NORTHWESTERN UNIVERSITY

The Affective Reasoner: A process model of emotions in a multi-agent system.

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY
Field of Computer Science

By
Clark Davidson Elliott

EVANSTON, ILLINOIS
June 1992

This research was supported in part by the Defense Advanced Research Projects Agency, monitored by the Air Force Office of Scientific Research under contract F49620-88-C-0058 and the Office of Naval Research under contract N00014-90-J-4117, by the Office of Naval Research under contract N00014-89-J-1987, and by the Air Force Office of Scientific Research under contract AFOSR-89-0493. This work was also supported in part by grant IRI-8812699 awarded to Andrew Ortony by the National Science Foundation. The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting. The Institute receives additional support from Ameritech, an Institute Partner, and from IBM.

ABSTRACT

The problem we have addressed in this dissertation is that of designing a pragmatic and rich computer representation of emotions that is at least congenial with psychological theory. Our solution has focused on implementing a platform for reasoning about emotions that supports the testing of such theories.

In the platform we model a multi-agent world and give simple affective life to agents in the form of rudimentary emotions and emotion-induced actions. In addition the agents are able to reason about emotion episodes that take place in one another's lives. The implementation includes representations for twenty-four emotion types (based on the work of Ortony et al., 1988) and 1400 emotion-induced actions. Agents have rudimentary personalities, including an *interpretive* component which causes them to construe the world in idiosyncratic ways leading to emotional states, and an *expressive* component which gives agents a unique profile for manifesting their emotions. Agents keep internal models of the concerns of other agents which allow them to explain the emotional episodes of other agents by classifying them as instances in which one or more of the twenty-four emotion types arise.

The implementation is simulation-based. It has been run with up to forty agents at a time. Situations arise in the modeled world, agents respond to some of these in their own unique, emotional, ways. Emotion-induced actions are generated which may be placed back in the simulation queue and further perturb the system. Other agents observe and explain the situations using both strong-theory reasoning based on a set of emotion rules, and weak-theory reasoning using a case-based heuristic classification system.

Acknowledgments

First and foremost I am indebted to my wife Jane for her unending support in seeing our growing family through this long process, and to her parents, Hideaki and Kiyoko Arao, for their daily assistance, and for their care and understanding.

To Professor Andrew Ortony I am deeply indebted for the faith he has placed in me as a researcher, for supporting me at the Institute for the Learning Sciences, and for teaching me to be a scholar. In addition I owe him a debt for the painstaking effort he has given in working with me on this dissertation and on our collaborative papers.

To Professor Lawrence Henschen I am indebted for the careful guidance he provided from my first days at Northwestern through graduation, for the strong formal background he gave me in AI, and for fostering a scholarly atmosphere among his many students.

At the Institute I am indebted to professors Lawrence Birnbaum for continually challenging me to understand new perspectives, Gregg Collins for his ability to make clear the most confusing problems in the shortest time, and Chris Riesbeck for teaching me incantations in LISP and for providing continual feedback about my work and about the state of AI. In addition, I wish to thank Alex Kass and Ray Bareiss.

I am also indebted to my colleagues at ILS, among them Eva Gilboa – my officemate, Peter Prokopowicz, Robin Burke, Mike Freed, Lucian Hughes and Will Fitzgerald. Unnamed, but not forgotten, are those ILS students who were here before me and helped me to learn by example, my classmates with whom I have had so many long and interesting discussions, and the newer ILS students with whom I have established ties.

In the Electrical Engineering and Computer Science department I am indebted to my friends Jung Kim, James Lu, Monica Barback and Hugh Devlin for their continued support.

I am deeply grateful to my colleagues at DePaul University who first taught me about AI and who have since supported me in so many ways. In particular I wish to thank Helmut Epp, who had faith in me from the start, Gary Andrus who has been a steadfast friend, and Henry Harr who was there at the beginning. Special thanks are due Joseph Casey who first started me on the path to scholarship. Among the others are Hon Wing Cheng, Gerald Gordon, Marty Kalin, George Knafl, David Miller, Rosalie Nerheim, Ed Pudlo, Steve Samuels and Jacek Witaszek. Heartfelt thanks are due Nicholas Lavrov, the best teacher I have ever known.

Lastly I want to thank my parents for their support and for always encouraging me to ask, *"Why?"*

To Jane, Nell and Peter.

Contents

List of Figures

Chapter 1

Introduction

1.1 The nature of the problem

Emotions are central to human motivation: they are both the precursors to, and an end result of, many undertakings. They glue society together. They feature prominently in the relationships we have, the stories we tell, and the plans we make. Why has so little been done to create working, experimental, models of emotion reasoning?

To answer this we must look at the nature of the problem. First, in one sense everyone *knows* what an emotion is, but few would venture a definition. Those that did would disagree. To quote Reber, "**emotion:** *Historically this term has proven utterly refractory to definitional efforts; probably no other term in psychology shares its nondefinability with its frequency of use"* [Reber, 1985]. Second, we must consider that any full treatment of emotions must consider biology: emotions are clearly something that we *feel.* Separating this out from the cognitive aspects we wish to model is a difficult task. Third, consider the stimuli for emotions, and how complex they are: How do we map a simple act, such as paying money, into a state of *joy*, or of *anger, fear, pride, pity,* etc? Here the meaning of the act is not one of a transaction having occurred, but rather one of the relevance of the event to a whole range of unseen goals, standards and preferences of the interpreting agent [Ortony *et al.*, 1988; Reeves, 1991; Bain, 1986]. Fourth, emotions are highly personal in nature: one man's meat is another man's poison. Emotions have little to do with fact and everything to do with interpretation. Lastly consider observability: We have only small clouded windows into the emotions actually being experienced, even our own. Facial expressions, body cues, inflection changes, choices of words, the subjective interpretation of behavior, and so forth are all we have to go on. The list of difficulties for even the most basic academic treatment is seemingly endless. So, we have an area of our science that on the one hand embraces almost all aspects of human-like reasoning but that on the other fails to accept even the most elementary categorization. Is it any wonder that few models exist?

A premise of this work is that emotions have much to do with intelligent reasoning, so that emotion modeling is of interest both from a generative standpoint, when intelligence is modeled, and from an understanding standpoint, when interaction with the human world is required. Such modeling will pay dividends when it is integrated into other systems such as multi-agent reasoning systems [Lesser, 1992; Elliott *et al.*, 1992], automated tutors [Van Lehn, 1988], language understanding programs [Reeves, 1991], software interfaces for computer supported cooperative work [Tatar, 1990], and so forth. In addition, the modeling platform built for this work can be used for testing psychological models: can theories be shown to produce results within the constraints of the real world [Rollenhagen and Dalkvist, 1989; Frijda and Swagerman, 1987]?

One way to explore emotion reasoning is by simulating a world and populating it with agents capable of participating in emotional episodes. This is the approach we have taken. For this to be useful we must have (1) a simulated world which is rich enough to test the many subtle variations a treatment of emotion reasoning requires, (2) agents capable of (a) a wide range of affective states, (b) an interesting array of interpretations of situations leading to those states and (c) a reasonable set of reactions to those states, (3) a way to capture a theory of emotions, and (4) a way for agents to interact and to reason about the affective states of one another. The Affective Reasoner supports these requirements.

Suppose, for example, that we take as our domain the world of some taxi drivers. We would want to be able to model emotions for these taxi drivers, and we would want them to be able to reason about the emotions that other drivers are having. First we need some taxi driving situations to arise. Next we need to have interpretations of these situations by the drivers give rise to emotions. Finally, we need action sets which allow the drivers to express those emotions. If we can do this, the stage will have been set for giving emotional life to our agents. Once this is done we can add a mechanism for reasoning about these emotional lives, preferably a mechanism that learns. Situations arise; the taxi drivers have emotional reactions to those situations; they reason about each others' emotions and about the expressions of those emotions.

This, in essence, is what the *TaxiWorld* version of the Affective Reasoner does: A simulation runs. Multiple agents[1] have emotional lives in response to each other and to the situations that arise in the simulated world. They express emotions from twenty-four different categories as a series of actions, which are in turn represented as new events in the world. Agents have personalities. Agents reason about and learn about each other. The system is rooted both in information processing psychology and in traditional Artificial Intelligence. Representations are symbolic, and logical reasoning plays a role in the strong-theory aspects of interpretation, while case-based

[1] *TaxiWorld* has been run with up to forty agents at a time.

reasoning is used to heuristically classify the weak-theory aspects of emotional expression. Within this scheme emotions are *cognitive appraisals of instigating stimuli in the closed "world" of the system, which may give rise to motivated actions.* They are, in the present incarnation of the Affective Reasoner, acute, short-lived states with immediate reactions.

1.2 Importance of the problem

Asking what applications there are for reasoning about the human emotion mechanism is somewhat akin to asking what applications there are for human commonsense reasoning. The more interesting problem is in finding appropriate applications for the limited emotion representation we have achieved, and the reverse, setting goals for our limited emotion representation based on what we would *like* to achieve.

Emotions as communication filters

As software design becomes more complex, and as reasoning/utility modules become more autonomous, we inevitably find ourselves having to coordinate a *society* of functional and descriptive units. But this often means that the proportion of time each module spends processing in the problem domain relative to the time spent processing communications with other modules decreases significantly. Another way to look at this is as the age old problem so well described by Brooks in *The Mythical Man Month* [Brooks, 1975]: As the complexity of the problems we tackle increases, an ever-greater burden is placed on our ability to let individual modules perform their tasks autonomously without adverse effects on the group. At some point the cost in communicating with other modules eclipses the cost of performing the work itself. Eventually, the cost of communicating becomes so great that little or no useful work can be performed. As computer-assisted environments scale up, this will become a central issue in AI. [Schank, 1992]. To quote Oatley:

> It seems certain that, as we understand more about cognition, we will need to explore autonomous systems with limited resources that nevertheless cope successfully with multiple goals, uncertainty about the environment, and co-ordination with other agents. In mammals, these cognitive design problems seem to have been solved, at least in part, by the processes underlying emotions. [Oatley, 1987]

There is a growing interest in distributed problem solvers using loosely coupled agents [Durfee *et al.*, 1989]. Such systems will proliferate in the future as a result of the connection of special-purpose computers and software packages via network links. A natural outgrowth of this is the interaction of human and computer expert partners, within the network. As such a system grows then the kinds of communication problems discussed by Brooks begin to arise, and an emotion-like system may help the intelligent nodes to operate as a society. In such a society, standards and principles would be shared by sub-groups of the society, the effects of actions on other agents would be monitored through observing their emotion manifestations as feedback, and concern frameworks would be used to filter out situations that were not likely to be of interest to the individual agents.

Lastly, as computers start to produce results when given less well-defined, more creative tasks, increased stress and importance will be placed on the dynamic nature of the human-computer interface needed for problem solving. Both the computer *and* the human user will have to be able to reassess goals as the problem state is altered [Rich, 1991]. There are two specific features in such an environment that call for the use of affective control components: (1) in an unpredictable world there is need for continual assessment of the state of that world with respect to the concerns of the observing agent, and (2) local control of action initiation is necessary to respond in a timely manner to both serendipitous opportunities and (abstract) preservation requirements.

Testing psychological theories

Another reason to model emotions is that we can use the representational system to test theories of emotion elicitation and response. In principle we do this by (1) seeing if theories embedded in the computer model produce plausible "behavior", (2) using different theories and comparing the resultant behavior with that generated previously, and (3) using the results to modify the theories. The Affective Reasoner provides a rich testing ground for this since it addresses issues in interpretations leading to emotions, in the expression of emotions, in the observation of emotion manifestations, and in rudimentary personality representation.

Representing stories

One way to view stories is that they all have emotional content. One of the reasons we tell stories is because it helps people to remember the content, and by implication to index that content. When looking for points in stories we certainly have to do lower-level commonsense processing such as indexing known scripts and so forth, but often these lead to a higher-level assessment in terms of the great emotion themes: *Who was angry and why? Who was the hero and why do we admire him?* and so forth.

In the Affective Reasoner, stories are broken down into the features comprising the antecedents of emotions. Rudimentary personalities are constructed that interpret these features in different ways, and in different combinations of ways. Stories, full of emotional content, unfold as the agents with these personalities interact, both with each other and with the environment. To approach story understanding from this perspective, one must construct representations of the *characters* in the story as unique individuals, with unique concerns. While this version of the Affective Reasoner does not do story understanding, per se, its agents *are* given some ability to build characterizations of other agents in the system.

In other words, while most story-understanding systems attempt to explain *what happened* from a commonsense point of view, our agents, instead, build personality representations of other agents with respect to how they *felt* and why. Here are two (hypothetical) examples illustrating the different emphasis on reasoning between the two approaches:

The story:

> *Tom's car would not start and consequently he had to to miss an appointment. He cursed at his car. Harry observed this episode.*

A more traditional system might make inferences, and take actions, such as the following:

→ Tom's car would not start. He had no way to get to the location of his appointment within the time constraints. He was not at his appointment. He was upset.

→ Tom should get his car fixed.

→ Harry has learned that Tom doesn't leave enough time before appointments.

→ Harry has learned that Tom's car doesn't work.

→ Harry suggests that Tom should get his car fixed.

→ Harry suggests that next time Tom should leave earlier to avoid this problem.

In this case commonsense reasoning allows inferences that the car is the mode of transportation and the system "understands" the relationship between the running car and getting to the appointment. By contrast, the Affective Reasoner would make a completely different set of inferences and actions, such as the following:

→ Tom was angry. He felt that his car was to blame. His goal of being at his appointment was thwarted.

→ Tom might do something bad to his car.

→ Harry can't imagine why Tom would be angry at his car since to his mind a car cannot be held responsible.

→ Harry has learned that Tom holds inanimate objects responsible for (perceived) causal actions.

→ Harry tells Tom to calm down, that the appointment is not that important.

→ Harry feels sorry for his friend Tom who is upset.

In this case affective reasoning allows the inference that if *Tom* is yelling at his car he holds it responsible for some action that has thwarted one of his goals. *Harry* does not blame cars, but he is able to form an internal picture of *Tom*, who does. The program has no concept of how the car is causal in blocking *Tom's* goal.

Obviously there is overlap between the two approaches. A commonsense reasoning system can be taught to do emotion reasoning, and an affective reasoning system must be able to use common sense to map from situations to the concerns of the individual. What is of interest here, however, is the emphasis on what is important. In the first case, an understanding of the domain, and how the agents think about the domain, is important. In the second case, it is only what the agents are *feeling* that is considered, and an immediate analysis of what led to those states. To put this another way, the first system would know about cars, and causality and being late. The second system would know about anger, about blame, about a goal being blocked, and about sympathy. The first system might learn that one needs to make sure his car is working to get to an appointment on time, the second that *Tom* generally gets mad at inanimate objects. The theme captured by the first approach might be considered, *methods of getting from one location to another are unreliable;* the theme captured by the second: *inanimate objects can fail to live up to expectations and some people act as though they hold them responsible for this*. To close this section, we again quote Oatley:

> ... understanding emotions is important for our understanding of human narrative and dialogue. The study of language has been, perhaps, the largest single area in the research progamme of cognitive science. Recently, work has begun on understanding pieces of language larger than the sentence, including stories and other kinds of narrative. As this work has proceeded it has become clear that understanding when emotions can occur, what the meaning of emotions might be to characters in stories, and what inferences can be made from the occurrence of emotions is essential. Summarisation of narratives and other computational aspects of the theory of language will be impossible without such understanding. People's so-called "folk theories" of emotions, as emobodied in language and culture, become relevant, because it is within such theories that inferences take place, on which the understanding of narratives depends [Oatley, 1987].

1.3 Comparison with other work

The role of emotions in information processing and in multi-agent systems has not been widely studied in AI. In this section we compare and contrast the Affective Reasoner with other emotion-reasoning systems including PARRY [Colby, 1981], BORIS and OpEd [Dyer, 1983], THUNDER [Reeves, 1991], SIES [Rollenhagen and Dalkvist, 1989], ACRES [Frijda and Swagerman, 1987] and AMAL [O'Rorke and Ortony, 1992]. In addition we look at the related work of Toda [Toda, 1982] which describes emotional robots.

One dimension by which these programs may be distinguished is the degree to which they perform commonsense (plausible) reasoning in the object domain as a way of mapping from situations that arise in the world to the emotion-initiating concerns of the individual. For example, commonsense reasoning tells us that a man who hates insects may get angry at someone else who leaves his screen door open during the summer. We make this inference because it is *plausible* that if the door is left open, insects will come in the man's house. In the Affective Reasoner, front-end commonsense reasoning is not performed. This sort of intelligence has long been studied in its own right and is beyond the scope of this research. In processing the screen-door story, the Affective Reasoner would rely on an explicit representation of the *avoid-insects* goal being thwarted through the actions of some responsible agent to get emotion inferencing started. An obvious extension to the Affective Reasoner would be to include commonsense reasoning about how one state can lead to another, and so forth, leading to an interpretation of relevance to the emotion machinery. In fact, the system has been designed specifically to allow for such future extension, but no effort has been spent on this in the current version.

Another way to look at this is through the "derivative" and "nonderivative" motivators that Sloman discusses in [Sloman, 1987]:

> Roughly, a motivator is derivative if it is explicitly derived from another motivator by means-ends analysis and this origin is recorded and plays a role in subsequent processing. A desire to drink when thirsty would be nonderivative, whereas a desire for money to buy the drink would be derivative.

The affective reasoner is concerned only with manipulations of the nonderivative motivators which lead directly to the emotion inference mechanism.

Among the emotion-reasoning systems, the Affective Reasoner and ACRES may be characterized as approaches that deal with the representational issues of emotions proper, and of the processes that generate them. By contrast OpEd, SIES and AMAL may be seen as more general approaches that have an emphasis on reasoning *within* the object domain about situations that give rise to emotions. BORIS and THUNDER sit somewhere in the middle. In other words, the Affective Reasoner and ACRES focus on the nonderivative motivators which lead to emotions whereas the other systems have varying degrees to which they also concentrate on the non-derivative motivators. Note also that the Affective Reasoner does have an associated system for indexing stories in an object domain according to the underlying initiators of emotions (see [Elliott, 1992]). In this sense it is similar to SIES.

Of these programs, only ACRES gives any treatment at all to the biological systems involved in the production of emotions, and then only cursorily. The remainder (the Affective Reasoner included) only deal with the *cognitive* aspects of emotions.

The programs in contrast

The earliest well-known computer model of emotion reasoning was Kenneth Colby's PARRY [Colby, 1981], a program that mimicked the behavior of a schizophrenic paranoid personality. A concise account of PARRY is provided by Ortony [Ortony, 1992] who writes,

> Colby's program, PARRY, accepted English-like input and responded as a patient in a psychiatric interview by attempting to construe inputs as directly or indirectly indicative of malevolence. If malevolence was detected, one or more of three affective responses *fear, anger, or mistrust* was triggered, depending upon the nature of the construed malevolence. Physical malevolence induced fear, psychological malevolence anger, and the induction of either also induced mistrust. The linguistic output was designed to offset the construed malevolence either through counterattack or withdrawal. In all cases, the nature of the construals and of the responses was determined by the constantly updated values of a number of key variables, including those for fear, anger, and mistrust.

The Affective Reasoner, like PARRY, also models agents who "have" emotions. In addition, both systems maintain internal dynamic states, which affect the way in which the modeled agent(s) construe the perceived world and manifest the emotions resulting from those construals. Starting with a psychological model, PARRY attempted to model a true paranoid personality. By contrast, the Affective Reasoner uses an *ad hoc* theory of rudimentary personality representation designed to be broadly applicable rather than focused on a specific psychological syndrome.

More recently, another system, BORIS [Dyer, 1983], analyzed text for affective responses to goal states and interpersonal relationships. In this system, Dyer used ACE structures (Affect as a Consequence of Empathy) to account for a number of inference mechanisms that included affective content. In particular he developed the idea that understanding affect can allow one to produce a causally coherent analysis of stories. In BORIS, Dyer does not make a distinction between principles, goals and preferences. This is important because the Affective Reasoner uses this distinction to resolve some ambiguities. For example, a person may turn himself in after committing a crime because it upholds a standard of honesty, even though it clearly violates his goal of being free. Such conflict between goals and principles are a common theme of human emotional life and give us important leverage for reasoning in the affective domain.

Dyer uses a six-slot frame structure to represent his "affects". By contrast, the Affective Reasoner uses a hierarchy of loosely-structured frames to represent interpretation schemas, and a simple nine-attribute relation (called the Emotion Eliciting Condition Relation) to represent emotion types. There are a number of reasons this latter approach is better suited to the type of reasoning we do. In the first place, frame hierarchies allow for the use of slot inheritance. This is useful when specifying the interpretation schemas used for appraising emotion eliciting situations. That is, properties may be shared by many interpretation schemas and yet can be specified once in an ancestor, and then inherited by children frames. For example, when a team scores a touchdown during a football game it may elicit emotions in the fans. However, much of what defines this as an emotion eliciting situation has to do with the nature of games in general, and at an even higher level, with entertainment. Interpretation schemas may inherit slots that specify the associated goals as *entertainment* goals instead of *health* goals, and descriptive slots such as *team-that-scored-the-goal*, etc. The use of frames allows a great deal of efficiency in making use of the system's stored knowledge about situations.

The frames used in this flexible representation of an agent's interpretation schemas should not be confused with the simple relations representing the emotions. Both BORIS and the Affective Reasoner use such relations to represent emotions, although interestingly BORIS represents these relations as frames. Here there is no room for inheritance: each attribute of the tuple is specified and has an associated value. To give inheritance to such an emotion specifier is to imply that one emotion is a child of some other emotion, and, as such, derives value from the parent. (Such a position is viable, but it is not one taken either by the Affective Reasoner or BORIS.)

One result that the BORIS work shares with the Affective Reasoner and the work by Ortony, et al. [Ortony *et al.*, 1988] is in suggesting valid, but generally unnamed, emotion types. These arise when certain configurations of the attributes in the Emotion Eliciting Condition relations (to use the Affective Reasoner's term) seem to give rise to a recognizable, but unnamed set of emotional states in the agent. Dyer refers to the case where X feels negative toward X as a result of a goal failure caused by Y and gives the example of a woman who felt foolish and castigated herself for having had her purse stolen by a thief [Dyer, 1983]. Ortony, et al. identify the class of emotions that arise when one's fears have been confirmed. They call these, appropriately, the *fears-confirmed* type of emotions. In the current Affective Reasoner research questions are raised about one's goals being blocked by the upholding of standards (i.e., "doing the right thing" which has a bittersweet, heroic quality), and having one's goals furthered by the violation of standards.

Related work has been done by Dyer on OpEd [Dyer, 1987], which understands newspaper editorial text. In the OpEd architecture, frames are used to represent conceptual dependency structures [Schank and Abelson, 1977] which form belief networks about the world. Inferencing is guided by knowledge of affect which is incorporated into the knowledge base. For example, in trying to understand the meaning of a piece of text, the word "disappointed" causes demons to be spawned that resolve pronoun references and search for blocked goals. This processing all falls into the category of plausible reasoning within the object domain which, as previously discussed, is beyond the scope of the Affective Reasoner's focus.

The THUNDER (THematic UNDerstading from Ethical Reasoning) program of Reeves [Reeves, 1991], is a story-understanding system that focuses on evaluative judgment and ethical reasoning. Schema called *Belief Conflict Patterns* are used to represent the different points of view in a conflict situation. This work is related to that of the Affective Reasoner in that the kind of moral reasoning needed to understand the stories that THUNDER takes as input involves the same sorts of knowledge required to generate the attribution and compound emotions (e.g., reproach and anger). One of Reeves's contributions is in the detailed working out of a set of moral patterns at the level of stories. Reeves develops the idea that *evaluators* of stories make many moral judgments about the characters in those stories using these patterns, and that this moral framework is often necessary for understanding text. For example, Reeves analyzes a story in which hunters tie dynamite to a rabbit "for fun". The rabbit runs under their car for cover and when the dynamite blows up it destroys their car. Reeves argues that to understand the irony in this story the evaluator must know that killing a rabbit with dynamite for entertainment is reprehensible. This allows the judgment that the chance destruction of the car is an example of morally satisfying retribution.

Clearly moral reasoning is closely related to the rise of emotions. In both THUNDER and the Affective Reasoner, the focus is on the point of view of individual agents within, or observing, a situation. Reeves writes: "*...using different value systems will produce different ethical evaluations.* For example, the actions of a high school student who killed himself after failing a test would be judged ethically wrong by a Catholic, but not by a Samurai..." ([Reeves, 1991], page 23). The Affective Reasoner has no text understanding mechanism, but it could *simulate* the suicide scene with various observers responding with different emotions, dependent upon their (moral) principles.

One area that has not been focused upon in great depth is the action generation component of emotion reasoning. Most action schemes in AI research focus on the logically defined needs of the agent (i.e., actions as a function of planning for goals). Since emotions themselves are so poorly understood and so seldom incorporated into experimental AI systems, it is understandable that little work is being done that uses them as motivations for action. One system that does function in this way, however, is the ACRES program (Artificial Concern REalisation System) developed by Frijda and Swagerman [Frijda and Swagerman, 1987] in the Netherlands.

The main premise of their research is that emotions provide a way of operating successfully in an uncertain environment (see also [Sloman, 1987]). They approach the design of ACRES from a functional standpoint: if an agent is to be able to respond functionally to its environment, what properties must the subsystem implementing that functionality have? To this end, in contrast to the Affective Reasoner's descriptive approach, the primary emphasis is placed on behavior potentials and the search for reasons to initiate them. To quote:

> The major phenomena are: the existence of the feelings of pleasure and pain, the importance of cognitive or appraisal variables, the presence of innate, pre-programmed behaviors as well as of complex constructed plans for achieving emotion goals, and the occurrence of behavioral interruption, disturbance and impulse-like priority of emotional goals [Frijda and Swagerman, 1987] (page 235).

Their architecture is designed to meet these demands:

> The system properties underlying these phenomena are facilities for relevance detection of events with regard to the multiple concerns, availability of relevance signals that can be recognized by the action system, and facilities for control precedence, or flexible goal priority ordering and shift. (page 235)

The Affective Reasoner also addresses some of these issues, although the emphasis is different. The Frijda-Swagerman action architecture arises from issues relative to processing needs, whereas ours arises from a description of actions. Similarities can be found on a number of fronts. Both systems incorporate the idea of the cyclic nature of emotion machinery: a situation arises, it is evaluated, it is either ignored or a response is initiated which is fed back to the modeled world. As with the Affective Reasoner, ACRES performs no reasoning about derivative motives. Instead, emotion triggers are directly wired into the architecture.

The Affective Reasoner and ACRES also both stress the importance of *concern realization* as a way of filtering out situations that are not of interest, and of interpreting those that are. Entities have multiple concerns and limited resources. To solve this problem both programs focus on *matching* as an efficient means of assessing emotion eliciting situations. To quote Frijda and Swagerman, "Concerns can be thought of as embodied in internal representations against which actual conditions are matched," [Frijda and Swagerman, 1987] (page 237), which is exactly the conceptual approach of the Affective Reasoner.

The two systems differ, however, in the way in which they use the realized concerns. Of primary importance to ACRES (but only of incidental importance to agents in the Affective Reasoner) is filtering out situations irrelevant to resource conservation (as most situations are). In addition, unlike the Affective Reasoner, ACRES stresses the correct interpretation of situations with respect to *urgency* (as distinct from importance- see [Sloman, 1987]). To approximate this behavior ACRES uses a system of interrupts.

There are a number of interesting issues that ACRES does not address. One of these is the maintaining of expectations with respect to future goals. The aging of expectations and the elicitation of associated prospect-based emotions is important for systems representing emotional behavior, as is the representation of the emotional status of other agents. ACRES does not maintain internal models of other agents, nor does it maintain a working memory for the storage of currently active relationships. Without these features it is not possible to reason about the emotional lives, and therefore motivations, of other agents with whom it must interact. A simple test of such a system is to see how well it fares on modeling a situation such as the following: Tom pities his friend Harry who is sad because his (Harry's) mother has had her garage burn down. Humans have no trouble with this because not only do we have an internal representation of our own concerns, but we are also apparently able to use some similar mechanism for representing the concerns of others, and thus "imagine" how they might feel with respect to some situation. In this case Tom must not only model Harry, but he must also model the concerns that Harry is modeling for his mother. (See section 2.8.3.)

The Affective Reasoner, in contrast to ACRES, incorporates representations of the concerns of others in its architecture (see section 2.8), and represents relationships between agents. These representations are incorporated into the emotion-generation process when initiating fortunes-of-others emotions (see section 2.3) and into the process that helps agents identify new situations as instances of a certain emotion type.

The System for Interpretation of Emotional Situations (SIES) of Rollenhagen [Rollenhagen and Dalkvist, 1989], and the AMAL system of O'Rorke [O'Rorke and Ortony, 1992] share a number of features. They both have taken on the burden of having to do non-derivative reasoning about the situations that give rise to emotions within the object domain. They both use retrospective reports about emotion-inducing situations as the basis for the structural content analysis that drives the reasoning systems. Lastly, they both are descriptive in nature. SIES attempts to better define emotion terms by exploring their similarities and differences through abstract representations. AMAL attempts to explain situations as emotion episodes through the use of abductive reasoning.

By contrast, the character of the Affective Reasoner, as a *simulation*, is quite different from these two programs. In the Affective Reasoner the process of representing emotions contributes not only to developing a language for describing emotion episodes but also to the building of intelligent agents that "have" emotions. In addition, while the Affective Reasoner uses a logic-based representation for reasoning about the eliciting conditions for emotions, similar to both SIES and AMAL, neither of these latter systems has anything like the case-based approach used in the Affective Reasoner to reason backwards from the manifestations of emotions to the emotions themselves.

The theoretical underpinnings of AMAL and SIES are quite different, although in practice this difference is minimized. AMAL (based in part on the work of Ortony, et al. [Ortony *et al.*, 1988]) is an extension of the *cognitive appraisal* approach where the antecedents of emotions are described independently of the particular situations from which they derive. In this approach it is situation *types* that give rise to emotions (e.g., an undesirable event giving rise to distress), rather than relatively specific situations (such as the death of a loved one). However, since AMAL is designed to be able to represent such real-world emotion-eliciting situations as those portrayed in student diaries [Turner, 1985], it must also be able to do extensive reasoning within the object domain. To represent these emotion anecdotes a *situation* calculus is used. SIES, on the other hand, is an attempt to combine the cognitive approach with a *situational* approach in which the antecedents of emotions are seen as arising in certain real-world situations (e.g., loud noises leading to fear, or separation from a loved one through death leading to sadness).

In his description of SIES, Rollenhagen gives many rules for emotion reasoning, but most of these are employed in describing an instance of a given emotion eliciting condition as embodied in a narrative description – that is, in giving structure to the derivative motivations of the agents. SIES should be understood as a *system* for analyzing emotion episodes. In this regard, much of the contribution of the system is in structuring the representation of the data. The program does not map situation features into specific eliciting conditions. Instead, inferences tying abstract situations into the personal concerns of the subjects are made at the time the researcher encodes each emotion eliciting situation.

The representation of emotion eliciting situations in the Affective Reasoner is from a different perspective, and is structured around the idea that agents may be seen as carrying internal schemas that match situations that arise in their world. Here the focus is on frames that contain enough information to discriminate one situation from another through *matching*. Varied interpretations of these situations, produced by unifying an interpretation frame with a situation frame during the course of the simulation, are seen as whole units. These interpretation units are configured in different ways such that they give rise to the various emotions.

Unlike SIES and the Affective Reasoner, AMAL [O'Rorke and Ortony, 1992] does do extensive, automated, plausible reasoning about derivative motivators. To do this it uses an abduction engine. AMAL uses a logic-based situation calculus framework to describe actual emotion episodes[Turner, 1985]. Using abductive reasoning, AMAL is able to construct plausible explanations for emotion instances.

One other important difference among the three systems should be noted. AMAL and SIES typify the approach that takes the logical structure of the preconditions for emotions to be simply an extension of the structure of the world (e.g., a noise causes Tom to startle, which causes him to drop a hammer on his foot, which causes him to feel pain, which causes his friend to feel sorry for him). Our approach is fundamentally different in that there is a definite demarcation between the internal strong-theory rules used to generate emotions for agents, and the object-domain rules used to map into them. In this approach, since we see emotion eliciting situations in the object domain as instances that may or may not match internal schemas representing the concerns of the agents, processing and data-entry tasks are very different in nature. While each approach can, of course, be reduced to a logic representation, the two languages for content theory representation are entirely different, and thus, in practice, so are the systems. In AMAL and SIES everything is represented as rules, in the Affective Reasoner the concerns of the users are represented as collections of features.

Toda's work on *Solitary Fungus-Eaters* [Toda, 1982] attempts to describe emotions as a situated functional component of society. In this case, Toda's society is that of simple (fantasy) uranium-mining robots that operate autonomously on a distant planet and eat native fungus to survive. The goal of these robots is to collect as much uranium as possible. If they consistently choose only to travel to locations containing uranium, however, they may not encounter enough fungus to keep themselves going and in such a case would "die". Society arises out of a need for a "coalitional bond" to help protect the robots against stronger predators, and so forth.

Toda's structural analysis of the human emotion system is based on his stated belief that emotion was a useful mechanism for a very different, primitive, society than our current one, and that just as our bodies have problems dealing with modern-day environmental stresses such as pollution, so does our emotional system have difficulties with such an (evolutionarily) unfamiliar environment. To illustrate his ideas about why an emotion system arose in the first place, giving insight into its nature, Toda details a system of *urges* necessary to solve coordination problems in his simple robot society.

For example, in response to the threat of a larger predator, a Fungus-eater may have a *Fear Urge* which causes him to send out a distress signal. This in turn triggers a *Rescue Urge* in other robots who then come to jointly attack the predator. Since a reward is necessary for this scenario to fit into the "individual-gain" paradigm of Toda's robot society, a *Gratitude Urge* is developed. Rewards, given out of gratitude by the rescued robot, may have to be postponed, however, if the rescued robot is short on resources. Since this delay may cause problems, a more immediate reward in the form of a *Love Urge* is developed, and so forth.

This major difference between this work and that done on the Affective Reasoner is of course that the latter is actually implemented as a running program. While Toda has given some level of detail for many aspects of his system (including vision and planning, not discussed here), there are nonetheless many gray areas in his description. Vision and planning aside, such social actions as *defecting from one's coalition partner and joining a more profitable coalition* are treated as primitives in the discussion. To implement even a small number of these primitives of the robot society would be a major task. In addition Toda's *urges*, intended to be thought of tentatively as simple emotions, are presented anecdotally and not as part of a cohesive, complete theory.

Toda's theoretical perspective shares with both the Affective Reasoner and ACRES the idea of emotions as mediators for controlled, immediate *responses* to situations (see also [Simon, 1967]). To this end, emotions take a functional role in the automated organism.

From this short survey it should be apparent that comparatively little work has been done in this area, and that what has been done is often so different from what has come before that comparisons are strained. PARRY is a system that attempts to model a diseased and limited mind. BORIS, OpEd and THUNDER use a content theory of emotions to aid in the understanding of text. ACRES attempts to model the dynamic utility of emotions. SIES provides a structure for mapping from the features of situations into the emotions. AMAL uses abductive reasoning, well suited to the understanding of emotion episodes, to explain emotion situations. Clearly these are diverse approaches only loosely tied together by the generic categorization of the work as "emotion research". One theme that all of these treatments share however, is that working out theories at the level of detail required for a running computer program forces the theorist to refine and completely specify all of the hidden "gray areas" of the theory. This is particularly important for this complex, and little understood subject.

1.4 TaxiWorld

The Affective Reasoner has three primary components: (1) the affective reasoning component, which is independent of any domain, (2) a world simulation based on an object-domain theory, and (3) a graphics interface based on the simulated world. Once an object-domain theory has been incorporated into the Affective Reasoner, and a graphics interface customized for that domain, it becomes a domain-specific system. In its current form the Affective Reasoner primarily manipulates agents that are intended to be interpreted as operating in a schematic representation of the Chicago area (figure 1.1). We call this *TaxiWorld*. In fact, other partially analogous domains can be simulated if the user simply reinterprets the meaning of the icons displayed. This is how the football game example discussed in chapter 3 was run.

TaxiWorld has approximately forty-five different sets of simulation events which produce emotion eliciting situations. Included are events that represent traffic accidents, rush hour delays, getting speeding tickets, picking up passengers, getting paid, having to wait a long time for a taxi, and so forth. In addition, there are approximately forty different rudimentary "personalities" which may be given to each agent that participates in one of these situations. For example, one taxi driver may get angry when he is given a small tip whereas another may just figure it is part of the job. Or, both may get angry but one will be rude to his passenger, whereas another may just smile and pretend that he does not care. There are roughly 150 different candidate interpretations of the forty-five situations. Different interpretations, or sets of interpretations, can give rise to one or more of twenty-four emotion types, each of which has about sixty possible action responses associated with it. Used in different combinations these components can yield tens of thousands of different emotion episodes.

In any research of this kind the question naturally arises as to what has actually been implemented and what has been run. In this dissertation only a few emotion episodes will be discussed in detail so as to allow us to focus on emotion representation issues. We provide at most one emotion episode example for each point. However, it should be understood that many different examples may actually have been run in the process of addressing any one particular problem. In addition, unless otherwise stated, illustrative examples in this dissertation are based on code that has been written and on actual simulated episodes.

The researcher using TaxiWorld has two forms of input into the system. First he or she statically configures the object domain (i.e., the world of taxi drivers and their passengers) through LISP files. Second, he or she dynamically manipulates the running simulation through menus and windows. Configuring the object domain has six parts: (1) Should the researcher wish to extend the number of situations represented then new simulation events must be entered into a LISP file, and event handlers written to process them. (2) Once the object domain (e.g., the world of taxi drivers) is stable, the researcher may choose to add new interpretations of the situations that arise in it by creating *construal frames* (see chapter 2) with which agents interpret situations. (3) New emotion manifestations may be created within the object domain. For example, in the world of taxi drivers we may wish to represent "cutting someone off" as an expression of anger. (4) New personalities may be created by grouping (old and/or new) construal frames into sets, and by grouping *temperament traits* (i.e., tendencies toward certain types of actions as expressions of emotion) into new sets. (5) Simulation sets may be constructed by specifying which of the represented situations are to arise, which types of agents are going to be present, and which personalities those agents will have. Lastly, (6) a content theory for reasoning about observed actions from cases may be recorded in a data file or through interaction with the

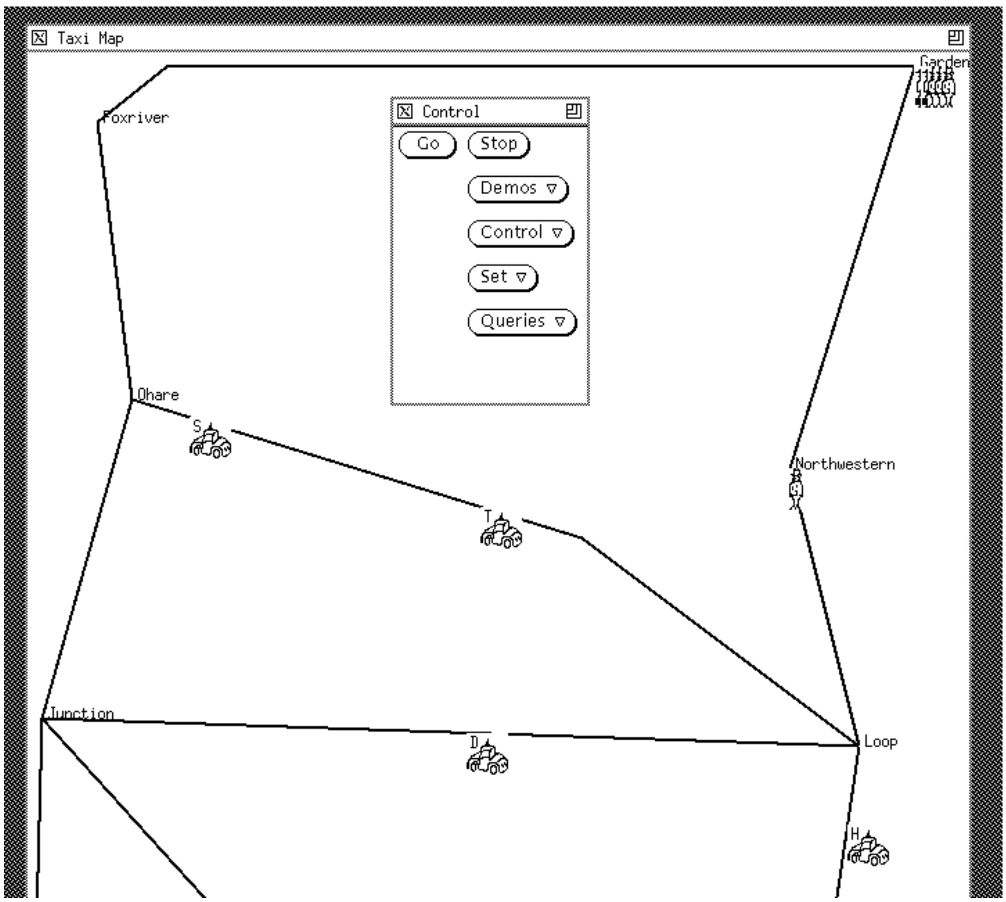


Figure 1.1: The TaxiWorld display.

running system. User input macros have been provided for (2) – (6).

Figure 1.1 shows a portion of TaxiWorld's map display. Interaction with TaxiWorld is through this interface. Except for *stop* and *go* each of the buttons on the control panel leads to a pop-up menu for manipulating different aspects of the system. This control mechanism allows the following functions, by button: (1) *stop* and *go*: start the simulation running or pause it, (2) *demos*: select a configuration of agents and situations for the simulation, (3) *control*: change the speed of the simulation, or enable and disable different features in the simulation such as whether or not agents negotiate and what types of traffic delays may arise, (4) *set*: set rudimentary global moods for the agents, and set the mode for the heuristic classification system (i.e., set the case-based reasoning system in *learn* mode, in *report-only* mode, or *off*), and (5) *queries*: look at a scrolling window containing summaries of the situations that have arisen so far.

Once the user has configured and started the simulation, the animated icons move around the screen as agents who "experience" the situations in the simulated world. The figure shows a simulation that has been running for a short time. Four taxi drivers, Tom, Dick, Harry and Sam are shown. Tom and Sam are traveling between Chicago's O'Hare Airport and the Loop, Dick is between the "Junction" and the Loop, and Harry is on his way to the Loop from the Museum of Science and Industry (not pictured). One passenger is waiting for a ride at Northwestern University, and five passengers are queued up waiting for rides from the Chicago Botanical Gardens. During rush hours the roads swell with traffic and the agents move more slowly. This occurs also with accidents. Highway patrol agents may also be displayed.

Selecting an agent with the mouse provides information about that agent. One option will bring up a scrolling window containing the current "physical" state of the agent: where he is, whether he has someone in the cab (if he is a taxi driver agent), how much gas he has left, where he is going, etc. Another option brings up an emotion-history scrolling window. Figure 1.2 shows one of these, in this case for Harry. Here we see summaries of two emotion instances, *distress* and *dislike*. These have arisen from two different construals of the same situation. Briefly, a new passenger has gotten into Harry's cab and was cheerful. Harry, whose "personality" type might be roughly described as *depressed grouch* has a goal and a preference that apply in such a case. First, he has no desire to be around happy people because they remind him of how unhappy he himself is. This leads to distress. Second, he just does not like happy people, he finds them distasteful, so he dislikes this passenger.

For maximum flexibility, the researcher has the option of communicating with the running system directly through LISP. Basic tools have been provided for this, in particular to control the asynchronous processes spawned in the simulation. The user can also interact with the case-based reasoner through LISP as well.

TaxiWorld is written in Common LISP and runs on a *SUN SPARC station 1* with 40MB of RAM. It uses a simple *SOLO* graphics interface. The TaxiWorld code is

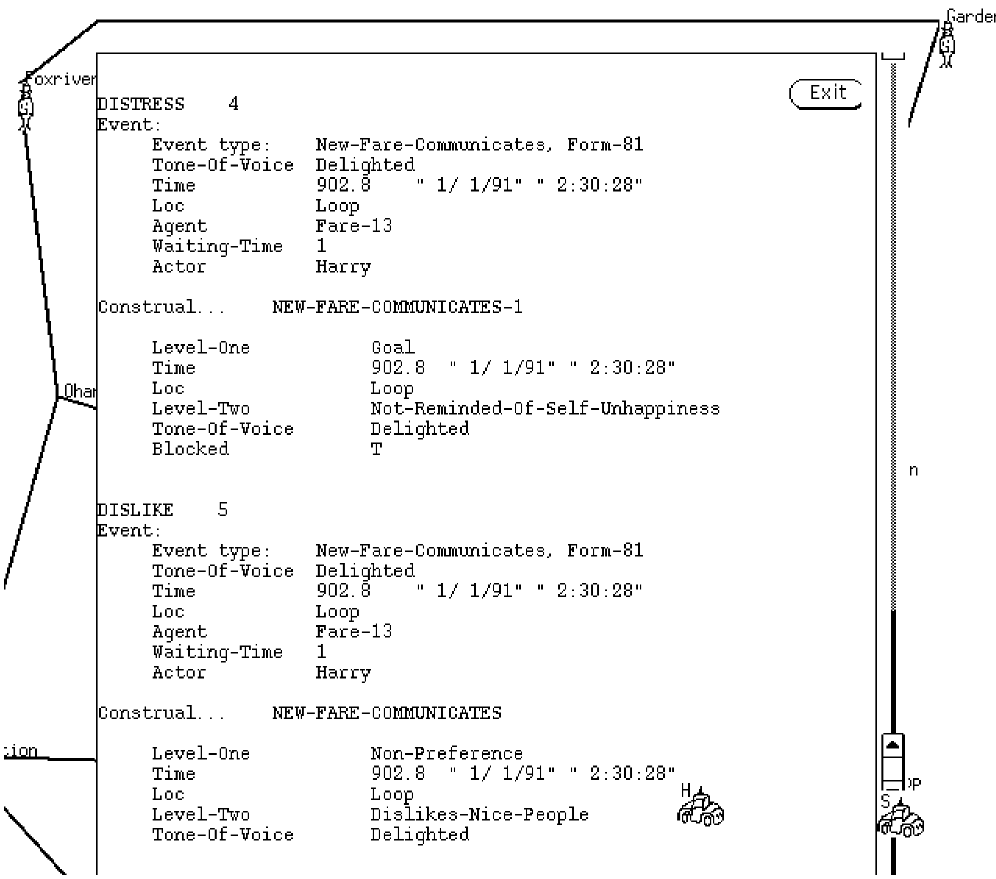


Figure 1.2: The emotion history scrolling window.

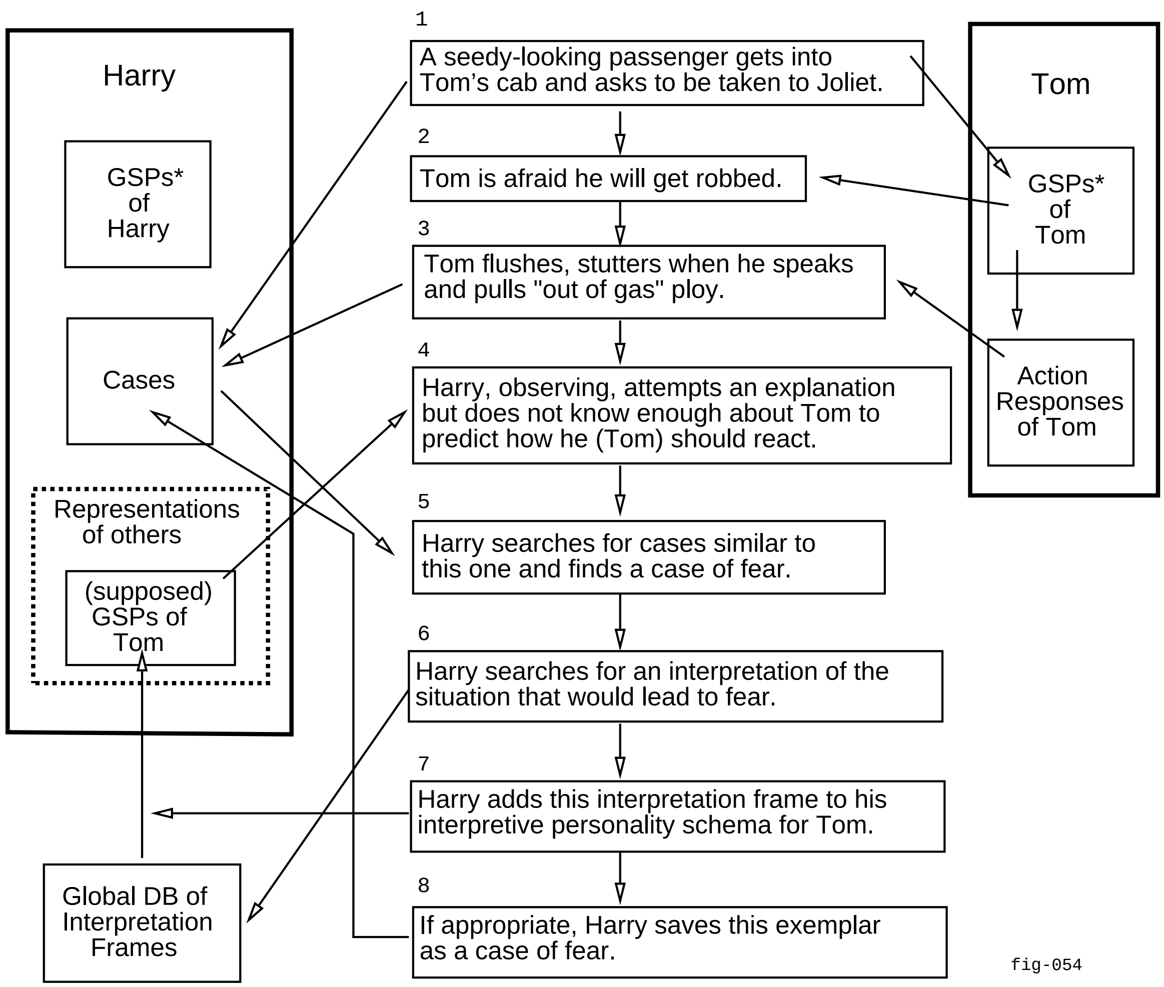
about fifteen thousand lines in length, and is combined with an additional fifteen thousand lines of slightly modified code based on Dvorak's Common LISP version of *Protos* [Dvorak, 1988].

1.5 Two examples from TaxiWorld

In this section we look at two examples of the kinds of episodes that occur in TaxiWorld. Figure 1.3 represents a simple episode where a taxi driver, Harry, observes another taxi driver, Tom, and learns something about his fears. This episode makes use of three agents (the third is Tom's passenger – not pictured), one situation (that arises in the Chicago area) and a single affective state, in a *TaxiWorld* simulation. In this episode we see the following: (1) Tom picks up a passenger headed for Joliet at the Museum of Science and Industry. (2) The passenger is seedy-looking, causing Tom to fear that he will be robbed. (3) This fear is expressed as a flushed expression on Tom's face, a stutter in his voice, and the statement that he does not have enough gas to drive to Joliet. (4) Another taxi driver, Harry, watches this episode. He does not know that Tom is inclined to be afraid of seedy-looking passengers whose destinations are Joliet. He has no representation of how Tom might construe such an event. However, (5) Harry *has* seen a case where another agent had a flushed face and a stutter in his voice. The previous case was known to be an example of *fear*. (6) Harry now reasons to come up with a good explanation: if Tom is experiencing fear, why might this be so? He "imagines" some possible interpretations of this event that might cause someone to be afraid and reasons that if Tom had the goal of retaining his money and/or maintaining his personal safety, then given that seedy-looking people have been known to rob cab drivers – thus taking away that money and perhaps causing harm, Tom might experience fear. Since this explanation fits, Harry drops his search. (7) He updates his internal representation of Tom as someone for whom retaining money is important, and who is afraid of unsavory passengers. (8) Harry also now has a new case which he may wish to save. The "I don't have enough gas" ploy, which was not a feature of the previous case of *fear*, may be explained to him as a *problem-reduction* strategy, or it may be left unexplained, depending on whether the system was being trained, or was running in report-only mode.

Harry now believes he has learned something new about Tom. If this knowledge is correct he will be better able to explain Tom's future actions in response to such a situation without resorting to a case search (i.e., he *knows* how Tom will interpret the situation). In addition he will be able to *predict* how Tom might react to such an event, even in the absence of any empirical evidence. On the other hand, if at a later time the knowledge proves to be erroneous it will be discarded and a search for an alternate explanation will be initiated.

The episode pictured in Figure 1.4 expands on the previous example. Here we see



* Note: GSP's refer to the Goals, Standards and Preferences of an agent.

Figure 1.3: Harry learns about Tom's fear.

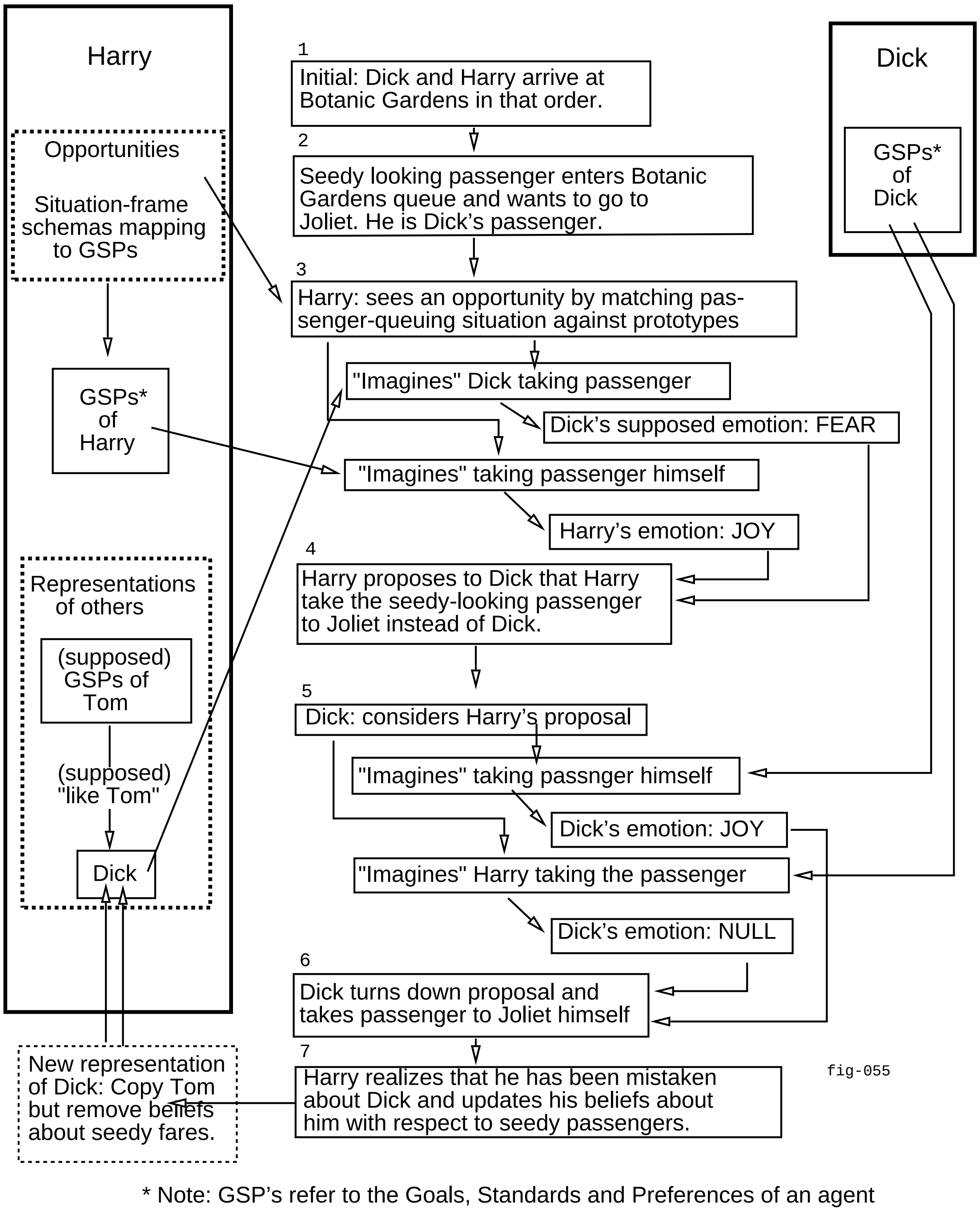


Figure 1.4: Negotiating about who takes the seedy-looking passenger to Joliet.

that some time later Harry has come to know Tom quite well.[2] He has met a new agent, Dick, who seems to respond to events the way Tom does, although he (Harry) has never seen Dick pick up a seedy-looking passenger headed for Joliet. Until he knows differently, Harry makes the assumption that Dick is just like Tom (i.e., with respect to the way he interprets situations). At one point, Harry (who has the goal of making money and likes to go to Joliet because, as a long trip, this furthers his goal), sees an opportunity. Here is the sequence of events: (1) Dick and Harry arrive at the Botanic Gardens in that order, giving Dick the right to the first passenger. (2) A seedy-looking passenger arrives at the cab stand and wants to go to Joliet. (3) Harry matches the passenger-arrival situation against an opportunity prototype.[3] He sees an opportunity to make a request of Dick, but "thinks it through" first to see if it is feasible. Using his representation of how Tom sees the world as a default for Dick, Harry believes that Dick, like Tom, has fear about going to Joliet with the seedy-looking passenger. On the other hand he believes that he will, himself, be happy about going on such a trip. (4) Harry proposes to Dick that Harry, instead, take the fare to Joliet. (5) For his part, Dick, considering this proposal, "imagines" that Harry gets to take the passenger. He has no emotional response to this. Next he "imagines" taking the passenger himself. As it happens, Harry's schema for Dick is incorrect: while Dick is indeed very much like Tom in many ways, he is unlike him in that he is not afraid of seedy passengers who want to go to Joliet. Consequently Dick does not see a threat to his goals, but rather only that if he *does* make the trip to Joliet he will make some money. In other words, he reasons that if he makes the trip he will be happy, and that if he does not make the trip he will simply experience a lack of emotion. (6) He therefore does not agree to Harry's proposal and instead takes the passenger himself. (7) Harry believes he has learned something new about Dick. Apparently Dick does not construe seedy passengers going to Joliet as threatening. Harry splits off his internal representation for the goals, standards and preferences of Dick from the representation of those for Tom and removes the offending interpretation frame from the latter.

In this example we see that Tom and Dick both have distinct emotional lives, and that Harry is able to reason about them. Tom has a *goal* of making (keeping) money and is able to experience *fear* as a result of believing that this goal may be threatened.

[2] Harry "knows" Tom only in the sense that in a system with a limited number of situation types and interpretations of those situation types, he knows which interpretations Tom will probably make.

[3] This simple "opportunity" has the form: the other cab driver has right to the passenger; the other driver may have a negative emotion about this; *self* has positive emotion about taking passenger; so make a proposal. The Affective Reasoner allows the user to attach an opportunity-detection mechanism to the construal frames used to interpret emotion eliciting situations. These mechanisms are used to recognize situations which might, if the proper action is taken, lead to other situations in which the construal frames apply. This simple mechanism allows us to initiate rudimentary negotiations between agents, which in themselves often give rise to emotions. Since this is not a sophisticated mechanism, and is not central to this research, it will not be discussed further.

Similarly, Dick also has a goal of making money and is capable of feeling *hope* over the prospect of taking a passenger on a long trip.

On the other hand, Harry, who is observing Tom and Dick, draws on his experience, and his internal schemas for the other agents, to explain their actions. In the first case, he has no clue as to how Tom might interpret the situation. He looks through some cases to see if he might make a guess as to what *stuttering, flushing* and *the out-of-gas ploy* might indicate. He has seen something similar when someone else was afraid, and he can explain the fear as a fear of getting robbed. Since this explanation of the situation is workable he assumes that this is part of how Tom interprets the world, and he now *adds this to his representation of Tom.* Until he has cause to do differently he will now always make this assumption about Tom.

In the future, Harry will test this knowledge whenever a similar situation arises by asking this question, *Given a similar emotion eliciting situation (seedy passenger going to Joliet), does the observed agent respond in a manner compatible with an emotion to which the assumed interpretation leads?* If the agent does, then Harry will be more confident that his assumed interpretation is correct; if he does not, then Harry will have to search for another explanation (i.e., one which leads to a different emotion).

In addition, Harry's representation of Tom is one he can draw on as a *default* personality type, for reasoning about a new agent, Dick. As long as Dick's actions can be explained in terms of Tom's personality, the default personality suffices and he uses it to "see the world through Dick's eyes". When an opportunity arises he believes that it is worth pursuing because he is able to "imagine" how Dick will perceive his offer. However, when Dick turns him down, Harry reasons that he has made an incorrect assumption about Dick and removes this supposition from his representation of how Dick sees the world. But this means that Dick no longer looks just like Tom, so Harry must split the two schemas apart. The old one still represents Tom, and the new one now represents Dick.

The rudimentary negotiation, the simple opportunity recognition, and the rudimentary default personality reasoning, all open research questions in their own right, are not what is important in these examples. What is of interest is the idea that there are units of appraisal, here represented as frames, which can be used as filters for the interpretation of situations, and that these appraisal units can be mixed and matched as necessary to construct rudimentary personalities for agents (i.e., to give them primitive affective life). Of additional interest is the idea that these appraisal units can also be used by an observing agent to construct an internal representation of the observed agents for generating explanations of their actions, and for predicting their emotional responses to future similar situations. Lastly, it is important to consider that agents manifest their emotions in ways that can be understood by observers: agents often communicate their emotions through their actions, or their lack thereof.

In the following chapters we attempt to illustrate some concerns that have to be addressed by emotion reasoning systems, how our representation addresses these concerns, and how it can be used to model a number of different multi-agent interactions.

Chapter 2

Overview

In this chapter we provide an overview of the Affective Reasoner. The discussion focuses on the functional role of the various modules in the system. Two of these modules (construal and action-generation) are quite elaborate, having important theoretical content, and so are only introduced here since they are discussed more fully in later chapters. The other modules are less elaborate, and are treated fully in this overview.

The sections are organized around the various stages the system goes through in processing a simulated situation. They are introduced in the order of the different processing stages. These stages, and the static representations which they use, are illustrated in figure 2.1. Processing flows from some initiating simulation event through emotion and action generation to the final stages simulating observation by other agents. Roughly speaking, this may be interpreted as follows: Something happens in the modeled "world", creating a situation. The agents that populate this world interpret the situation in terms of their individual concerns. Each interpretation is reduced to a nine-attribute relation, and some variable bindings. Specific configurations of the relation lead to the production of specific emotional states in the agents (i.e., their emotional responses to the situation). These emotional states, in turn, are manifested as action responses. Agents can observe each other's responses to situations and attempt to explain them in terms of emotions the other agent may have been experiencing. An observing agent then finds an explanation for that emotion based on varied interpretations of the original situation. Once this is done, the observer places the schema that lead to the successful interpretation (i.e., the interpretation that lead to what was believed to be the other agent's emotion) into a database representing the concerns of the other agent. This Concerns-of-Other database may then be used to predict and explain the other agent's responses to future similar situations.

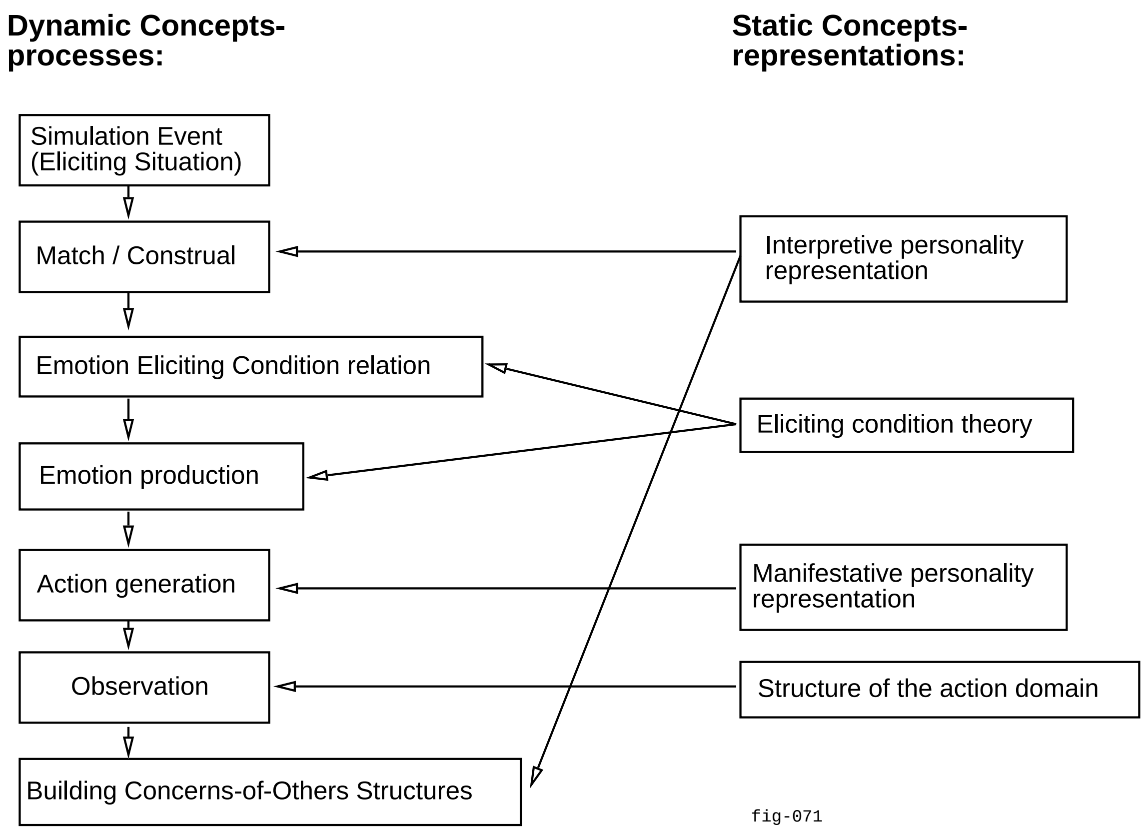


Figure 2.1: Processing stages and related representations

2.1 Fundamental concepts

This research has roots in both Artificial Intelligence and Cognitive Science. Its interdisciplinary nature can lead to confusion over the use of terms, such as *personality*. Accordingly we will, in this section, make explicit our intended meanings. In addition, we also discuss some central concepts, such as the emotion eliciting condition theory on which the construal process is based [Ortony *et al.*, 1988].

Rudimentary personalities. To computer scientists *personality* means something like *an aggregate of qualities that distinguish one as a person.* (Webster). To psychologists however, *personality* means something rather different, something like *co-occurring classes of trans-situational stable traits.*

In the Affective Reasoner, the agents in the system have rudimentary "personalities" (at least computer scientists might consider this to be so) that distinguish them from one another. These rudimentary personalities are divided into two parts: the disposition agents have for interpreting situations in their world with respect to their concerns, and the temperament that leads them to express their emotions in certain ways. We have chosen the terms **interpretive personality component** and **manifestative personality component** to denote these two parts of an agent's unique makeup. By the former we mean the *rudimentary "personality" which gives agents individuality with respect to their interpretations of situations* (i.e., their uniquely individual concerns). By the latter we mean *the rudimentary "personality" which gives agents individuality with respect to the way they express or manifest their emotions.*

Situations that lead to emotions. In the Affective Reasoner, some simulation events create situations that can initiate emotion processing on the part of the agents involved. These we call *emotion eliciting situations* or simply, **eliciting situations**. An example of such an *eliciting situation* is the conceptual "arrival" of some agent at a location, which might give rise to an emotion such as relief or distress. *Eliciting situations* are not to be confused with *eliciting conditions* which derive from the emotion eliciting condition theory of [Ortony *et al.*, 1988] and are discussed below.

Goals. This term has a very broad meaning. Here we use it in its simplest sense: a state of affairs that an agent desires to have come about. In this dissertation, unless otherwise specified, most goals will be considered to be equivalent in structure.[1] Clearly this is not actually the case. Some goals are preservation goals, some are never satisfied (i.e., living a good life), some can be partially achieved by achieving some other goal (i.e., saving another $100 towards a house), and so forth. In general, however, these distinctions have more to do with goal generation, interaction, and

[1]But see chapter 3 for a discussion of this.

retirement than they do with the way goals fit into our underlying cognitive theory. Consequently, for most of the discussion it will suffice to define the term *goal* to mean simply, *a desired state of affairs that, should it obtain, would be assessed as somehow beneficial to the agent.*

Object domain. Emotion reasoning can be performed as well by one person as by another. Furthermore, people can have emotions about almost anything and in almost any circumstance. People have emotions about something important like the state of their finances, but they might also have them about something as apparently inconsequential as the state of their shoelaces. For this reason emotions may be considered an *abstract* domain that operates within *object* domains. The language of the emotion domain is abstract: goals, standards, preferences, and so forth. The language of an object domain is specific: money, shoelaces, etc. The Affective Reasoner can be used to do abstract emotion reasoning within any object domain that can be modeled. One such object domain is the world of taxi drivers, as represented by the TaxiWorld version of the Affective Reasoner. By *object domain* then we mean, *that domain in which the eliciting situations for emotions are described, and in which emotion-based actions are manifested.*

The emotion eliciting condition theory. The emotion eliciting condition rules we use for the strong-theory reasoning component of the Affective Reasoner are based on the work of Ortony et al. [Ortony *et al.*, 1988]. They specify twenty-two emotion *types* based on valenced reactions to situations construed either as being goal-relevant *events*, as *acts* of accountable agents, or as attractive or unattractive *objects*. The extended and adapted twenty-four *emotion-type* version of the emotion eliciting condition theory that we have used in the Affective Reasoner is outlined in figure 2.2 (table format based on [O'Rorke and Ortony, 1992]). Each of these twenty-four emotion states has a set of eliciting conditions. When the eliciting conditions are met, and various thresholds have been crossed, corresponding emotions result. A key element of the theory is that the way emotion eliciting situations map into these eliciting conditions depends on the interpretations of the individual agent. For example, suppose that my team wins a basketball game with a basket at the buzzer, and your team loses. I may experience *joy* at the event, whereas you may experience *distress*. In both cases we share the same sets of eliciting conditions for our emotions and the emotion eliciting situation is the same (i.e., the ball went in the basket just before the buzzer); it is only the interpretation or construal of the situation which is different.

The emotion types are simply categorizations of selected patterns of emotion eliciting conditions. They have been given English names roughly corresponding to an intensity-neutral label for the *type* of emotions represented by the specific configura-

Figure 2.2: Emotion types

Group	Specification	**Name** and Emotion Type
Well-Being	appraisal of a situation as an *event*	**joy**: pleased about an *event* **distress**: displeased about an *event*
Fortunes-of-Others	presumed value of a situation as an *event* affecting another	**happy-for**: pleased about an *event* desirable for another **gloating**: pleased about an *event* undesirable for another **resentment**: displeased about an *event* desirable for another **sorry-for**: displeased about an *event* undesirable for another
Prospect-based	appraisal of a situation as a prospective *event*	**hope**: pleased about a prospective desirable *event* **fear**: displeased about a prospective undesirable *event*
Confirmation	appraisal of a situation as confirming or disconfirming an expectation	**satisfaction**: pleased about a confirmed desirable *event* **relief**: pleased about a disconfirmed undesirable *event* **fears-confirmed**: displeased about a confirmed undesirable *event* **disappointment**: displeased about a disconfirmed desirable *event*
Attribution	appraisal of a situation as an accountable *act* of some agent	**pride**: approving of one's own *act* **admiration**: approving of another's *act* **shame**: disapproving of one's own *act* **reproach**: disapproving of another's *act*
Attraction	appraisal of a situation as containing an attractive or unattractive *object*	**liking**: finding an *object* appealing **disliking**: finding an *object* unappealing
Well-being / Attribution	compound emotions	**gratitude**: admiration + joy **anger**: reproach + distress **gratification**: pride + joy **remorse**: shame + distress
Attraction / Attribution	compound emotion extensions	**love**: admiration + liking **hate**: reproach + disliking

tions of the emotion eliciting conditions. It is important to note that these names (e.g., *joy* and *anger*) given to the emotion *types* are not to be mistaken for the specific emotions to which they usually refer. For example, *annoyance* is one of the anger *type* emotions, as is *rage*, because they both follow from interpretations of a situation as an undesirable *event* coming about as a result of someone else's blameworthy *act*.

Emotion eliciting conditions leading to emotions fall into four major categories: those rooted in the effect of *events* on the *goals* of an agent, those rooted in the *standards and principles* invoked by an accountable *act* of some agent, those rooted in *tastes and preferences* with respect to *objects* (including other agents treated as objects), and lastly, selected combinations of the other three categories. Another way to view these categories is as being rooted in an agent's assessment of the *desirability* of some event, of the *praiseworthiness* of some act, of the *attractiveness* of some object, or of selected combinations of these assessments.

Emotion types. As discussed above, when we speak of *an emotion* generated by the system we are really talking about an emotion of that *type*. It is usually not convenient (i.e., it obscures the meaning of the text) to talk about the "emotion type of anger" and so forth. However, it should be understood that throughout this text when we refer to some emotion, as though by name, we are actually referring to some unspecified emotion characterized by that named emotion *type*.

Goals, standards and preferences databases. These are referred to in the text as **GSP**s. They are the hierarchical frame databases used to represent an agent's concern structure. They are organized around the three categories of an agent's concerns. They hold most of the information used to define an agent's *interpretive personality component*. When an agent is created, the GSP database must be filled in to give it a unique set of concerns. When an eliciting situation is interpreted using this database, attributes of the eliciting condition theory are bound to features in the situation. See chapter 3 for discussion.

Representing concerns of others. An agent may represent the concerns of some other agent by keeping a partial GSP database for that other agent, and filling it in as knowledge about the agent is acquired. Since these databases represent the concerns of other agents, with respect to an observing agent, they are known as *Concerns-of-Others*, or *COO*, databases. Since they are generally learned by the agent, COOs are usually incomplete, and may contain erroneous interpretation schemas as well. See section 2.8 for a discussion.

Emotion Eliciting Condition Relations. The process that matches frames in the GSP (and COO) databases against an eliciting situation will, if the match is successful, reduce the eliciting situation to a set of bindings. Some of the bindings represent values for two or more of the nine attributes of a special relation known as the *emotion eliciting condition relation*. Different patterns of bindings for the attributes in this relation, and different values, give rise to different emotions. In the text these are abbreviated as *EEC* relations. See section 2.3 for discussion.

Action response categories. Once an agent is in an emotional state, it will manifest this emotion in one way or another. Some of these manifestations naturally fall into one functional category, while others fall into another. For example, *trembling* and *breaking into a cold sweat* may be categorized by the *somatic* characteristics they share. We have specified approximately twenty different groups (there is some variation between the emotions) to which the various action responses belong. See figure 4.1 for a complete listing of the action response categories for *gloating*, and section 2.6.1 and chapter 4 for discussion.

Frame types. Construal frames, which interpret emotion eliciting situations (also represented as frames) must be retrieved from an agent's GSP database at the time the situations arise. The retrieval of candidate construal forms is a feature-indexing problem. A satisfactory solution to this problem is beyond the scope of the current research. We sidestep it in the Affective Reasoner by simply giving each eliciting situation a *type*, such as an *arrival-at-destination* frame type, or a *passenger-pays-driver* frame type. Construal frames of a particular type are candidates for interpreting eliciting situations of the same type.

The structure of simulation events. The Affective Reasoner is designed around a simulation engine. Initial *simulation events* are placed in the system's event queue. When the simulation is started these *simulation events* are processed, and spawn further *simulation events* which are then placed in the event queue, ad infinitum. In this manner, once the simulation is started, it continues until it is halted. Some of the simulation events simply drive the system, moving icons around the screen and so forth, and are not of theoretical interest. Other simulation events have *eliciting situation frames* attached to them which, when matched against the concerns of agents, can initiate emotion processing.

2.2 The construal process

The construal process is covered in detail in chapter 3. Here we discuss how a matcher is used to identify and instantiate internal schemas called *construal frames* which are used to interpret eliciting situations in the simulated world for the simulated agents.

When a simulation event occurs which has an attached *eliciting situation* frame, agents appraise the situation's relevance to their concerns by trying to match it against internal schemas representing the eliciting conditions for the various emotions. Has a goal been achieved? If so, is the goal important enough to generate an emotion? Has a standard been violated? Who is responsible for the blameworthy act? The answers to such questions result in an *interpretation* of the event with respect to the eliciting conditions for emotions in terms of the observing agent's concerns. If the interpretation is such that the eliciting conditions have been met, and certain feature thresholds have been reached (e.g., was *enough* money lost...), then an emotion will result.

The internal schemas which map emotion eliciting situations into the eliciting conditions for an emotion for some agent are represented as frames, called *construal frames*. These construal frames are in an inheritance hierarchy. Leaf node frames inherit slots from ancestors, so that many attributes need only be specified once, in some ancestor. The frame system is based on XRL [Charniak *et al.*, 1987] but has been extended so that slots may contain pattern-matching variables and attached procedures. Since variables allow us to create generalized frames, many different instances of a certain situation type may match the same construal frame, albeit with a different set of resultant bindings for the variables produced by the match.

Figure 2.3 shows how inherited slots are used to match eliciting situation frames. In this figure the slots A through E come from three different frames. *Frame 3* is the leaf node construal frame, and should be thought of, conceptually, as containing all five slots. These five slots match the slots for the eliciting situation, pictured at the bottom of the illustration. Note that the hierarchical nature of the GSP database is purely for the convenience of capturing an agent's concerns in frames. There is no functional reason for the hierarchy (i.e., there is no run-time processing dependent upon the hierarchy); the hierarchies of frames could be compiled by collapsing them into "flat" representations of the leaf node frames, so that inherited slots would be propagated down and included in each of the containing frame's leaf-node descendants.

Once candidate frames have been selected as potential matches for an eliciting situation (using the situation frame type as discussed above), the matcher comes into play. This matcher uses a specialized unification algorithm, and is discussed in chapter 3. If a match between the emotion eliciting situation frame and a particular construal frame succeeds then the eliciting situation is considered to be of concern to the agent, as interpreted by the construal frame. At the same time, a set of bindings is produced from the unification of the pattern-matching variables and the features of the situation, and any attached procedures (see below). The feature values contained in these bindings, and specifications within the construal frame, are used to determine the exact nature of the eliciting situation with respect to the agent's concerns. In addition, these bindings are later used to specify the possible responses

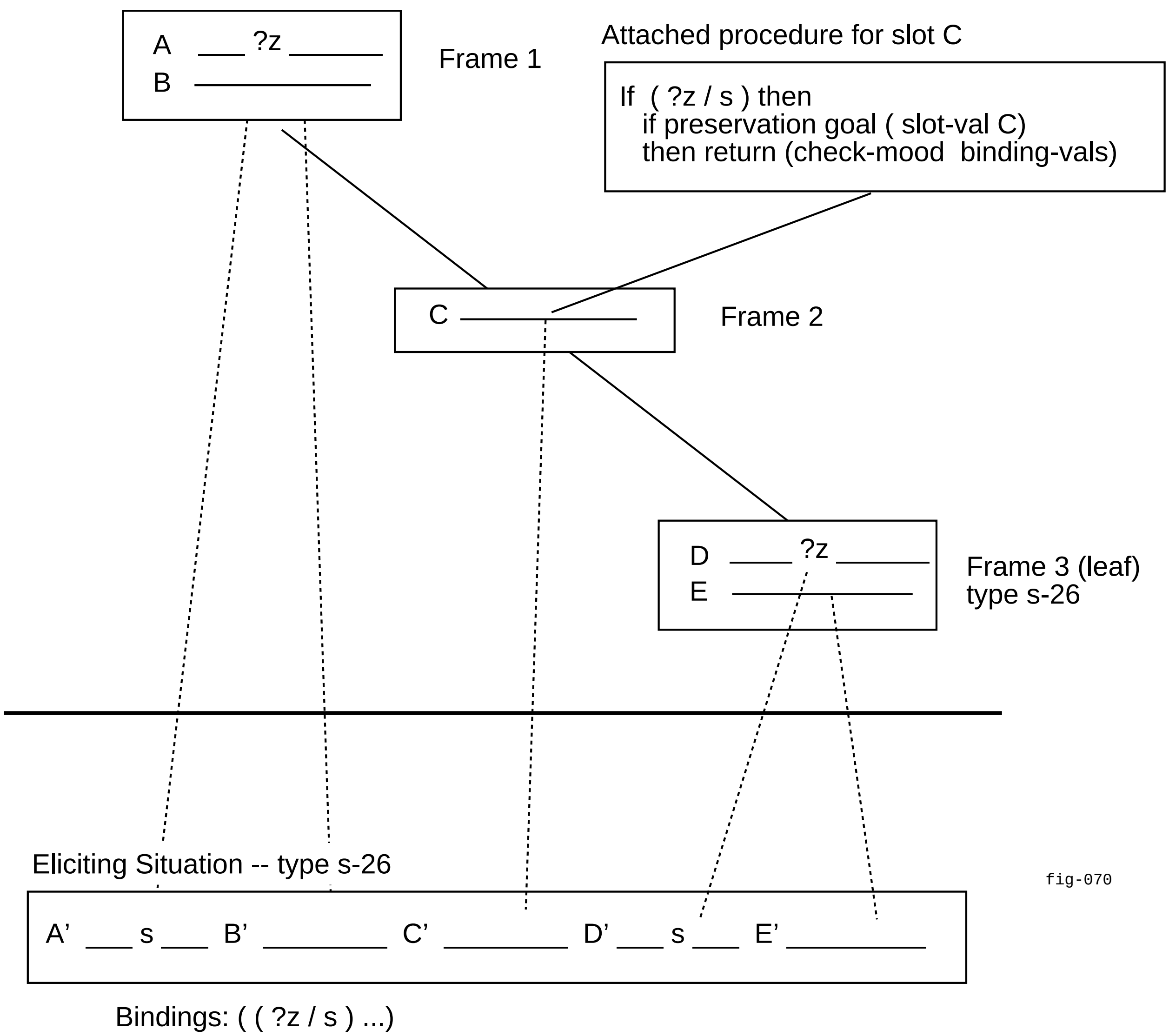


Figure 2.3: Inheritance of slots in a construal frame

as well. Output from the match process is either an indication that the match has failed, or an instantiated construal frame containing bindings for eliciting condition attributes such as desirability for the agent, praiseworthiness of the act, attractiveness of the object, and so on.

In figure 2.3, *Frame 2* is shown with a procedure attached to slot C. Such procedures allow working memory values to be incorporated into the match process. They also allow fine-tuning of the match process by calling predicate functions which may base decisions on the current set of bindings. Finally, these attached procedures may also contribute to the current set of bindings, since the (new) set of bindings is returned from the procedure calls.

Frame 1, slot A and *Frame 3, slot D* are shown as sharing a common variable *?z*. In the eliciting situation, this feature is instantiated as s in slots A' and D'. During the match, since s unifies with s, this portion of the unification process succeeds and a binding of the variable *?z* to the constant s is created and added to the bindings list.

An important point to note is that within this context, *no event has meaning to an agent until after it has been filtered through the concerns of that agent.* Without going into the philosophical foundations of this argument (but see [Regoczei and Hirst, 1991] for a discussion of this) it should be evident that people work this way: extraneous cognitive and perceptual information is filtered out of the input stream, and the relevant situations that do pass this filter are sent on, along with the interpretation of *why* they are relevant. Not all interpretations are relevant to the human affective machinery (i.e., the information that a room is dark and that the light needs to be turned on is not likely to cause an emotional response), but a significant amount of binding to affective inference structures occurs at the time eliciting situations are assessed. For example, affective states are intertwined with expectations. To form a match between a stored expectation and some current situation for the purpose of (dis)confirming the expectation, one must bind features of the new situation to the expected facilitation or blocking of the stored goals, to the expected upholding or violation of the stored standards, and so forth.

2.3 Emotion Eliciting Condition relations

Once an initial interpretation has been made, *Emotion Eliciting Condition* (EEC) relations are constructed. These are, essentially, a set of features derived from eliciting situations and their interpretation which, taken as a whole, may meet the eliciting conditions for one or more emotions. Different patterns of features lead to different emotions. The source of these features varies. Some features, such as the names of agents, are directly derived from the situation. Other features, such as the *desirability* of the event, are derived from matching the situation against the agent's concern

Figure 2.4: The Emotion Eliciting Condition relation

self	other	desire-self	desire-other	pleasing-ness	status	evaluation	responsible agent	appeal
(*)	(*)	(d/u)	(d/u)	(p/d)	(u/c/d)	(p/b)	(*)	(a/u)

Key to attribute values	
abbreviation	**meaning**
*	some agent's name
d/u	desirable or undesirable (event)
p/d	pleased or displeased about another's fortunes (event)
p/b	praiseworthy or blameworthy (act)
a/u	appealing or unappealing (object)
u/c/d	unconfirmed, confirmed or disconfirmed

structure (i.e., they are contained in the bindings produced when a construal frame from the agent's GSPs is matched against the situation frame); still other features, namely *pleasingness* and *status,* are dependent upon dynamic information and so must be derived partially from working memory. The complete set of features comprising the EEC relation is shown in figure 2.4, and is discussed below:

Self. This attribute represents the identity of the agent experiencing the emotion. The value for this attribute is derived from the situation. It is always bound to an agent's name. If two agents are involved in some situation, and it is relevant to the concerns of each of them, then two sets of emotion eliciting condition relations will be produced, with *self* bound to a different agent in each of the different sets. Three agents will yield three sets, and so on.

Other. This attribute represents the identity of some other agent about whose fortunes the *self* agent may have an emotion. Such an emotion will result only when *pleasingness* (for *self* – see below) is also bound, on the basis of a relationship stored in working memory. The value for this attribute is derived from the situation. When the attribute is bound, it is always bound to an agent's name, but it is never the same value as *self*.

Desire-self. This attribute represents the *self* agent's assessment of the eliciting situation as an event. If he interprets the situation as one where a goal of his is blocked then the value of this attribute is *undesirable*; if he interprets it as one where a goal is achieved the value is *desirable*. If he does not interpret the eliciting situation as relevant to its goals then this attribute is not bound.

The value for this attribute is derived from the agent's construal of the eliciting situation.

Desire-other. This attribute represents the *self* agent's assessment of the eliciting situation as an event relevant to the desires of another agent. If a *self* agent has a Concerns-of-Other representation for some other agent (either specific or default – see section 2.8), he can interpret that agent's reaction to a situation by "imagining" what it is like for the other agent. If it is perceived that a goal of the other agent is blocked the value of this attribute is *undesirable*, if it is perceived as having been achieved, the value is *desirable*, otherwise the attribute is not bound. The value for this attribute is derived from the agent's construal of the eliciting situation.

Pleased. This attribute represents the valence of the *self* agent's response to the emotions of another agent. It is used strictly in situations in which two conditions hold: (1) an eliciting situation perceived as a goal-relevant event "happens" to another agent (bound to the attribute *other*), and (2) the agent bound to the attribute *self* is related to the agent bound to *other* either by *friendship* or *animosity*. Once the value of *other* has been determined from the eliciting situation, the value of *desire-other* has been determined from a Concerns-of-Other representation (chapter 2.8), and the relationship (friendship or animosity) has been determined from working memory, the value of *pleased* can be determined. For example, if an agent has a friend, and the agent observes some situation in which he perceives his friend to have achieved one of his (the friend's) goals, the agent can be pleased about it. This attribute is derived from the *desire-other* attribute in conjunction with working memory.

Status. When bound, this attribute represents the status of an expectation. When a situation perceived as an event takes place, in most cases it has no *status* attribute associated with it (i.e., the status attribute is not bound). The question of expectations does not arise – an event just happens and that is the end of it. On occasion, however, events may lead to expectations on the part of agents. They may also confirm or disconfirm prior expectations. The representation of such expectations is problematic and is treated in depth in section 3.3. Briefly, a situation perceived as an event may be *unconfirmed,* meaning that the *self* agent perceives it as possible, but as not yet actually having taken place. Once an agent has had this perception, a schema representing the expected outcome is stored in the agent's expectation database. At this point, if some situation matching this schema comes about, then the situation will be interpreted as having been *confirmed* or *disconfirmed* depending on the nature of the confirming situation. Some of the values for this attribute are taken from the situation; some are derived from prior expectations (see sections 3.2 and 3.3).

Evaluation. This attribute represents the *self* agent's assessment of the eliciting situation as containing a praiseworthy or blameworthy act. If the agent perceives that such an act has been performed by some (not necessarily other) agent then he may evaluate this act as in accordance with, or perhaps in conflict with, his principles. If it is in accordance with his principles the *evaluation* attribute is bound to *praiseworthy*. If it is in violation of his principles the attribute is bound to *blameworthy*. Otherwise, the attribute is not bound. The value for this attribute is derived from the agent's construal of the eliciting situation.

Responsible Agent. This attribute represents the agent that the *self* agent holds responsible for a perceived praiseworthy or blameworthy act. If it is bound, it is always bound to some agent's name. It may be bound to the same name as the *self* attribute. The value for this attribute is taken from the agent's construal of the situation.

Appealingness. This attribute represents the *self* agent's assessment of the eliciting situation as containing an attractive or repulsive object. If this attribute is bound its value is either *appealing* or *unappealing*, according to the agent's tastes. The value for this attribute is taken from the agent's construal of the situation. Note that the object itself is not relevant to the generation of emotions and so is passed in the bindings list rather than in the emotion eliciting condition relation.

We now present two simple examples. In the first we suppose that Tom has the goal of making it through the day without getting a speeding ticket. If he achieves his goal then this situation is construed by him as *desirable*, and if he fails to achieve the goal it will be construed as *undesirable*. (We assume the simple case where he has no expectations one way or the other.) Let us suppose that the latter case obtains, and that *Tom* is stopped by a policeman and given a traffic ticket. The EEC relation derived from construing this as an undesirable situation would have *?self* bound to *Tom* and *?desire-self* bound to *undesirable*, and is shown in figure 2.5. This configuration of features is necessary (but not necessarily sufficient – see section 2.4.2), to meet the eliciting conditions for a *distress* emotion (see section 2.1). The features for which no value is shown may be thought of as bound to the value *none*, although this is not strictly necessary. In general, each different interpretation of an eliciting situation will produce a different EEC relation. Thus, for example, interpretations of an eliciting situation regarding the attractiveness of an object, which involve determination of the *appeal* attribute, will not be mixed with interpretations of the situation as containing a praiseworthy or blameworthy act, which involves determination of the *evaluation* and *responsible-agent* attributes. This is discussed later in more detail in the sections on compound and multiple emotions (sections 2.4.1 and 2.4.3).

Figure 2.5: An Emotion Eliciting Condition relation for *distress*.

self	other	desire-self	desire-other	pleasing-ness	status	evaluation	responsible agent	appeal
Tom		u						

Figure 2.6: An Emotion Eliciting Condition relation for *pity*.

self	other	desire-self	desire-other	pleasing-ness	status	evaluation	responsible agent	appeal
Harry	Tom		u	d				

In the second example, Tom again gets a speeding ticket. This time however, he has a friend, Harry, who observes the situation. It is Harry's point of view with which we are concerned: he is *displeased* about the bad fortune of his friend Tom, and feels sorry for him. To represent this second point of view, we need a second Emotion Eliciting Condition relation. Such a relation is shown in figure 2.6. In this case *?self* is bound to *Harry, ?other* is bound to *Tom, ?desire-other* is bound to *undesirable,* and *?pleasingness* is bound to *displeased.* This configuration is necessary and sufficient to meet the eliciting conditions for *pity.*

Note that it is not necessary or even likely that Harry has goals with respect to Tom not getting a ticket. He is only interested in his friend Tom's general welfare. Probably this eliciting situation is not of direct interest to Harry at all, although it is possible that he has, for example, made a bet about Tom getting speeding tickets.[2] In any case Harry's own direct personal goals can be seen as distinct from the (possibly conflicting) goals he may have with regard to Tom's fortunes.

Many goals, standards and preferences may be specified in a hierarchy so that features may be inherited. The interpretation schema of a *getting-speeding-ticket* situation might be represented as a *set* of goals, where each goal is a subset of a larger goal. In this approach the situation itself is considered to block a low-level goal, *getting no speeding tickets.* This in turn may be a subgoal of a *retain money* goal which may be a subgoal of, respectively, *retain resources, increase profits, be wealthy* and *be secure,* which is a high-level goal. Inherited features might be that *money is quantifiable,* that *amount of money lost/found is important in calculating thresholds,* that *money goals can have both positive and negative outcomes,* and so forth. The concepts represented in these slots, both local and inherited, are used to interpret an eliciting situation and to reduce it to the nine attributes of the Emotion Eliciting Condition relation.

[2]We can, for example, imagine a situation in which Harry has made a bet that his friend Tom *will* get a speeding ticket. In this case, if Tom does, in fact, get a speeding ticket then Harry might be displeased for Tom but pleased for himself.

Figure 2.7: Compound emotions

Goal	**Standard**	**Emotion**
achieved	upheld	gratification (self), gratitude (other)
blocked	violated	remorse (self), anger (other)

One final point is important. The number of EEC relations and the number of different emotion instances following an eliciting situation can both vary. Sometimes one eliciting situation can generate more than one EEC relation, even for the same agent, and sometimes one EEC relation can generate more than one emotion. These issues are discussed in depth in the following section.

2.4 Generating emotions

Once a set of EEC relations has been generated for an agent, the relations are used for generating emotions. This section discusses how this is done within the context of the underlying cognitive theory. There are a number of complex issues. The first of these has to do with the generation of mixed and multiple emotions. Since the discussion is lengthy it will have a section devoted to it. Following this we show how reasoning about multiple emotions, and other complicating issues, are dealt with to produce actual emotion instances from the set of eliciting conditions.

2.4.1 How compound emotions are generated

Compound emotions are generated when an agent is seen as being accountable for some blameworthy or praiseworthy act that has like-valenced consequences with respect to the goals of some, not necessarily different, agent. The four possible compound emotions are shown in figure 2.7.

As an example, consider again the eliciting situation where Tom gets a speeding ticket. Suppose that *Tom* blames the policeman for giving him a ticket by invoking a principle that says, in effect, that *Policemen should not stop motorists for speeding when they are traveling at the same speed as everyone else.* Taken separately, the blocking of the get-no-speeding-tickets goal leads to *distress*, as discussed above, and the violation of the principle leads to *reproach*. Together, however, this combination represents the eliciting conditions for *anger*, and this emotion replaces the other two.

In the Affective Reasoner, a construal leading to a compound emotion is represented as a single compound EEC relation derived from combining two individual EEC relations into one. In the speeding ticket example, this would mean replacing

Figure 2.8: An Emotion Eliciting Condition relation for *anger*.

self	other	desire-self	desire-other	pleasing-ness	status	evaluation	responsible agent	appeal
Tom		u				b	policeman	

the EEC relation derived from the construal of a goal being blocked and the EEC relation derived from the construal of a standard being violated with a single EEC relation representing both. This EEC relation is shown in figure 2.8.

2.4.2 Subsumption of constituent emotions

Does anger really subsume distress? Do compound emotions always subsume their constituent emotions? That is, in feeling anger does a person *also* feel distress and reproach? This is a difficult question. Unfortunately, since we are implementing a platform that generates discrete instances of emotions, we cannot finesse this issue. Either they do or they do not. There can be no middle ground until the eliciting condition theory is extended, and the EEC relations extended. For our purposes here we have made the arbitrary decision to *replace* the constituent emotions with the compound emotion.

There are some technical details to be considered with respect to this issue. Previously we stated that the EEC relation shown in figure 2.5 was necessary, but not necessarily sufficient to meet the conditions for *distress*. All of the attributes necessary for *distress* (i.e., *?self* and *?desirability*) are bound, but if, in addition, the attributes for *?evaluation* and *?responsible-agent* are bound then the generation of *distress* may be blocked. For this reason, the presence of the first two attributes is not enough to guarantee the generation of the *distress* emotion. Similarly, the presence of bindings for *?self*, *?evaluation* and *?responsible-agent* is necessary for the generation of the standards-based emotions, but not sufficient since the elicitation of these emotions may be blocked by the presence of a binding for *?desire*. In general, necessary conditions are those specified above, while sufficient conditions further require that *?evaluation* not be bound to a similarly-valenced value for the goal-based emotions, and that *?desire* not be bound to a similarly-valenced value for the attribution-based emotions.

Except for these two cases the bindings for the remaining attributes are ignored. For example, a binding for *appealingness* in an EEC relation has no bearing on the generation of either the goal-based or attribution-based emotions, or the compound emotions. The EEC relations in figure 2.5 and figure 2.6 have, respectively, seven of nine and five of nine attributes with no bindings shown. In most cases these attributes will, in fact, have no bindings. This is because construal frames tend to calculate and bind only those values necessary for the particular type of interpretation they represent. For example, a construal frame for attraction interpretations is not likely to create bindings for *?responsible-agent*, and a construal frame for fortunes-of-others interpretations is not likely to have bindings for *?desire-self*. However, sometimes bindings for these unrelated attributes do show up in an EEC relation (a common unrelated attribute is the one for *?other*). When they do they are simply ignored.

2.4.3 Multiple emotions

An agent may have many concerns relevant to a single type of situation. Situations seen as events may facilitate or interfere with several goals at once or may achieve one goal while interfering with another. The situations perceived as containing a praiseworthy or blameworthy act may uphold or violate several principles at once, or may both uphold *and* violate different principles at the same time. Situations seen as containing objects may both attract and repulse at the same time for different reasons. These different, effectively simultaneous, construals lead to multiple emotions.[3]

There are no restrictions placed on which construal frames may be placed in an agent's GSPs. Any construal frame may coexist with any other construal frame. This means that for any given situation EEC relations may be produced which lead not only to multiple emotions, but to *conflicting* emotions. We see this not as a weakness, but rather as a requirement of emotion reasoning. It is, after all, consistent with the way humans perceive the world. We have all had situations which lead to happiness on our parts, which nonetheless have a bittersweet quality to them because there are aspects of sadness as well. Most of us will also have had the experience of happiness over the achievement of some goal, while also feeling ashamed of the way in which we achieved it. These are examples of conflicting emotions, and a system that reasons about emotions needs to be able to represent them.

A special class of multiple emotions is that where an agent has the *same* emotion, but for different reasons. An agent might be happy, for example, when he wins at a game of cards. On the one hand he might be happy because he is competitive and simply likes to win when he competes. On the other hand he might have made a bet on the outcome and have achieved a money goal as well. The generation of these emotions is straightforward, but an associated activity, that of *explaining* the emotion is more difficult. Why was the agent happy? Is he happy twice for two different reasons, or is he just twice as happy for a combination of reasons? This may seem an esoteric point, but from an implementation standpoint it has to be resolved, especially with respect to the following discussion on the generation of multiple instances of compound emotions. In the Affective Reasoner, we have simply generated two instances of the same emotion in response to a single eliciting situation.

[3]Ortony et al. claim that these multiple construals are not, and cannot be, simultaneous, but rather that agents oscillate between them. The Affective Reasoner does not make this distinction, although such oscillation could be represented.

2.4.4 Problems with discrete instances of simultaneously occurring emotions

The general problem is that the emotion types are treated as discrete, non-overlapping categories. An agent is either in one state, another state, or both. This affects the explanation of emotion instances. For example, suppose that one of the concerns of some agent, Tom, is that he has a job he likes and another is that he has a job that pays a good wage. A prospective employer calls him and offers him a well-paying job with interesting work. Clearly Tom will be happy about this. In the Affective Reasoner paradigm Tom will be happy because he now has a well-paying job, and he will be happy because he is going to have interesting work.

Now suppose instead that Tom gets a job that pays very well indeed, but that the work is not very interesting. In this case he might have mixed emotions: happy that he will be getting a good salary, but sad that he must now do boring work. In the Affective Reasoner paradigm this situation is handled by generating two distinct emotions for Tom.

So far, the approach we have taken (i.e., generating distinct emotion instances for each EEC relation) is reasonably consistent with the sophistication level of the rest of the system. The approach breaks down however when we come to explanations of action generation for the emotions. Clearly when one is manifesting emotions in response to a situation, then those manifestations should represent an *integrated* response to the situation and to the dominant emotion. It would be ridiculous, for example, to have an explanation that said, in effect, that Tom was *smiling* because he got a well-paying job, but was *laughing* because he got an interesting job. Certainly Tom is both *laughing* and *smiling* simply because he is happy, and he is happy both because he got an interesting job and it pays well.

We address this problem superficially in the conflict resolution component of the action generation module (see section 4.6). The basic idea of this component is to remove actions which are not compatible with one another. While this helps to avoid outlandish behaviors on the part of the system, it does not address the integration issue at all.

One further point is that the makeup of the manifestation of mixed emotions is presumably dependent upon the intensity of those emotions. One emotion may take precedence over another if it is much stronger. In addition, conflicting emotions may, in some cases, tend to counteract the manifestation of each of those emotions. For example, someone who is both happy and fearful at the same time (e.g., when finally getting a chance to bat in the big leagues) may have an intense bodily response expressing both emotions simultaneously. He is unlikely, however, to cower out of fear, because he is also happy; he is unlikely to simply relax in bliss because he is also afraid. Except for the superficial treatment of conflicting expressions of emotions, the development and implementation of a theory of the expression of multiple emotions

Figure 2.9: Explanations for Tom's *gratitude*.

The $d \times a$ approach
not telling secret helped get a good paying job
not telling secret helped get an interesting job
not telling secret helped get a job close to home
not betraying friend helped get a good paying job
not betraying friend helped get an interesting job
not betraying friend helped get a job close to home

is beyond the scope of this work.

2.4.5 Multiple compound emotions

Lastly we must consider the problem of multiple compound emotions. Multiple emotions of the same type can lead to problems when they are subsumed by compound emotions. Suppose that the agent, Tom, who is looking for a job, has three goals: (1) he wants a well-paying job, (2) he wants an interesting job, and (3) he wants a job close to home. In addition he has confided in a friend that he has a felony conviction in his past that he did not mention to a prospective employer who is offering just such a job. He has two standards associated with this confidence: (1) a person should never betray a friend, and (2) a person should not "tell" a secret. The prospective employer calls the friend for a reference and the friend does not mention the prior felony conviction. Tom gets the job. Figure 2.9 shows the possible explanations for Tom's gratitude.

These explanations for the source of Tom's gratitude illustrate the $d \times a$ approach where d represents the desire-based construals and a represents the attribution-based construals. This approach is not very good. While each of the explanations might itself be valid (i.e., such eliciting conditions might well lead to *gratitude*), none of them seems to be what Tom is likely to think of as the cause of his emotions. Still it seems possible that Tom is grateful to his friend for several different reasons.

A second approach, the $d + a$ method, is to group the desire-based construals together and the attribution-based construals together for different explanations, as shown in figure 2.10. The first three explanations represent Tom having gratitude because he construes his friend's actions (all of them taken as a group) as "something admirable" which he assesses in the light of three of his goals. The fourth and fifth explanations represent Tom having gratitude because he construes his friend's act of not telling a secret as helping to achieve his goals (all of them – with respect to this situation – taken as a group) and construes his friend's act of "not betraying him"

Figure 2.10: Grouped explanations for Tom's *gratitude*.

The $d + a$ approach
not telling secret and not betraying helped get a good paying job
not telling secret and not betraying helped get an interesting job
not telling secret and not betraying helped get a job close to home
not betraying friend helped get a good paying job, interesting and close to home
not telling secret helped get a good paying job, interesting and close to home

similarly helping him to achieve each of his three goals. One can imagine someone thinking: "I am grateful because he did some things which were admirable and that helped me get a well-paying job, it helped me get an interesting job and it helped me get a job close to home", and "I am grateful because he was trustworthy with respect to my secret, and that helped me get some things I desired. I am also grateful because he was a good friend to me, and that also helped me get those things I desired." This approach seems palatable and yet still seems to contain more power as an explanation than, "I am grateful because he did some admirable things that helped me get some things I wanted." For this reason, it is the approach we have adopted in the Affective Reasoner.

Obviously this is a matter open to debate: at what point do humans cease to discriminate individual goals and standards and start to group them together? Fortunately, given the coarse-grained nature of the emotion reasoning performed by the Affective Reasoner, this turns out to be not particularly critical to the working of the system. Such multiple compound emotions rarely arise in practice. This issue is important theoretically, however, because it highlights the problems one encounters when taking such a rigidly discrete approach to the classification of emotions and of emotion instances.

2.4.6 The domain-independent Rules

The Emotion Eliciting Condition Theory based on [Ortony *et al.*, 1988] is represented in a separate database as a set of high-level emotion rules. One result of this design choice is that *all* emotion-based processing must, at some point, pass through these rules. In other words, to get from an eliciting situation to the creation of some new simulation event based on the emotional response to the eliciting situation, at least one of the high-level emotion rules must be used.

This has two benefits. First, it helps us to assess the underlying eliciting condition theory: since all emotion processing is based, at least partially, on the rules representing the theory, if the theory is lacking we will soon know it, when we try to represent those situations where its representational power fails. Another way to think of this is as follows: if the system is successful in its attempt to realistically

Figure 2.11: An instantiated emotion template for *anger*.

Tom is angry at Harry	
slot	**binding**
emotion	anger
agent	Tom
situation	sim-event-241...
bindings	((?time / 10:30 A.M.) (?responsible-agent / Harry)...)
responsible-agent	Harry

represent a rudimentary emotion system, then this tends to verify the underlying theory. If the system is unable to successfully represent certain aspects of such a rudimentary emotion system, then we may be able to trace its failings to a failing in the underlying theory.

Second, if the emotion eliciting condition theory has to be changed then only a simple modification to this strictly formatted rule-base need be made to change the system as well. There is exactly one of these high-level rules for each of the twenty-four emotion classes, so changes to the emotion eliciting condition theory for one of the emotion classes would mean, literally, changes to only one rule.

Each of the twenty-four rules has a left-hand side which can be matched by certain configurations of EEC relations, and a right-hand side containing an *emotion template.* When the left-hand side is instantiated by some instance of an EEC relation then an emotion template is bound using the bindings created during the construal process. This emotion template represents an emotional state on the part of the agent interpreting the situation and is the output from the construal phase.

The emotion template has slots for (1) the name of the emotion type, (2) the name of the agent having the emotion, (3) the initiating situation ID number, (4) the bindings created during the construal process, and (5) the additional identification of some agent, in the case of both the attribution emotions and the Fortunes-of-Others emotions. Figure 2.11 shows an instantiated *emotion template* for *anger*.

Once an *emotion template* is generated in response to an eliciting situation, the construal phase of the system has completed its processing. The next section reviews the steps we have illustrated so far.

2.5 Summary of emotion generation

We have described how eliciting situations may arise from the simulation events. These situations may or may not be of concern to one or more agents. If they are, then varied interpretations of the situation may be made, depending upon the makeup of each agent's *interpretive personality component*. These interpretations are reduced to Emotion Eliciting Condition relations, which in turn are used to generate instantiated emotion templates.

Figure 2.5 illustrates the different sources for these emotion templates in the Affective Reasoner, and serves as a summary for this part of the chapter. We now present a brief description of the steps in the diagram.

1. A situation is created when a simulation event is popped off the queue and the state of the simulated world is altered in a way that might be of concern to one or more agents.

2, 3. The frames representing the goals, standards and preferences (GSPs) of some agent are matched against the eliciting situation frame. Working memory is attached to slots in the GSPs and can alter the success of the match, as well as the resultant bindings that are generated.

4, 5. When a match succeeds for some construal frame (and its inherited properties) then the situation is considered to be relevant to the agent's concerns. Since an agent may simultaneously have many different construals of the same situation, multiple interpretations of that situation may result. When all matches fail, the situation is not considered relevant to the agent's concerns and is ignored.

6. If the situation is relevant to the concerns of the agent, bindings will have been created during the match process. In addition to those bindings created when slots in the situation frame are unified with pattern matching variables in slots of the construal frame (including inherited slots), additional bindings may come from attached procedures and from working memory.

7. A basic Emotion Eliciting Condition Relation (EECR) is created for each construal of the situation.

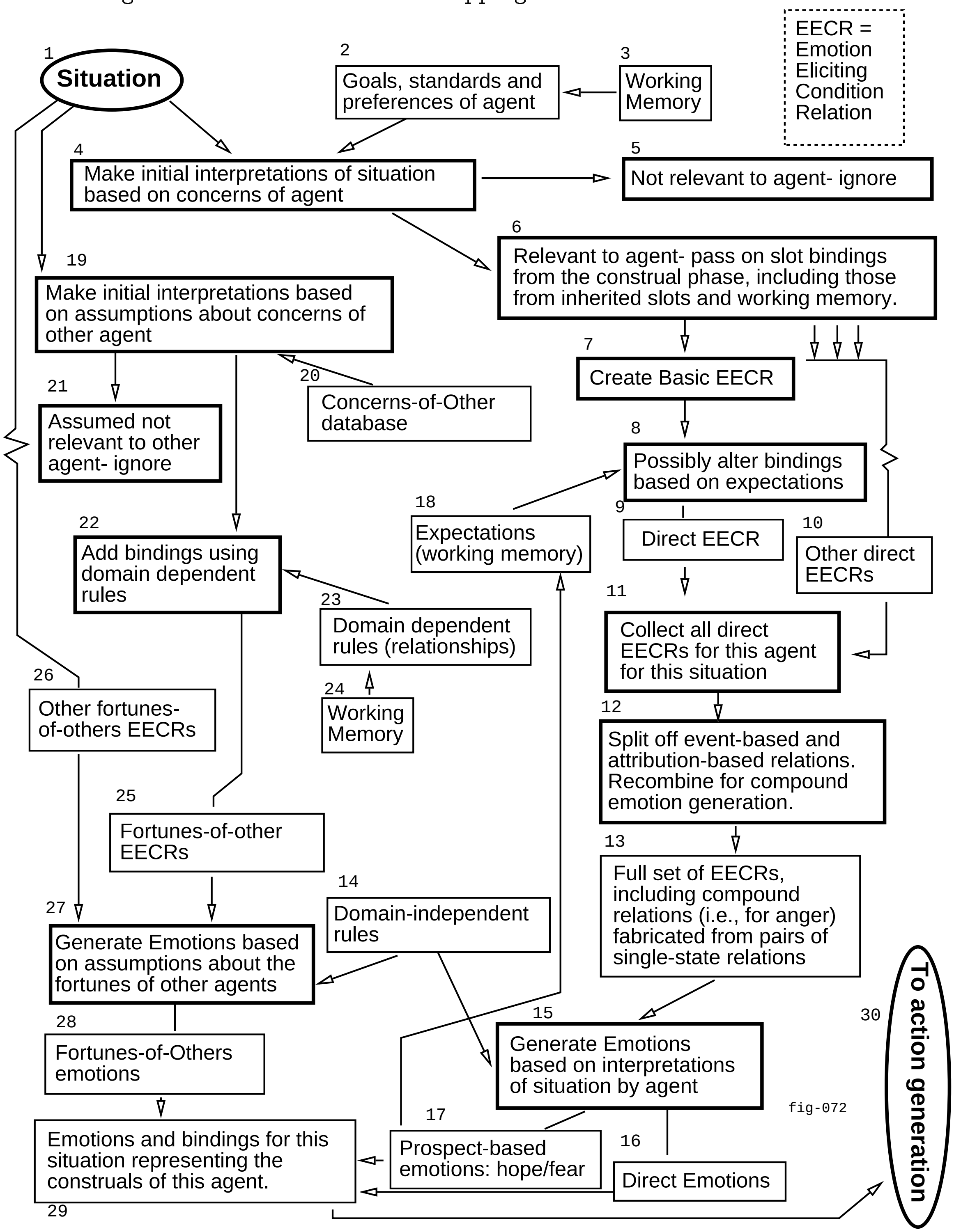
8, 9. The *status* attribute of the EECR may be changed to *confirmed* or *disconfirmed* if the situation is relevant to a stored expected outcome.

10, 11. Since there may be multiple construals there may be multiple EECRs. These are collected together before further processing.

12, 13. The event-based construals and attribution-based construals are split off into two (non-intersecting) sets. They are recombined, as applicable, to form the compound-emotions EECRs, thus subsuming the component EECRs. When recombination is *not* applicable (e.g., in the case of a blameworthy act helping the agent to achieve a goal) the original EECRs are passed along instead.

14. The domain-independent rules contain the backbone of the emotion eliciting condition theory. All emotions are generated using these rules. When an EECR matches the left hand side of one of these rules, an emotion instance is generated.

Figure 2.12: Structure of the mapping from a situation to emotions



15, 16, 17, 18. Those emotions not about the fortunes of others are the *direct* emotions. A subclass of these is the prospect-based emotions of *hope* and *fear*. These latter emotions stem from unconfirmed events and so generate *expectation* frames. These frames are stored, and may be used to interpret future situations.

19, 20, 21. For each other agent involved in the situation, the target agent's representations of their concerns (i.e., the Concerns-of-Other databases the target agent maintains for them) are used to interpret the situation, similar to the process in steps 2, 3 and 4.

22, 23, 24. If the situation is determined (partly by assumption) to be of concern to the other agent, then the domain-independent rules are used to determine the meaning for the target agent. Is he pleased about the outcome for the other agent? Relationships between the agents are stored in working memory and may be altered as the simulation progresses.

25, 26, 27, 28. Each of the resulting fortunes-of-other EECRs is used to produce an emotion with the domain-independent rules.

29, 30. The *direct* and *fortunes-of-others* emotions are all collected into a group and passed to the action generation module. Included in each emotion instance is set of bindings from both the original match and the intermediate processing.

2.6 From emotions to actions

This topic, *Action generation*, is covered in detail in chapter 4. We discuss it briefly here to place it in larger context of the system. Many questions about the theoretical reasons for implementation decisions are raised, but are not addressed here since they will be covered in the later chapter.

In this section we use the terms *temperament trait, response action*, and *emotion manifestation*, all of which need clarification. Borrowing from Reber [Reber, 1985], we define *temperament traits* to be the enduring characteristics of an agent that can serve an explanatory role in accounting for the observed regularities and consistencies in an agent's behavior with respect to the manifestation of emotions. The *temperament traits* may be thought of as descriptive theoretical entities, whereas their representation, the structures making up the *response action categories*, may be thought of as the prescriptive mechanism used for generating such manifestations.

Once an emotion state has been established for some agent, processing of his action response (i.e., the manifestation of his emotion) is initiated. Responses are divided into about twenty response categories, described later in this section. These categories are based on the work of Gilboa and Ortony [Gilboa and Ortony, 1991]. The categories are descriptive in nature and are intended to cover the entire set of possible responses, irrespective of the nature of the eliciting situation. While the response categories are often the same for each of the emotions (e.g., both anger and joy have *communicative verbal* response categories in which they can be manifested), the *actions* associated with each category are emotion-specific and, of course, must be relevant to the object domain. The actions may be simple tokens (such as <*laugh*>) which are not further constrained by the eliciting situation, they may be templates, such as <*laugh at ?other-agent*>, or they may be mini-plans such as <*(the plan for) get-revenge*>. The latter two are dependent upon the eliciting situation and the construal process for binding values. Our current research has not emphasized the mini-plan form. Response actions may themselves be simulation events and may thus initiate and modulate further processing.

The Affective Reasoner's emotion manifestation lexicon may be viewed as a three-dimensional array whose cells are filled with response actions. The first dimension of the array is the twenty-four emotion classes. The second dimension is the approximately twenty emotion manifestation categories. The third dimension represents a partial intensity ordering of the several response actions for each of the parent cells.[4]

Theoretically, the response-action dimension may be further broken down into two tiers, creating a hierarchy: non goal-directed responses (including expressive and information processing responses), and goal-directed responses (including affect-oriented emotion regulation and modulation, and plan-oriented responses). Since there are no inherited properties from these additional levels, such a theoretical division does not affect processing. Future work would, however, make use of this further breakdown of the domain.

2.6.1 Action Response Categories

Each of the categories that comprise the second dimension of the emotion manifestation space discussed above may be enabled or disabled for a given agent at a given point in the simulation run. If a category is enabled, then actions in that category, for a specific emotion, may be selected as manifestations of that agent's emotions. Taken together the particular pattern of enabled and disabled response categories gives agents their unique *manifestative personality* components. Consequently, the individual categories represent the potential *temperament traits* of an agent. Lastly, these temperament traits are usually enabled or disabled for all of the positive emotions at once, and for all of the negative emotions at once. The categories we employ, based on [Gilboa and Ortony, 1991] are as follows:

> **Somatic responses.** Bodily manifestations such as trembling, going into shock, feeling overall pleasure.

[4] This intensity ordering is not actually being used in the current version of the Affective Reasoner.

Behavioral responses directed towards an inanimate object. Responding to the environment with attention or action focused on an inanimate object: slamming a door, kicking something.

Behavioral responses directed towards an animate object. Responding to the environment with attention focused on an animate object such as hitting someone, pushing someone, etc.

Behavioral responses that are unfocused. Responding through actions without attending to another agent or object, e.g., jumping up and down, and sitting quietly.

Communicative non-verbal responses. Intentionally revealing one's emotional state through non-verbal actions such as smiling, winking, throwing arms up in the air, and shaking one's fist.

Communicative verbal responses. Intentionally revealing one's emotional state through words: "rubbing it in", telling others, etc.

Evaluative self-directed attributions. Ascribing qualities pertaining to one's social status, inherent worth, or moral standing, qualities such as superiority, invincibility, stupidity, and so on.

Evaluative agent-directed attributions. Ascribing qualities pertaining to some other agent's social status, power, inherent worth or moral standing.

Obsessive attentional focus. Excluding other contexts, concepts or agents so as to continually monitor or evaluate some aspect of the emotion or emotion-inducing situation. For example, focusing exclusively on the other agent, a blocked goal, future consequences, and so on.

Repression. Excluding unacceptable desires and impulses from consciousness. For example, denying positive valence.

Suppression. Conscious intentional attempts to exclude thoughts or feelings, such as laughing when sad, showing compassion when gloating, etc.

Reciprocal. Causing the other responsible agent to feel a similarly valenced emotion if not a similar emotion. Examples include threatening and thanking.

Reappraise self. Changing one's assessment of his role, status or inherent value. For example, assessment of one's self as a winner, as a loser, or as more or less powerful.

Reappraise situation. Changing one's assessment of the situation so as to see it as modifiable or insignificant.

Other-directed emotion modulation. Changing another agent's emotional state by, for example, inducing embarrassment, or threatening him.

Situated plan initiation. Responding to the event with an immediate logical response such as running away, ducking, or telling of love.

Full plan initiation. Ruminating to construct or select a complex response and then initiating it. A classic example here is planning revenge.

2.6.2 Action summary

Figure 2.17 shows an overview of the action generation component of the Affective Reasoner. What follows is annotation for the various elements in the illustration:

1, 2, 3, 4, 5. An eliciting situation arises. This is passed directly to the modules that build Concerns-of-Others representations, as discussed in section 2.7 below. In addition, features are extracted and passed, along with features of the action response, to the feature reader for the observation component of the Affective Reasoner, also discussed in section 2.7.

6, 7. As discussed in section 2.4, emotions are generated in response to eliciting situations using the *interpretive personality* components of agents. These emotion instances are passed, along with the bindings created during the construal process, to the action generation module.

8. The emotion instance activates a portion of the action database representing that particular emotion, for that particular agent.

9, 10, 11. Depending upon the composition of *manifestative personality* component for the agent, certain *temperament traits* will be enabled.

12, 13. (Usually) one action is selected for each enabled temperament trait, for the selected emotion, using the bindings generated from the construal process. Together these selected actions comprise the set of candidate actions.

14, 15. The set of candidate actions may contain incompatible manifestations (e.g., *jumping up and down* and *quietly reflecting*). A database of *conflict sets* is used to identify and resolve such conflicts. Incompatible manifestations are removed using a pseudo-random selection algorithm.

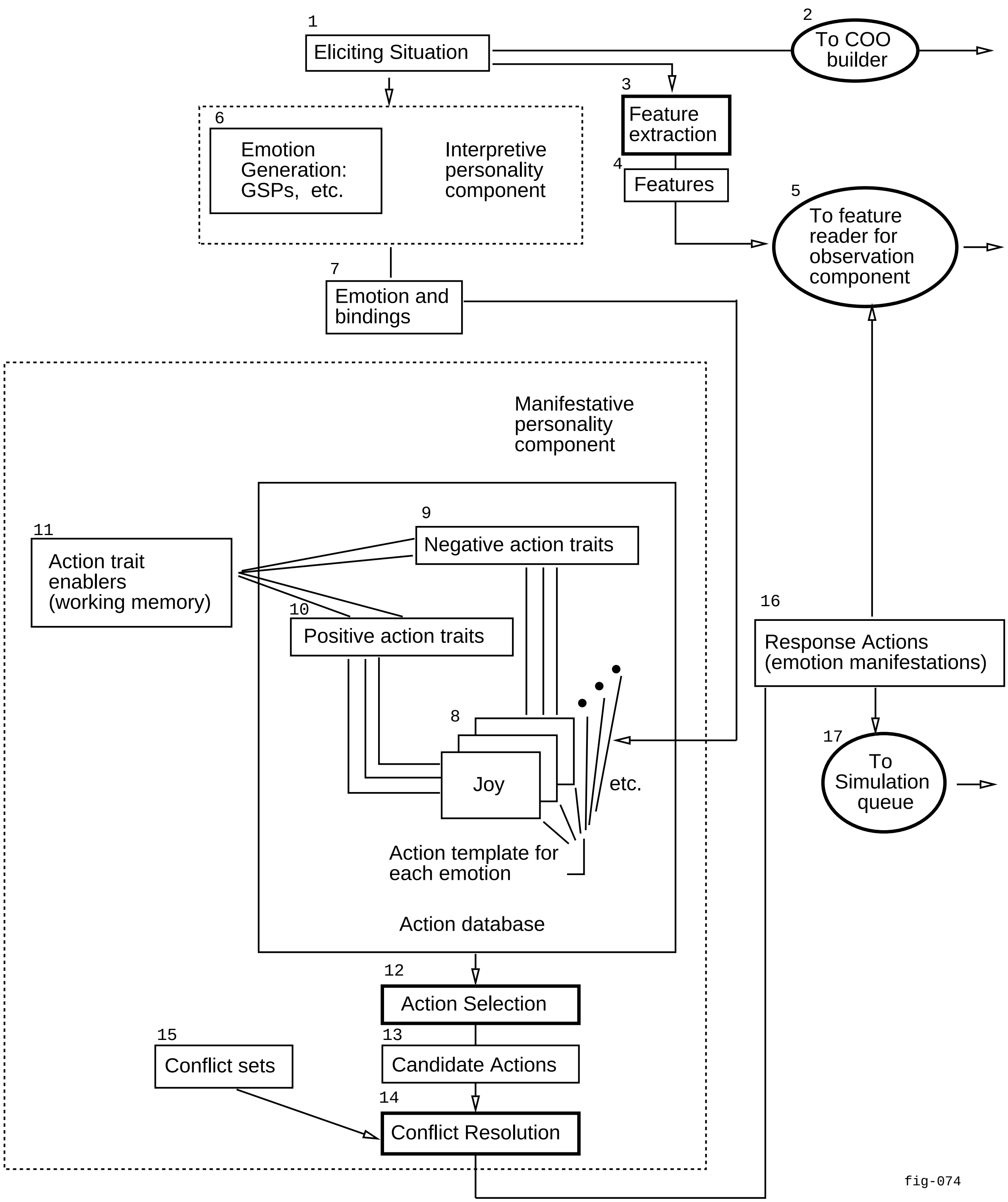


Figure 2.13: The structure of action response generation

16, 17. The response actions (emotion manifestations) are fed back into the simulation queue to possibly modulate future processing. These response actions are also combined with selected features from the eliciting situation and passed to the feature reader for the observation component.

2.7 Observing actions

Because this area of the Affective Reasoner is not fully developed, it will be treated briefly, and only in this section. The observation mechanism described here is closely tied to building Concerns-of-Others structures, a topic covered in the next section. Indeed, much of the motivation for observing the actions of other agents is to decide what their concerns are, for future reference.

Within the context of this section, we define an *emotion episode* as an emotion eliciting situation taken together with an intermediate emotion and any resultant action responses on the part of the participating agents. For an observing agent the episode has two constituent parts: that which the observer can "see", meaning the features of the eliciting situation together with the features of the agent's emotion manifestation, and that which the observing agent cannot "see", meaning the construal of the eliciting situation which lead to some emotion. It is the abduction of the second of these constituent parts which is the subject of this section.

2.7.1 Motivating the Protos approach

The problem of how to observe emotion episodes and make plausible assessments of what the intermediate emotions were is a classification problem. In other words, the question we are asking is: *what emotion makes this episode make sense?*

An emotion episode is classified by recognizing that its attributes imply that it is an instance of a particular emotion class more strongly than they imply that it is an instance of any contrasting emotion class. To classify reliably, agents in the Affective Reasoner must have an accurate representation of the correspondences between attributes and concepts (i.e., between eliciting situation features and emotion-based action responses of some agent, and the emotion classes).

Since our agents are dynamic entities, with varied rudimentary personalities and varied moods, we need a dynamic mechanism for understanding their action responses to eliciting situations. Agents are also very complex: the number of different responses to eliciting situations is large, even in such a limited system as the present implementation. Because of this, to store a useful static representation of a classification system would be prohibitive.

This suggests that our classification mechanism should adapt to current needs of the agents in the system, giving less priority to knowledge that is not used often, and a greater priority to knowledge that is. The mechanism should learn only that small portion of the structure of the emotion manifestation domain that is necessary to do its task in a specific set of circumstances, without being encumbered with knowledge it will not use. To meet these needs, and to perform the sort of flexible reasoning required in any weak-theory domain, we chose to use *Protos*, an exemplar-based knowledge acquisition program developed by Ray Bareiss [Bareiss, 1989]. To quote:

> Protos learns as a byproduct of performing classification under the guidance of a human teacher. When presented with the description of an entity (i.e, a *case*) to be classified, Protos attempts to recall a previous case and to explain its similarity to the new case. When it cannot correctly classify or adequately explain a case, Protos interacts with the teacher to obtain the correct classification and an explanation of why it is correct. It learns by selectively retaining cases and their explanations. [Bareiss, 1989] (page 4)

Protos interacts with the user as the simulation proceeds. It is through this interaction, with the user taking the role of the "emotion manifestation domain expert", that Protos acquires its knowledge about emotion manifestations as they arise in the context of the simulation. The sum of the interaction between the user and Protos is the sum of transference of domain knowledge from teacher to automated student.

Protos' job, in this application, is to give observing agents the ability to make judgments about what emotion is taking place in any given episode. Within the closed process of situation-emotion-action, input is a structure containing features of the situation[5] and features of the emotion manifestation (i.e., the action responses of the agent having the emotion). Output is an assessment of which emotion took place. Additionally, depending on the mode of operation, other input and output may arise which is external to the simulation, as Protos asks for help in classifying cases and assessing the role of features in those cases, and as Protos offers justification for its classification of a new case.

2.7.2 Overview and two examples

The purpose of the observation and classification mechanism is to permit an observing agent to conjecture what emotion some other agent is experiencing. This information is then used as leverage for explaining *why* the other agent acted a certain way in

[5]There is no sophistication in such feature selection. Names and roles of the involved agents are included, and the type of event which has taken place. Obviously a more sophisticated feature selection algorithm would allow for more structure in the case base. While quite interesting, this is not central to the current work.

response to a situation. For example, when Harry sees Tom swearing and shaking his fist after getting a speeding ticket then Harry may look at past cases and discover that other times when someone was swearing and shaking their fist they were angry. Once Harry makes the assumption that Tom is angry he can ask the question, *Why is Tom angry?*, which in turn leads him to reason about Tom's *interpretive personality* component.

Features of a situation and features of an agent's responses to those situations are observed by some other agent. These features are then compared with past cases to see how to classify the episode. Is it a case of *pride?* Is it a case of *fear?* Once a classification has been made, and verified through one of the methods given below, the information used to classify the case is also used to store it for future use. Parts of the new case which are not understood have to be explained to the system by the teacher.

Figure 2.14, for example, shows how a stored exemplar of anger can be adapted to match a new case. Here the observing agent has stored two previous cases of anger. In the first case an angry agent has been observed *shouting, shaking his fist, turning red*, and *scowling*. In the second case an angry agent has been observed *swearing, shaking his fist* and *trembling*. Furthermore, the observing agent knows that both *shouting* and *swearing* are *communicative, expressive, non goal-oriented* emotion manifestations for anger. He also knows that *shaking fist* is highly indicative of anger. A new case is encountered, containing the features, *swearing, shaking fist* and *turning red*. Using the reminding of *shaking fist* the agent retrieves the two previous cases. Substituting *swear* for *shout* in exemplar one (since they have the same expressive function for anger), a match can be created for classifying the case.

Figure 2.15 shows how feature links can be used as censors to represent negative associations between features and categories. In this example *swearing* shows up as a feature in past cases of both anger and joy (from time to time agents may swear as an expression of joy). This is a stronger reminding of anger than it is of joy, but the second feature, *laughing* is not compatible with anger and so causes Protos to choose the first exemplar as the closest match.

When a classification has been made and verified, some features may be unaccounted for. In this case Protos asks the teacher for help. The unaccounted for features are discussed in the light of the present case. Protos has thirty-one predefined relational links, and ten qualifiers. These are used to link features in a case to the classification category. Since Protos is a general purpose representational system a number of relational links, such as *acts on* are not useful for emotion manifestation classification. Others, such as *suggests* and *is mutually exclusive with* are more appropriate.

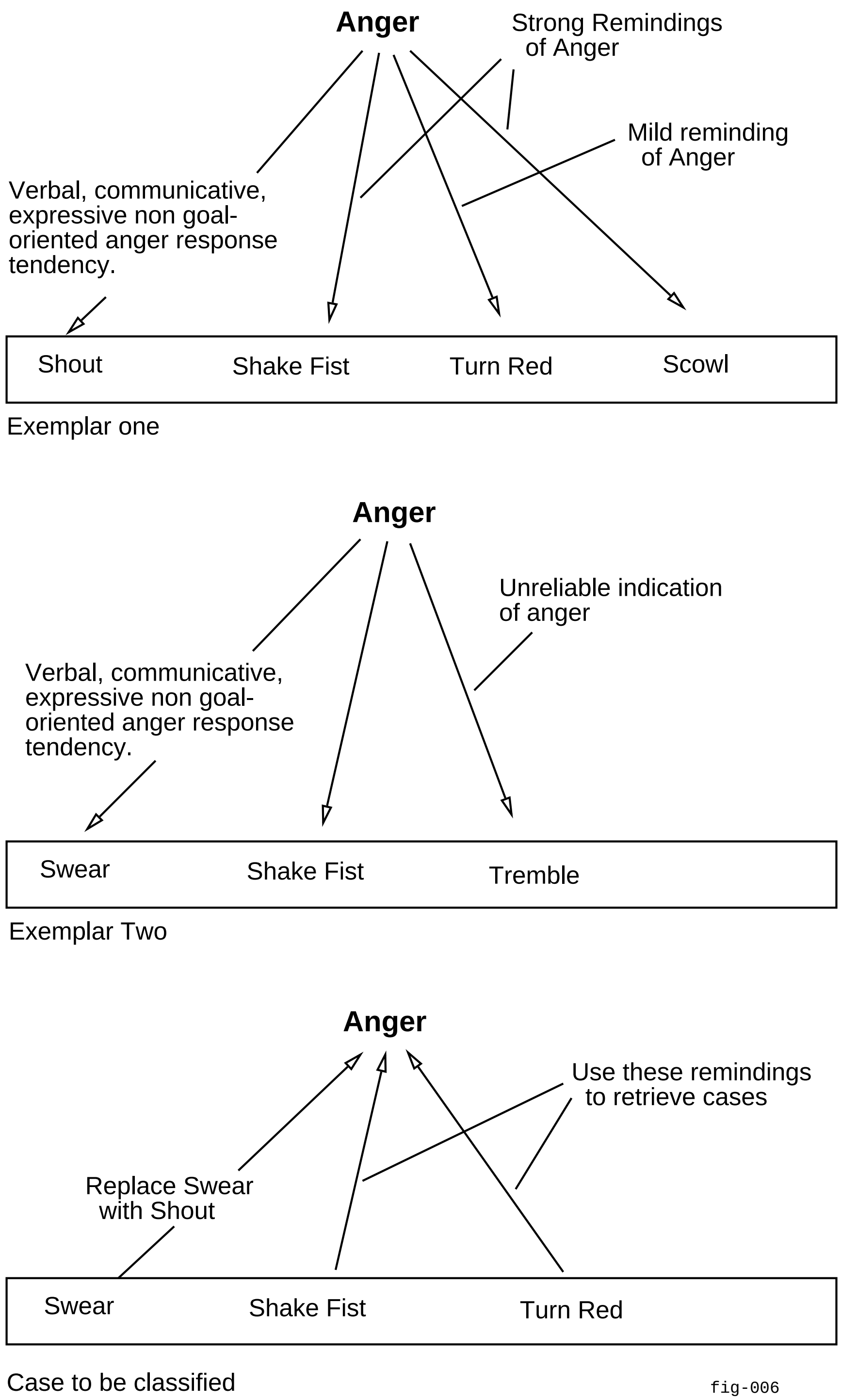


Figure 2.14: Using a case to retrieve anger

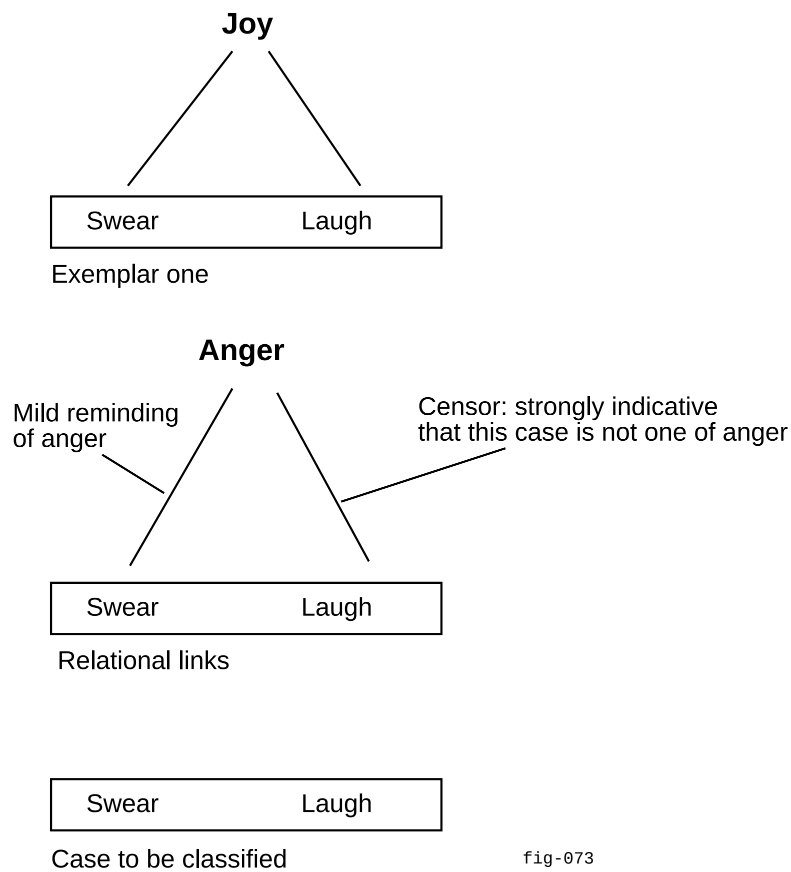


Figure 2.15: Remindings and censors

2.7.3 Observation modes

Observation can take place in one of two different modes.[6] In the first mode the observing agent is simply told by the system what emotions are present in the other agent, as though he were "asking" the observed agent himself, and as though the observed agent always knew the correct answer (certainly not the case with humans). This information is then used for the construction of Concerns-of-Others databases, discussed below in section 2.8. Protos is not used in this mode.

In the second mode the agent uses Protos to attempt to determine what emotion is present, reasoning from past cases about the features present in the eliciting situation and the other agent's response actions. Feedback is given by the "teacher" (in this case the user) and includes both assessment of Protos's emotions classification (i.e., *was the classification correct?*) and clues about the functional, causal, correlational, mutual exclusion, etc. relational links from observed features to the emotion category as well. For example, using Protos to operate on past cases, an observing agent may decide that another agent who is shaking his fist in the air is expressing anger. Protos first wants to know, *Is this correct, that the agent is experiencing anger?* Next Protos wants to know, if it does not already know from previous cases, *what is the relationship between shaking one's fist and anger?* Lastly, Protos wants to know, *how predictive is this feature of anger?* Any new emotion manifestation domain knowledge that Protos has learned in this interaction with the teacher is then stored as part of the current case.

2.7.4 Organizing the emotion manifestation domain as cases

What is a workable, expressive lexicon for emotion manifestation within a limited domain? How can we use the representational power of Protos to classify and store old cases so that they can be used effectively for understanding new cases? To answer these questions fully, we would need a broad content theory of the manifestations of emotions. In addition we would need to capture that content theory in a structural representation of each object domain. (For example, what does it mean when one taxi driver "cuts off" another driver? In what way might this express anger?)

In other words, we must first decide what the relational links are from the attributes of the emotion manifestation domain back to the emotion categories. What is the relationship, for example, between the *non-goal-directed, expressive, behavioral-toward-animate* emotion manifestation category (see section 2.6.1) and anger? Second, we must then decide upon links from constituent tokens in these categories (i.e., the observable manifestations). For example, what is the relationship between

[6] Actually, the Affective Reasoner also has been used to reason in two additional modes, which involve empirical learning. Since the development of these two modes is still in a preliminary stage it will not be discussed further.

the *scowling* attribute and anger? Lastly our content theory must include features from the object domain. For example, what is the relationship between a *taxi driver scowling at another driver* and anger? We do not propose such a content theory here. However, we do suggest a start on a theory that has, at least, given us minimal functionality in this component of the Affective Reasoner.

As will be discussed in section 4.2 we track approximately 1440 different actions for the twenty-four different emotion categories, about sixty apiece. There are obviously many different cases that can be generated from different combinations of these actions. To fully explore the ideas presented here one must then develop a very large library of cases. This, at present, is beyond the scope of this work. What we do present, however, is a six step account of how we have used Protos to classify such cases in a useful way.

Step one. As discussed previously in section 2.6.1, action responses to emotions have been divided into about twenty categories. These categories are purely descriptive in nature but each may be thought of as being associated with some biological or cognitive function. For example, *somatic* responses may well serve the purpose of preparing the body for some sort of action or trauma, and *expressive* responses may serve the purpose of communicating with others so that they can help to meet the needs of the expressing agent. We may use this in building up our structure for organizing emotion manifestations. As discussed in section 4.4 we have generally limited emotion manifestations to single actions within one of the action response categories. We then make a *functional* relational link from the emotion to the action response category in Protos, applicable to each action response token in that category. For example, *shouting* may serve as a *verbal expressive* manifestation of anger. In other words, we are viewing *shouting* as serving some *verbal expressive* "need" of the agent which has arisen as a result of his anger.

Using such an organizational scheme gives us some return in increased expressive power of our cases. Suppose that we have some old cases, each with a different *verbal expressive* action response token. When we attempt to classify a new case we find that it does not match any of the old cases. However, because we have given some structure to the emotion manifestation domain based on the supposed functional nature of the manifestations, we are able to follow links to other cases and substitute one *verbal expressive* response token for another, thus creating a better match.

Step two. The second step in organizing the emotion manifestation domain is setting up the strengths of remindings from response actions to emotion categories. This is extremely useful since most emotions have some *prototypical* manifestations, some *typical* manifestations, some *unlikely but possible* manifestations, and so forth. If we observe some agent *shouting* when the lottery results come out we may not be sure what he is shouting about; if we observe him *shouting* and *scowling* and *shaking his fist* we may be relatively confident that he is angry; if we observe him *shouting* and *smiling* we may believe, instead, that he is joyful. In this case *shouting* is ambiguous, but *scowling, shaking fist* and *smiling* are less so, and help to disambiguate the input.

Step three. The third step is to add mutual exclusion links. *Laughing* and anger are generally mutually exclusive, as are *crying* and reproach. In addition, the mutual exclusion relational link may be qualified, so we can accurately express the idea that some feature is *usually* mutually exclusive with an emotion class, etc.

Step four. The fourth step is to include some causal and correlational links from features anecdotally present in the eliciting situation. *Getting a speeding ticket* may be considered to cause anger (without specifying *why*); the presence of Tom, an agent, may *usually* co-occur with *distress*, and so forth.

In the Affective Reasoner this is handled by including features from the eliciting situation in the structure that holds the action response features. Including features from eliciting situations in the case base provides another way of reasoning from the eliciting situation to the resulting emotions. This does not take the place of reasoning using the Concerns-of-Others representation of the *interpretive personality* of the observed agent however, because, at best, reasoning from observed features will allow us to use correlations to make predictions about those features and the resulting emotions. It ignores the strong-theory reasoning that makes use of the causal nature of the emotion eliciting conditions. In simple terms, using heuristic classification to identify emotion types based on features of eliciting situations may tell us what emotion has taken place, but it will not tell us why. This is discussed further in section 2.7.4.

Step Five. The fifth step is to run the simulation with certain personalities (both *interpretive* and *manifestative*) present in the agents. The makeup of these agents then is such that they each tend to have certain emotions, and they each tend to express them in a certain way. This allows observing agents to build case-bases from repeated observations. In Protos, the repeated use of the case library for classification alters the structure of the library so that certain *prototypical* exemplars are given precedence. This means that agents will form content theories of emotion manifestation tuned to their particular environments. This prototypicality of exemplars is flexible, however, and will change if the moods of the agents change so that the agents either begin to experience a different set of emotions or manifest the emotions they are already having in a different way.

Step six. Lastly, we may choose to keep case bases from previous simulation runs. This process, over time, develops a large set of cases for reasoning about emotions, and tends to de-emphasize the situation-specific knowledge. This case base corresponds to "built-in" knowledge about emotion manifestations and emotion-initiating situations that may be given to an agent at the start of a simulation, as though that agent had a long history of emotional interactions with others on which to draw.

Implicit representation of manifestative personalities

As discussed in section 2.8, consider that agents in the Affective Reasoner maintain (imperfect) representations of the concerns of other agents (i.e, copies of what is assumed to be the other agents' *interpretive personalities*). This allows them to explain why other agents respond in the way they do to situations in their world. It also allows them to predict how another agent may respond to a situation, which in turn might give rise to a fortunes-of-others emotion on the part of the observing agent. In simple terms, this allows an agent to reason as follows: "I know how Tom will interpret this situation, therefore I know what emotional state he is in." Different agents have different concerns. Representing the different concerns of the various agents allows an observing agent to represent them as different individuals with respect to situations that arise in the simulation.

Absent from this model of other agents are the unique ways in which the individual agents manifest their emotions. In fact, no explicit model of this is kept. Such knowledge would be equivalent to knowing that, *Tom shakes his fist when he gets angry,* and *Harry always wrings his hands when he gloats.* However, since an agent's name is always among the features of a situation it is still possible to associate an agent with typical cases of a certain emotion. While it is true that an observing agent cannot reason from an explicit representation of some agent's *manifestative personality* to explain or predict actions, he nonetheless has some way of associating that agent with certain actions, and tying those associations into an emotions classification, through the inclusion of eliciting situation features in his case base.

For example, we may see that Tom often glares when he is angry. Since we do not have any representation of Tom's *manifestative personality* component we cannot store the information that Tom often glares as an expression of anger. This prohibits us from reasoning backwards from this particular manifestation to the emotion, except in the general case. On the other hand, we *do* have knowledge that says that the features Tom and *glaring* are often highly correlated with cases of anger. Should some other agent than Tom be glaring, we have less confidence that anger is present. In this way we have linked Tom to cases where he is glaring in anger without having an explicit representation of this for Tom.

The second point to consider with respect to the storing of situation features in cases is that this does not take the place of keeping *interpretive personality* representations for other agents. As will be discussed below in section 2.8, one of the important aspects of these representations is that they allow an observing agent to explain emotion-based responses in terms of *emotion eliciting conditions* and their mapping to *eliciting situations* within the object domain. Simply storing situation features in cases does not allow for this type of explanation to be made.

2.8 Personalities of others

In a world of interacting agents it is useful for an agent to keep a model of other agents with whom he interacts. Emotions are signs of how those other agents have interpreted the world around them, and therefore (a) how they are are likely to react to similar situations in the future, and (b) what sort of agents they are (e.g., laid-back, irritable, etc.).

2.8.1 Representing the concerns of others

For an agent to understand how another agent is likely to construe a situation, he must understand that situation from the other agent's point of view. Since in the Affective Reasoner frames representing an agent's potential interpretations of the world are stored in his GSP database it follows that an observing agent must also have some internal representation of other agents' GSPs. This knowledge is captured in knowledge structures known as *Concerns-of-Other* (COO) databases. They are, essentially, imperfect copies of other agents' GSPs, and represent their concerns as modeled by an observer.

Thus, in addition to the GSP database representing an agent's own concerns, a COO database must also be maintained for each other agent the original agent is internally modeling. When an eliciting situation arises, an agent filters it through his GSP database producing an interpretation of that situation with respect to his own concerns. Using the same process, the agent may instead filter the eliciting situation through the COO database for some other agent, producing an interpretation from what the observing agent perceives as the other agent's point of view.

A perfect COO representation, of course, would be an exact duplicate of the other agent's GSP database, and would always lead to the same interpretations that the observed agent has. However, since, as discussed above in section 2.2, GSPs are built out of interpretation modules (i.e., frames), partial COOs can be created incrementally. Even though they are imperfect, these partial representations are useful since they allow the agent to interpret at least selected events correctly. For example, Harry might know that Tom is a passionate Cubs fan, and that if the Cubs lose he will be upset, and yet not know more about him. Still, if the Cubs do lose, and Tom is jumping up and down, then Harry probably knows why.

Since the Affective Reasoner was developed as a general research platform, several options are available with respect to the establishment of COO databases. They may be constructed at start-up time as part of the initial composition of agents, or they may be learned as the simulation proceeds and as agents come into contact with each other. In the former case a number of difficulties are avoided, such as having to work out the details of when agents are permitted to observe each other; in the latter case, many useful knowledge acquisition issues can be studied. For example, if the purpose of the system is to store knowledge about interesting agents and study emotions that arise when they interact, then the domain-analysis investment required for setting up the COO learning process will have little return. On the other hand, if one is studying user-modeling from an emotion perspective, such a component could be very useful.

2.8.2 Collecting construal frames for COOs

When the Affective Reasoner is set up so that agents learn about one another's concerns through interactions, COOs are built up incrementally by locating and incorporating construal frames that seem to explain another agent's emotional states in response to observed situations. For example, when Harry sees that Tom has always been unhappy when the Cubs lose he might infer that Tom construes some aspect of this situation as blocking one or more of his goals. Harry might then try to determine exactly which goals are involved: is Tom a Cub's fan, or has he just been betting on them to win lately? Perhaps his brother plays for the team? In the following algorithm, which describes this process, we assume that an agent has already discovered the emotion present in the other agent.[7] He now attempts to explain that emotion in terms of the eliciting situation, and possible construals of that situation. To do this, the observing agent consults first his COO for the observed agent (if he has one), and then, if necessary, a global database of available construal forms (see section 2.8.4). Here is the algorithm for incrementally building COO representations.

1. Locate the *Concerns-of-Other* representation (COO) for the other agent. If one does not exist or it does not contain an interpretation for this *type* of eliciting situation[8] then go to 5,

2. Filter the event through the COO, producing an emotion. If this emotion is the same as the emotion that actually was present in the observed agent then the COO has probably given a correct interpretation of the eliciting situation so go to 8,

3. Since the interpretation produced by the COO is incorrect (i.e., the emotion based on the COO's interpretation of the eliciting situation does not match the emotion known to be present in the agent whose concerns it is supposed to represent) the construal frame used to make the interpretation should not be part of the concern structure for the agent. Remove it from the COO,

[7]Obviously, to make inferences about *why* an agent is in some emotional state we must first know what that state is. This is, of course, a problem in its own right, and was discussed in section 2.7.

[8]All eliciting situations are typed. Construal frames which interpret them have the same type (see section 2.2).

4. Mark the construal frame ineligible for this agent. Eligible frames are those frames that can produce interpretations for this *type* of event. Ineligible frames are previously eligible frames which have been found to produce incorrect interpretations,

5. Search through the global (or default) database for the next eligible interpretation of this event,

6. Evaluate the event using the new interpretation as a filter. If the resultant emotion is not the same then go to 4,

7. Add the construal frame to the COO,

8. Generate an explanation based on the current construal frame.

Once a COO has been established for some other agent it can be used for two purposes. First, it is now possible for an observing agent to have emotions based on the fortunes of the second agent. In the Affective Reasoner this may come about if the agents are in one of the following three (possibly only unidirectional) relationships: *friendship, animosity* and *empathetic unit*.[9] For example, if the observing agent knows that a second agent is a Cubs fan, then if they are *friends* he can feel sorry for the second agent when the Cubs lose. On the other hand, if they are *adversaries* then the observing agent can gloat when the Cubs lose. Lastly, should the bond between the two agents be so strong in some situation that the observing agent temporarily takes some of the second agent's concerns on as his or her own then an *empathetic unit* has been formed. The observing agent will temporarily suspend his or her own GSP database, using his or her COO for the second agent to generate direct emotions instead. Note that even in this case the observing agent might actually be wrong about the import of a particular situation for the observed agent, since it is the observing agent's *representation* of the observed agent's concerns which are being used to generate emotions, not the actual GSPs of the observed agent. The second use of COOs is that once they are established it is possible to explain, and sometimes predict, the emotional responses of other agents based on the eliciting condition rules, as in the previously discussed case of Tom the Cubs fan, which opened this section.

2.8.3 Satellite COOs

Modeling the simple concerns of other agents, as specified in the previous section, is still incomplete. Consider the following story:

[9] These three relationships have a very specific meaning here. *Friendship* means that an agent will tend to have similarly valenced emotions in response to the emotions of another agent. *Animosity* means that the emotions will tend to be oppositely valenced. *Empathetic unit*, the most controversial of the three, means that the particular situation is seen "through the eyes" of the other agent, so that the emotions are experienced as the agent's own.

A rookie quarterback is, as usual, sitting on the bench during a football game. His brother and a woman friend of his brother are in the stands. Suddenly the starting quarterback goes down with a knee injury. The woman smiles because she is happy for her friend whose brother will now be placed in the game.

The following sets of concerns and relationships must be considered for each of the agents:

1. The actual concerns of the rookie quarterback, i.e., his GSPs. Inferred in the story is that he will be pleased about achieving a *getting-to-play* goal.

2. The supposed concerns of the rookie quarterback as represented by his brother (i.e., the COO representing the brother's beliefs about the GSPs of the rookie quarterback).

3. The relationship between the rookie quarterback and his brother. Specifically the *friendship* relationship, or even an *empathetic unit* relationship.

4. The *friendship* relationship between the woman and the brother.

5. The supposed concerns of the brother as represented by the woman. This must include, recursively, her *supposed* supposed concerns of the brother for the quarterback as well, and the supposed empathetic relationship between the brother and the quarterback. In other words, the woman must have a belief that the brother will believe that the quarterback will be happy about the starting quarterback's injury. Furthermore, she must believe that the relationship between the brother and the quarterback is such that a positive outcome for the quarterback maps to a positive outcome for the brother.

Because the story gives no clues as to the emotional states of either the quarterback or his brother it should be obvious that neither the actual concerns of the quarterback nor those of his brother are necessary for understanding the episode. To make this clear, consider the following possible continuation to the story:

...But the smile quickly fades away when the brother says, "Oh no, I *told* him he shouldn't have drunk that case of beer at lunch."

Clearly the woman's beliefs leading to emotional states and action expressions of those states are not dependent upon all of the actual facts. Similarly, even if her understanding of the facts is correct, this still does not mean that her emotions have to be in line with them. Consider the following alternate continuation to the story,

> Instead of playing the rookie quarterback, however, the coach puts in a third-string quarterback. The woman, who unbeknownst to the brother had consumed a case of beer with the rookie quarterback at lunch and was sworn to secrecy about it, is relieved. The brother, however, feels terrible for the rookie quarterback because he will not get to play and consequently is very unhappy. The woman is sorry to see him in this state.[10]

In this case the woman *knows* that the brother's beliefs are incorrect, and she does not share them, but she still is capable of having emotions based on the brother's fortunes, which in turn are based on those incorrect beliefs.

It can be seen then, that to accurately represent the fortunes of another agent, to have emotions regarding those fortunes and to accurately interpret their actions with regard to those fortunes, we must not only represent the concerns of the other agent, but also his representations of the concerns of those who are important to him.

The design of the construal mechanism for agents in the Affective Reasoner allows us to create such twice-removed points of view. Essentially, every distinct *interpretive personality* representation is structurally and functionally the same. It does not matter whether the representation is to be used as a GSP by the system, or as a COO or satellite COO by the agent. Thus the emotion machinery that applies to GSPs for the generation of direct emotions will also apply to COOs used for the generation of the fortunes-of-others emotions and to satellite COOs used for representing an agent's beliefs about another agent's beliefs.[11]

The process for making use of each of these GSP and COO databases is essentially the same in all cases. The eliciting event, act or object is filtered through each respective GSP or COO database to produce an interpretation with respect to the antecedents of emotions. In the direct case, this will lead to emotions the system generates for the agent. In the once-removed case it will lead to an interpretation based on *imagining* what it is like for the other agent (possibly incorrectly), which, when combined with a relationship may yield a fortunes-of-other emotion in the observing agent. In the twice-removed case, when combined with beliefs about relationships,

[10]Although this is not the point, these anecdotes, where the feelings of the other agent are not in accord with the known facts, and yet where the observing agent responds to those feelings, are somewhat hard to come by. In general this is because if the observing agent knows that the observed agent will soon find out the facts, the observing agent is much less likely to base his emotions on the temporary happiness or unhappiness of the other agent. This adds an element of secrecy, or of quirky twists of fate (e.g., someone dying before they find out) which in themselves almost always complicate the situation and the resulting emotions. There seems to be some difference between the negatively and positively valenced emotions with respect to this as well. One is more likely to be sad that a friend is temporarily unhappy because he misunderstands a situation that will ultimately make him happy, than one is to be happy that a friend is temporarily happy because he misunderstands a situation that will soon make him sad.

[11]We do not address the issue of how the representation used for reasoning about the concerns of another agent and the representation of one's own concerns may actually be fundamentally different in character.

the interpretation may lead to a belief about the emotional state of the other agent, which in turn may also then lead to fortunes-of-others emotions.

2.8.4 Defaults

In some cases little may be known about another agent. Nonetheless, one may feel sorry for a stranger, and one certainly may wish to explain the actions of strangers. Thus we must give agents a mechanism by which they may still reason about the emotions of other agents, even if little is known about them.

Since, for the purpose of generating emotions in the Affective Reasoner, one GSP database is as good as another, and since even the component construal frames may be mixed at will, we may use a system of defaults for reasoning under uncertainty. Two of these are rather obvious. The first is a system-wide default GSP which corresponds to the knowledge source one might consult in addressing such questions as, *How might a typical agent interpret this event?* The interpretations produced by such a default database are useful when producing explanations such as, *when someone is hit they get mad*, and *losing money increases distress.* The next obvious default GSP is an agent's own GSP database, which corresponds to the knowledge source one consults when asking, *How would I interpret and react to this event?* The resulting emotional states can then be projected onto the other agent.

After these two defaults, the path is less clear. In the Affective Reasoner, when modeling some new agent, an agent may make use of a previously existing COO for some third agent as a default. As discussed in section 1.5, as long as this COO is suitable it remains in use. When the COOs diverge (i.e., when one of the construal frames in the existing COO is found to be incorrect for the new modeled agent) then a copy of the existing COO is made and the offending construal frame is removed. This becomes the current representation of the COO for the new agent. The use of COOs in this manner corresponds roughly to reasoning that since *Agent A* seems just like *Agent B,* then assume he is alike in all ways until learning differently.

Nor are we restricted to using only one COO when searching for an explanation. For agents then, the order of precedence is as follows: (1) search through the COO for the other agent to look for an interpretation of some eliciting situation; if there is none, or it is found to be in error, then (2) search through a COO for some other agent that appears to be similar to this one, if one exists; next, (3) search through the system default GSP database to see *how a typical agent* would interpret the situation; failing in this, then (4) search through one's own GSP database to see *how I might interpret* this situation, and lastly (5) search through the global shared database of construal frames for all possible interpretations of the situation.

Building the frame-hierarchies that represent the concerns of other agents is an iterative process. Figure 2.16 diagrams this process, as an observing agent iteratively builds his representation of some other, *observed*, agent. He starts with a default

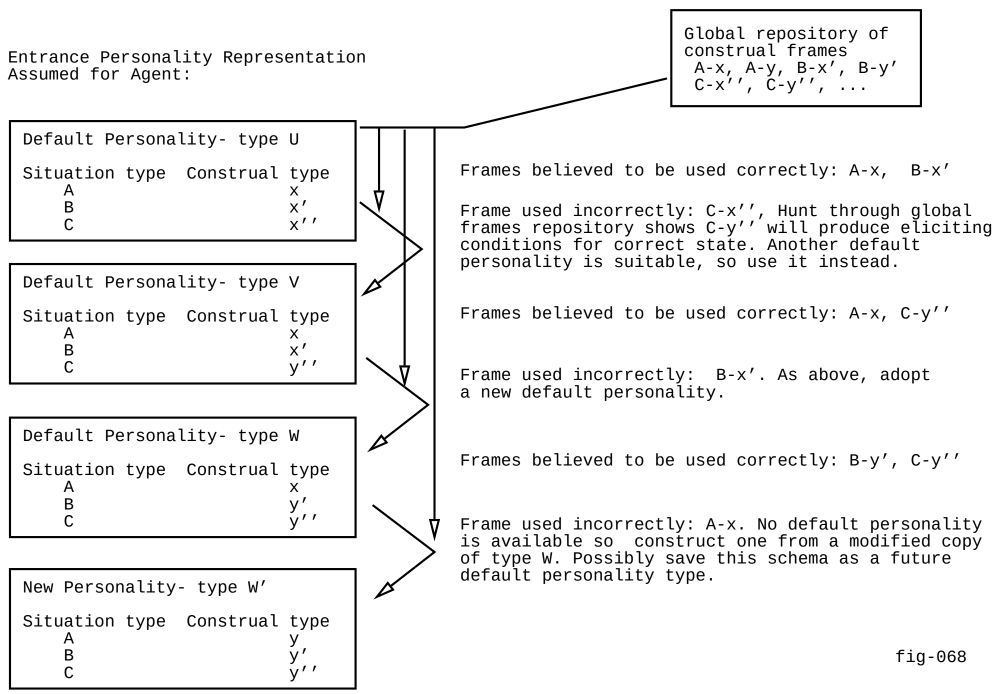


Figure 2.16: Selecting and adapting an interpretive personality structure for another agent.

interpretive personality representation provided by the system. As long as interpretations of situations filtered through this construal database are consistent with the observed features, and as long as the "teacher" of the system has not complained, then no changes are made. It is not until some construal is found to be in error that we make changes to the database.

In this example, the original default personality is given as the abstract type $\mathcal{U}$. This produces the suitable construal of x for a situation of type A (e.g., situation type A might be *getting a speeding ticket*, construal x for A might be having a retain-money goal blocked), and the suitable construal, x', for a situation of type B. A construal of type x'' is *not* suitable for situation type C however, presumably because it does not lead to the affective state believed to have resulted from an evaluation of the situation. This initiates a search in the global repository of construal frames for an interpretation that will lead to the known affective state. The observing agent finds a construal frame, y'', for situation type C which does produce the correct affective state, and replaces (or possibly complements) the existing construal frame x''.

Rather than do this outright, however, the observing agent first searches for an existing personality type that contains both the frames already used correctly and the new one as well. As shown in the diagram, he is able to adopt a personality of type $\mathcal{V}$ that has these properties. Following a similar episode involving the situation type B he has to adopt a personality of type $\mathcal{W}$. Finally, he must create a new personality type, $\mathcal{W}'$ to model the observed agent. In the Affective Reasoner default personality types may be *paranoid, stingy, altruistic* and so on.

2.8.5 Summary of observation component

Figure 2.17 gives an overview of the processes the Affective Reasoner goes through as an agent builds COO representations for other agents. Below are annotations for nodes in the illustration:

1, 2, 3. Features, derived from the eliciting situation and from response action generation, are read by a feature reader and formatted as a case for Protos.

4, 5. Protos analyzes the case and classifies it as belonging to one of the twenty-four emotion classes. A case-library is developed through interaction with the user and features are placed in a conceptual structure indicating their relationship to the emotion classification. Output from Protos is, conceptually, an abduction of an emotion that would plausibly explain the features. This is mode two.

6, 7. In mode one the abduction step is skipped and the observed agent is directly queried about which emotion lead to his actions.

8. The hypothesized emotion is passed to the module that maintains Concerns-of-Other (COO) representations for "other" agents.

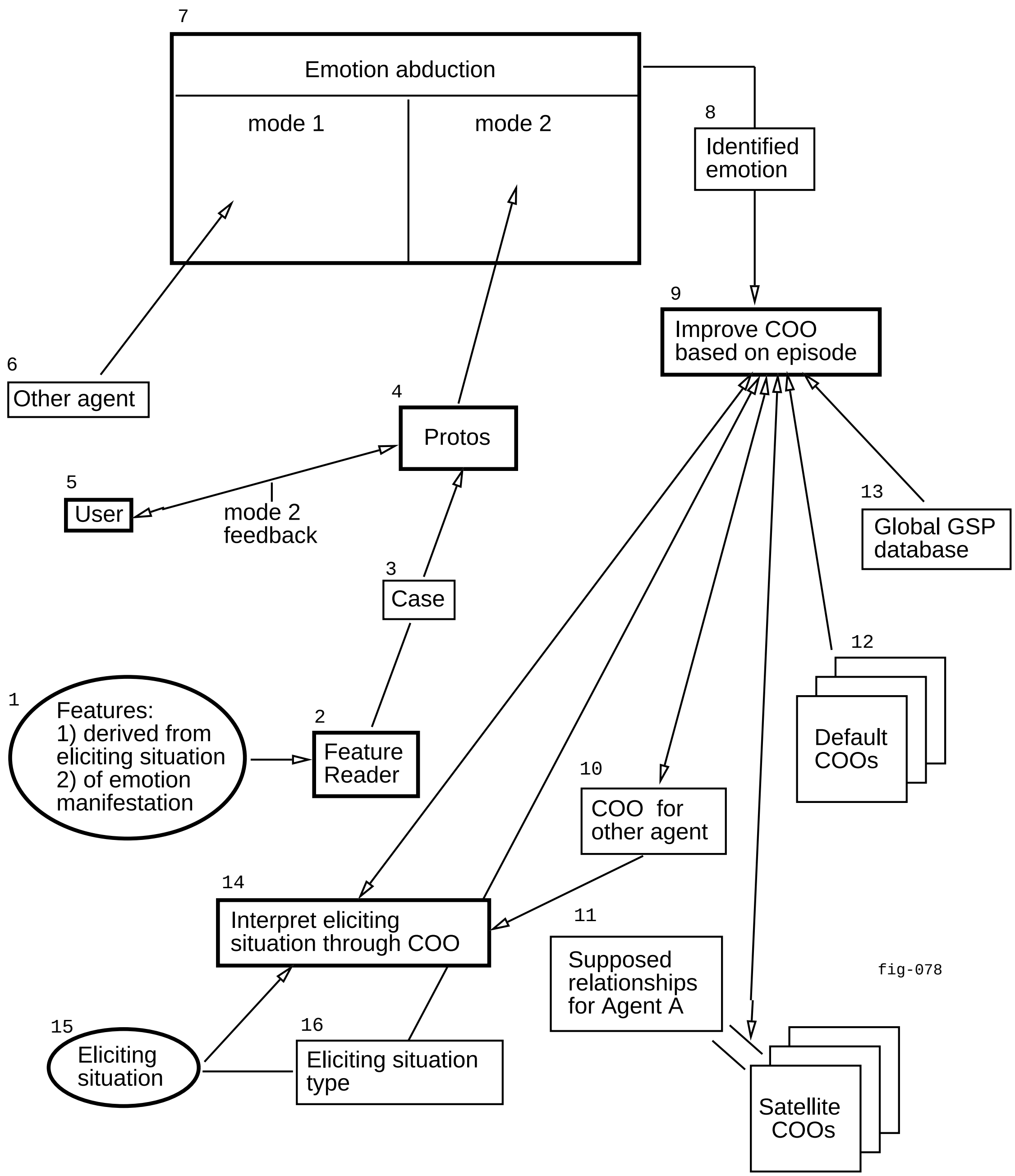


Figure 2.17: Reasoning from cases to build Concerns-of-Others representations

9 – 16. The eliciting situation is filtered through the "eyes of the other agent" using the COO representation for that agent. If the resulting emotion is consistent with that passed from the abduction module, then the COO is assumed correct, and no further processing is done. If no interpretation is made by the COO, or if the interpretation made is not consistent with the emotion passed from the abduction module, then the COO is updated. A search is made through first default COOs, and then a global database of all construal frames, for another construal frame for the situation that will produce an interpretation matching the abduced emotion category. When one is found it is used to update the COO for future processing. The *eliciting situation type* is used as a simple index into the various GSP (COO) databases.

2.9 Summary

In this chapter we have examined the fundamental concepts in the Affective Reasoner. We have traced its processing of emotion eliciting situations from the moment of instantiation by the simulation engine through interpretation by agents, through the generation of emotions based on those interpretations, and through emotion-induced actions. Two subjects covered in detail in later chapters, *construal* and *response actions* were introduced. In addition, a number of shorter topics were treated in full, including an introduction to the emotion elicitation theory of Ortony, et al. [Ortony *et al.*, 1988], the building of emotion eliciting condition relations (EEC relations), the Affective Reasoner's representation of multiple and compound emotions, reasoning about emotion episodes by observing agents, and the construction Concerns-of-Others (COO) databases by which agents represent their beliefs about the personalities of others.

Chapter 3

Construal

A *construal* of a world event is an interpretation of that event with respect to the concerns of some agent. Construals can vary from one agent to another. For example, a last-second touchdown in a football game may be seen as a great moment by a fan of the winning team, as a disaster by a fan of the loosing team, and as of no concern to someone who has no interest in football. In the Affective Reasoner a *construal* is represented as a generalized internal schema (i.e., a *construal frame*) bound to a ground instance of some *eliciting situation* frame.

In this section we look at an extended example of a such a touchdown situation in a football game from the point of view of three different agents *Tom, Dick* and *Harry*. We examine several different instantiations of the situation and trace the many different ways each can be construed by the three agents, leading to an array of distinct emotions. These examples have been run in TaxiWorld by reinterpreting the meaning of the icons.

3.1 Simple goal-based construals

Represented in figure 3.1 are three different construal frames used to interpret the first scenario of the football situation, one for each of the three agents, Tom, Dick and Harry. These three construal frames are matched against the situation frame (event-262) leading to the three different construals (i.e., of *goal-achieved, goal-blocked*, and *preference encountered*.

As can be seen, Tom is a Northwestern football fan. Most touchdown situations are of interest to him if one of the teams playing is Northwestern. Specifically, the construal frame shown will only be invoked if Northwestern is playing and there is almost no time left on the clock.[1] If Northwestern scores a last-second touchdown

[1]This frame has been simplified from that used in the actual simulation of this episode in the system. Many additional considerations have to be taken into account.

Tom's *heroic-finish* goal will be achieved and may generate *joy*. On the other hand, should Illinois score the last-second touchdown, then Tom's goal would be blocked and he might experience *distress*.

Dick is an Illinois fan. His construal of the situation will mirror Tom's: if Northwestern wins, his goal is blocked; if Illinois wins, it is achieved. It is important to note that any instantiated situation of this type that is of interest to one of the agents, Tom or Dick, is of interest to the other. Furthermore, the situation would be of similar interest (the game was decided in the last second with a heroic finish; someone won and someone lost), although the interpretations of the situation with respect to the *heroic-finish* goal would be diametrically opposed.

The third agent, Harry, on the other hand, has no interest in the touchdown, per se. He is, however, interested in the situation. His entertainment preferences run to sunny days and festive crowds. Since the situation meets both of these requirements he may experience an enjoyment emotion.

Here is a walk-through of the matching process for the agent Tom:

- the *type* of eliciting situation frame *event-262* is *touchdown* so construal frames which interpret situations of type *touchdown* are retrieved from Tom's interpretive personality database. In this case there is only one construal frame for interpreting *touchdown* situations.

- *?time-left, ?t1, ?t2, ?t1s* and *?t2s* are all bound to the appropriate slot values. Note that this is the simplest type of match (other than constants) but that *any* type of match template is allowed and that a standard unification procedure is used. Bindings are passed along to the next slot and unification must proceed within the new constraints.

- A predicate function is called on the substitution of the given inequality. That is, the variable *?time-left* is replaced by its value in the set of bindings and the inequality is evaluated. A true value allows the match to continue, a false value terminates the process and justifies an unsuccessful match.

- A predicate function is called on the substitution of the next expression which tests for either *?t1* or *?t2* being bound to the constant *Northwestern.* If this is the case then the predicate succeeds, otherwise the match fails.

- *blocked* is assigned a value of true or false depending on who wins the game. In either case the situation is of concern to Tom. If the goal is blocked this situation may produce a negative emotion, if it is achieved it may produce a positive emotion. Note that although it is not the case here, for some goals there is an intermediate value which says that the goal is neither blocked nor achieved. For example, suppose a taxi driver gets a fifteen percent tip. Since this

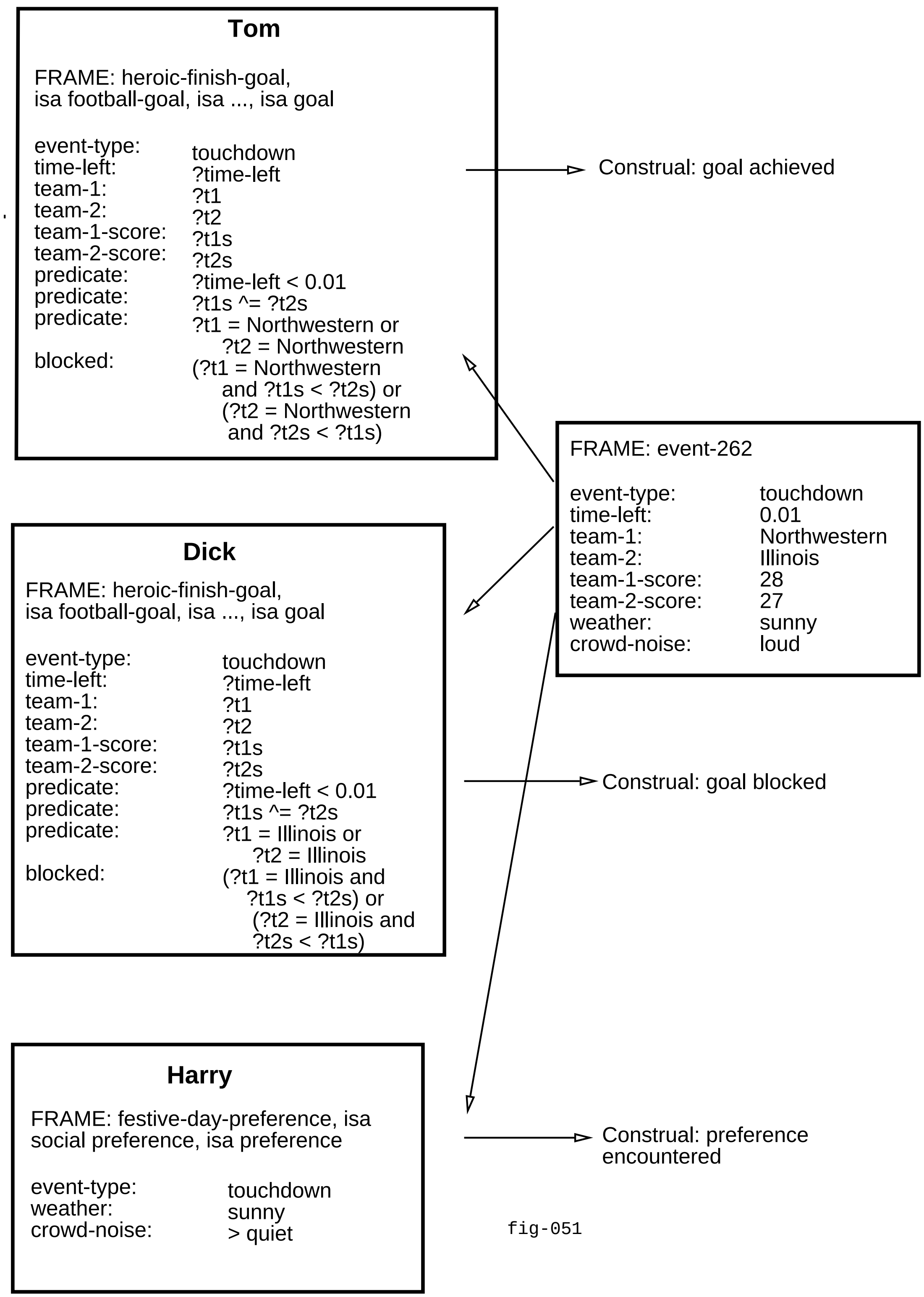


Figure 3.1: Tom, Dick and Harry construe a last-second touchdown.

is the usual amount, the chances are that he will be neither happy nor unhappy about this. A construal frame for interpreting a *get-tip* situation would not match unless the tip were significantly less or more than normal. This is an important point for content theory development. One might ask, if the goal is neither blocked nor achieved then who cares if the eliciting situation is otherwise relevant to this goal frame? As it turns out, however, it is useful to consider such situations to be of *interest* to the agent (since all other features of the situation are matched) but not of *concern* with respect to the goal. This is relevant when we consider *expectations* with respect to goals (see section 3.3 below).[2]

Inherited information may come from the GSP frame hierarchy. This is useful when we want to add interpretive value to a set of construals, value which does not come from the situation itself. In this case we see that this heroic-finish goal is a *football goal*. We might then wish to pass along the information that this is not a very important goal in the larger arena of life, or that it is a *sports* goal, which might affect the nature of how we express our emotions about it.

The construals (i.e., the bound interpretation frames) produced by interpretations on the part of the three agents shown in figure 3.1 are used to create the Emotion Eliciting Condition relations described in section 2.3. Going back to Tom's construal of Northwestern's last-second heroic finish we have:

self	other	desire-self	desire-other	pleased	status	appraisal	appeal	responsible agent
Tom	none	desire	none	none	none	none	none	none

Emotion Eliciting Condition Relation for Tom with respect to Northwestern's last-second touchdown.

Assuming that there were no threshold or other calculations that aborted the construal process, this configuration of the Emotion Eliciting Condition relation would lead to an affective state of *joy,* since being *pleased about an event that is desirable (for one's self)* meets our preconditions for that state. On the other hand for Dick the Illinois fan, we have:

self	other	desire-self	desire-other	pleased	status	evaluation	appeal	responsible agent
Dick	none	undesire	none	none	none	none	none	none

EEC Relation for Dick with respect to Northwestern's last-second touchdown.

[2]Consider, for example, the case of the taxi driver who is *expecting* to get no tip at all because he has offended his passenger. When he gets a fifteen percent tip after all, he will be happy (relieved). In this case the tip is of interest because of the prior expectation of not getting it.

This construal, subject to the above constraints, leads to *distress*, since being *displeased about an event that is undesirable (for one's self)* meets the preconditions for that state.

3.2 Prospect-based construals

Now let us change this situation slightly so that the confirmation status of the situation comes into play. Suppose, for example, that Northwestern were to appear to have scored the last-second go-ahead touchdown as in the above story, but that this time the officials are meeting to discuss whether or not the ball-carrier had actually crossed the goal line. The outcome of the game is on the line. In this case the status is *unconfirmed* until the officials make their final ruling. Tom and Dick both believe that it is likely that the touchdown will stand. In this case the construals produced from the situation matching process for each agent will be the same as before, except that the status attributes will be changed from *none* to *unconfirmed.* Thus the EEC relation for Tom would be:

self	other	desire-self	desire-other	pleased	status	evaluation	appeal	responsible agent
Tom	none	desire	none	none	unconf.	none	none	none

EEC Relation for Tom with respect to Northwestern's unconfirmed last-second touchdown.

leading to *hope*. The EEC relation for Dick would be:

self	other	desire-self	desire-other	pleased	status	evaluation	appeal	responsible agent
Dick	none	undesire	none	none	unconf.	none	none	none

EEC Relation for Dick with respect to Northwestern's unconfirmed last-second touchdown.

leading to *fear*.

3.3 The confirmation emotions

One of the central issues to consider when thinking about the confirmation emotions is that in reality the goals and expectations on which they are based are constantly changing. An expectation that one's team is going to lose the game may alter one's goals: anything other than losing (e.g., tying, or having the game called) can now become a desired outcome. In addition, expectations may vary in strength (e.g., *possibly* versus *extremely likely*). A proper treatment of these problems is beyond the scope of this research. Instead we have made some simplifying assumptions that have allowed us to implement a pragmatic form of reasoning about the confirmation and disconfirmation of expectations. In this section we discuss this approach.

To continue our example from the previous section, let us look at what happens when the status of some situation which was previously *unconfirmed* becomes either *confirmed* or *disconfirmed.* To do this we look at what happens when the officials make their final ruling. If they decide that the touchdown stands, then Tom's and Dick's expectations are *confirmed*, leading, respectively, to the *satisfaction* and *fears-confirmed* emotions; if the officials decide that the touchdown is to be called back, causing Northwestern to lose the game, then Tom's and Dick's expectations are *disconfirmed* leading to, respectively, *disappointment* and *relief*.

Let us consider the implications of implementing such a scheme. The critical issue is how to handle the transition state between a situation being *unconfirmed* and it becoming *confirmed* or *disconfirmed*. To illustrate this using our example, suppose that the belief about the last-second touchdown was established much earlier than just before the officials were to rule. We might consider, for example, that the last three times these two teams had met, Northwestern had won the football game with a touchdown in the last second. Now, at the beginning of play in the first quarter, Tom and Dick are, respectively, *hopeful* and *fearful* that Northwestern will again win in a heroic finish. The problem then is to decide when, if ever, this expectation has been met or shown to be invalid.

3.3.1 Active confirmation and disconfirmation of expectations

In the Affective Reasoner all action is controlled by the simulation queue. This means that (dis)confirmation of expectations must be initiated by a simulation event. If the simulation event gives rise to a confirmed or disconfirmed version of the expected situation then the expectation is retired and a confirmation emotion is generated. We call this *active* confirmation or disconfirmation. On the other hand, the expectation might be retired by a simulation event that does not give rise to the expected situation (e.g., is retired by a time limit). This might simply cause the expectation to disappear (i.e., as though forgotten) or to confirm or disconfirm the expectation, giving rise to a confirmation emotion. In such cases we refer to this as *passive* confirmation or disconfirmation. These two approaches are discussed in this section and the next.

To illustrate, consider again the football example. Suppose that expectations have been set up because Tom (the Northwestern fan) and Dick (the Illinois fan) both believe that Northwestern will win the football game with a touchdown in the final seconds. A process that performs *active* (dis)confirmation must examine all situations that arise subsequent to the creation of the expectation to see if they apply. Since most situations do not apply, it must filter them out. For example, when looking for (dis)confirmation of the last-second-finish expectation it must filter out any situation which is not of type *touchdown*, since a touchdown situation is what Tom and Dick are anticipating. It must also filter out all touchdown situations that do not take place at the end of the game, since only those that do are relevant to the expectation. Lastly it must filter out all last-minute touchdowns that do not change the outcome of the game such that Northwestern wins.[3] But we have seen this before: the active matching of expectations is *exactly the same process* as that of matching concerns. Every eliciting situation frame must be considered, at some level, as a potential (dis)confirmation of the expectations that have been created by the agents in the system, just as every eliciting situation frame must be considered as a potential initiator of emotions.

This is, in fact, exactly how active (dis)confirmation of expectations is treated by the Affective Reasoner. To implement this, it would seem that a separate database of expectations (i.e., one designed to be used as match schemas for eliciting situation frames and modeled after the GSP hierarchies of agents), would suit our purposes. However, it is clear that every expectation frame is derived from a previously instantiated GSP frame, specifically a goal-based one. This means that instead of building a separate hierarchy of frames for the expectations that have been created we can instead use the original GSP hierarchy and simply keep auxiliary information to supplement that database. Clearly, if a situation can be unified with an instantiated expectation frame then it can also be unified with the original GSP schema (i.e., construal frame) from which the instantiated expectation frame was derived. This means then, that we may greatly reduce our processing by considering for expectation matches *only those situations which have been shown to match the original concern from which the expectation was derived.* Since concern matching must be done anyway, we have saved processing expense by using it also as a filter for expectation matches.

Here then is the general idea. We match some original emotion eliciting situation and get an expectation frame which is implicitly represented as a set of bindings from the original construal. If we can match the original construal frame again,

[3]This illustration points out how appropriate qualitative reasoning would be as an extension to this system.

within the constraints of these bindings, we may be able to resolve our expectation. To illustrate, we may imagine a situation in which Tom tells Dick, the Illinois fan, that Northwestern almost always manages to win by scoring a touchdown in the final seconds. This sets up an expectation in Dick, represented as a set of bindings for the status-unconfirmed touchdown situation. Later, when an actual touchdown situation arises, an attempt is made to unify the bindings created from matching the eliciting situation frame against the touchdown-event construal frame with the bindings representing the expectation. If the two sets of bindings can be unified then a confirmation emotion is generated, such as relief that Northwestern did not manage to beat Illinois in the final seconds, or having Dick's fears confirmed that Northwestern did, in fact, win in the final seconds. Otherwise, if the construal frame matches but the bindings are not compatible with those representing the stored expectation frame, then a direct emotion (such as distress) or prospect-based emotion (such as hope) is generated.

Figure 3.2 illustrates this for the general case. Here we see that expectation frames are stored as a set of bindings and a pointer back to the original construal frame that generated them. When, later, a new eliciting situation of the same type (in this case type s-37) is generated it is matched against the appropriate construal frame. In addition, since there has been an expectation generated by an interpretation previously made by this construal frame (i.e., the implicitly represented expectation frame stored in *bindings set 4*, for *situation 41*), additional unification is performed to see if the bindings generated from the current match are compatible with the bindings from the previous match. Depending on the results of this additional unification a confirmation emotion may be generated.

To illustrate why these original bindings are necessary, consider the following, which might happen if the original bindings were ignored:

> *Tom is fearful that Harry, whom he does not like, will also show up at the game. However, Harry has given his ticket to Dick, whom Tom does like. When Dick shows up, Tom's fears that Harry will show up are confirmed.* [sic]

This could occur if the bindings, which specify that the agent expected to show up is Harry, were ignored. Thus the match must be consistent with the bindings that instantiate the original expectation. However, some modifications are necessary because we do not really want to match the original situation again, but rather some *similar* situation which takes place in the future. The bindings for the two situations will differ in three areas: (1) in some specialized bindings such as temporal features, (2) in the *status* value of the situation, and (3) in the value of *?blocked-violated*. We now take a closer look at these three modifications to the stored bindings:

> **Specialized bindings.** Suppose that some eliciting situation arose at time $t1$, and that it was construed as an *unconfirmed situation*, so that an agent was

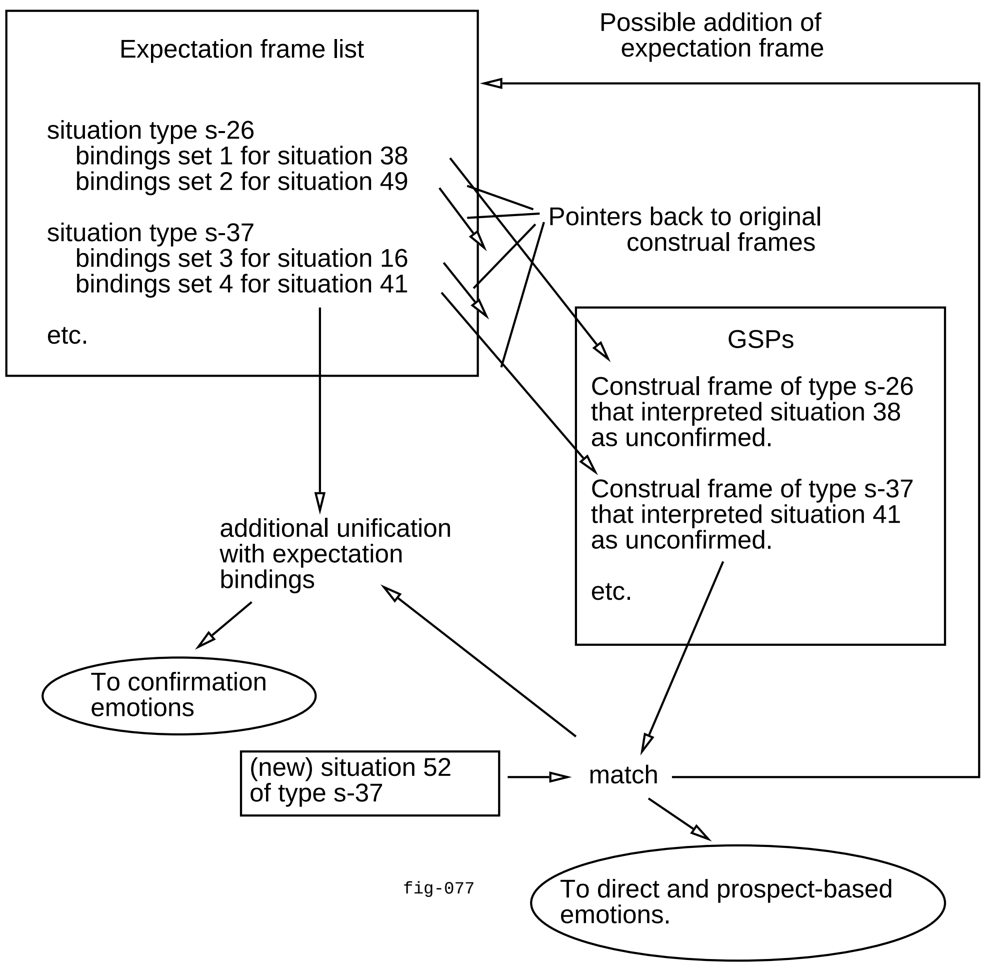


Figure 3.2: Maintaining expectation lists for the confirmation emotions.

worried (or hopeful) about it. Later, at time $t2$, a situation of the same type and with the same features occurs but is construed as *confirmed* by the agent. Since the features are the same, the match between the expectation frame and the construal of the new situation almost succeeds, but ends up failing because $t1 \neq t2$. To avoid this problem in the Affective Reasoner the matcher is modified so that members of a specified set of features (such as *?time*) are not required to participate in the unification process. In our example, the match would now succeed and the $t2$ situation would lead to one of the confirmation emotions on the part of the agent.

Status attribute. In the original eliciting situation taking place at $t1$ the status attribute was bound to the constant, *unconfirmed,* leading to the creation of an expectation. However, when (dis)confirming this expectation the status attribute must be matched in a special way, in fact using the exact opposite of the normal matching procedure. In this case a successful unification of the constant *unconfirmed* would cause the match to fail, while either *confirmed* or *disconfirmed* would cause it to succeed. Intuitively this is correct: if the situation is (still) *unconfirmed* it will not help to resolve the expectations raised by the original *unconfirmed* situation; if it is *confirmed* or *disconfirmed*, it may.

?blocked-violated value. We have already indicated that some goals and standards may, in addition to specifications for *blocked* and *achieved*, have a third specification of neither *blocked* nor *achieved*, i.e., *nil.* Here is where this becomes important. A *?blocked-violated* value of *nil* is a good outcome causing *relief* when one is in a state of *fear*, since the expectation that the goal will have been blocked has been disconfirmed. On the other hand *nil* is a bad outcome causing *disappointment* when one is a state of *hope* since the expectation that the goal will have been achieved is now *disconfirmed.* In all cases, if the value of *?blocked-violated* in the stored bindings for the expectation frame is matched exactly in the new situation frame, the expectation is *confirmed,* otherwise it is *disconfirmed.*

For example, suppose that Tom has the goal of having his team win. Consider two scenarios. In the first he is *expecting* his team to loose, and consequently is *fearful* about this outcome. In our implementation this means that the construal frame for this goal has matched an *unconfirmed* version of this situation, where the goal has been blocked, leading to both the generation of the prospect-based emotion of *fear* and the creation of an expectation frame. If later the game ends in a tie, then Tom will experience *relief* because the expectation of a negative outcome has been disconfirmed. This disconfirmation arises because the value of *?blocked-violated* in the expectation frame (i.e., *blocked*) does not match the value of *?blocked-violated* in the actual outcome (i.e., *nil*).

In the second scenario Tom is expecting his team to win, and consequently is *hopeful* about this outcome. Similar to the above, we represent this as an unconfirmed construal, leading to the generation of the prospect-based emotion of *hope* and the creation of an expectation frame. If the game later ends in a tie then the disconfirmation of Tom's expectation will lead to *disappointment*.

These two examples illustrate how, in the Affective Reasoner's paradigm, a goal-relevant situation that ordinarily would not lead to an emotion can still lead to relief and disappointment through the confirmation or disconfirmation of a prior expectation.

3.3.2 Passive confirmation and disconfirmation of expectations

Now we look at the harder problem, that of (dis)confirming an expectation *passively* (i.e., by default). Going back to our football example, where the agents have formed the expectation that Northwestern will score a touchdown in the final seconds, we can see that if the game is over and this has not occurred then the expectation has been disconfirmed by default[4]. We need only set a time limit on the expectation equal to the end of the game and at that moment, if nothing has happened to intervene, expire the expectation. We do this by putting a simulation event into the simulation queue equal to that time. This simulation event then gives rise to an emotion eliciting situation at the appropriate time which in turn leads to the *disappointment* and *relief* emotions of Tom and Dick, respectively.

What are the problems with this approach? First we have a simulation detail to consider. Clearly an expiration event must be created within the simulator. This event must necessarily take an *active* role in causing the (dis)confirmation emotions, and it must point back to the original construal frame creating the expectation. In this case the end of the football game must somehow be linked to an expectation for a touchdown situation, two possibly unrelated frames. This can be a messy process.

Next, consider the related problem that since the time when the football game will end may not be known in advance (i.e, if the simulation events representing the game are created dynamically as the game progresses), the *end-of-football-game* event cannot be placed directly in the time-sorted simulation queue. Some fix is required, such as an end-of-football-game cleanup event. As just discussed, this simulation event would have to point to expectation frames (i.e., construal frames) that were to be activated or deactivated when the football game ended. But this means that the events in the simulated world might have to be modified every time some new relevant construal frame was added to a personality. For example, suppose that we decided

[4]An example of confirmation by default would be, say, when we are expecting to not get a phone call from someone we have asked to call us, and in fact we do not.

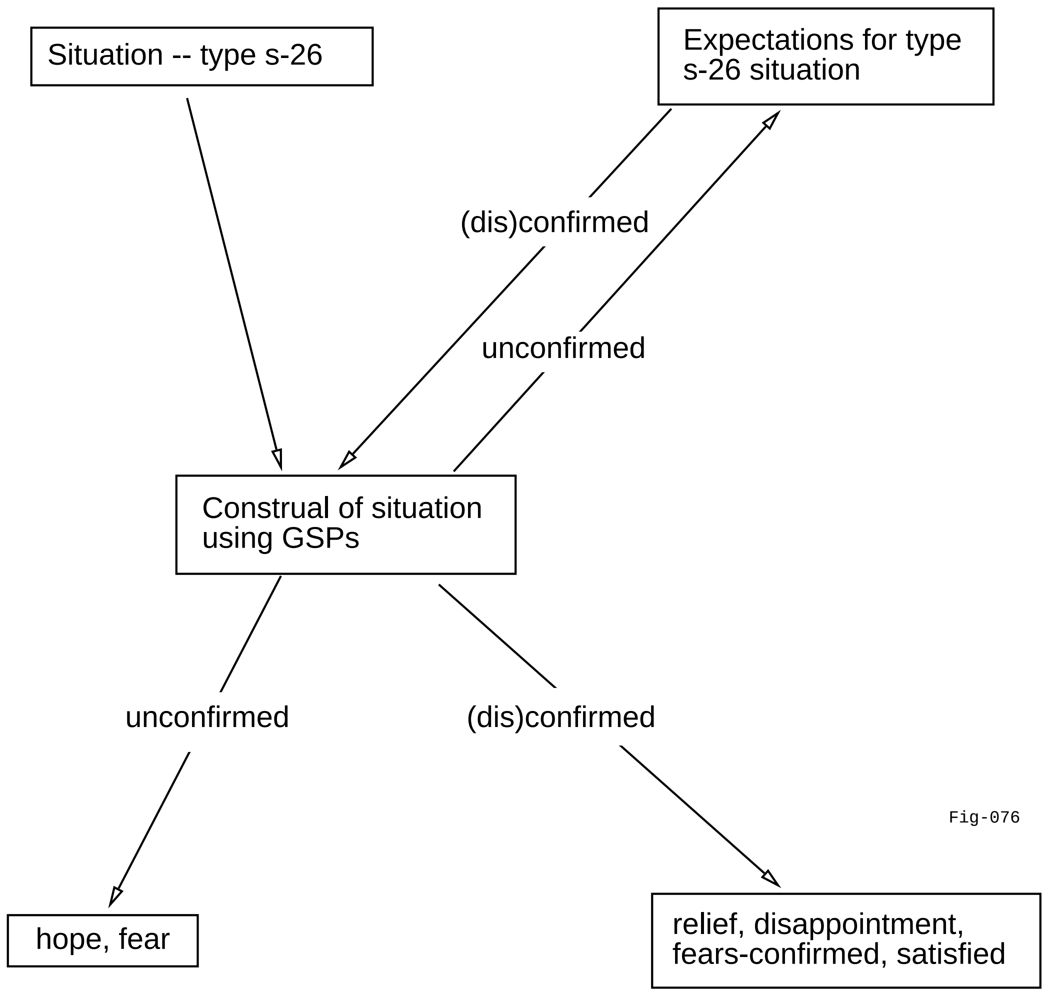


Figure 3.3: Prospect-based and confirmation emotions

to make one of our agents worried about traffic when the football game was over. The end-of-football-game simulation event would now have to be altered to trigger (dis)confirmation of this additional worry. This is clearly not an attractive solution.

Now, consider the hardest problem of all, that most cases of passive (dis)confirmation are not associated with a specific time at all. Instead, the prospect-based emotions just fade away over time. Suppose, for example, that rather than fear that one's football team is going to lose *this* game, one has a more generalized fear that his team is going to let him down. When will enough games have passed that this fear is allayed? Two games? Four games? Six minutes and five seconds into the first quarter of the fourth game, ...?

Lastly, we also must consider cases where the prospect-based emotions end abruptly because some other emotional episode occurs and takes precedence. If Tom learns that his mother is dying, for example, he is unlikely to continue experiencing anxiety over the outcome of a football game. In this we see the essence of the problem: For the *active* (dis)confirmation of situations we can effectively look for an occurrence of the situation we are expecting, in either the blocked-goal or achieved-goal (and possible *nil*) versions. For *passive* (dis)confirmation we have no way of knowing what it is that will trigger interaction with the expectation, yet this interaction must be initiated by that unknown situation leading to "awareness" on the part of the agent.

How is this handled in the Affective Reasoner? *Active* expectation (dis)confirmation is handled using the matching process as specified, and by keeping a list of currently active expectations for each agent. *Passive* expectation (dis)confirmation is simply by time limit. A simulation event is created to expire the expectation at a certain time in the future. If the prospective situation has already been (dis)confirmed then nothing happens, otherwise (dis)confirmation takes place by default, possibly subject to some additional constraints. This difficult area is ripe for a detailed structural analysis, but is currently beyond the scope of this research.

Figure 3.3 illustrates our use of expectations in generating emotions. A situation may be *unconfirmed* and so give rise to the prospect-based emotions of hope and fear. At the same time, this will create an expectation, which is stored as a set of bindings from the original construal. Later, if the situation is either confirmed or disconfirmed this will give rise to one of the confirmation emotions, depending on the actual (dis)confirmed situation and the prior expectation about it.

3.4 Standards-based construals

In this section we continue to examine different ways in which the last-second touchdown situation may be construed. We have seen how such a touchdown situation can be relevant to goals both directly and in terms of prospective outcomes. We have also seen how simple preferences can be invoked and how different perspectives can be used to interpret the same situation. Now let us turn to construals based on interpretations of eliciting situations as being the praiseworthy and blameworthy acts of agents, i.e., the *standard-based* appraisals of eliciting situations.

To this end, let us imagine that the Illinois coach has said the following to his team just before the last play of the game: "I am sure that they are going to run the football on the right side. We could fade back into a zone defense and probably stop them before the end zone, but I want to show them our stuff and beat them on the line." When Tom and Dick see the way the defense lines up they both know that the coach is taking a needless chance just to make a macho point to the Northwestern team. The defense fails and Illinois loses a game they might otherwise have won.

Tom, the Northwestern fan, is happy that his team has won. In addition he is scornful of the Illinois coach's decision which he felt was stupid. The EEC relations for these interpretations are:

self	other	desire-self	desire-other	pleased	status	evaluation	appeal	responsible agent
Tom	none	desire	none	none	none	none	none	none

self	other	desire-self	desire-other	pleased	status	evaluation	appeal	responsible agent
Tom	none	none	none	none	none	blame	none	coach

EEC Relations for Tom with respect to the Illinois coach's blameworthy act, violating Tom's standard but achieving his goal, leading to the mixed emotions of *joy* and *reproach*.

Such a relation would lead to mixed emotions: (1) *joy* over winning the game (as discussed above), and (2) *reproach* over the violation of one of Tom's standards, namely the *win-at-all-costs* standard that he feels all football coaches should have.

Dick (the Illinois fan), on the other hand, is not happy about his team losing the game. Nonetheless, he respects the coach for playing what he considers to be in the true macho spirit of the game. The EEC relations for these interpretations are:

self	other	desire-self	desire-other	pleased	status	evaluation	appeal	responsible agent
Dick	none	undesire	none	none	none	none	none	none

self	other	desire-self	desire-other	pleased	status	evaluation	appeal	responsible agent
Dick	none	none	none	none	none	praise	none	coach

EEC Relations for Dick with respect to the coach's praiseworthy act, upholding Tom's standard but blocking his goal, leading to the mixed emotions of *distress* and *admiration*.

In this case Dick has mixed emotions: (1) *distress* over losing the game, and (2) *admiration* for the Illinois coach for upholding his (Dick's) standard.

Now let us suppose that the standards held by Tom and Dick are reversed, with Tom adopting the standard that coaches should "stick it to them" whenever possible, and Dick the standard that coaches should "win at all costs". In this case the Illinois coach's decision was *praiseworthy* from Tom's point of view, and *blameworthy* from Dick's point of view. Since a standard has been upheld and a goal achieved, the preconditions for both *joy* and *admiration* have been met on Tom's behalf. However, taken as a whole, the preconditions for the compound emotion of *gratitude* have also been met, and this takes precedence.[5] Likewise, since a goal has been blocked and a standard violated on Dick's behalf, the compound emotion of *anger* takes precedence over *distress* and *reproach*. Here is the EEC relation for Dick's *anger*:

self	other	desire-self	desire-other	pleased	status	evaluation	appeal	responsible agent
Dick	none	undesire	none	none	none	blame	none	coach

EEC relation for coach's blameworthy act blocking to Dick's goal, leading to anger.

In figure 3.4 we see the two construal frames that lead to this EEC relation and how they match the original situation. As can be seen, some of the features in the original situation are not relevant to either of these particular concerns (e.g., *weather* and *crowd-noise*), and some are relevant to only one (e.g., *time-left* is important to the goal of having a heroic finish, but is not important to the standard dictating a win-at-all-costs style of play). In addition, some of the construal frames capture bindings for features that are not used in the construal process (e.g., *team1* and *team2* in the second construal frame). These bindings may be used later, however, when forming action-responses.

In general, each construal frame that matches an eliciting situation frame will produce its own Emotion Eliciting Condition relation. If one eliciting situation is of concern to an agent with respect to three different goals or three different preferences, etc., then three EEC relations will be produced, from which in turn may be derived three different emotion instances, *even of the same type.* This holds in all cases but for those EECs producing compound emotions (such as the examples of *gratitude* and *anger* above). In the first case a single construal produces a single EEC. In the latter case each *pair* of construals that belong to the relation illustrated in figure 3.5 produces a single EEC relation, as discussed in section 2.4.5. Briefly, there will be $a+b$ EEC relations for compound emotions where a is the number of *standards* construal

[5] Actually, the system would probably be wrong to make this interpretation since it is more likely that in this case the agent would simply feel admiration and happiness. On the other hand it certainly is plausible that one might feel gratitude towards a hero who has temporarily given life a special quality. This is a common occurrence in the lives of performers when audience members thank them for their work. Further study is needed.

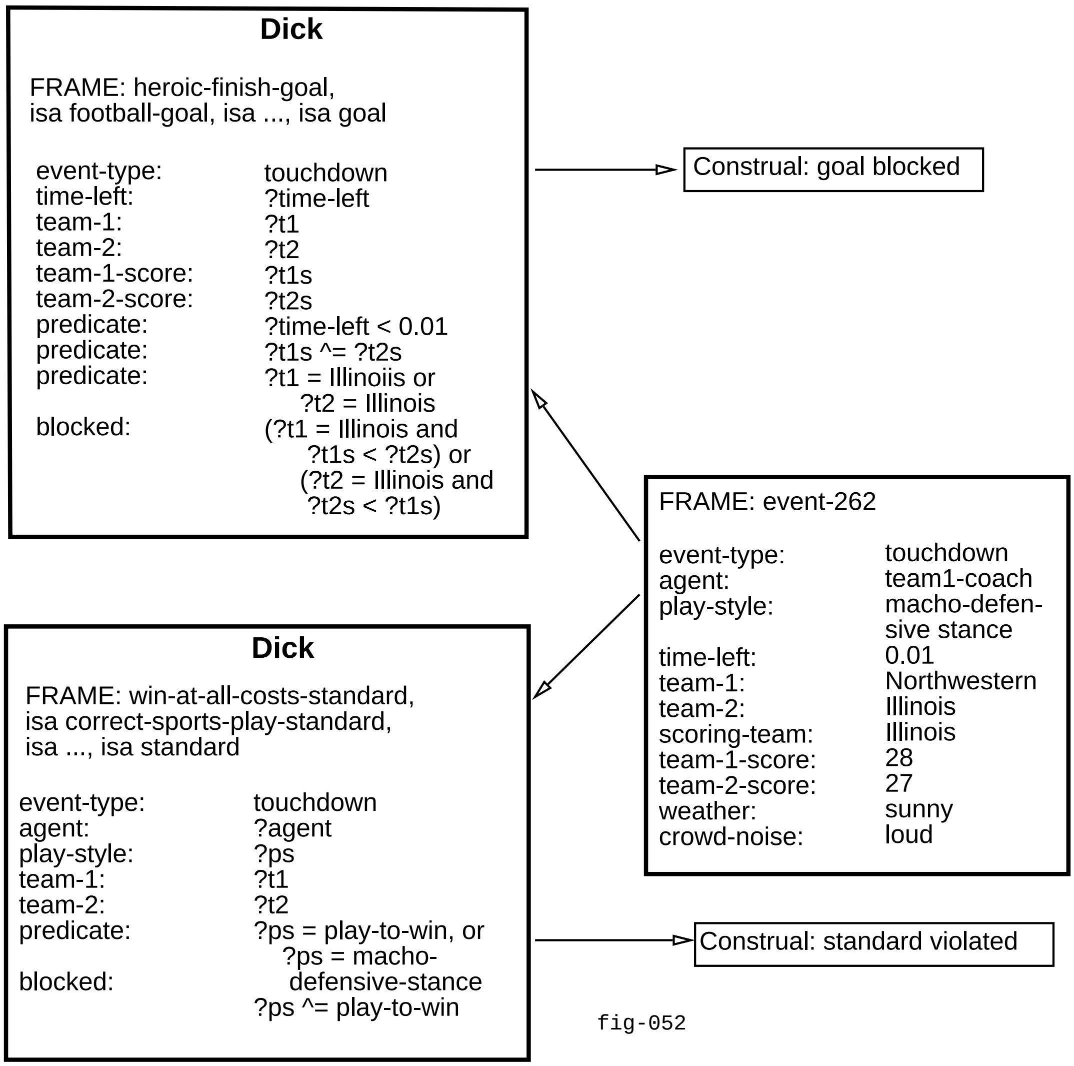


Figure 3.4: Dick construes situation in a way that leads to *anger*.

Goal	**Standard**	**Emotion**
achieved	upheld	gratification (self), gratitude (other)
blocked	violated	remorse (self), anger (other)

Figure 3.5: Compound emotions

frames matching an eliciting situation frame, and b is the number of *goals* construal frames matching the eliciting situation frame, subject to the above constraints.

We have seen how Tom and Dick can have multiple and compound emotions in response to a single situation. To close this section we now take this one step further and see how a single situation can lead to *conflicting* emotions. Suppose, for this argument, that Tom figured that Northwestern would lose the football game. Banking on this he bet against them. Now, although he is happy that his team won in such a dramatic fashion, he is also distressed about losing his bet. On the one hand a goal of his has been achieved, on the other hand a different goal has been blocked. Figure 3.6 illustrates this.

3.5 Serendipity and preservation construals

Many situations can be interpreted in either a positive or a negative way by an agent, depending on the bindings for only a few of the many constituent features of the situation. For example, suppose the following situation arises in the life of a taxi driver: He delivers a passenger at some location. The ride was fine. He gets paid. In most cases, since he wants to make money, if he gets a big tip he will be happy and if he gets a small tip he will be unhappy.

In the scheme discussed so far in this chapter, these opposing, but otherwise similar, construals (e.g., of the tip situation) could each be represented as a distinct construal frame: one leading to a positive emotion and the other leading to a negative emotion. Obviously, however, both situations share many features. To make use of this in the Affective Reasoner we have, in general, represented both situations with the same construal frame, but have added a procedure to calculate the value of an additional attribute, *blocked-violated*, representing the blocking of goals or the violation of standards. Thus the match process may be thought of as containing two steps. The first is determining whether or not a situation is of *interest* to the agent. The second is determining the valence of the construal. This valence is represented as a *true, false*, or *nil* value for the *?blocked-violated* attribute where *nil* may indicate either a positive or a negative construal depending on prior expectations.[6] The *blocked-violated* function that computes this value does not participate in the determination of relevance, it only adds a value for the *?blocked-violated* variable to the bindings which are used to create the Emotion Eliciting Condition relation.

Representing positive and negative construals in this way has the benefit of halving the computational load involved in the matching process by avoiding the un-

[6]Note that this has not been implemented for preferences. Positively and negatively valenced construals of situations seen as objects (i.e., the liking and disliking of objects) are not usually in negation relationships. In fact, it is the quality of *needing no reason* for liking or disliking something that sets preferences apart from goal-based and attribution-based emotions.

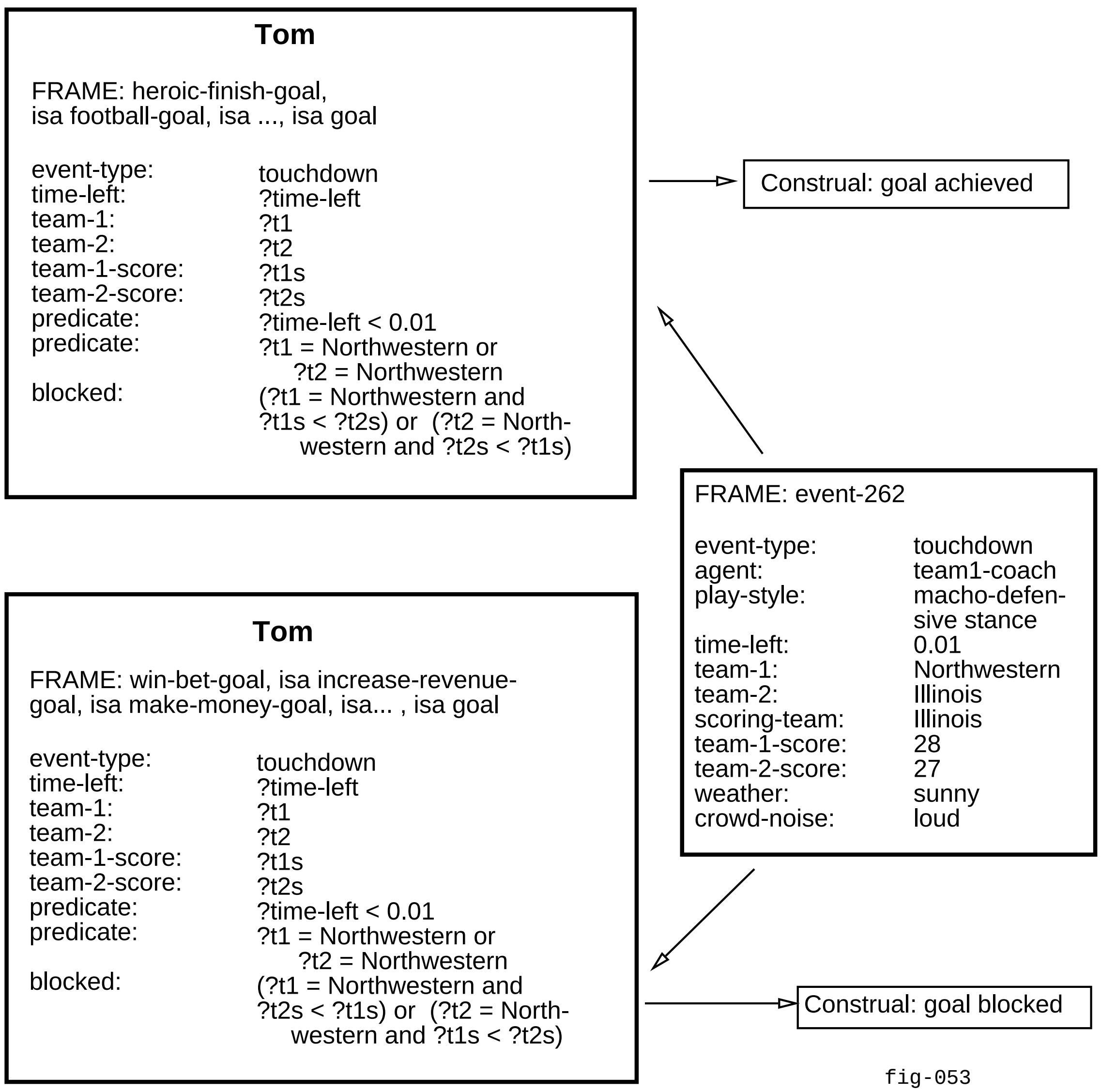


Figure 3.6: Tom's multiple construals lead to *joy* and *distress*.

necessary duplication of data. However, this solution is not without problems of its own. Some goals can only be achieved (and some standards only upheld) and never blocked (violated). Other goals can only be blocked (standards violated) and never achieved (upheld). We call these *serendipity* and *preservation* goals and standards, respectively. They are related to the idea of preservation goals ([Schank and Abelson, 1977]) but are simpler and are driven by more practical concerns. When the match process is performed using a serendipity construal frame, if *?blocked-violated* is bound to *false* (i.e., not blocked) the match continues, else it fails. When the match process is performed using a preservation construal frame, if *?blocked-violated* is bound to *true* (i.e., blocked) the match continues, else it fails. Here is a story that illustrates the need for a serendipity goal:

> *A wealthy-looking stranger walks up to John on the street and gives him a thousand dollars in cash, explaining that he just likes to make people happy. John was overjoyed.*

In this case John experiences *joy* because he has a goal of getting money. On the other hand, if the stranger does *not* give him a thousand dollars he would not experience *distress*. Here is a story that illustrates the need for a serendipity standard:

> *Sarah leapt out from the crowd, rushed into the raging fire, and pulled the strange dog out to safety. John admired her for her bravery.*

John has a standard that says that endangering one's self for the benefit of others is worthy of approval. However, if Sarah did not rush into the fire John would not have blamed her for it. Here are two stories that illustrate the need for preservation goals and preservation standards, respectively.

> *John is walking down the street, and a flower pot falls on his foot. John is unhappy about it.*

In this case John experiences *distress*, because he has a (preservation) goal of maintaining his good health. On the other hand, if he walks down the street and the flower pot does *not* fall on his foot, he is unlikely to experience *happiness* about it.

> *John is walking down the street when he sees two strangers passing each other on the next block. The first stranger robs the second stranger and then disappears. John is reproachful about the robber's evil actions.*

In this case John feels *reproach* for the robber. On the other hand, should the two strangers have passed without incident then John is unlikely to feel *admiration* for the would-be thief.

One might argue that this is really a matter of the *surprisingness* of the eliciting situation, and that if an outcome is *usual* it should not produce a valenced reaction, (see [Ortony and Partridge, 1987] and [O'Rorke and Ortony, 1992] for a discussion of this). In the previous examples such a treatment would suffice. However, there are cases where the *surprisingness* approach is not as suitable. Here is a story that illustrates this:

> *John is looking out his window and sees a neighborhood youth walking down the street eating candy. Usually the youth throws his candy wrapper on someone's lawn. This time, as is no surprise, when the youth passes a neighbor's house he does exactly that, again. John is reproachful.*

Here John experiences *reproach* for the youth, because John has a standard about littering. On the other hand, if the youth were to put the trash in his pocket John would be surprised but would probably not experience any emotion at all. In such a case, it is the *unsurprising* behavior that leads to an emotion, and the *surprising* behavior which leads to none.

3.6 Summary

In this chapter we have discussed a *touchdown* event at a football game from a number of different points of view. This example was used to explore the goal-based and standards-based emotions and to discuss how multiple and even conflicting emotions can arise for a single agent. The difficulties in representing *expectations* was discussed in light of the prospect-based emotions of hope and fear, and the confirmation emotions of relief, fears-confirmed, disappointment and satisfaction. The mechanism used to store expectations was discussed. Lastly, special classes of goals and standards which lead only to either the positive or negative emotions, but not both, were discussed and the *serendipity* and *preservation* construal frames used to interpret them were introduced.

Chapter 4

Response Actions

4.1 The nature of the task

When we think of someone as "being emotional" we are usually referring to their thoughts and actions following some emotion-inducing situation. These response states are initiated and controlled in the Affective Reasoner by the *manifestative personality component.* Few attempts have been made in AI to decompose emotion-based action, probably because of the difficulty of the task. Here are some of the obstacles to an effective treatment of this domain.

First, what are the functional considerations of response-action processing? Are the actions to be merely expressive in nature, or are they going to drive the system? Perhaps they will be used as modulators of a logic-based action generation system? Each approach has merit, and the best system will support them all. In one system, for example, we might use emotion feedback only to adjust the communication levels between agents. In another we might instead implement agents so that they only act when driven to do so by their emotions.

Second, we must decide on the complexity of the actions that we are going to implement. Emotional responses may be as simple as *turning red in the face* or as detailed as *planning revenge* against a traitor. A full treatment of the abstract emotion domain must embrace a wide spectrum of such responses.

Third, if agents are engaged in purposeful behavior it can be assumed that they are guided by some implicit or explicit plan. We must ask ourselves then: How can emotion-based action help agents to further their goals? What is the *purpose* of the emotion machinery and how can its effectiveness be measured? If we define the result of planning simply as the sequencing of actions in time, the question is obvious – how do we integrate emotion-based action into that sequence such that the new sequence is more beneficial to the agent (or to the society of agents as a whole) than was the old sequence?

Lastly, we must consider taxonomic issues. *Smiling* can be an appropriate action response when one feels *joyful*. It can also be an expression of *pride,* of being *happy-for* someone, and so forth. In addition *smiling* can mask the repression of negative emotions such as when one smiles to cover for being *sad.* On the other hand, other actions usually *are* associated with a particular emotion (e.g., glaring at someone is associated with anger). The expression of emotions is rife with such ambiguities: one emotion can initiate many expressions, one expression can have many initiating emotions. In addition, almost any action can serve the *purpose* of emotional expression, but this is entirely dependent upon the context in which the action takes place. For example, we might make friends with someone because we are lonely (a *problem-reduction plan-initiation* emotion manifestation) or we might do so simply because it is good for business (a purely non-emotional motivation). In summary, then, we have a *weak-theory* domain, that is, one where some relationships consistently hold (e.g., shaking one's fist usually implies anger) but where most only occasionally hold (e.g., anger does not entail shaking one's fist).

4.2 The three-dimensional nature of the action database

Figure 4.1 (inspired by [Gilboa and Ortony, 1991]) gives a breakdown of the action response categories and their theoretical groupings for the emotion *gloating.* The high-level categories (i.e., *goal-directed, expressive, information-processing,* and so forth), are not presently used for action generation.[1] They instead are employed simply as a way of organizing the spectrum of responses. One possible use of the theory underlying these categories, however, might be as a basis for a functional representation of emotions as initiators of purposeful, *motivated* actions. In this regard, the *information processing* actions, for example, might be seen as representing aspects of an agent's attempts to "understand what is going on."

The high-level categories are arranged, as far as possible, from the more spontaneous to the more planned (e.g, *expressive* actions tend to be less planned than those for *emotion regulation and modulation*). Within these high-level categories the particular action response categories are similarly arranged (e.g., *somatic* expressive responses tend to be more spontaneous than *communicative-verbal* expressive responses). Lastly, within the action sets themselves the tokens (and mini-plans) are arranged from the least intense to the most intense (e.g., *smiling* as a *non-goal directed, expressive behavioral-toward-animate* expression of *gloating* is less intense than is *laughing*). In all cases, these are at best partial orderings since different contexts will affect the ratings of the categories and actions, and in any case the ratings are

[1] But see section 2.8 on the structure of the action response with respect to heuristic classification for a discussion of how these may be used in abduction.

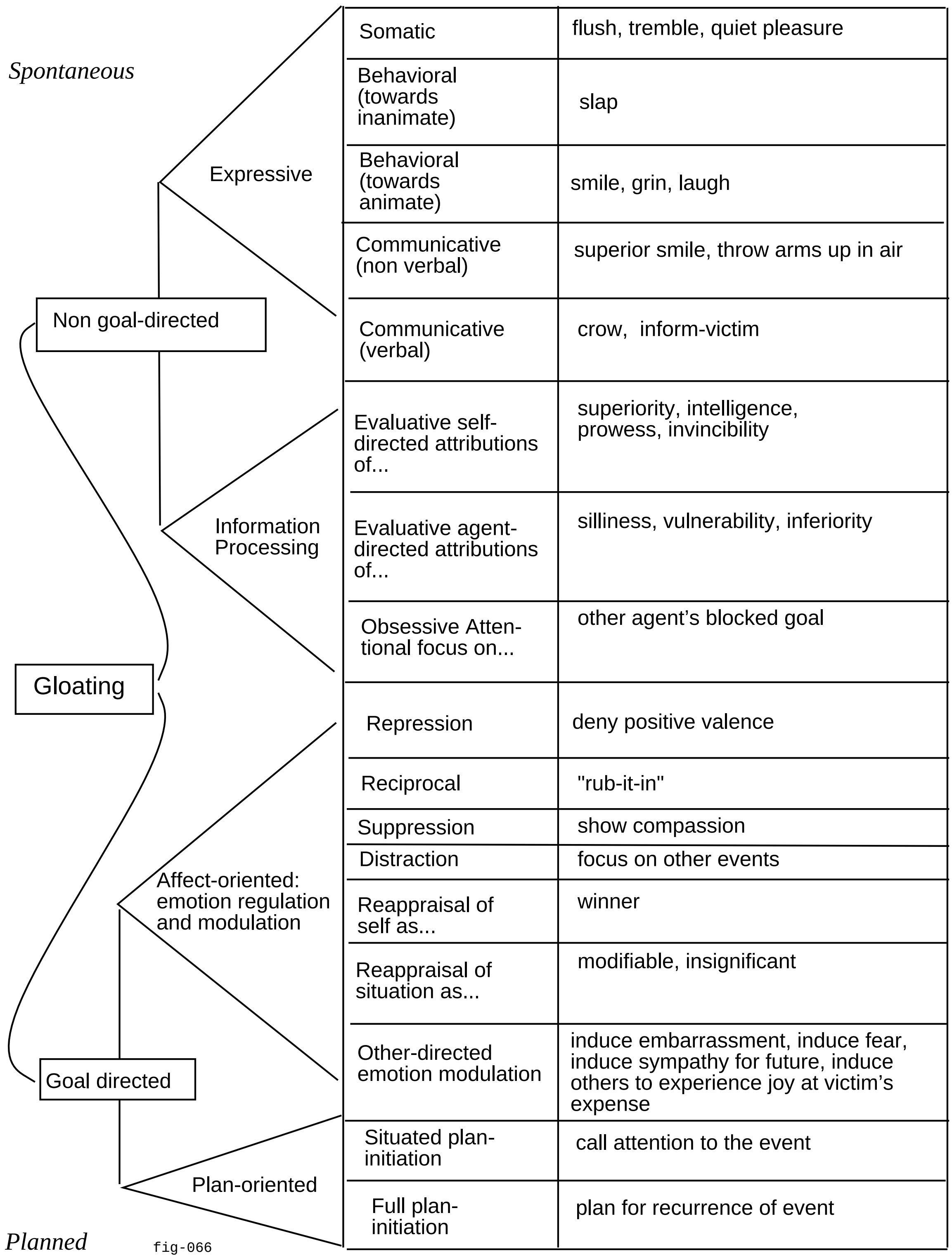


Figure 4.1: Action Response Categories for *gloating*.

purely subjective.[2]

One important point to consider is that this is NOT an attempt to list action words that might be associated with a particular emotion. If it were then we might have the following sorts of entries for *gloat*: boast, glory, delight, jubilate, exult, and so forth. While this would be interesting with respect to understanding and generating text, it is not useful when attempting to characterize some internal state. The point is this: there are many synonyms for a single action expression of an emotion. Since we are making the attempt to generate response actions that have some (possibly abstract) *purpose* we are interested in reducing the number of redundant emotion expressions, and in increasing the range of purposeful emotion expressions allowed. Thus, while *slapping* some inanimate object is a rather obscure way of expressing the *gloating* emotion, it has more value to the system than the more understandable *exult* which is simply another way of saying *crow.*

The actual response actions themselves may be simple tokens, such as <*flush*>, they may be templates requiring variable instantiation such as <*laugh at ?other-agent*>, or they may be full plan initiations such as <*plan for recurrence of event*>.[3] Plans in the Affective Reasoner are simply "canned" responses since planning is not the focus of this research. Nonetheless, emotionally motivated plan *invocation* has been addressed, and at least may serve as a starting point for the integration of other work.

Once an action has been selected it is instantiated using the set of variable bindings generated by matching the original emotion eliciting situation frame. For example, bindings may be accrued for *?other-agent* during the original match process. This binding may then be used to instantiate the action response template <*laugh at ?other-agent*>, described above.

Figure 4.2 shows the full structure of the first portion of the action response mechanism. As can be seen, the structure for all twenty-four emotion categories is similar, although only that for *gloating* is shown here. In this example we see that *superior smile* and *throwing arms up in the air* are two ways of non-verbally communicating *gloating* and that *inducing embarassment* and *inducing others to experience joy at the victim's expense* are two ways of *modulating the emotions of others* as an expression of the same *gloating* emotion.

As shown here, the action response space can be broken down into the elements of a three-dimensional array. The first dimension is the twenty-four emotions, the second

[2]Note that it would be difficult to defend any such ordering, even if it were based on actual studies, since the degree of *spontaneity* in an action relative to other actions is meant to at least approximate some functional attribution, not our assessment of it. The same is true of the intensity ordering.

[3]In the actual system actions are represented as tokens until after the exclusion set processing stage (see section 4.6), but, except for the simple actions, have pointers to the actual templates and plans.

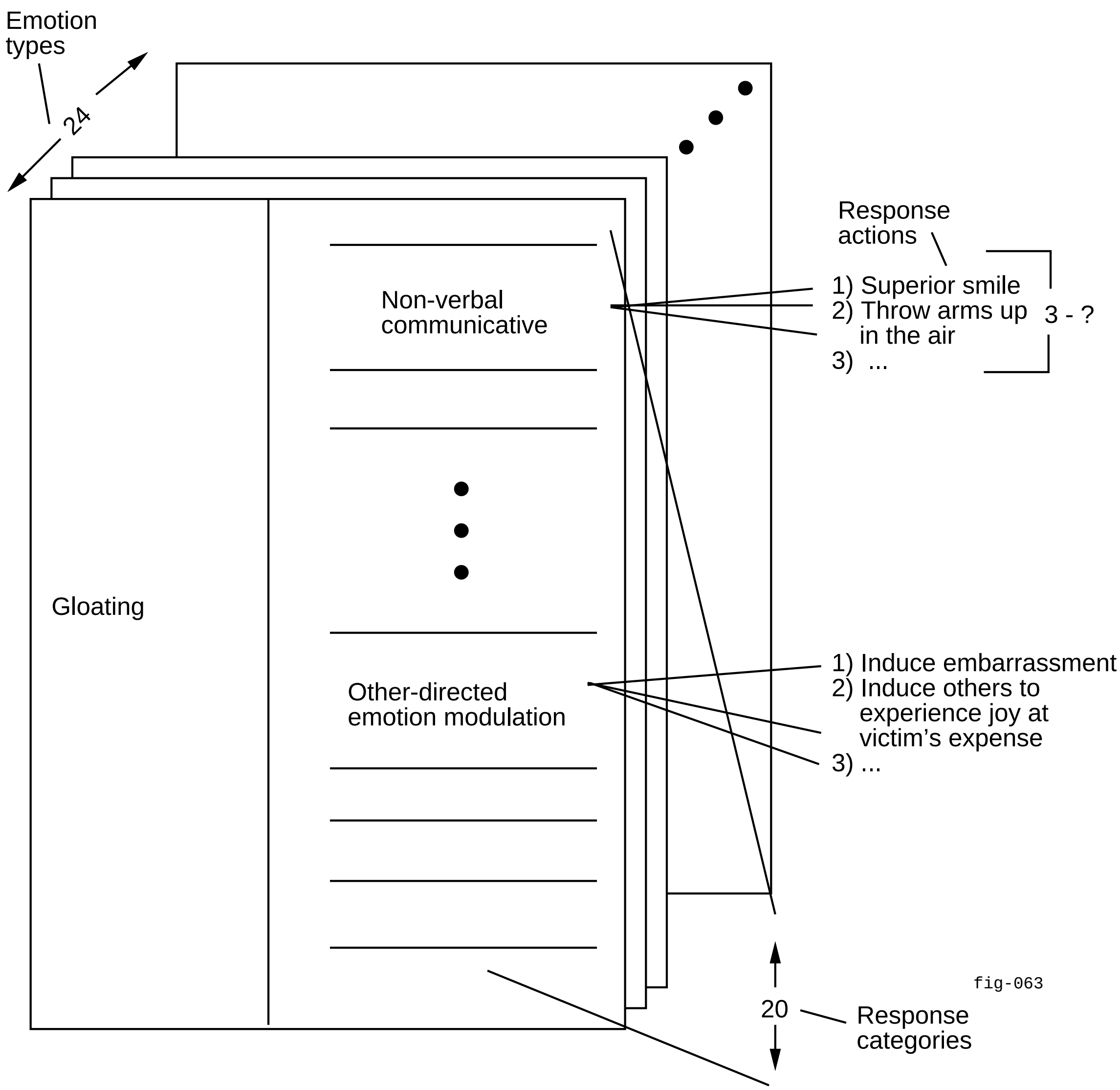


Figure 4.2: Structure of the action response

is the twenty action response categories and the third is the actions themselves. To understand the magnitude of the action classification task, consider that given even a modest three action tokens per response action category, we have 24 emotions times 20 action response categories times 3 actions giving 1440 actions to be tracked within the system. Even allowing only 3 actions to be selected as simultaneous manifestations of some emotion we have 1440^3 action combinations, not counting the almost limitless number of instantiations of templates containing variables.

4.3 The structure of response action selection

Having laid the foundation for expressing many different action sets in response to emotions, we now need a method for grouping these responses into those that are typical for the personality we wish to represent. At a minimum we want to be able to represent such broad characterizations of temperament as *shy* and *outgoing*. Better yet, we would like to have an even more specific characterization. For example, we might wish to be able to represent an agent who tends to be obsessively focused on other agents' views of him, who tends to have somatic responses to stressful situations, who rarely initiates plans in response to his emotions, and so forth. One way to think of this is to imagine an actual person within the context we wish to simulate and then describe how he might respond to different situations. The twenty or so action response categories provide a reasonable set of options for making such a description.

For example, an agent might be described roughly as *highly excitable, warm, insecure, illogical* and *shy*. For such a personality we might choose to emphasize the temperament traits *somatic* – positive and negative emotions (highly excitable), *behavioral (towards animate)* – positive emotions (warm), and *evaluative (self-directed attributions)* – negative emotions (insecure), and so forth, but would be less likely to choose the temperament traits, *full plan-initiation* – because (illogical), or *verbal-expressive* – because (shy), etc.

In the interests of simplifying the task of specifying these action personality types, a concession has been made (but see section 4.5.2 for other, more complex, approaches to this): temperament traits are selected across the entire range of positively valenced or negatively valenced emotions (i.e., in the second dimension of the action response space). For example, if an agent is represented as tending to be *repressive* with respect to *anger*, then the agent is also represented as being *repressive* with respect to the other negative emotions such as *reproach, remorse*, and so forth. Clearly there are personalities which we might wish to represent which are excluded by such an arrangement. On the other hand it is easy to imagine that a shy person is shy with respect to expressions of all of his positive emotions, that a cold person tends to deny the feeling of any positive or negative emotions, and so forth.

In many cases the expressive temperament traits chosen (or emphasized) will be the same for both the positively and negatively valenced emotions. In other words, if one is open in his expressions of negatively valenced emotions he will also tend to be open with respect to his positively valenced emotions. However, activation of traits for the positive emotions and for the negative emotions has been separated in the Affective Reasoner to give more flexibility in the specification of personalities, and, specifically, for the representation of *moods* (for a discussion of which see section 4.5.2 below). It should be understood that the decision to force the selection of temperament traits to be the same for all twelve emotions in each of the two sets of emotions (i.e., the positive set and the negative set) is *not* a restriction imposed by the architecture of the system. At the time the personality of an agent is created, or his mood altered, each temperament trait enabled or disabled is ultimately selected, by name, in the rete-like structure representing the action selection database for each emotion. The problem with doing this manually, however, is that one must then specify each of the 480 temperament trait/emotion duples when defining a personality. This has not, so far, proven to be worth the effort.

One must be careful not to confuse a predisposition to have certain emotions with a predisposition to express the emotions that one has. For example, to describe someone as being an *angry* person only suggests that he is likely to experience the *anger* emotions, a matter for the construal process to address. No claim is made about this agent's tendency to express the anger emotions in a *retaliatory* fashion, to have *somatic* responses, and so forth. In addition, no claim is made about the nature of an agent's expressions of the positively valenced emotions.

Once the system template has been created representing the full spectrum of *potential* emotion expression paths it must be personalized for each agent. This entails copying the entire database and then activating temperament traits according to the action personality profiles of the individual agents. Figure 4.3 illustrates this process. The emotion types, temperament traits, and so forth are combined to form the system template database. This is copied in full for each new agent in the system.[4]

Finally, at setup time, or whenever an agent's mood or manifestative personality is changed, the temperament traits representing the desired response action profile for that agent are activated.

[4]The database is copied as a *framework* for holding agent-specific information in the form of bindings, as discussed later in this chapter. However, even this is somewhat misleading. Many elements that are copied are really only pointers to shared structures. In addition, some design considerations have had to do with implementing agents so that they run on different machines and communicate through a server.

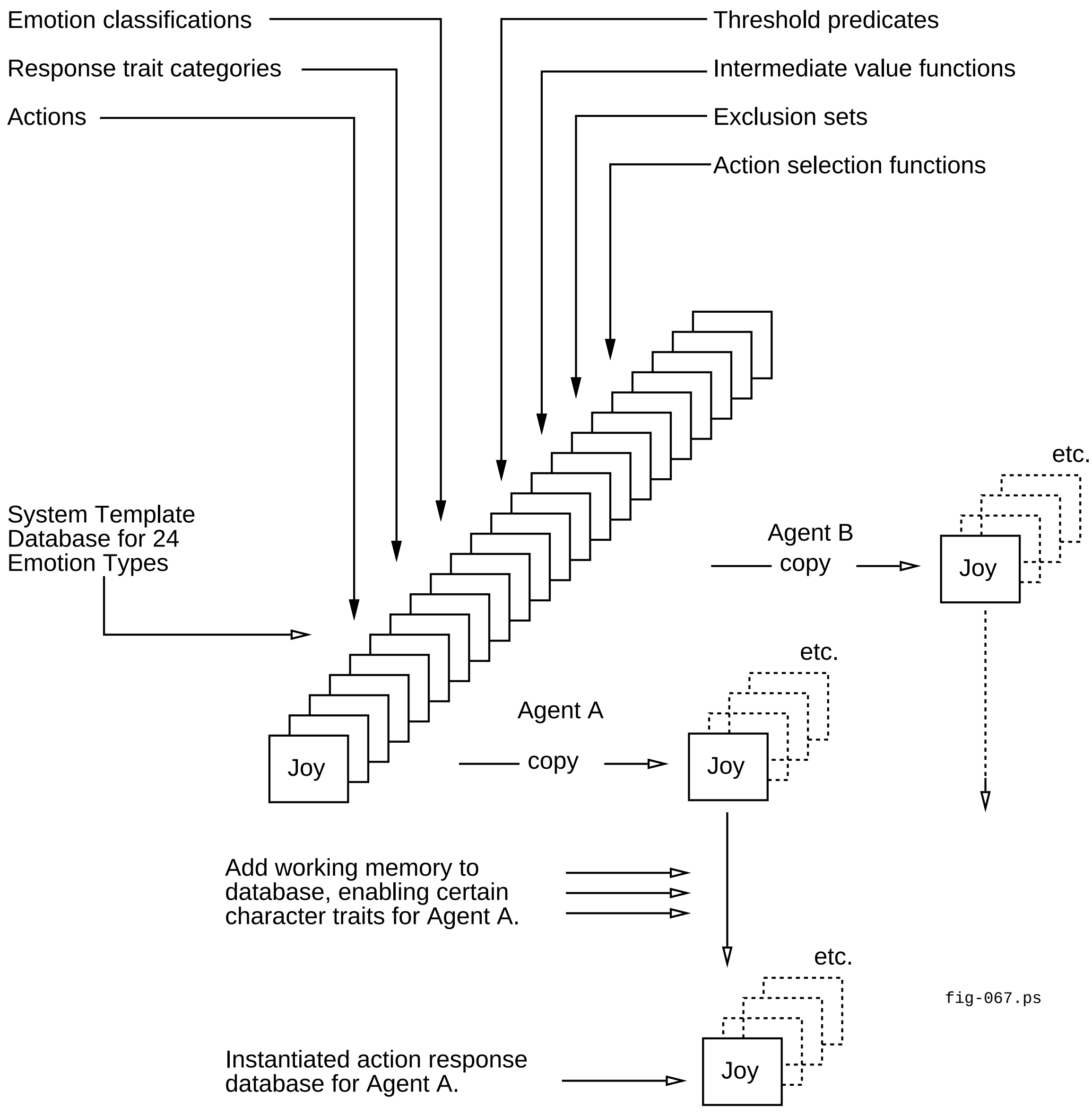


Figure 4.3: Creation of personalized action database.

4.4 The functional spectrum of response categories

Because of the complexity of the relationship between emotions and action responses, we have implemented a constraint on the range of action profiles represented wherein at most one token from each action category (i.e., *somatic, repressive,* etc.) is allowed to participate in the action set expressing any given instance of an emotion.[5] The justification is this: our initial purpose is to create interesting, realistic personalities. To reason effectively about their actions we need to have some understanding of the structure of their responses to events. It is sufficient, at least for the time being, to be able to represent a significant subset of all possible action personalities. Excluding some personalities from representation in the Affective Reasoner is not desirable, but neither is it prohibitive for most applications. (The converse of course, including personalities that one could never encounter, would be a much greater problem.) The following list gives some of the benefits of this approach.

Limiting cases. While there are still a great many ways of expressing each emotion category, this technique places some limits on the total number of cases likely to occur, and organizes the cases so that they will be more easily represented by prototypes.

Consistency. By organizing the token choices in each category, some consistency problems can be avoided. For example, some tokens are not compatible (e.g., induce sympathy and induce fear). This tends to fall out from the idea that expressions in each action category are *purposeful* in nature, and that a single expression will serve that category's purpose; alternate expressions are thus not necessary.

Intensity spectrum. As discussed above, even though not used in the present system, the tokens are arranged in a partial ordering by intensity. Selecting one token suggests the selection of one intensity. If, in the future, the system is altered to allow more than one token from each category, then equivalence classes would need to be created for equal intensity-valued tokens and multiples limited to tokens from a single class.

Interesting action sets. This is a very coarse-grained system with respect to the subtleties of human emotional expression. Given the limited representational resources available, we must use the relatively few tokens to their greatest expressive value. Spreading expressive purpose over different expressive categories helps to achieve this.

[5] But see item 7 in the annotations for figure 4.6 for other selection methods.

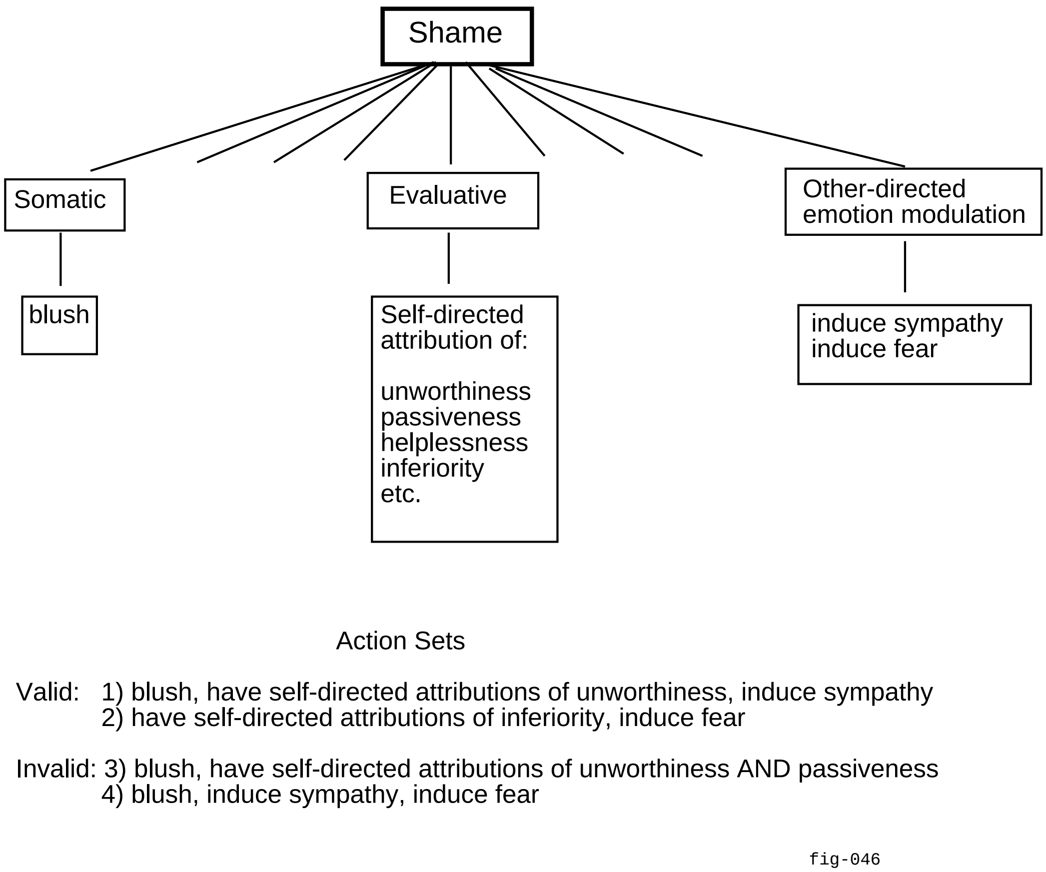


Figure 4.4: Selecting a single action from each response category.

The only apparent problem is that some valid action sets may not be represented. For example, *glancing downward* and *having head bent* (both *communicative, non-verbal* actions) are obviously compatible but they will not appear together in this system in its current configuration.

Figure 4.4 illustrates how this constraint works. Action tokens may be selected from one or more of the action categories. In set (1) a token has been chosen from each of the three listed categories producing the action set, *blush, have self-directed attributions of unworthiness, induce sympathy.* In set (2) tokens have been selected from two of the three listed categories producing, *have self-directed attributions of inferiority,* and *induce fear.* Set (3) is not possible in the Affective Reasoner because it includes both *have self-directed attributions of unworthiness* and *have self-directed attributions of passiveness*, each of which is in the *evaluative* category. Set (4) is not possible because it includes *induce sympathy* and *induce fear,* each of which is in the *other-directed emotion modulation* category.

4.5 The action database entries

In this section we discuss the format of the databases comprising the action component of agents' rudimentary personalities. We have already discussed the three-dimensional structure of the database and the way the basic template is copied for each new agent created. We now look in greater detail at the actual database entries themselves. For this discussion we focus on a single entry, the *communicative, nonverbal* entry for a single emotion type, *shame.* To put this in perspective there are approximately 480 such entries in an agent's full response-action database.

First we look at the basic format of the entry, explaining each of its clauses. Next we instantiate each of these clauses in sequence with the characteristics of *shame*. Following this we look at a conceptual view of the configured entry. Lastly, we discuss how the final form of this *shame* entry is tuned to represent the individual action response characteristics of agents in the system.

4.5.1 The basic layout of an action database entry

The basic layout of an action database entry is shown in figure 4.5. Here we see a set of clauses that will first be instantiated to represent one of the action response categories for one of the emotions (e.g., *communicative, nonverbal*), will then be copied into a new agent's structure representation, and then will either be activated or left dormant when the agent's action personality component is tuned. During action generation for the agent the processing mechanism reads these clauses from the top down, collecting bindings as it goes. In this processing, should any of the clauses 3 - 7 fail to have been activated by the addition of working memory, action generation fails for this temperament trait as an expression of the parent emotion.

Annotations for figure 4.5

1 Action template. This will be the output from processing this database entry. Each time this database entry is used for producing actions a copy of this template (i.e., *(action ?action-set ?bindings)*) is instantiated with the final action set selection and the current bindings. The instantiated template is placed in a temporary queue, along with other actions selected for other emotion-type / action-character-trait duples for discrimination by, first the conflict resolution mechanism, and then the action-generation calling function.

2 Index key. This key indexes the action database entry. Hooks have been left for future improvements to the current, crude, indexing scheme. When looking for responses to the *shame* emotions, for example, this database entry will be searched.

1. (action ?action-set ?bindings)
2. (emotion-name) ;; e.g., *shame*
3. (nil has-trait negative communicative-non-verbal)
4. (nil shame-communicative-non-verbal ?acts)
5. (nil special-wmA ?special1 ?special2 ...)
6. (nil special-wmB ?special3 ?special4 ...)
7. ...
8. (threshold-predicate)
9. (?intermediate-varA ((lambda (var1 var2...) (function)) ?var1 ?var2...))
10. (?intermediate-varB ((lambda (var3 var4...) (function)) ?var3 ?var4...))
11. ...
12. (?act ((lambda (acts intermediate-varA...) (function)) ?acts ?intermediate-varA))

Figure 4.5: Basic Format of an action database entry.

3 Trait selection clause. This clause activates the database entry for an agent when a matching working memory token is added to the agent's action database. A discussion of the importance of this clause follows in section 4.5.2.

4 Name of database entry. This is the name of the emotion-type / temperament trait duple that this database entry represents. *?acts* will be bound to the eligible actions for this duple when working memory is updated during the initialization process. This is a *persistent match* clause, so called because once it has been altered by a change to working memory, that change persists until it is further updated or actively removed.[6]

5-7 Optional additional persistent match clauses. Any number of special working-memory clauses may be added here. Each of these clauses is again a *persistent match* clause. The *special-wmA* token is simply an identifying constant; the *?special1, ?special2,...* variables are additional parts of the template, and will cause bindings to be stored when matches are made. For example we might include the clause,

```
(nil physical-state ?what ?valence ?intensity)
```

In this case, clauses such as *(physical-state head negative severe)* will match when added to the database. The bindings *((?what / head) (?valence / negative) (?intensity / severe))* will be stored for future reference. In general, any match configuration is allowed, including any number of variables. Each time working memory that matches this clause is added, a new variable binding will be stored for each of the variables. As with the *?acts* variable, these working memory bindings are stored in place of the *nil* list entry at the beginning of the clause. In all cases these bindings are cumulative, so that if more working memory elements are added that match these clauses, more bindings are stored. Working memory may only be *explicitly* removed by working memory identifier.

8 Threshold predicate. If used, it is a function that evaluates to *true* or *nil*. If a *true* value is returned the matching process continues and an action is selected. If a *nil* value is returned processing ceases and no actions are returned for this database entry.

9-11 Additional variable binding functions. Sometimes it is useful to add additional bindings to the bindings list. These clauses allow the researcher to list the additional variables and record the functions that determine their values.

[6] For the current discussion, the *nil* token at the beginning of all the persistent match clauses should be ignored. They are merely place holders for a list position that will be used for storage of the currently active bindings for variables in the clause, and do not participate in the match.

12 Temperament trait action selection. The *?act* variable is bound to the final selection of actions for this database entry. *?acts,* which contains the candidate actions, *must* be passed, along with any other variables needed by the choosing function. In the degenerate case this function simply makes a random choice from the list of candidates.

4.5.2 The trait-selection clause

As stated above, this is a powerful and important clause in the database. With the current version of the action theory in the Affective Reasoner, this clause sees only basic use, but the design of the system allows a much more powerful theory to be expressed. As an example of the simplest form of its use, suppose the following directive were to be issued to the simulator:

```
(add-wm-element '(has-trait negative communicative-non-verbal)
                (name-to-action-database 'Tom))
```

This would cause working memory to be updated for *Tom's* action database, specifically the *communicative-non-verbal* entries for each of the twelve negative emotion types. Note that the *empty* bindings list will be saved since there are no variables in the match template for this clause. However, the nature of the propagation path through the action database hierarchy is such that some binding set is necessary for each clause, even if it is empty, for the path to succeed. If more than one binding set exists then more than one path through the hierarchy succeeds.

A number of interesting features are present in this part of the action database entry. First, it should be obvious that this is simply a binary control switch to activate and deactivate a temperament trait for an agent as per the current action theory. However, this need not necessarily be binary. One might specify instead that actions from a given trait were to become *more likely* or *less likely* to be selected. For example, the following assertions:

```
(has-trait negative communicative-non-verbal very-likely)
```

and

```
(has-trait negative communicative-non-verbal possibly)
```

might be inserted in the database matching the the template:

```
(has-trait negative communicative-non-verbal ?likelihood)
```

In this case *?likelihood* is bound and then passed to some later function (e.g., the threshold predicate of clause 8) for evaluation. Similarly we might also record *?arousal-level* and *?attentional-focus-level,* and so forth, here for the agent, thus affecting *which* action is actually selected from this database entry, instead of merely *whether* an action is selected. In other words, precisely the manner in which this trait is activated determines its action selection characteristics.

Second, note that the token *negative* is included in the trait-activation template. As discussed, this is a way of tuning personalities differently for the negative emotion types and positive emotion types. It also provides a crude method of allowing an agent's *moods* to be tuned. As an example of the first case consider the attempt to create an action personality for an agent, *Tom,* who never has anything nice to say. In this case we would issue the following directive:

```
(add-wm-element
     '(has-trait negative communicative-verbal)
      (name-to-action-database 'Tom))
```

which would allow us to select verbal communications as an expression of the negative emotions. We would explicitly NOT issue the following, however:

```
(add-wm-element
     '(has-trait positive communicative-verbal)
      (name-to-action-database 'Tom))
```

By not doing so we prohibit verbal communications as expressions of the positive emotions, which is what we wanted. Note that we are not representing someone who always sees the world in a bad light, which is strictly a matter for the interpretive personality component to address. Instead, we are only tuning the expressions of affective states which already exist.

With respect to moods, one can imagine that a generally negative mood might tend to activate certain temperament traits in the negative emotions and certain other traits in the positive emotions. For example we might wish to issue the following mood-tuning directives:[7]

```
(add-wm-element
    '(has-trait negative somatic)
     (name-to-action-database 'Tom))
```

and

```
(remove-wm-element
    '(has-trait positive communicative-verbal)
     (name-to-action-database 'Tom))
```

for someone who grows sullen when they are in a bad mood (i.e., to increase the likelihood of a *somatic* response for a negative emotion, and remove the possibility of a *verbal-communicative* response for a positive emotion).

[7]This form is used only for clarity of discussion. Working memory must actually be removed by *explicit* working memory identifier.

Nor are we limited to such a two-dimensional treatment of moods and personalities. Obviously we can, in principle, represent any grouping relating temperament traits to moods and personalities within the emotion space. As an illustration, we might focus one set of working-memory updates on the *standards-based* emotions, allowing us, for example, to specify that an agent will tend to initiate plans as a response to *admiration, reproach, gratitude, anger, pride* and so forth while not making any specification for responses to *distress, joy, pity* and the like. Such an update might be achieved by the following:

```
(add-wm-element
    '(has-trait standards-based full-plan-initiation)
     (name-to-action-database 'Tom))
```

matching:

```
(nil has-trait standards-based full-plan-initiation)
```

but not

```
(nil has-trait goal-based full-plan-initiation)
```

We can also combine these two (or more) dimensions giving us such clauses as:

```
(add-wm-element
    '(has-trait standards-based positive full-plan-initiation)
     (name-to-action-database 'Tom))
```

matching:

```
(nil has-trait standards-based positive full-plan-initiation)
```

but not

```
(nil has-trait goal-based positive full-plan-initiation)
```

or

```
(nil has-trait standards-based negative full-plan-initiation)
```

Lastly we raise the possibility of the grouping of temperament traits strictly by mood. Just as we use the temperament traits to define the overall individual characteristics of an agent's action personality, so might it also be useful to characterize moods the same way.

As a coda to this section we note that, as previously stated, action working memory in the Affective Reasoner is cumulative. This means that if some clause is added to working memory more than once, a new path through the affected database entries may be found for each additional working memory element. This opens up possibilities for interesting intensity, duration and mood calculations based on independent forms of input. For example, some process $\mathcal{P}$ in the simulation may cause a working-memory item to be added to *Tom*'s action database. Another process $\mathcal{Q}$, not coordinated with $\mathcal{P}$, adds the same working memory item. A duplicate path through the action database entries is created. This means that particular tokens are twice as likely to be selected as before to express some particular affective state.[8] This can be useful for representing the cumulative effects of certain kinds of interaction between the agents and their environment without having to control the interaction of asynchronous events.

For example, suppose that the user has set up the simulation so that at 5:00 P.M., all the agents become more belligerent. Included in this is an increased likelihood of a *verbal expressive* response for the negative emotions. In the hours preceding this 5:00 P.M. global mood change Tom, an agent, has had a series of goals blocked which, independently, puts him in a bad mood. This also enables the *verbal expressive* response path for the negative emotions in Tom. Tom now has two paths to *verbal expressive* responses (or three if he had one enabled before the two changes just mentioned). This increases the likelihood that Tom will have one of these emotion manifestations in the output response action set for any of the negative emotions. No coordination between prior manifestative personality settings, global mood changes and local mood changes is necessary.

4.5.3 The conceptual view of an instantiated database entry

Figure 4.6 illustrates the conceptual layout of an instantiated action database entry for *shame* as it would appear in the system template database.[9] In other words, this is a piece of the profile of an action database entry for all agents, before any personalized tuning has been performed. The filling in of this template comes in three phases.

The first phase is the creation of the static portion of this system-wide database. This includes the creation of each of the twenty-four emotion types (e.g., *joy, shame, ...*), the approximately twenty character-trait response categories for each of those emotion types (e.g., *joy-somatic, joy-behavioral-towards-animate, ... shame-somatic, shame-behavioral-towards-animate, ...*), threshold predicate functions for each emotion-type / character-trait duple (to determine if a given situation and emotion actually merits a response), intermediate variable binding functions (for adding to the bindings list)

[8]Actually the algorithm is more complex and more interesting than this. Additional paths through the action database give weight to certain *sets* of action tokens. This also then affects the likelihood of the set passing the exclusion set conflict resolution algorithm (discussed in section 4.6). Also, paths through the database are multiplicative, so adding two working-memory items, in series, actually produces *one* path through the database, adding two sets of two, in series, produces *four* paths, two sets of three, *six* paths, and so forth.

[9]This is conceptual in the sense that it has been simplified and re-ordered to avoid having to explain uninteresting implementation details.

and a final action-selection function for selecting one (although possibly more) actions to express each emotion-type / character-trait duple (e.g., a function to choose between *head-bent* and *eyes-cast-downward* as an expression of *shame-communicative-non-verbal*).

The second phase is the addition of system working memory to the existing static template. This should also be considered as functionally static since working memory is altered before the agents are created, and since these working memory elements are not removed.[10] In this stage the action tokens (and templates and mini-plans) for each of the emotion-type/character-trait duples are added to the database and stored in each of the associated 480 *?acts* variables. For example, this is where the *?acts* variable for *shame-communicative-verbal* is bound to *mutter, apologize, etc.*.

The third phase of preparing the action database is where the true dynamic working memory is updated. When a new agent is created for a simulation run, the system-wide database produced in the first two phases is copied whole into this agent's structure representation. Two sets of working memory modifications are performed on this copy to create the individual agent's action personality.

(1) the action personality keys are used to enable specific paths through the action hierarchy. If an agent, *Tom,* has the temperament trait *somatic,* for example, then the statement,

```
(add-wm-element
    '(has-trait somatic)
     (name-to-action-database 'Tom))
```

would be issued enabling *somatic* action responses to be selected for the agent *Tom*. (2) the special *persistent match* clauses are tuned. As stated above, these are the optional clauses where additional working memory items can be stored. Given a physical state clause, we might, for example, specify that *Tom* is feeling sick, or even that he is asleep.

To illustrate how these three phases work we will now build our example database entry for *shame.* The following list of annotations shows how this is done.

Annotations for figure 4.6

Line 3: Temperament trait activation clause. As discussed this clause stores an empty binding when the temperament trait has been activated.

Line 4: Identifier for temperament trait database entry. After issuing the following working memory update directive:

[10] This was implemented as working memory for the convenience of the variable binding mechanism, and to leave a hook for large scale dynamic changes of action context (e.g., from that of being at home to that of being at work) in future versions of the system.

1. (action ?action-set ?bindings)

2. (shame)

3. (*has-trait negative communicative, non verbal (nil)*)

4. (*shame-communicative-non-verbal ((head-bent glance-downward))*) ;; var: ?acts

5. (*physical-state ((hungry) (very))*) ;; vars: ?phys-state, ?phys-intensity

6.
```
((lambda (phys-state phys-intensity)  ;;  <- intensity function
          (cond ((and
                     (is-bound phys-state phys-intensity)
                     (eql phys-state 'hungry)
                     (intensity-gt phys-intensity 'medium)) nil)
                (t t))) ?phys-state ?phys-intensity)
```

7.
```
    (?duration
      ((lambda (time-of-event phys-state phys-intensity)
         [code to determine duration of emotion])
       (?time-of-event ?phys-state ?phys-intensity)))
```

8.
```
    (?intensity-curve
      ((lambda (time-of-event phys-state phys-intensity duration)
         [code to determine intensity curve of emotion])
       (?time-of-event ?phys-state ?phys-intensity ?duration)))
```

9. (?act (lambda (acts intermediate-varA...)

Figure 4.6: Conceptual layout of an action database entry for *shame, communicative (non verbal).*

```
(add-wm-element
    '(shame-communicative-non-verbal (head-bent glance-downward))
     (name-to-action-database 'Tom))
```

we see that the *?acts* variable has been replaced with two action tokens, *head-bent* and *glance-downward.* It is from this set of two tokens that we will select an appropriate action for the *communicative-non-verbal* temperament trait expression path for the *shame* emotion. The clause was added in phase one, and the action tokens were added in phase two.

Line 5: Optional persistent memory clause. Here we see that an additional working memory clause has been added. This particular clause allows us to keep track of the current physical state (or states) of the agent. In this case working memory has been added so that *hungry* and *very* are bound to the variables *?phys-state* and *phys-intensity*, the effects of making a (phase three) assertion such as:

```
(add-wm-element '(physical-state (hungry) (very)))
```

In other words, the *clause* was added to the database in phase one, and the *tokens* representing the state of *Tom* being *very hungry* were added in phase three. Note that a typical use of these additional persistent-match clauses is to simultaneously alter a number of database entries. One can imagine that significant changes in the physical state, such as being hungry, will affect the expression of a number of different emotions at once.

Line 6: Threshold function. Next we add a threshold predicate which, in our example, says that if the agent is more than a little bit (*medium*) hungry, then the predicate will fail and he will not act to express *shame* in a *communicative-non-verbal* way. A projected use of the threshold predicate is for intensity reasoning (i.e., to determine thresholds under which no action is taken to express the emotion), although as seen here, it can be used for any sort of filtering. The threshold predicate may, of course, contain conjunctive and disjunctive reasoning and so may actually be a composite of many different tests. This threshold function is added to the database in phase one and, since it has no working memory component, is not altered in later phases.

Line 7-8: Additional variable binding functions. Next we add some functions for adding bindings to the bindings list. These are evaluated in series, so the result of one binding may be used in the determination of another binding. Obviously these bindings could be performed all at once within a single function, but since one of the stated functions of the Affective Reasoner is to do psychological modeling and simulation it was clearly a better choice to make the creation of each binding as explicit as possible. For example, here we first bind *?duration* to some value using the time of the original situation (this is in the set of bindings passed from the original situation match occurring during the construal process), the physical state of the agent and the intensity of that physical state. Next we bind *?intensity-curve* to a value using all of the previous arguments with the addition of the *?duration* variable.

To understand the structure of these functions (and the temperament trait action selection function which follows) consider how they are evaluated. First, the bindings are used to instantiate the function arguments. In this case it means that *?time-of-event, ?phys-state, ?phys-intensity* and later, *?duration* are bound to values from prior processing of the event and its construal. The function is then called with these values as arguments.[11]

Line 9: Temperament trait action-selection function

Lastly we add a function for selecting which of the actions will represent this temperament trait category (in this case the *shame-communicative-non-verbal* category). As explained above in section 4.4, in most cases only a single action token is selected. This is arbitrary however, and is under the control of this function. If intensity values are to be used, their influence will come into play in this function. Using the simplest intensity algorithm, for example, we would have intensity range from 1 to N and the action tokens ordered by intensity from 1 to N. An intensity of n would cause action token n to be selected. As discussed in section 4.2 however, it is not generally possible to fully order the action tokens by intensity within a temperament trait category for some emotion. Neither is it a simple matter to construct a meaningful intensity function within a domain.

Alternatively, we might wish to select more than one action for this emotion / temperament trait path. Additional theory is needed to specify relationships such as *often co-existing* and so forth with respect to relationships between these actions. It should be understood also that this complicates the task of reasoning about *observed cases* of action expressions as well.

Should it happen that the selection criteria for a given set of response actions does not give precedence to one action over another within a given response category, an action is chosen at random. Until we have a better understanding of the relationship between emotions and actions this randomness of action choice is the best we can do. How can we say what it is that determines that we shake our head as a non-verbal expression of reproach in one instance, and throw up our arms

[11]The data input macros relieve the researcher from having to repeatedly enter the variable names however.

in another? On the other hand, it seems likely that if one action, such as *smiling*, serves more than one function (i.e., behavioral-toward-animate, non-verbal-communicative, other-directed-emotion-modulation, etc.) in the action response array its likelihood of selection will be increased. The Affective Reasoner's current selection algorithm supports this.

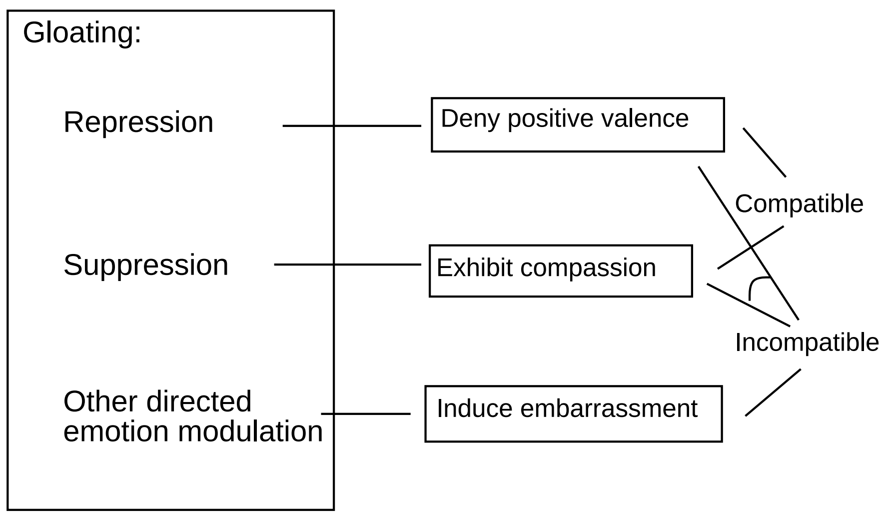
Finally, we must consider the problem of selecting actions when one is experiencing mixed emotions. If *shouting* is expressive of both *fear* and *joy*, then it seems likely one might well shout when experiencing both emotions simultaneously.[12] At present this problem is left open, and actions from each of the emotions are simply combined in the action event.

4.6 Action conflict sets and exclusion sets

Once a set of actions has been selected in response to an emotion there is one more step required, that of removing incompatible actions from the action set or of choosing between competing "factions" of sets of incompatible actions. Restricting each category of action tokens to one representative eliminates a great many conflicts, but conflicts can still occur. Manifestations of emotions that are typical for an expressive, outgoing person tend to conflict with those of a withdrawn, solitary person, and so forth. To address this problem in the Affective Reasoner, we have taken a two-step approach. The first step is to group the action tokens under consideration into named equivalence classes. Figure 4.7 shows portions of three such equivalence classes for the *gloating* response: *opposite-valenced* responses, *active/expressive* responses, and *withdrawn* responses. The second step is to identify conflicts between the sets such that a member of one set may not co-occur in the final action set with a member of the second set. In the example we see that the set of tokens in the equivalence class for *active/expressive* responses is not compatible with the set of tokens in the equivalence class for *withdrawn* responses. Thus *none* of the tokens in the first equivalence class may appear in an output action set with any of the tokens in the second equivalence class. The sets of tokens in these equivalence classes are known as *action exclusion sets.* In this example we have captured the idea that a person may deny being happy about the misfortune of someone else and at the same time may exhibit compassion (although not necessarily feel anything like compassion), but the person cannot do either of these while also attempting to overtly embarrass the victim.

A token may appear in more than one action exclusion set. For example, the action token *smile* may appear only as a *suppression* manifestation of the emo-

[12] These mixed and even conflicting emotions are actually quite common. For example, one can imagine being happy to have won the regional semi-finals in the metropolitan opera tryouts, but fearful of having to perform at the finals. Note that Ortony, et al. ([Ortony *et al.*, 1988]) treat these as *oscillating* rather than simultaneous.



Named equivalence classes:

Opposite valenced responses (repression, suppression, distraction):
deny positive valence, exhibit compassion, re-evaluate event as inconsequential, ...

Active/expressive responses:
laugh, induce embarrassment, shout, ...

Introverted responses:
quiet pleasure, superior smile, ...

fig-065

Figure 4.7: Action exclusion sets

tion *distress.* As such it would naturally appear in an action exclusion set for *suppressive/distraction-effecting* tokens, to be contrasted with the more direct *communicative* responses. However, it might also appear in an action exclusion set for its *active* quality and would be excluded from appearing with the tokens in the *withdrawn* action exclusion set. (E.g., people tend not to laugh when they are being abnormally quiet). Exclusion sets are created *ad hoc* to restrict output response action sets to viable candidates for each of the emotion types. Here is an example representation of exclusions sets for *distress*:[13]

```
(table-value 'distress *exclusions*)
(make-exclusion
 :sets
 '((WITHDRAW tired be-quiet low-energy withdraw quiet-voice)
   (ACTIVE hyperactive be-active be-entertained smile laugh)
   (SUPPRESS laugh smile be-active be-entertained)
   (EXPRESS cry drooping-posture frown withdraw))
 :incompatible '((WITHDRAW ACTIVE) (SUPRESS EXPRESS)))
```

The use of action exclusion sets is only one of the ways that the Affective Reasoner has to make decisions about the final version of the output action set. As discussed in section 4.5.3, processing may also be fine-tuned for each of the emotion categories through the use of the action selection function. Clearly, the resolution of many conflicts is dependent upon the context in which the action is performed. In general, this is a difficult problem, and it is of necessity only cursorily addresssed in the Affective Reasoner.

4.7 Summary

In this chapter we examined the generation of emotion-based actions for the agents in the Affective Reasoner. The spectrum of action responses to emotion eliciting situations has been reduced to approximately twenty *action response categories*. Each category groups together responses that share functional qualities. These categories, together with the twenty-four emotions create a grid within which the actions themselves are placed. Actions consist of simple tokens, templates containing variables, and simple mini-plans. There are roughly 1400 such actions in the system. The structure resulting from this description of emotion-based action is used as a template for the action databases of the individual agents.

Agents have a *manifestative personality component* which controls which *temperament traits* are active for each of the agents. Each temperament trait is associated with one of the action response categories and activates a set of paths through either

[13] Some of these tokens are pointers to mini-plans containing variables and system events.

the set of positively valenced emotions or through the set of negatively valenced emotions. When a temperament trait is active for some agent he may manifest actions associated with the corresponding action response category. The set of potential response actions generated is reduced so that the actions do not conflict with one another.

The capabilities of the expressive system beyond the current state of the action theory were discussed. Actions can be ordered by intensity. Working memory can be used to capture moods. An increased *likelihood* of a specific action being selected to express an emotion may be effected. Relationships more complex than simple negative and positive valence may be established between the activation of temperament traits and the emotions types.

Chapter 5

Conclusion

A major goal in developing the Affective Reasoner was to build a platform for modeling the interactions between multiple agents operating in an environment in which some of their goals are shared, and others are unique. The uniqueness of their goals was achieved by endowing each agent with a rudimentary "personality", both with respect to their motives, standards and preferences, and with respect to certain behavioral characteristics. A second important goal was to construct a system for reasoning about the emotions and emotion-induced actions of these agents. Both of these goals required the design and development of an emotion-representation scheme, realized in the Affective Reasoner through such data structures as *construal frames, GSPs, COOs, temperament trait* propagation networks and so forth.

A third goal was to design the Affective Reasoner in a manner at least congenial to psychological theory, where such theory was relevant. To this end, we drew upon a psychological theory of the elicitation of emotions ([Ortony *et al.*, 1988]) which characterizes distinct emotions in terms of the ways in which agents construe situations, and have expanded a preliminary effort of Gilboa and Ortony, which attempts to impose some structure on emotion-based action [Gilboa and Ortony, 1991].

With this machinery, we have been able to make a start not only on the problem of how to model the emotional reactions that agents might have to situations that they encounter in their "world", but also on how such reactions might be manifested in behavior, with concomitant consequences in that world. The system was designed so as to allow agents to build representations of other agents, representations we called *Concerns-of-others* databases. These representations are both based on, and contribute to a capacity to reason about one another's concerns, reasoning that is prompted by one agent observing the "behavior" of another. In this sense the modeled world contains not only the static features of the vicinity in which the agents are operating, but also representations of the agents themselves, including their dynamically updated representations of others.

In implementing a theory as a computer program, many details have to be addressed which may result in the theory becoming specified more precisely. An example of this in the Affective Reasoner is the specification of the emotion elicitation rules. In addition, it was found necessary to add two new emotion types. On the other hand pragmatic constraints of building a running system sometimes necessitate that issues important from a theoretical perspective be finessed. So, in the Affective Reasoner a number of simplifying assumptions had to be made in order to complete the cycle of situation interpretation, emotion generation, response-action generation, and reasoning about such emotion episodes from the point of view of an observer. An ideal computer model of this cycle would certainly include, for example, considerations of emotion intensity and duration, but it simply was not practical to deal with such issues in addition to those already being implemented.

To reason effectively about both the situations that lead to emotions and the actions that derive from them we have adopted two different approaches. The strong-theory aspects of emotion elicitation suggested the rule-based approach we used for mapping the features of modeled situations into emotion representations. The weak-theory aspects of emotional expression prompted us to incorporate a case-based reasoning system and to create a simple theory of the structure of emotion manifestations within the modeled world.

This research is as much an effort in Cognitive Science as it is in Artificial Intelligence, which suggests two ways in which it might be of use. First, if the design and implementation of the Affective Reasoner is sufficiently flexible, it might serve as a helpful vehicle for comparing the implications of different theories pertaining to emotion elicitation, emotion-induced action, and the relation between idiosyncratic cognitive and behavioral characteristics of agents on the one hand and their affective lives on the other. Second, it may be that elements of this work can contribute to other areas of AI in which some form of emotion reasoning is important.

Bibliography

[Bain, 1986] William M. Bain. A case-based reasoning system for subjective assessment. In *Proceedings of the National Conference on Artificial Intelligence*, pages 523–527, 1986.

[Bareiss, 1989] Ray Bareiss. *Exemplar-Based Knowledge Acquisition, A Unified Approach to Concept Representation, Classification, and Learning.* Academic Press, Inc., 1989.

[Brooks, 1975] Frederick P. Brooks. *The Mythical Man Month.* Addison-Wesley, 1975.

[Charniak *et al.*, 1987] Eugene Charniak, Christopher K. Riesbeck, Drew McDermott, and James R. Meehan. *Artificial Intelligence Programming, Second Edition.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

[Colby, 1981] Kenneth M. Colby. Modeling a paranoid mind. *The Behavioral and Brain Sciences*, 4(4):515–560, December 1981.

[Durfee *et al.*, 1989] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Cooperative distributed problem solving. In *The Handbook of Artificial Intelligence*, volume 4, pages 83–146. Addison-Wesley, 1989.

[Dvorak, 1988] Daniel L. Dvorak. Common lisp version of protos. Electronic copy of common lisp code provided by Ray Bareiss, 1988.

[Dyer, 1983] Michael G. Dyer. *IN-DEPTH UNDERSTANDING: A Computer Model of Integrated Processing For Narrative Comprehension.* MIT Press, 1983.

[Dyer, 1987] Michael G. Dyer. Emotions and their computations: Three computer models. *Cognition and Emotion*, 1(3):323–347, 1987.

[Elliott *et al.*, 1992] Clark Elliott, Lawrence Henschen, and James Lu. Adapting a cognitive modeling platform for use in the control of automated reasoning. unpublished, 1992.

[Elliott, 1992] Clark Elliott. Representing stories in the Affective Reasoner. Unpublished manuscript, April 1992.

[Frijda and Swagerman, 1987] Nico Frijda and Jaap Swagerman. Can computers feel? theory and design of an emotional system. *Cognition and Emotion*, 1(3):235–257, 1987.

[Gilboa and Ortony, 1991] Eva Gilboa and Andrew Ortony. The structure of emotion response tendencies. Work in progress, 1991.

[Lesser, 1992] Victor R. Lesser. A retrospective view of FA/C distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics- Special Issue on DAI*, January 1992.

[Oatley, 1987] Kieth Oatley. Editorial: Cognitive science and the understanding of emotions. *Cognition and Emotion*, 3(1):209–216, September 1987.

[O'Rorke and Ortony, 1992] Paul O'Rorke and Andrew Ortony. Explaining emotions. Submitted for publication, 1992.

[Ortony and Partridge, 1987] Andrew Ortony and Derek Partridge. Suprisingness and expectation failure: what's the difference? In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987. IJCAI.

[Ortony *et al.*, 1988] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The Cognitive Structure of Emotions.* Cambridge University Press, 1988.

[Ortony, 1992] Andrew Ortony. Emotion modeling. In Stuart C. Schapiro, editor, *The Encyclopedia of Artificial Intelligence*, pages 446–448. John Wiley and Sons, New York, 1992.

[Reber, 1985] Arthur S. Reber. *Dictionary of Psychology.* Penguin Reference Books, 1985.

[Reeves, 1991] John F. Reeves. Computational morality: A process model of belief conflict and resolution for story understanding. Technical Report UCLA-AI-91-05, UCLA Artificial Intelligence Laboratory, 1991.

[Regoczei and Hirst, 1991] Steven Regoczei and Graeme Hirst. The corporation as mind: Lessons for ai. Manuscript submitted for publication, 1991.

[Rich, 1991] Elaine Rich. Planning, reacting and communicating. In press, 1991.

[Rollenhagen and Dalkvist, 1989] Carl Rollenhagen and Jan Dalkvist. Cognitive contents in emotions: A content analysis of retrospective reports of emotional situations. Technical report, Department of Psychology, University of Stockholm, 1989.

[Schank and Abelson, 1977] Roger C. Schank and Robert Abelson. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

[Schank, 1992] Roger Schank. Where's the AI? *Artificial Intelligence Magazine*, 1992.

[Simon, 1967] Herbert A. Simon. Motivational and emotional controls of cognition. *Psychological Review*, 74:29–39, 1967.

[Sloman, 1987] Aaron Sloman. Motives, mechanisms and emotions. *Cognition and Emotion*, 1(3):217–234, 1987.

[Tatar, 1990] Deborah Tatar, editor. *CSCW '90 Los Angeles: Proceedings of the Conference on Computer-Supported Cooperative Work*. The Association for Computing Machinery, New York, 1990.

[Toda, 1982] Masanao Toda. *Man, Robot and Society*. Martinus Nijhoff Publishing, Boston, 1982.

[Turner, 1985] Terence J. Turner. Diary study of emotions: Qualitative data. Unpublished, 1985.

[Van Lehn, 1988] Kurt Van Lehn. Student modeling. In Martha C. Polson, J. Jeffery Richardson, and Elliot Soloway, editors, *Foundations of Intelligent Tutoring Systems*. Lawrence Erlbaum Associates, 1988.