI ARMY HEADQUARTERS IN LEBANON.

(truth HYPOTHEZIZED-IN) -(node EMOTIONALLY-INITIATES.464)-

Inferred: THE FANATIC BELIEF IN THE RELIGION MENTALLY INITIATED THE PREFERENCE OF THE GOAL OF THE BOY TO ACHIEVE THE DAMAGE OR DESTRUCTION OF THE ISRAELI ARMY HEADQUARTERS IN LEBANON OVER THE GOAL OF THE BOY TO PRESERVE THE LIFE STATE OF THE BOY.

(truth HYPOTHEZIZED-IN) -(node EMOTIONALLY-INITIATES.460)-

Inferred: THE BOY DECIDED TO DO THE CAR BOMBING TO SATISFY THE GOAL OF THE BOY TO ACHIEVE THE DAMAGE OR DESTRUCTION OF THE ISRAELI ARMY HEADQUARTERS IN LEBANON AT THE EXPENSE OF THE GOAL OF THE BOY TO PRESERVE THE LIFE STATE OF THE BOY BECAUSE OF THE PREFERENCE OF THE GOAL OF THE BOY TO ACHIEVE THE DAMAGE OR DESTRUCTION OF THE ISRAELI ARMY HEADQUARTERS IN LEBANON OVER THE GOAL OF THE BOY TO PRESERVE THE LIFE STATE OF THE BOY.

(truth HYPOTHEZIZED-IN) -(node XP-SACRIFICE.449)-

Finished installing XP-RELIGIOUS-FANATIC.445

THE BOY DID THE CAR BOMBING.
because THE BOY WAS A RELIGIOUS FANATIC.

AQUA uses a simple template-based natural language generator to describe concepts in memory. A slightly more readable form of the above hypothesis was described in section 5.3.1. The final hypothesis is represented as follows:

```
(define-instance XP-RELIGIOUS-FANATIC.445
  (isa      (set: {xp-religious-fanatic}))
  (truth   (value [hypothesized-in]))
  (status  (value [question]))

  (explains (value [actor.333]))
  (action   (value [car-bombing.20]))
  (agent    (value [boy^bomber^lebanese-teenager.18]))
  (role     (value [actor.333]))

  (pre-xp-nodes (set: {actor.333
                       boy^bomber^lebanese-teenager.18
                       car-bombing.20
                       main-result.325
                       destroyed-state.26
                       side-effect.317
                       death-state.104})))
```

```

(internal-xp-nodes (set: {volitional-agent.465
                        headquarters.106
                        alive-state.165
                        preservation-goal.457
                        goal-ordering.458
                        religion-category.448}))

(xp-asserted-nodes (set: {religion.446
                        religious-belief-state.459
                        a-damage.456}))

(enemy          (value [volitional-agent.465]))
(target        (value [headquarters.106]))

(main-result   (value [main-result.325]))
(good-state    (value [destroyed-state.26]))
(satisfied-goal (value [a-damage.456]))

(side-effect   (value [side-effect.317]))
(bad-state     (value [death-state.104]))
(violated-state (value [alive-state.165]))
(violated-goal (value [preservation-goal.457]))

(goal-ordering (value [goal-ordering.458]))

(religion      (value [religion.446]))
(belief-state  (value [religious-belief-state.459]))

(links         (set: {emotionally-results.462
                    emotionally-initiates.464
                    emotionally-initiates.460
                    xp-sacrifice.449}))

```

This represents the instantiated hypothesis corresponding to the religious fanatic XP shown in section 5.3.1. Notice that "interesting" slots in the boy frame have been tagged with questions:

```

-----
(define-instance BOY^BOMBER^LEBANESE-TEENAGER.18
  (isa          (set: {boy bomber lebanese-teenager}))
  (truth        (value [in]))
  (status       (value [story-instance]))
  (age          (value [teens-age.78]))
  (gender       (value [male.0]))
  (nationality  (value [lebanese.161]))
  (goals        (questions (set: {a-damage.456})))
  (religious-belief (questions (set: {religious-belief-state.459})))
  (religion     (questions (set: {religion.446})))

```

In comparison with the minimal boy~bomber~lebanese-teenager.18 frame shown on page 163, this frame now has questions attached to it, which determine the slots that are interesting for the purposes of understanding this story.

Abstract XPs are applied in the same manner as stereotypical ones. As an example, consider the XP that the boy didn't know that he would die as a result of his action:

 Trying to explain why THE BOY DID THE CAR BOMBING.
 using the XP: XP-NOT-KNOW-OUTCOME

Checking if XP-NOT-KNOW-OUTCOME.397 is applicable

WHO DID THE CAR BOMBING?
 Known (IN) -(node ACTOR.333)-

IS IT TRUE THAT THE BOY EXISTS?
 Known (IN) -(node BOY~BOMBER~LEBANESE-TEENAGER.18)-

DID THE BOY GO ON A SUICIDE BOMBING MISSION?
 Known (IN) -(node CAR-BOMBING.20)-

WHAT IS THE STATE OF HEALTH OF THE BOY?
 Known (IN) -(node DEATH-STATE.104)-

DID THE CAR BOMBING RESULT IN THE DEATH OF THE BOY?
 Known (IN) -(node SIDE-EFFECT.317)-

DOES THE BOY WANT TO PRESERVE THE LIFE STATE OF THE BOY?
 Known (IN) -(node PRESERVATION-GOAL.405)-

IS IT TRUE THAT THE BELIEF THAT THE CAR BOMBING RESULTED IN THE
 DEATH OF THE BOY EXISTS?
 Not known -(node BELIEF.403)-

Indexing EQ for explanation:

IS IT TRUE THAT THE BELIEF THAT THE CAR BOMBING RESULTED IN THE
 DEATH OF THE BOY EXISTS?

Suspending explanation:

THE BOY DECIDED TO DO THE CAR BOMBING DESPITE THE VIOLATION OF THE
 GOAL OF THE BOY TO PRESERVE THE LIFE STATE OF THE BOY BECAUSE
 THE BOY DID NOT KNOW THAT THE CAR BOMBING WOULD RESULT IN THE
 DEATH OF THE BOY.

XP-NOT-KNOW-OUTCOME isn't applicable to ACTOR.333

The above transcript also illustrates how understanding tasks are suspended in memory. At this point, the boy~bomber~lebanese-teenager.18 frame has a new question attached to it in the

beliefs slot. This question must be answered before the above XP can be used. If AQUA finds out whether the boy knew about the negative outcome, the question will be answered and the above XP will either be confirmed or refuted, depending on the answer.

In this story, the religion questions attached to the boy~bomber~lebanese-teenager.18 frame are answered, causing the religious fanatic hypothesis to be refuted.

 Processing word: THE
 Processing word: BOY

Searching for referent for the description THE BOY

Merging concepts:
 THE TEENAGE LEBANESE BOMBER
 THE BOY

 Processing word: WAS
 Processing word: A
 Processing word: SHIITE
 Processing word: MOSLEM

Attaching THE SHIITE MOSLEM RELIGION to the RELIGION slot of THE BOY

 Answering question: WHAT IS THE RELIGION OF THE BOY?
 with: THE BOY IS A SHIITE MOSLEM.

This was an HVQ for XP-RELIGIOUS-FANATIC.445

Pending HVQs:
 WHAT IS THE RELIGIOUS ZEAL OF THE BOY?
 DOES THE BOY WANT TO ACHIEVE THE DAMAGE OR DESTRUCTION OF THE
 ISRAELI ARMY HEADQUARTERS IN LEBANON?

 Processing word: BUT
 Processing word: NOT
 Processing word: A
 Processing word: RELIGIOUS
 Processing word: FANATIC
 Processing word: *PERIOD*

 Answering question: WHAT IS THE RELIGIOUS ZEAL OF THE BOY?
 with: THE BOY DOES NOT BELIEVE FANATICALLY IN THE SHIITE
 MOSLEM RELIGION.

This was an HVQ for XP-RELIGIOUS-FANATIC.445

Refuting hypothesis:
 THE BOY DID THE CAR BOMBING.
 because THE BOY WAS A RELIGIOUS FANATIC.

Finally, AQUA reads that the boy was blackmailed into the suicide bombing. This answers the question "Why did the boy do the suicide bombing?" Since this is a novel explanation, AQUA learns new situation and stereotype indices for `xp-blackmail` as described in chapter 6.

Processing word: HE
 Processing word: WAS
 Processing word: RECRUITED
 Processing word: FOR
 Processing word: THE
 Processing word: MISSION
 Processing word: THROUGH
 Processing word: BLACKMAIL

Word type: EB
 EB Definition: BLACKMAIL

Instantiating concept: BLACKMAIL

Attaching THE BLACKMAIL to the GOAL-SCENE slot of THE RECRUIT

Answering question: WHAT IS THE GOAL SCENE OF THE RECRUIT?
 with: THE BLACKMAIL.

Trying explanations suggested by THE TERRORIST GROUP BLACKMAILED THE
 BOY TO DO THE CAR BOMBING.
 to explain THE BOY DID THE CAR BOMBING.

Trying to explain why THE BOY DID THE CAR BOMBING.
 using the XP: XP-BLACKMAIL

Novel explanation for A CAR BOMBING
 -- XP-BLACKMAIL!

Copying XP-BLACKMAIL to XP-BLACKMAIL-CAR-BOMBING

Applying XP-BLACKMAIL-CAR-BOMBING.1666 to ACTOR.333

Installing NODES for XP-BLACKMAIL-CAR-BOMBING.1666

THE TERRORIST GROUP.
 THE MOP.
 THE STATE.
 THE TERRORIST GROUP THREATENED THE BOY.
 THE BOY WANTS TO PREVENT THE STATE.
 THE BOY DID NOT VOLUNTEER FOR THE CAR BOMBING.

Installing HVQs for XP-BLACKMAIL-CAR-BOMBING.1666

DID THE TERRORIST GROUP PERSUADE THE BOY BY THREATENING TO DO THE
 MOP THAT WOULD RESULT IN THE STATE?
 DOES THE BOY WANT TO PREVENT THE STATE?
 DID THE BOY VOLUNTEER FOR THE CAR BOMBING?

Installing LINKS for XP-BLACKMAIL-CAR-BOMBING.1666

Finished installing XP-BLACKMAIL-CAR-BOMBING.1666

THE BOY DID THE CAR BOMBING.
 because THE TERRORIST GROUP BLACKMAILED THE BOY INTO DOING THE CAR
 BOMBING.

This was a novel explanation!

Building new stereotype from BOY^BOMBER^LEBANESE-TEENAGER.18

Inferring GOALS from XP-BLACKMAIL-CAR-BOMBING (successful):

THE ACTOR WANTED TO PRESERVE HIS OWN LIFE.
 THE ACTOR DID NOT WANT TO DESTROY A PHYSICAL OBJECT
 BELONGING TO THE ISRAELI ARMY.
 THE ACTOR WANTED TO AVOID SOMETHING.

Inferring GOAL-ORDERINGS from XP-BLACKMAIL-CAR-BOMBING
 (successful):

THE GOAL OF THE ACTOR TO AVOID SOMETHING WAS MORE IMPORTANT
 THAN THE GOAL OF THE ACTOR TO PRESERVE HIS OWN LIFE.

XP-RELIGIOUS-FANATIC failed because:

THE BOY DID NOT BELIEVE FANATICALLY IN THE SHIITE MOSLEM
 RELIGION.

Inferring BELIEFS from XP-RELIGIOUS-FANATIC (failed):

THE ACTOR DID NOT BELIEVE FANATICALLY IN A RELIGION.

Built new VOLITIONAL-AGENT stereotype (STEREOTYPE.1679):

Typical goals: NOT-VOLUNTEER (CAR-BOMBING) (in)
 AVOIDANCE-GOAL (STATE) (question)
 P-LIFE (in)
 A-DAMAGE (ISRAELI-ARMY-HEADQUARTERS) (out)

Typical goal-orderings:
 AVOIDANCE-GOAL (STATE) over P-LIFE (question)

Typical beliefs: RELIGIOUS-ZEAL = NOT A FANATIC (in)

Typical features: AGE = TEENAGE AGE (hypothesized)
 GENDER = MALE (hypothesized)
 RELIGION = SHIITE MOSLEM RELIGION (hypothesized)
 NATIONALITY = LEBANESE (hypothesized)

Indexing XP-BLACKMAIL-CAR-BOMBING
 Situation index = CAR BOMBING
 Stereotype index = STEREOTYPE.1679 (the ACTOR)

Unanswered questions:

DID THE BOY WANT TO PREVENT A STATE MORE THAN THE BOY WANTED TO
 PRESERVE THE LIFE STATE OF THE BOY?

New questions for XP-BLACKMAIL-CAR-BOMBING:

DID THE ACTOR WANT TO PREVENT A STATE MORE THAN THE ACTOR WANTED TO
 PRESERVE THE LIFE STATE OF THE ACTOR?

Predicting scenes for THE BLACKMAIL

...

 Processing word: *PERIOD*

Built representations for:

THE TERRORIST GROUP BLACKMAILED THE BOY TO DO THE CAR
 BOMBING.
 THE BOY WENT ON A SUICIDE BOMBING MISSION.

In conclusion, here is AQUA's summary of the blackmail story:

Finished processing story.

Main events:

 THE TERRORIST GROUP RECRUITED THE BOY TO DO THE CAR BOMBING.

Explanations:

THE TERRORIST GROUP WANTED THE CAR BOMBING DONE BUT BY SOMEONE ELSE.

(hypothesized-in) -(node XP-RECRUIT.590)-

Verification questions:

DOES THE TERRORIST GROUP WANT TO ACHIEVE THE BOY DOING THE CAR BOMBING?

DOES THE TERRORIST GROUP WANT TO AVOID THE TERRORIST GROUP DOING THE CAR BOMBING?

 THE BOY ENTERED THE EXPLOSIVE-LADEN CAR.

Explanations:

THE BOY DID THE ENTER TO ENABLE THE DRIVE.

(hypothesized-in) -(node XP-INSTRUMENTAL-SCENE.802)-

 THE BOY DROVE THE EXPLOSIVE-LADEN CAR.

Explanations:

THE BOY DID THE DRIVE TO ENABLE THE CAR BOMBING.

(hypothesized-in) -(node XP-INSTRUMENTAL-SCENE.1427)-

 THE BOY WENT ON A SUICIDE BOMBING MISSION.

Explanations:

THE TERRORIST GROUP BLACKMAILED THE BOY INTO DOING THE CAR BOMBING.

(in) -(node XP-BLACKMAIL-CAR-BOMBING.1666)- (Novel explanation)

Verification questions:

DID THE TERRORIST GROUP PERSUADE THE BOY BY THREATENING TO DO THE MOP THAT WOULD RESULT IN THE STATE?

DOES THE BOY WANT TO PREVENT THE STATE?

DID THE BOY VOLUNTEER FOR THE CAR BOMBING?

Refuted explanations:

THE BOY WAS A RELIGIOUS FANATIC.
 (out) -(node XP-RELIGIOUS-FANATIC.445)-

AQUA can be asked to print out the questions raised by the story, as well as novel explanations encountered, if any. These questions and explanations are indexed in memory. They are listed here only for illustration; while reading a story, AQUA has no way to list all its questions save for walking through its entire memory. (Here, AQUA's "English" has been annotated for clarity.)

Pending interesting questions:

DID THE BOY VOLUNTEER FOR THE CAR BOMBING?

(i.e., Was the decision to perform the car bombing made out of free choice?)

DID THE TERRORIST GROUP PERSUADE THE BOY BY THREATENING TO DO THE MOP
 THAT WOULD RESULT IN THE STATE?

DOES THE BOY WANT TO PREVENT THE STATE?

(i.e., Did the terrorist group persuade the boy by threatening to do an
 action that would result in a state that the boy wanted to prevent?)

WHY IS THE BOY THE OBJECT OF THE RECRUIT?

(i.e., Why was a boy recruited for the mission?)

Pending uninteresting questions:

WHY DID THE TERRORIST GROUP RECRUIT THE BOY?

Assumed: THE TERRORIST GROUP WANTED THE CAR BOMBING DONE BUT BY SOMEONE
 ELSE.

WHY DID THE TERRORIST GROUP PLAN THE RECRUIT?

Assumed: THE TERRORIST GROUP WANTED TO DO THE RECRUIT.

WHY DID THE BOY ENTER THE EXPLOSIVE-LADEN CAR?

Assumed: THE BOY DID THE ENTER TO ENABLE THE DRIVE.

WHY DID THE BOY DRIVE THE CAR?

Assumed: THE BOY DID THE DRIVE TO ENABLE THE CAR BOMBING.

WHY DID THE TERRORIST GROUP PLAN THE CAR BOMBING?

Assumed: TERRORIST GROUPS TYPICALLY PLAN CAR BOMBINGS.

Novel explanations:

XP-BLACKMAIL-CAR-BOMBING

Indices:

Category index = XP-SACRIFICE

Situation index = CAR BOMBING

Stereotype index = STEREOTYPE.1679 (the ACTOR)

Pending questions:

DID THE ACTOR WANT TO PREVENT A STATE MORE THAN THE ACTOR
 WANTED TO PRESERVE THE LIFE STATE OF THE ACTOR?

In effect, AQUA has formulated a set of questions about this particular car bombing mission, as well as about car bombings in general, that it will try to answer when it reads car bombing stories in the future. For example, suppose AQUA reads the following story immediately after the previous one:

S-25: JERUSALEM — A young girl drove an explosive-laden car into a group of Israeli guards in Lebanon. The suicide attack killed three guards. ...

The driver was identified as a 16-year-old Lebanese girl. ... Before the attack, she said that a terrorist organization had threatened to harm her family unless she carried out the bombing mission for them. She said that she was prepared to die in order to protect her family.

As described in chapter 6, AQUA retrieves the new `xp-blackmail-suicide-bombing` and applies it to the story. This results in the XP being elaborated.

The first paragraph of the story is read in a manner similar to that illustrated by the previous story. The transcript below picks up at the point when AQUA begins to explain the motivations behind the suicide attack:

Processing story:

Built representations for:

THE YOUNG GIRL.

THE EXPLOSIVE-LADEN CAR.

THE GROUP OF ISRAELI GUARDS IN LEBANON.

THE YOUNG GIRL WENT ON A SUICIDE MISSION.

THE SUICIDE MISSION RESULTED IN THE DEATH OF THE YOUNG GIRL.

THE SUICIDE MISSION RESULTED IN THE DEATH OF THE ISRAELI GUARDS.

Processing THE SUICIDE MISSION

Running interesting heuristics

No pending questions about THE SUICIDE MISSION.

Running interesting heuristics on ACTOR role:
THE YOUNG GIRL DID THE SUICIDE MISSION.

Asking anomaly detection questions for ACTOR.2010

Was THE SUICIDE MISSION instrumental to another action?
Does THE YOUNG GIRL typically do SUICIDE MISSIONs?
Did THE YOUNG GIRL want the outcome of THE SUICIDE MISSION?

Characterizing the outcome: ((DEATH-STATE) (DEATH-STATE))
of MOP: THE CAR BOMBING
from the point of view of: THE YOUNG GIRL (the ACTOR)

DOES THE YOUNG GIRL WANT TO ACHIEVE THE DEATH OF THE ISRAELI
GUARDS?
No relevant GOALS found

DOES THE YOUNG GIRL WANT TO ACHIEVE THE DEATH OF THE YOUNG
GIRL?
Anomaly! THE YOUNG GIRL WANTS TO ACHIEVE THE DEATH OF THE
YOUNG GIRL.

Characterized outcome as a BAD outcome

Retrieved abstract XPs:

XP-SACRIFICE
XP-NOT-KNOW-OUTCOME

Retrieved stereotypical XPs:

XP-RELIGIOUS-FANATIC
XP-BLACKMAIL-SUICIDE-BOMBING

So far, the processing is similar to that done on the previous story. However, at this point, the new XP that was created during the previous story has been retrieved. When this XP is applied to the current story, AQUA discovers that there is a pending question attached to it. Since the answer to this question is not yet known, it becomes an HVQ for the new hypothesis.

Applying XP-BLACKMAIL-SUICIDE-BOMBING to THE YOUNG GIRL DID THE
SUICIDE BOMBING.

Instantiating pending questions:

DID THE YOUNG GIRL WANT TO PREVENT A STATE MORE THAN THE YOUNG GIRL WANTED TO PRESERVE THE LIFE STATE OF THE YOUNG GIRL?

Installing NODES for XP-BLACKMAIL-SUICIDE-BOMBING.2561

THE TERRORIST GROUP.
 THE MOP.
 THE STATE.
 THE TERRORIST GROUP THREATENED THE BOY.
 THE BOY WANTS TO PREVENT THE STATE.
 THE BOY DID NOT VOLUNTEER FOR THE CAR BOMBING.

Installing HVQs for XP-BLACKMAIL-SUICIDE-BOMBING.2561

DID A TERRORIST GROUP PERSUADE THE YOUNG GIRL BY THREATENING TO DO A MOP THAT WOULD RESULT IN A STATE?
 DID THE YOUNG GIRL WANT TO PREVENT A STATE MORE THAN THE YOUNG GIRL WANTED TO PRESERVE THE LIFE STATE OF THE YOUNG GIRL?
 DID THE YOUNG GIRL VOLUNTEER FOR THE SUICIDE MISSION?

Installing LINKS for XP-BLACKMAIL-SUICIDE-BOMBING.2561

Finished installing XP-BLACKMAIL-SUICIDE-BOMBING.2561

In addition, AQUA uses `xp-religious-fanatic`, which has also been retrieved, to build a religious fanatic hypothesis. However, since the blackmail hypothesis has a pending question attached to it, it is more interesting than the religious fanatic hypothesis based on the "need to learn" criterion for interestingness discussed in chapter 7. In computational terms, this means that AQUA would prefer an interpretation of the story which confirmed this hypothesis, as discussed in section 5.1.2.

To continue the example, this story goes on to say that a terrorist organization had threatened to harm her family unless she carried out the terrorist mission for them.

Answering question: DID THE YOUNG GIRL WANT TO PREVENT A STATE MORE THAN THE YOUNG GIRL WANTED TO PRESERVE THE LIFE STATE OF THE YOUNG GIRL?
 with: THE YOUNG GIRL WANTED TO PROTECT THE FAMILY OF THE YOUNG GIRL MORE THAN THE YOUNG GIRL WANTED TO PRESERVE THE LIFE STATE OF THE YOUNG GIRL.

Answering question: DID A TERRORIST GROUP PERSUADE THE YOUNG GIRL BY THREATENING TO DO A MOP THAT WOULD RESULT IN A STATE [that the girl wanted to prevent]?
 with: THE TERRORIST GROUP PERSUADED THE YOUNG GIRL BY THREATENING TO HARM HER FAMILY.

Confirming hypothesis:

THE YOUNG GIRL DID THE SUICIDE MISSION.
because THE TERRORIST GROUP BLACKMAILED THE YOUNG GIRL INTO
DOING THE SUICIDE MISSION.

Answering pending question:

DID THE ACTOR WANT TO PREVENT A STATE MORE THAN THE ACTOR
WANTED TO PRESERVE THE LIFE STATE OF THE ACTOR?

Elaborating XP-BLACKMAIL-SUICIDE-BOMBING:

Elaborating XP-ASSERTED-NODES:

THE ACTOR WANTED TO PROTECT THE FAMILY OF THE ACTOR MORE THAN
THE ACTOR WANTED TO PRESERVE THE LIFE STATE OF THE ACTOR.

Trying to explain why THE YOUNG GIRL WANTED TO PROTECT
THE FAMILY OF THE ACTOR MORE THAN THE YOUNG GIRL WANTED
TO PRESERVE THE LIFE STATE OF THE GIRL?

No explanations found

New questions for XP-BLACKMAIL-SUICIDE-BOMBING:

WHY DID THE ACTOR WANT TO PROTECT THE FAMILY OF THE ACTOR
MORE THAN THE ACTOR WANTED TO PRESERVE THE LIFE STATE OF
THE ACTOR?

Since no explanations are known for the newly added node, this in turn becomes a new question about the elaborated XP. The question is seeking a reason for the unusual goal-ordering of the actor, in which protect-family is given a higher priority than p-life.

Thus AQUA's question-driven understanding and learning process is essentially one of question transformation. AQUA starts out with questions and ends up with new questions along with answers to some of the old questions. This is shown in figure 8.9.

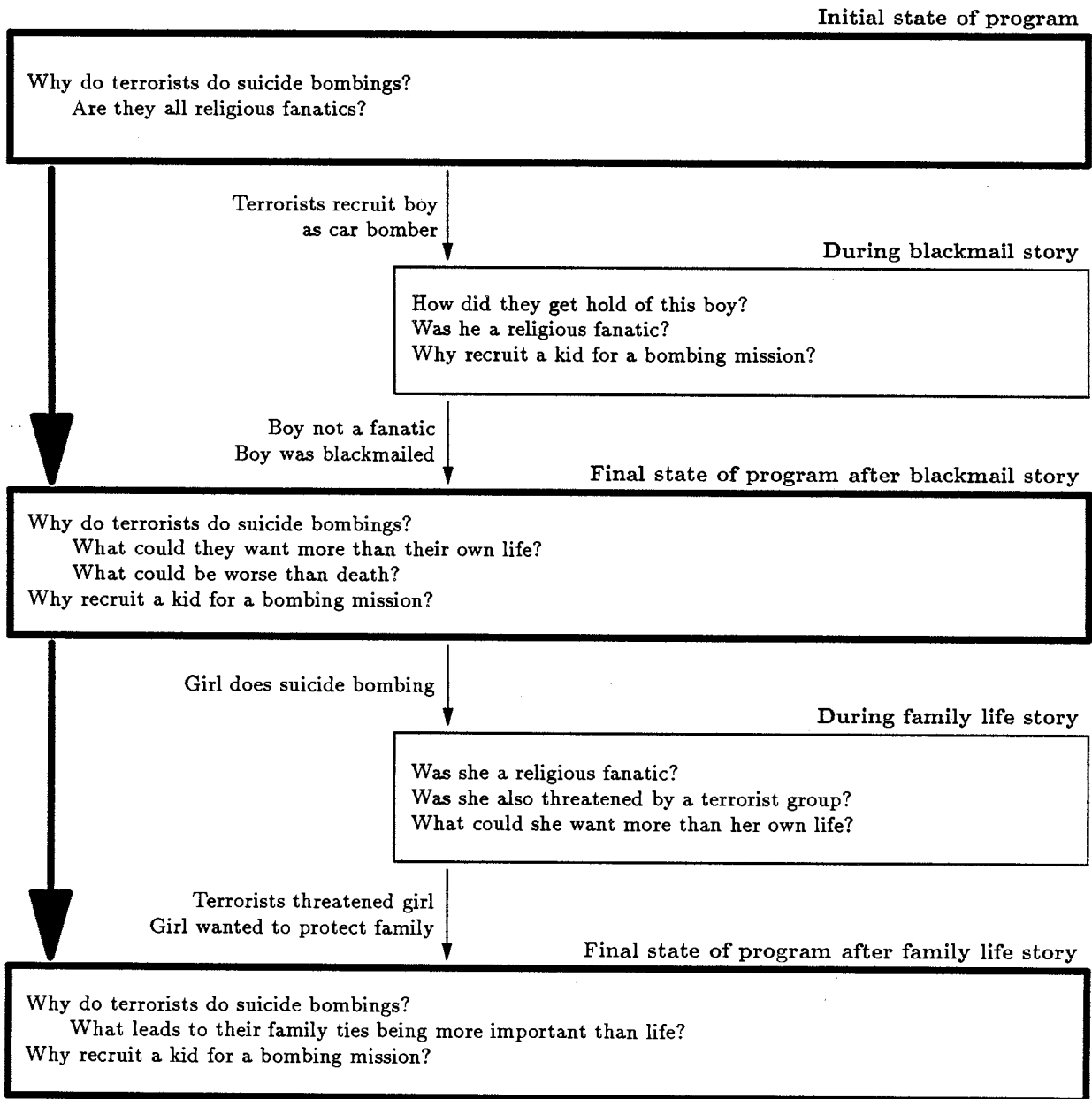


Figure 8.9: Questions in, questions out. Question-driven understanding is a process of asking questions and trying to answer them by reading a story. The understander starts out with a set of questions. At it reads, some of these questions are answered, and new questions are raised. After reading the story, the understander is left with a new set of questions that are the starting point for reading future stories.

Chapter 9

Conclusions

This dissertation presented a theory of question-driven understanding, the major claims of which are summarized below:

- Questions are subgoals of understanding tasks. The theory of questions is based on a taxonomy of understanding tasks that questions arise from. In particular, explanation questions arise from the underlying theory of explanation.
- For the purposes of understanding goal-based stories, explanation questions are based on planning/decision models of volitional explanations.
- Questions are indexed in memory.
- Questions determine what to look for in the story. They focus the understander's attention on the interesting or relevant aspects of the story, i.e., those that either answer questions or raise new ones.
- Questions may be answered opportunistically. When questions are answered, the understander resumes the reasoning task that was awaiting that piece of information.
- Questions (and therefore memory) are dynamic.
- Novel answers to questions provide an opportunity to learn, which in turn affects how the understander interprets future stories.
- Not all questions are equally interesting. The understander uses a set of heuristics to focus its attention on those questions that would allow it to improve its model of the domain.

We also presented a story understanding program, AQUA, which implements the theory of question-driven understanding. AQUA learns about terrorism in the Middle East by reading newspaper stories about terrorist incidents. The program is "interested" in terrorism in the sense that it has questions about terrorism. It tries to answer these questions by reading newspaper stories. In doing so, some of its questions are answered, and new questions are raised.

Let us discuss the range of stories that can be handled by the current implementation of AQUA, as well as the theoretical scope of this approach in general.

9.1 Current implementation of AQUA: Capabilities and limitations

Frames in AQUA's memory are organized into 11 disjoint hierarchies. Each hierarchy is headed by the "natural category" of that class of concepts. Categories are distinguished by the criterion that two concepts in different categories can never be viewed as one another or matched against one another. For example, it does not make sense to view a goal as a volitional-agent, or to view a mop as an xp.

As currently implemented, AQUA's memory consists of about 700 frames, including about:

- 15-20 abstract XPs, organized around the explanation questions discussed in section 4.1.1.
- 10 stereotypical XPs.
- 50 MOPs, most of which deal with the kinds of actions encountered in suicide bombing stories.
- 250 relations, organized as shown in figure 8.6.

In addition, there are about 20 interestingness heuristics, most of which are represented procedurally.

The range of stories that AQUA can handle is limited only by the XPs in memory. AQUA can understand several variations of 10 basic types of stories, one for each stereotypical XP that it has. For example, AQUA can understand stories about religious fanatics, depressed teenagers, Kamikazes, and so on. The basic story can be varied to include different actors, different actions, and so on, as long as these actors and actions are represented in memory. Since text-level processing has received the least attention, AQUA does not handle complicated variations in sentence construction. AQUA currently reads five or six different newspaper stories about terrorism, which have been simplified to fit within the constructions that AQUA can deal with. AQUA also reads variations on these stories, such as story S-51 below, and a few other stories that have not been taken from actual newspapers.

In addition, AQUA can also understand stories which involve one or more of its abstract XPs, such as stories about people performing actions to achieve their goals, people planning actions that they don't want to perform themselves by using other actors, and so on. Any of the 50 or so MOPs represented in memory can occur as the underlying action, and of course a story could involve more than one such action. Thus AQUA can understand a large range of basic "goal-based" stories, although it has actually been run on about ten such stories.

AQUA learns from stories in which an XP provides an explanation for an action which the XP is currently not indexed under. For example, AQUA could read a variation of the blackmail story in which a gullible teenager is tricked into performing a suicide mission without knowing its outcome:

S-51: Terrorists recruit boy as car bomber.

A 16-year-old Lebanese got into an explosive-laden car and went on a suicide bombing mission to blow up the Israeli army headquarters in Lebanon. ...

The teenager was a Shiite Moslem but not a religious fanatic. He thought he was being recruited as a limousine driver. He did not know that the deadly mission would result in his own death.

This story is initially processed in the same manner as the blackmail story. However, in this case it is *xp-not-know-outcome* that is confirmed, resulting in a new XP being learned under the *suicide-bombing* situation index.

The obvious limitation of the program as currently implemented is the limited range of text-level processing it can handle. AQUA does not learn new MOPs, nor does it learn new XPs that are not incremental variations on old ones that it already knows. Thus the MOPs and XPs in memory also provide a constraint on the stories that can be processed. Although the abstract XPs that have actually been implemented are fairly complete for the kinds of motivational stories that AQUA is designed to deal with, additional XPs for planner-actor relationships would be required to understand stories with complex goal interactions. The motivational aspects of most of the suicide bombing stories that have appeared in the newspapers over the past several years seem to fall within variations of the religious fanatic and coercion themes, but AQUA would need several more stereotypical XPs and MOPs in order to read a wider range of terrorism or other stories.

Theoretical (as opposed to implementational) limitations of the program are discussed in the next section.

9.2 Boundary conditions: Capabilities and limitations

To illustrate the scope of question-driven understanding, let us consider AQUA would handle the following types of stories. These stories can be viewed as "boundary conditions" for the program, since they represent interesting borderline cases with which the theory can be tested.

9.2.1 No stereotypical XP in memory

If a story presents a situation for which no stereotypical explanation pattern is known, AQUA can not formulate a good explanation that is specific to that type of situation. However, AQUA can still use its abstract XPs to form a general explanation, based on the type of anomaly that occurred.

For example, suppose AQUA reads a story about a suicide bombing in Ireland without any XPs for suicide bombing in memory. Based on the goal violation anomaly, AQUA builds a general hypothesis such as "The bomber wanted to destroy the target more than he wanted to preserve his own life." In the absence of a more specific explanation, this is the best that AQUA could hope to do. This hypothesis constitutes a reasonably good understanding of the story.

In addition, if the story provides a specific explanation for the novel situation, AQUA learns a new stereotypical explanation and indexes it in memory. The blackmail story is an example of this. The new explanation is only as good as the story that provided it, of course, but it will be refined incrementally as it gets used in different situations.

9.2.2 Story too bizarre

Although AQUA can understand novel stories, stories that are too deviant fall outside its range. As discussed in chapter 7, small deviations from expectations are optimal in creating interest. A story

that is too bizarre, in the sense that it does not relate to anything that the program knows about, is difficult to understand precisely because the program does not have the knowledge structures to even begin to process the story. Since the story does not raise questions that are within the scope of AQUA's knowledge structures, AQUA has no reason to be interested in the story.

AQUA learns through incremental modification of its XPs. Thus in order to learn from a novel story, it should be possible to understand these novel aspects using the kinds of modifications that AQUA is capable of performing. A story that is too deviant would be processed only superficially since it would be about concepts that can't be related what AQUA already knows.

9.2.3 Story fits perfectly

At the other extreme, a story that fits perfectly within existing knowledge structures is easy to understand. But by the same token, such a story is not very interesting since it does not say anything new. When AQUA reads a story that fits well with what it already knows, no new questions are raised. The processing questions that arise are easily answered. Although AQUA can read these stories, therefore, it will not have learned anything new as a result, nor would it have asked a new question.

9.2.4 Questions arising from other goals

As discussed in chapter 1, AQUA has one underlying goal: to learn and improve its model of its domain. This gives rise to intellectual or cognitive goals, which are knowledge goals or questions that then drive the understanding process. These questions determine AQUA's interests; AQUA is interested in certain aspects of terrorism by virtue of having questions about these aspects.

However, one can be interested in something by virtue of other goals that are outside the scope of AQUA's goals. AQUA's questions do not model knowledge goals that arise from other extrinsic goals that AQUA does not have. There would be no reason for AQUA to be interested in stories relating to goals that it does not have.

9.2.5 How many questions does AQUA ask?

Finally, let us consider the issue of how AQUA's questions change as it reads several stories about suicide bombing. Since AQUA has more knowledge about the domain, we would expect it to be able to ask more questions about a new story that it reads. On the other hand, fewer stories would be novel since AQUA would already know a lot about that domain, and so fewer new questions would be raised.

In other words, as AQUA gets more "expert" in its domain, most common stories that it sees would fit in fairly well with what it already knows, and so these stories are not very interesting to it; they raise very few questions. On the other hand, AQUA can ask more and better questions about stories that are novel. Furthermore, the unanswered questions that are pending in AQUA's memory would be more sophisticated than the ones that it started out with, in the sense that they would reflect a better understanding of the domain.

9.3 Conclusion: "Goal-based programs"

AI programs intended as cognitive models should be "goal-based," in the sense that they should be able to represent and reason about their own "cognitive goals." These goals arise from the basic tasks of the program, such as understanding, planning, education, etc., and are indexed in the program's memory. Processing (understanding, planning, etc.) is then viewed as the opportunistic pursuit of these goals. Furthermore, since goals are dynamic, the process theory must deal with the issues of learning and change.

In this dissertation, I have tried to demonstrate what this would mean in the case of an understanding system. Of course, most programs do have "goals" in the sense their processes are trying to achieve some cognitive task. The point is that these goals should be available for reasoning to the system itself. The system should be able to take these goals into account when making processing decisions. AQUA is an attempt to improve the kinds of reasoning that an understander can perform with its questions, so that questions begin to be less like predictions and more like real goals.

The advantages of this view have been discussed throughout the dissertation, and can be summarized as follows:

- Cognitive modelling: Since people have goals and interests, it makes sense to model them explicitly since these goals and interests play a central role in cognitive processes. In fact, they underlie the basic reasons why people understand, plan, and so on in the first place.
- Functional advantages: The goals of the system can be used to focus attention on interesting or relevant facts, and limit the inferences drawn from them. Furthermore, the explicit representation of program goals facilitates the discovery of unexpected opportunities to satisfy these goals.
- System design: This model provides a nice framework in which to integrate other cognitive processes that could be used as plans to pursue knowledge goals, such as case-based reasoning, analogy or tweaking. Thus this is a clean implementation of a system within integrated processing paradigm.

Finally, as an extension, this framework allows us to address the issue of "knowledge planning." If a system does have different plans at its disposal, it can try to plan for its knowledge goals. For example, in order to satisfy a knowledge goal, it might select the best inference strategy to use at any given time, or decide to read the newspaper to find the information it needs, or even ask the user. Or it might simply discard the knowledge goal and not bother to satisfy it. Ultimately, the decision to perform a piece of reasoning should be made actively by the program instead of being hardwired by the programmer.

Research in this paradigm would then consist of identifying a set of processing decisions, and criteria for making these decisions along with the kinds of knowledge required by these criteria. This is not very different from what researchers are doing now; the difference is that these criteria would be available to the program itself instead of existing only in the minds of the researchers. The theory of questions and question-driven understanding is a step towards this goal.

Bibliography

- [Abelson, 1973] R. P. Abelson. The Structure of Belief Systems. In R. C. Schank and K. Colby, editors, *Computer Models of Language and Thought*, W. H. Freeman and Co., San Francisco, 1973.
- [Abelson and Carroll, 1965] R. P. Abelson and J. D. Carroll. Computer Simulation of Individual Belief Systems. *American Behavioral Scientist*, (8):24-30, 1965.
- [Adelson, 1988] B. Adelson. Purpose Guided Analogical Learning and Reasoning. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, pages 300-301, Cognitive Science Society, Montreal, Canada, August 1988.
- [Alterman, 1986] R. Alterman. An Adaptive Planner. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 65-69, AAAI, Philadelphia, PA, August 1986.
- [Anderson, 1982] R. C. Anderson. Allocation of Attention During Reading. In A. Flammer and W. Kintsch, editors, *Discourse Processing*, pages 292-313, North Holland Publishing Company, Amsterdam, 1982.
- [Austin, 1962] J. L. Austin. *How to Do Things with Words*. Harvard University Press, Cambridge, MA, 1962.
- [Bain, 1986] W. M. Bain. *Case-based Reasoning: A Computer Model of Subjective Assessment*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, 1986. Research Report #470.
- [Baird and Hidi, 1985] W. Baird and S. Hidi. The Effect of Factual Importance and Interestingness on Recall from Naturally-Occurring School Texts. In *Annual Meeting of the American Educational Research Association*, New Orleans, LA, 1985.
- [Birnbaum, 1986] L. Birnbaum. *Integrated Processing in Planning and Understanding*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, December 1986. Research Report #489.
- [Birnbaum and Collins, 1984] L. Birnbaum and G. Collins. Opportunistic Planning and Freudian Slips. In *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Cognitive Science Society, Boulder, CO, 1984.
- [Birnbaum and Selfridge, 1979] L. Birnbaum and M. Selfridge. *Problems in Conceptual Analysis of Natural Language*. Technical Report 168, Yale University, Department of Computer Science, New Haven, CT, October 1979.
- [Bobrow *et al.*, 1977] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd. GUS, A Frame Driven Dialog System. *Artificial Intelligence*, 8(1), 1977.
- [Brewer, 1983] W. F. Brewer. Form, Content, and Affect in the Theory of Stories. *The Behavioral and Brain Sciences*, 6:595-596, 1983.

- [Brewer and Lichtenstein, 1982] W. F. Brewer and E. H. Lichtenstein. A Structural Affect Theory of Stories. *Journal of Pragmatics*, 6:473-486, 1982.
- [Brown and Burton, 1975] J. S. Brown and R. R. Burton. Multiple Representations of Knowledge for Tutorial Reasoning. In D. G. Bobrow and A. M. Collins, editors, *Representation and Understanding*, Academic Press, New York, 1975.
- [Burstein, 1988] M. Burstein. Analogical Explanations Combining Inconsistent Mental Models. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, pages 301-302, Cognitive Science Society, Montreal, Canada, August 1988.
- [Carbonell, 1979] J. G. Carbonell. *Subjective Understanding: Computer Models of Belief Systems*. Ph.D. thesis, Yale University, New Haven, CT, January 1979. Research Report #150.
- [Charniak, 1977a] E. Charniak. A Framed PAINTING: The Representation of a Common Sense Knowledge Fragment. *Cognitive Science*, 1(4):355-394, 1977.
- [Charniak, 1977b] E. Charniak. Ms. Malaprop: A Language Comprehension Program. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, IJCAI, Cambridge, MA, August 1977.
- [Charniak, 1983] E. Charniak. Passing Markers: A Theory of Contextual Influence in Language Comprehension. *Cognitive Science*, 7(3), 1983.
- [Charniak, 1986] E. Charniak. A Neat Theory of Marker Passing. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 584-588, AAAI, Philadelphia, PA, August 1986.
- [Charniak, 1988] E. Charniak. Story Comprehension, Abductive Unification and Non-Monotonic Equality. *Artificial Intelligence*, 34(3), April 1988.
- [Colby, 1973] K. Colby. Simulations of Belief Systems. In R. C. Schank and K. Colby, editors, *Computer Models of Language and Thought*, W. H. Freeman and Co., San Francisco, 1973.
- [Cullingford, 1978] R. Cullingford. *Script Application: Computer Understanding of Newspaper Stories*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, 1978. Research Report #116.
- [de Beaugrande, 1982] R. de Beaugrande. The Story of Grammars and the Grammar of Stories. *Journal of Pragmatics*, 6:383-422, 1982.
- [Dehn, 1989] N. Dehn. *Computer Story Writing: The Role of Reconstructive and Dynamic Memory*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, 1989.
- [DeJong, 1979] G. F. DeJong. *Skimming Stories in Real Time: An Experiment in Integrated Understanding*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, May 1979. Research Report #158.
- [DeJong, 1983] G. F. DeJong. An Approach to Learning from Observation. In *Proceedings of the International Machine Learning Workshop*, pages 171-176, University of Illinois at Urbana-Champaign, Monticello, IL, June 1983.
- [DeJong and Mooney, 1986] G. F. DeJong and R. J. Mooney. Explanation-Based Learning: An Alternative View. *Machine Learning*, 1(2):145-176, 1986.
- [Diettrich and Michalski, 1981] T. G. Diettrich and R. S. Michalski. Inductive Learning of Structural Descriptions: Evaluation Criteria and Comparative Review of Selected Methodologies. *Artificial Intelligence*, 16:257-294, 1981.
- [Domeshek, 1988] E. A. Domeshek. Understanding Stories in Their Social Context. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, Cognitive Science Society, Montreal, Canada, August 1988.

- [Doyle, 1979] J. Doyle. A Truth Maintenance System. *Artificial Intelligence*, 12:231-272, 1979.
- [Dyer, 1982] M. G. Dyer. *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, May 1982. Research Report #116.
- [Falkenheimer-87, 1987] B. Falkenheimer-87. Scientific Theory Formation through Analogical Inference. In *Proceedings of the Fourth International Workshop on Machine Learning*, Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1987.
- [Fodor, 1983] J. A. Fodor. *The Modularity of Mind*. The MIT Press, Cambridge, MA, 1983.
- [Fodor, 1985] J. A. Fodor. Precis of The Modularity of Mind. *The Behavioral and Brain Sciences*, 8:1-42, 1985.
- [Fodor et al., 1974] J. A. Fodor, T. Bever, and M. Garrett. *The Psychology of Language: An Introduction to Psycholinguistics and Generative Grammar*. McGraw-Hill, New York, NY, 1974.
- [Goldstein and Roberts, 1977] I. Goldstein and R. Roberts. NUDGE: A Knowledge-based Scheduling Program. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, IJCAI, Cambridge, MA, August 1977.
- [Granger et al., 1986] R. H. Granger, K. P. Eiselt, and J. K. Holbrook. Parsing with Parallelism: A Spreading-Activation Model of Inference Processing During Text Understanding. In J. L. Kolodner and C. K. Riesbeck, editors, *Experience, Memory and Reasoning*, chapter 14, pages 227-246, Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.
- [Hammond, 1986] K. J. Hammond. *Case-Based Planning: An Integrated Theory of Planning, Learning and Memory*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, October 1986. Research Report #488.
- [Hammond, 1988] K. J. Hammond. Opportunistic Memory: Storing and Recalling Suspended Goals. In J. L. Kolodner, editor, *Proceedings of a Workshop on Case-Based Reasoning*, pages 154-168, Defense Advanced Research Projects Agency, Morgan Kaufmann, Inc., Clearwater Beach, FL, May 1988.
- [Hayes-Roth and Hayes-Roth, 1979] B. Hayes-Roth and F. Hayes-Roth. A Cognitive Model of Planning. *Cognitive Science*, 2:275-310, 1979.
- [Hayes-Roth and Lesser, 1976] F. Hayes-Roth and V. Lesser. Focus of attention in a distributed logic speech understanding system. In *Proceedings of the IEEE International Conference on ASSP*, Philadelphia, PA, 1976.
- [Heider, 1958] F. Heider. *The Psychology of Interpersonal Relations*. Volume XV of *Current Theory and Research in Motivation*, John Wiley and Sons, New York, 1958.
- [Hidi and Baird, 1986] S. Hidi and W. Baird. Interestingness — A Neglected Variable in Discourse Processing. *Cognitive Science*, 10:179-194, 1986.
- [Hidi et al., 1982] S. Hidi, W. Baird, and A. Hildyard. That's Important, But Is It Interesting? Two Factors in Text Processing. In A. Flammer and W. Kintsch, editors, *Discourse Processing*, pages 63-75, North Holland Publishing Company, Amsterdam, 1982.
- [Hunter, 1989] L. E. Hunter. *Knowledge Acquisition Planning: Gaining Expertise Through Experience*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, January 1989. Research Report #678.
- [Hunter and Silbert, 1987] L. E. Hunter and J. A. Silbert. Progress Report on IVY: A Learning System for Intelligent Information Retrieval in Pathology. In *Proceedings of the Artificial Intelligence in Medicine Workshop*, Seattle, WA, July 1987.

- [Kass, in preparation] A. Kass. *Developing Creative Hypotheses by Adapting Explanations*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, in preparation.
- [Kass and Owens, 1988] A. Kass and C. Owens. Learning New Explanations by Incremental Adaptation. In *Proceedings of the 1988 AAAI Spring Symposium on Explanation-Based Learning*, AAAI, 1988.
- [Kass et al., 1986] A. Kass, D. Leake, and C. Owens. *SWALE: A Program That Explains*, pages 232-254. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.
- [Kelley, 1967] H. H. Kelley. Attribution Theory in Social Psychology. In D. Levine, editor, *Nebraska Symposium on Motivation*, pages 192-238, University of Nebraska Press, Lincoln, 1967.
- [Kintsch, 1980] W. Kintsch. Learning from Text, Levels of Comprehension, or: Why Would Anyone Read a Story Anyway? *Poetics*, 9:87-98, 1980.
- [Kolodner, 1980] J. L. Kolodner. *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, November 1980. Research Report #187.
- [Kolodner, 1988] J. L. Kolodner, editor. *Proceedings of a Workshop on Case-Based Reasoning*. Morgan Kaufmann, Inc., Clearwater Beach, FL, May 1988.
- [Kolodner et al., 1985] J. L. Kolodner, R. L. Simpson, and K. Sycara. A Process Model of Case-Based Reasoning in Problem Solving. In A. Joshi, editor, *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 284-290, IJCAI, Los Angeles, CA, August 1985.
- [Leake, in preparation] D. Leake. *Evaluating Explanations*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, in preparation.
- [Leake and Owens, 1986] D. Leake and C. Owens. Organizing Memory for Explanation. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Cognitive Science Society, Amherst, MA, August 1986.
- [Lebowitz, 1980] M. Lebowitz. *Generalization and Memory in an Integrated Understanding System*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, October 1980. Research Report #186.
- [Lehnert, 1978] W. G. Lehnert. *The Process of Question Answering*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1978.
- [Luftig and Greeson, 1983] R. L. Luftig and L. E. Greeson. Effects of Structural Importance and Idea Saliency on Discourse Recall of Mentally Retarded and Nonretarded Pupils. *American Journal of Mental Deficiency*, 87:414-421, 1983.
- [Lytinen, 1984] S. L. Lytinen. *The Organization of Knowledge in a Multi-lingual Integrated Parser*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, November 1984. Research Report #340.
- [Marcus, 1980] M. Marcus. *A Theory of Syntactic Recognition for Natural Language*. MIT Press, Cambridge, MA, 1980.
- [Martin, in preparation] C. E. Martin. *Direct Memory Access Parsing*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, in preparation.
- [McDermott, 1974] D. V. McDermott. *Assimilation of New Information by a Natural Language-Understanding System*. Master's thesis, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, MA, 1974. Technical Report 291.

- [McDermott, 1981] D. V. McDermott. Artificial Intelligence Meets Natural Stupidity. In J. Hauge-land, editor, *Mind Design*, MIT Press, Bradford Books, Cambridge, MA, 1981.
- [Minsky, 1975] M. Minsky. A Framework for Representing Knowledge. In P. H. Winston, editor, *The Psychology of Computer Vision*, chapter 6, pages 211-277, McGraw-Hill, New York, 1975.
- [Mitchell *et al.*, 1986] T. M. Mitchell, R. Keller, and S. Kedar-Cabelli. Explanation-Based Generalization: A Unifying View. *Machine Learning*, 1(1):47-80, 1986.
- [Mooney and DeJong, 1985] R. J. Mooney and G. F. DeJong. Learning Schemata for Natural Language Processing. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 681-687, IJCAI, Los Angeles, CA, August 1985.
- [Mostow and Bhatnagar, 1983] J. Mostow and N. Bhatnagar. FAILSAFE - A Floor Planner that uses EBG to Learn from its Failures. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 249-255, IJCAI, Milan, Italy, August 1983.
- [Norvig, 1987] P. Norvig. Inference in Text Understanding. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 561-565, AAAI, Seattle, WA, July 1987.
- [Pazzani, 1988] M. J. Pazzani. Selecting the Best Explanation for Explanation-Based Learning. In *1988 Spring Symposium Series: Explanation-Based Learning*, pages 165-169, AAAI, Stanford, CA, 1988.
- [Ram, 1984] A. Ram. *Modelling Characters and their Decisions: A Theory of Compliance Decisions*. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL, August 1984. Technical Report T-145.
- [Ram, 1987] A. Ram. AQUA: Asking Questions and Understanding Answers. In *Proceedings of the Sixth Annual National Conference on Artificial Intelligence*, pages 312-316, American Association for Artificial Intelligence, Morgan Kaufman Publishers, Inc., Seattle, WA, July 1987.
- [Ram, 1989] A. Ram. Incremental learning of paradigmatic cases. In *Proceedings of the Second Workshop on Case-Based Reasoning*, Defense Advanced Research Projects Agency, Pensacola Beach, FL, May 1989.
- [Ranney and Thagard, 1988] M. Ranney and P. Thagard. Explanatory Coherence and Belief Revision in Naive Physics. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, pages 426-432, Cognitive Science Society, Montreal, Canada, August 1988.
- [Rees *et al.*, 1984] J. Rees, N. Adams, and J. Meehan. *The T Manual*. Yale University, Department of Computer Science, New Haven, CT, fourth edition, 1984.
- [Rieger, 1975] C. Rieger. Conceptual Memory and Inference. In R. C. Schank, editor, *Conceptual Information Processing*, North-Holland, Amsterdam, 1975.
- [Riesbeck, 1975] C. K. Riesbeck. Conceptual Analysis. In R. C. Schank, editor, *Conceptual Information Processing*, North-Holland, Amsterdam, 1975.
- [Riesbeck and Martin, 1985] C. K. Riesbeck and C. E. Martin. *Direct Memory Access Parsing*. Technical Report 354, Yale University, Department of Computer Science, New Haven, CT, January 1985.
- [Schank, 1972] R. C. Schank. Conceptual Dependency: A Theory of Natural Language Understanding. *Cognitive Psychology*, 3(4):552-631, 1972.
- [Schank, 1979] R. C. Schank. Interestingness: Controlling Inferences. *Artificial Intelligence*, 12:273-297, 1979.
- [Schank, 1982] R. C. Schank. *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge University Press, 1982.

- [Schank, 1986] R. C. Schank. *Explanation Patterns: Understanding Mechanically and Creatively*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.
- [Schank and Abelson, 1977] R. C. Schank and R. Abelson. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.
- [Schank and Birnbaum, 1984] R. C. Schank and L. Birnbaum. Memory, Meaning and Syntax. In T. Bever, J. Carroll, and L. Miller, editors, *Talking Minds: The Study of Language in Cognitive Science*, pages 209-251, MIT Press, Cambridge, MA, 1984.
- [Schank and Ram, 1988] R. C. Schank and A. Ram. Question-driven Parsing: A New Approach to Natural Language Understanding. *Journal of Japanese Society for Artificial Intelligence*, 3(3):260-270, May 1988.
- [Schank et al., 1986] R. C. Schank, G. Collins, and L. Hunter. Transcending Inductive Category Formation in Learning. *The Behavioral and Brain Sciences*, 9(4), 1986.
- [Schank et al., 1978] R. C. Schank, M. Lebowitz, and L. Birnbaum. *Integrated Partial Parsing*. Technical Report 143, Yale University, Department of Computer Science, New Haven, CT, 1978.
- [Schmidt et al., 1978] C. Schmidt, N. Sridharan, and J. Goodson. The Plan Recognition Problem. *Artificial Intelligence*, 11(1,2):45-83, 1978.
- [Searle, 1969] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.
- [Segre, 1987] A. M. Segre. *Explanation-Based Learning of Generalized Robot Assembly Tasks*. Ph.D. thesis, University of Illinois at Urbana-Champaign, Urbana, IL, January 1987. Technical Report UILU-ENG-87-2208.
- [Simpson, 1985] R. L. Simpson. *A Computer Model of Case-based Reasoning in Problem-solving: An Investigation in the Domain of Dispute Mediation*. Ph.D. thesis, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, GA, 1985.
- [Sperber and Wilson, 1986] D. Sperber and D. Wilson. *Relevance: Communication and Cognition. Language and Thought Series*, Harvard University Press, Cambridge, MA, 1986.
- [Sussman, 1975] G. J. Sussman. *A Computer Model of Skill Acquisition*. Volume 1 of *Artificial Intelligence Series*, American Elsevier, New York, 1975.
- [Sycara, 1987] E. P. Sycara. *Resolving Adversarial Conflicts: An Approach Integrating Case-based and Analytic Methods*. Ph.D. thesis, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, GA, 1987.
- [Turner, 1986] S. Turner. Using Preconditions in a Case-Based Advisory System. In *Proceedings of the IEEE 1986 International Conference on Systems, Man and Cybernetics*, IEEE, Atlanta, GA, 1986.
- [Wilensky, 1978] R. Wilensky. *Understanding Goal-Based Stories*. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, 1978.
- [Wilensky, 1980] R. Wilensky. *Points: A Theory of Story Content*. Research Report UCB/ERL/M80/17, Berkeley Electronic Research Laboratory, Berkeley, CA, 1980.
- [Wilensky, 1983] R. Wilensky. *Planning and Understanding*. Addison-Wesley, Reading, MA, 1983.
- [Wilensky, 1986] R. Wilensky. Knowledge Representation — A Critique and A Proposal. In J. L. Kolodner and C. K. Riesbeck, editors, *Experience, Memory and Reasoning*, chapter 2, pages 15-28, Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

- [Wilks, 1973] Y. Wilks. An Artificial Intelligence Approach to Machine Translation. In R. C. Schank and K. Colby, editors, *Computer Models of Language and Thought*, W. H. Freeman and Co., San Francisco, 1973.
- [Wilks, 1977] Y. Wilks. What Sort of Taxonomy of Causation Do We Need for Language Understanding. *Cognitive Science*, 1:235, 1977.
- [Winograd, 1972] T. Winograd. *Understanding Natural Language*. Academic Press, New York, 1972.
- [Woods *et al.*, 1972] W. A. Woods, R. M. Kaplan, and B. Nash-Webber. *The Lunar Sciences Natural Language Information System: Final Report*. BBN Report 2378, Bolt, Beranek and Newman, Cambridge, MA, 1972.